



## Aplicação para gestão de equipamentos de laboratório

**LUÍS DA SILVA COUTINHO SEIXAS BRÁS**

Junho de 2022

# **Aplicação para gestão de equipamentos de laboratório**

**Luís da Silva Coutinho Seixas Brás**

**Dissertação para obtenção do Grau de Mestre em  
Engenharia Informática, Área de Especialização em  
Engenharia de Software**

**Orientador: Paulo Baltarejo**

**Co-orientador: Marco Rodrigues**

Porto, junho 2022



# Dedicatória

*<Dedicado à minha mãe, avó e namorada>*



# Resumo

A indústria está em grande transformação, impulsionada pelo desenvolvimento e utilização de tecnologias facilitadoras cada vez mais evoluídas e ágeis. Esta transformação é transversal apesar de alguns setores serem mais afetados do que outros.

Esta revolução, impulsionada pela digitalização dos processos, está a provocar alterações profundas, não só na indústria, mas também na sociedade, na economia, nos valores, na forma como se interage, numa maior customização dos produtos e na forma como os serviços são utilizados. É assim preciso entender as oportunidades e como a adoção de tecnologias disruptivas pode aumentar a produtividade, criar valor e bem-estar social.

Esta dissertação procura apresentar uma solução para o processo de gestão de equipamentos. Trata-se de um processo manual e ineficiente, pelo que se enquadrava perfeitamente como candidato à realização de uma transformação digital. O intuito final será tornar o processo mais organizado, estruturado e eficiente. Deste modo foi feito o levantamento de requisitos do cliente para perceber quais seriam as funcionalidades a implementar.

Após a análise das necessidades fez-se um levantamento das tecnologias operacionais e de informação que poderiam ser utilizadas para desenvolver a solução.

Como resultado da pesquisa foi possível idealizar uma solução (nomeadamente a arquitetura e design) que permite dar respostas às necessidades iniciais do cliente e digitalizar o processo existente.

Em seguida são selecionadas as tecnologias e apresentada a implementação da solução segundo a arquitetura e design sugeridos.

Por fim é avaliada a solução com base num cenário de qualidade no qual foram definidos critérios de cumprimento para cada um dos requisitos e escalas de medição. A aplicação desta metodologia permitiu averiguar a qualidade da aplicação desenvolvida.

Palavras-chave: Indústria 4.0, Tecnologias Operacionais de Localização, Tecnologias de Informação, Gestão de Equipamentos, IoT



# Abstract

The creation and use of increasingly developed and agile enabling technologies is driving a substantial revolution in the industry. This shift impacts all sectors, although some are more affected than others.

This revolution, fuelled by the digitalization of processes, is creating significant changes not only in industry, but also in society, in the economy, in the moral values, the way we connect and utilize services. As a result, it's essential to consider the opportunities and how disruptive technologies might boost productivity, create value and social well-being.

This thesis seeks to provide a solution to the equipment management process. This process it's a time-consuming and inefficient process, what makes it ideal for a digital transformation. The end goal is to make the process more organized, systematic, and efficient. As a result, a survey of the client requirements was conducted to determine which functionalities would be incorporated.

Following the requirements assessment, a survey of operational and information technologies that could be employed to design the solution was conducted.

As a result of the research, a solution (namely architecture and design) that matches the client's initial needs while also digitizing the existing process was devise.

After this, the technologies are selected, and the implementation of the solution is presented according to the suggested architecture and design.

Finally, the solution is evaluated based on a quality scenario in which compliance criteria were defined for each of the requirements and measurement scales. The usage of this methodology allows the quality of the developed application to be ascertained.

Keywords: Industry 4.0, Operational Location Technologies, Information Technologies, Asset Management, IoT



# Agradecimentos

Ao meu orientador do Instituto Superior de Engenharia do Porto, Professor Paulo Baltarejo, pela sua disponibilidade e sugestões ao longo da realização da tese.

Ao Instituto de Ciência e Inovação em Engenharia Mecânica e Engenharia Industrial (INEGI) pela oportunidade da realização do projeto curricular e pelas boas condições de trabalho para o desenvolvimento das competências profissionais e pessoais durante o tempo de estágio.

À equipa de desenvolvimento de produto e sistemas por toda a disponibilidade, ajuda e boa disposição transmitida ao longo da realização da dissertação.

À minha mãe e à minha avó por todo o apoio dado para que tudo decorresse da melhor forma ao longo da minha vida. A disponibilidade e conselhos que ajudaram na minha evolução como pessoa.

A todos os meus colegas e amigos do curso de Engenharia Software que ajudaram na discussão de problemas ao longo do meu curso.

Este trabalho foi apoiado pelo Fundo Europeu de Desenvolvimento Regional (FEDER) através de uma subvenção do Programa Operacional para a Competitividade e Internacionalização de Portugal 2020 Acordo de Parceria (PRODUTECH4S&C, POCI-01-0247-FEDER-046102).

A todos o meu obrigado.



# Índice

<b>1</b>	<b>Introdução</b> .....	<b>1</b>
1.1	Contexto e Problema .....	1
1.2	Objetivos .....	2
1.3	Abordagem .....	2
1.4	Estrutura da dissertação .....	2
<b>2</b>	<b>Contexto e Análise do problema</b> .....	<b>4</b>
2.1	Contexto.....	4
2.2	Problema .....	5
2.3	Abordagem ao problema .....	6
2.3.1	Processo de elicitação .....	8
2.4	Sumário .....	9
<b>3</b>	<b>Estado de arte</b> .....	<b>10</b>
3.1	Digitalização dos processos .....	10
3.2	Indústria 4.0 .....	11
3.2.1	Internet das coisas (IoT).....	12
3.3	Tecnologias operacionais e tecnologias de informação .....	13
3.3.1	Tecnologias operacionais.....	13
3.3.2	Tecnologias de informação .....	25
3.4	Sumário .....	34
<b>4</b>	<b>Análise de valor</b> .....	<b>36</b>
4.1	Definição da análise de valor.....	36
4.2	Modelo NCD .....	37
4.2.1	Identificação de oportunidade .....	38
4.2.2	Análise de oportunidade .....	38
4.2.3	Geração de ideias .....	39
4.2.4	Seleção da ideia .....	42
4.2.5	Definição da conceção.....	45
4.3	Valor .....	47
4.4	Proposta de Valor .....	48
4.5	Sumário .....	49
<b>5</b>	<b>Arquitetura e desenho da aplicação</b> .....	<b>50</b>
5.1	Descrição geral .....	50
5.1.1	Perspetiva do produto e funcionalidades .....	50
5.1.2	Restrições de implementação e design .....	51

5.1.3	Diagramas de caso de uso.....	51
5.1.4	Stakeholders .....	52
5.2	Requisitos do sistema.....	53
5.3	Processos de negócio .....	54
5.4	Modelo de domínio.....	54
5.5	Requisitos de interface externa.....	55
5.6	Requisitos não funcionais .....	56
5.7	Product Backlog .....	57
5.8	Arquitetura sugerida.....	58
5.9	Desenho da interface <i>front-end</i> .....	62
5.10	Sumário.....	66
<b>6</b>	<b>Desenvolvimento e construção da solução .....</b>	<b>67</b>
6.1	Seleção de Tecnologias.....	67
6.2	Implementação dos requisitos.....	68
6.2.1	Autenticação de utilizadores .....	68
6.2.2	Criação de um novo equipamento .....	71
6.2.3	Visualizar equipamentos existentes .....	74
6.2.4	Criação de uma requisição.....	75
6.3	Sumário.....	78
<b>7</b>	<b>Experimentação e avaliação da solução.....</b>	<b>79</b>
7.1	Abordagem de avaliação .....	79
7.1.1	Hipótese de investigação .....	79
7.1.2	Indicadores e Fontes de Informação.....	80
7.1.3	Metodologias de avaliação .....	80
7.2	Testes realizados ao sistema .....	80
7.2.1	Testes unitários .....	81
7.2.2	Testes de integração .....	82
7.2.3	Testes <i>end 2 end</i> .....	84
7.3	Análise do questionário e resultados obtidos .....	85
7.4	Sumário.....	87
<b>8</b>	<b>Conclusão .....</b>	<b>88</b>
8.1	Contexto e problema .....	88
8.2	Abordagem e solução.....	89
8.3	Objetivos realizados .....	89
8.4	Trabalhos futuros .....	90



# Lista de Figuras

Figura 1 - Folha para requisição de equipamentos laboratoriais.....	5
Figura 2 - Workflow da execução das diferentes tarefas no processo de levantamento de requisitos (Wieggers and Beatty, 2013) .....	7
Figura 3 - Técnicas de elicitação sugeridas por característica do projeto (Wieggers and Beatty, 2013).....	8
Figura 4 - Estrutura hierárquica de uma fábrica na era da Indústria 4.0 (Melzer, 2016).....	11
Figura 5 - Exemplo de um objeto inteligente utilizado para monitorizar a saúde dos trabalhadores de obras nas estradas (Kortuem <i>et al.</i> , 2010).....	13
Figura 6 - Alinhamento das tecnologias de IT e de OT ((Price, 2020)) .....	13
Figura 7 - Classificação de tecnologias sem fio com base no alcance (Ibrahim, 2019) .....	14
Figura 8 - Funcionamento do RFID ((Mendes, 2018)) .....	15
Figura 9 - Papéis de diferentes dispositivos BLE .....	16
Figura 10 - Apple Airtags ( <i>Apple AirTag Silver MX532AM/A</i> , 2021).....	18
Figura 11 - Tecnologias de WAN de baixo consumo (LPWAN) (Ibrahim, 2019).....	18
Figura 12 - Arquitetura LoRaWAN (Ibrahim, 2019).....	19
Figura 13 - Sistema de GPS (National Geographic Society, 2022) .....	20
Figura 14 - Exemplo de tags de código barras (esquerda) e QR code (direita).....	24
Figura 15 - Representação da arquitetura em camadas (Richards, 2017) .....	29
Figura 16 - Representação esquemática de uma arquitetura de microserviços (Richards, 2017) .....	30
Figura 17 - Representação da arquitetura orientada a eventos ( <i>A Quick Guide to Understand the Event-driven Architecture</i> , 2019) .....	30
Figura 18 - O processo de inovação pode ser dividido em 3 partes: <i>Fuzzy Front End</i> , <i>New Product Development</i> e <i>Commercialization</i> (Koen <i>et al.</i> , 2002) .....	37
Figura 19 - Modelo NCD (Koen <i>et al.</i> , 2002).....	37
Figura 20 - Representação esquemática solução 1 .....	40
Figura 21 - Representação esquemática solução 2.....	41
Figura 22 - Árvore hierárquica de decisão .....	42
Figura 23 - Matriz de comparação de critérios (MCC) .....	43
Figura 24 - Vetor Próprio.....	43
Figura 25 – Matriz de Comparação de custo.....	44
Figura 26 - Matriz de comparação de tempo de desenvolvimento .....	44
Figura 27 - Matriz de comparação de automatização do processo .....	45
Figura 28 – Matriz de prioridade (MP) .....	45
Figura 29 – Ranking das soluções.....	45
Figura 30 - Estrutura da matriz House of Quality (Corporation, 2008) .....	46
Figura 31 - House of Quality para a solução proposta desenvolver .....	47
Figura 32 - Proposta de valor segundo modelo de <i>value proposition</i> de Osterwalder.....	49
Figura 33 – Diagrama de casos de uso .....	52
Figura 34 – Modelagem do processo da requisição de equipamento .....	54

Figura 35 - Modelo de domínio da solução.....	55
Figura 36 - Vista física nível 2 .....	59
Figura 37 – Vista lógica de nível 2 .....	60
Figura 38 -Vista lógica de nível 3 – <i>Back-end</i> .....	60
Figura 39 - Vista lógica de nível 3 – Front End .....	61
Figura 40 – Página de login e de boas-vindas .....	63
Figura 41 – Página de Requisições do utilizador (esquerda) e menu lateral (direita) .....	64
Figura 42 – Página de registo de novo equipamento .....	64
Figura 43 – Página da lista de equipamentos .....	65
Figura 44 – Página de criação de requisição .....	65
Figura 45 – Página de Login (esquerda) e de boas-vindas (direita) implementadas .....	69
Figura 46 – Tipos de mensagens de aviso .....	69
Figura 47 – Diagrama de sequência de autenticação .....	70
Figura 48 – Menu principal (esquerda) e página de criação de novo equipamento(direita) ....	71
Figura 49 - Diagrama de sequência de criação de equipamento.....	73
Figura 50 – Página de visualização de equipamentos (vista de <i>desktop</i> ) .....	75
Figura 51 – Cartão do equipamento a requisitar (esquerda) e acessórios deste (direita) .....	76
Figura 52 – Página de requisição de equipamentos (vista <i>desktop</i> ).....	76
Figura 53 – Página com lista de requisições do utilizador (vista <i>desktop</i> ).....	76
Figura 54 - Diagrama de sequência de criação de requisição.....	77
Figura 55 – Teste unitário criado para o <i>Room service</i> (criação de sala).....	82
Figura 56 – Resultados obtidos com testes unitários .....	82
Figura 57 – Estrutura dos testes de integração .....	83
Figura 58 – Execução Teste de integração .....	83
Figura 59 – Estrutura de testes <i>End 2 End</i> .....	84
Figura 60 - Dashboard com resultado da execução dos testes end 2 end .....	85



# Lista de Tabelas

Tabela 1 - Boas práticas da engenharia de requisitos (adaptado (Wieggers and Beatty, 2013)) ..	7
Tabela 2 - Resumo das tecnologias de localização de curto alcance .....	22
Tabela 3 - Resumo das tecnologias de localização de longo alcance .....	23
Tabela 4 - Resumo das principais características do tipo de desenvolvimento de aplicações Móveis.....	28
Tabela 5 - Principais características dos padrões arquiteturais.....	31
Tabela 6 - Escala fundamental - Níveis de importância de comparações (Saaty, 1990) .....	43
Tabela 7 - Valores de IR para matrizes quadradas de ordem n .....	44
Tabela 8 – Relação benefícios e sacrifícios .....	48
Tabela 9 – Perfis dos utilizadores.....	53
Tabela 10 – <i>Product backlog</i> .....	58
Tabela 11 – Tecnologias selecionadas divididas por camadas.....	68
Tabela 12 – Sugestões de melhoria .....	85
Tabela 13 – Relação entre questões, requisitos e fator de cumprimento obtido .....	86
Tabela 14 – Valores obtidos no cálculo da qualidade do sistema desenvolvido .....	87



# Lista de Excertos de código

Excerto de código 1 - Classe de erro desenvolvida .....	70
Excerto de código 2 - Componente de validação de autenticação .....	72
Excerto de código 3 - Função para guardar imagens .....	74
Excerto de código 4 - Função de validação de sobreposição de intervalos temporais .....	78



# Acrónimos e Símbolos

## Lista de Acrónimos

<b>AHP</b>	Analytic Hierarchy Process
<b>AoA</b>	Angle of Arrival
<b>API</b>	Application Programming Interface
<b>BLE</b>	Bluetooth low energy
<b>CPS</b>	Cyber-Physical Systems
<b>CSS</b>	Chirp Spread Spectrum
<b>CSS</b>	Cascading Style Sheets
<b>DRY</b>	Don't Repeat Yourself
<b>DTO</b>	Data Transfer Object
<b>E2E</b>	End to end
<b>FFE</b>	Fuzzy Front End
<b>GPS</b>	Global Positioning System
<b>HOQ</b>	House of Quality
<b>HTML</b>	Hypertext Markup Language
<b>HTTP</b>	Hypertext Transfer Protocol
<b>IC</b>	Índice de consistência
<b>IEEE</b>	Instituto de Engenheiros Eletricistas e Eletrônicos
<b>INEGI</b>	Inovação em Engenharia Mecânica e Engenharia Industrial
<b>IoT</b>	Internet of Things
<b>ISEP</b>	Instituto Superior de Engenharia do Porto
<b>IT</b>	Information Technology
<b>ISO</b>	Organização Internacional de Normalização
<b>JSON</b>	JavaScript Object Notation

<b>LoRa</b>	Long Range
<b>LoRaWAN</b>	Long Range Wide Area Network
<b>LPWAN</b>	Low-Power Wide-Area Network
<b>MCC</b>	Matriz de comparação de critérios
<b>MIT</b>	Massachusetts Institute of Technology
<b>NB-IoT</b>	Narrowband IoT
<b>NCD</b>	New Concept Development
<b>NPD</b>	New Product Development
<b>OCP</b>	Open close principle
<b>OT</b>	Operational Technology
<b>PWAs</b>	Progressive Web Appps
<b>QEF</b>	Qualitative Evaluation Framework
<b>QR code</b>	Quick Response code
<b>RC</b>	Razão de consistência
<b>RFID</b>	Radio frequency identification
<b>RSS</b>	Relative Sinal Strength
<b>ToF</b>	Time of Flight
<b>UWB</b>	Ultrawideband
<b>UX</b>	User experience
<b>WebRTC</b>	Web Real-Time Communications
<b>WLAN</b>	Wireless local area network

## Lista de Símbolos

<b><math>\lambda_{max}</math></b>	Maior valor próprio da matriz A
-----------------------------------	---------------------------------





# 1 Introdução

Este documento contempla a entrega do projeto de dissertação de Mestrado de Engenharia de Software do Instituto Superior de Engenharia do Porto, e tem por objetivo apresentar o estudo e desenvolvimento de uma aplicação para a gestão de utilização de equipamentos laboratoriais de modo a tornar o processo mais eficiente.

## 1.1 Contexto e Problema

O projeto é desenvolvido no Instituto de Ciência e Inovação em Engenharia Mecânica e Engenharia Industrial (INEGI) no concelho do Porto, que tem na sua base um conjunto de unidades especializadas por área científica e tecnológica, suportando atividades de investigação. O INEGI dispõe de várias salas laboratoriais com armazenamento de equipamentos que são essenciais para a realização de diversas atividades no âmbito das Engenharias Mecânica, eletrónica, entre outras. Devido à expansão destas instalações a nível de infraestruturas laboratoriais há necessidade de desenvolver uma solução para melhorar o processo de gestão de requisição dos equipamentos.

Atualmente, para fazer a requisição de um equipamento é usual inserir manualmente a informação numa folha de papel com uma estrutura de recolha de dados pré-estabelecida. É prática corrente não fazer a atualização da requisição quando o equipamento solicitado pelo operador não é devolvido no prazo determinado, seja por esquecimento ou necessidade de prolongamento do tempo de utilização. Além disso, como não é mantido um registo de outras solicitações de equipamentos, não sendo possível saber se determinados equipamentos são utilizados com mais frequência do que outros ou se é necessário adquirir novos. O procedimento existente também não é particularmente adequado para operadores inexperientes que não estão familiarizados com os nomes das ferramentas, o que pode resultar na utilização de nomenclatura incorreta.

## 1.2 Objetivos

O objetivo será assim criar uma solução informática para a gestão de requisição dos equipamentos laboratoriais de modo que os colaboradores consigam realizar as seguintes operações:

- Identificar o equipamento que procuram a partir de informação multimédia ou textual (principalmente útil para utilizadores inexperientes);
- Digitalizar um processo manual atualmente existente e disponibilizar a informação em formato digital de modo que esta possa ser acedida e processada a partir de meios computacionais;
- Conseguir localizar o equipamento (qual a sala e em que contentor está armazenado ou na posse de quem se encontra, no caso de estar requisitado);
- Agendar ou requisitar os equipamentos para determinado intervalo de tempo (sem que haja sobreposição).

Com a solução final espera-se que os colaboradores da Organização obtenham um maior controlo sobre o processo de gestão e requisição dos equipamentos e que esta também permita obter novas informações resultantes do armazenamento do histórico de requisições e seu processamento por meio da aplicação desenvolvida que ajuda a identificar necessidades de manutenção ou aquisição de novos equipamentos.

## 1.3 Abordagem

De modo a atingir os objetivos da tese de mestrado foi efetuado um conjunto de procedimentos, designadamente contextualização do problema e levantamento dos requisitos do cliente; seleção das tecnologias que podem ser utilizadas para a resolução do problema; análise das possíveis soluções idealizadas; apresentação de uma arquitetura e desenho de uma possível solução; desenvolvimento e construção da aplicação e uma metodologia para fazer a sua avaliação. Estes foram os procedimentos realizados para a construção do sistema e explanados neste documento.

## 1.4 Estrutura da dissertação

A dissertação divide-se em 8 capítulos mais anexos, sendo os *Outcomes* abordados do capítulo 2 ao capítulo 7. Estes capítulos são:

- Contexto e análise do problema;
- Estado de arte;
- Análise de valor;
- Arquitetura e desenho da aplicação;
- Desenvolvimento e construção da solução;

- Experimentação, avaliação e implementação da solução.

Cada um destes capítulos tenta dar resposta a um conjunto de objetivos definidos como *Outcomes*, sendo eles:

- *Outcome 1*: Interpretar o problema a resolver, adotando ciências e boas práticas de engenharia informática;
- *Outcome 2*: Sintetizar o conhecimento existente de ciências e boas praticas de engenharia informática para a resolução do problema;
- *Outcome 3*: Avaliar diferentes abordagens para a resolução do problema;
- *Outcome 4*: Desenhar uma solução para o problema adotando ciências e boas práticas de engenharia informática;
- *Outcome 5*: Construir a solução para o problema aplicando ciências e boas práticas de engenharia informática;
- *Outcome 6*: Avaliar a solução desenhada / implementada aplicando ciências e boas práticas de engenharia informática.

Desta forma no capítulo 2 (contexto e análise do problema) é apresentado e discutido o problema recorrendo a técnicas de engenharia de requisitos.

No capítulo 3 (Estado de arte) é feito o levantamento das tecnologias existentes e que podem ser utilizadas para dar resposta ao problema em causa.

No capítulo 4 (Análise de valor) são idealizadas e avaliadas diferentes abordagens para a resolução do problema. É também aplicado o modelo New Concept Development (NCD) para ajudar na concretização da solução.

No capítulo 5 (Arquitetura e desenho a aplicação) é desenvolvida a especificação arquitetural e apresentado o desenho da interface do utilizador para a solução escolhida.

No capítulo 6 (Desenvolvimento e construção da solução) são selecionadas as tecnologias estudadas e apresentado com mais detalhe a forma como foi implementada a solução final.

Por fim, no capítulo 7 (Experimentação, avaliação e implementação da solução) foi estruturada uma metodologia para avaliar a aplicação informática e posta em prática contra a solução desenvolvida.

## 2 Contexto e Análise do problema

Este capítulo fornece uma contextualização e descrição mais aprofundadas do problema para melhor compreensão deste. No final do capítulo espera-se obter resposta para o Outcome 1 – Interpretar o problema a resolver, adotando ciências e boas práticas de engenharia informática.

Para melhor conhecer o problema foram realizadas entrevistas com os elementos da empresa que poderão vir a ser futuros utilizadores da ferramenta. Estas entrevistas foram preparadas antecipadamente com várias questões que tiveram em consideração boas práticas e técnicas de engenharia de requisitos e de engenharia informática. Os resultados das entrevistas realizadas podem ser consultados no capítulo Anexos, sendo este capítulo reservado para a discussão e análise do problema (de modo a não tornar o mesmo demasiado extenso), bem como algumas considerações que foram tidas em conta para a formulação das perguntas.

No final deste capítulo procura-se responder às seguintes questões:

- Qual o contexto em que o problema se insere?
- Qual o problema que se pretende resolver?
- Qual a abordagem adotada para interpretar o problema a resolver?

### 2.1 Contexto

Nos últimos anos, o INEGI tem consolidado a sua posição como parceiro da indústria nacional e internacional em diversas áreas de interesse, estando o desenvolvimento de produtos e sistemas inseridos na área de engenharia industrial e gestão de projetos de I&D. Para a execução dos diferentes projetos e atividades do negócio, a empresa dispõe de várias salas laboratoriais onde se encontram armazenados os equipamentos. Dada as amplas áreas de atividade da empresa (desde a área das engenharias mecânicas até as engenharias eletrónicas, entre outras) os equipamentos que dispõem também apresentam diferentes características e necessidades de armazenamento da sua informação.

Desde o ano passado (2021) que foi possível assistir a um reforço na infraestrutura laboratorial e experimental das instalações da empresa (Seara.com, 2021). Este plano de investimento estratégico iniciado em 2018 e que se estende até final de 2025, irá não só reforçar os meios laboratoriais e experimentais (com a aquisição de novos equipamentos), como também poderá criar a necessidade de admissão de novos colaboradores. Este aumento a nível estrutural implica uma melhor gestão da requisição e utilização dos ativos atuais bem como dos que poderão vir a ser adquiridos no futuro, de modo a tornar as operações mais eficientes (por exemplo, diminuindo o tempo de procura), ter uma maior perceção dos equipamentos existentes e uma melhor organização e gestão dos mesmos.

## 2.2 Problema

Atualmente a requisição de equipamentos laboratoriais é realizada de forma manual por inserção da informação numa folha de papel com uma estrutura de informação pré-definida (a Figura 1 ilustra essa folha afixada num armário do laboratório). Muitas vezes o equipamento que o operador requisitou não é devolvido no intervalo de tempo indicado, quer por esquecimento, quer por necessidade de prolongamento do tempo de utilização do mesmo, sendo que nestas situações é prática comum a folha de registo não ser atualizada.

Para além deste facto, não é guardado um histórico das requisições do equipamento, não permitindo assim, de forma imediata, perceber se há equipamentos que são mais requisitados que outros e detetar a eventual necessidade de aquisição de novos. O processo atual é, ainda, pouco amigável para operadores inexperientes e que desconhecem o nome das ferramentas, podendo levar a nomear ferramentas de forma errónea. Este procedimento de requisição do equipamento é por si só ineficiente e não escalonável.

ID Equipamento	Data Requisição	Data Receção	Responsável	Data Entrega	Data Devolução	Distribuição/Projeto
CONECTOR	2/01/2022	17/25		01/05	17/31	LADDA
DIAPHRAGMA	01/04	21/00	ALICE			LADDA
SACOS COMPACTOS			ALICE			
SACOS COMPACTOS			ALICE			
FORÇA	11/05		ALICE	11/05		ALICE
EXPANSÃO	17/05	17/00	ALICE	17/05	18/00	ESCALDO
EXPANSÃO						
EXPANSÃO	24/05	24/00	ALICE	24/05	24/00	LADDA
EXPANSÃO	28/06	28/00	ALICE	28/06		
EXPANSÃO	28/06	28/00	ALICE	28/06	17/15	
EXPANSÃO						
EXPANSÃO	14/07	14/00	ALICE	14/07	14/00	ALICE
EXPANSÃO	14/07	14/00	ALICE	14/07	14/00	ALICE
EXPANSÃO	14/07	14/00	ALICE	14/07	14/00	ALICE

Figura 1 - Folha para requisição de equipamentos laboratoriais

## 2.3 Abordagem ao problema

De modo a obter uma melhor interpretação do problema e dos requisitos necessários para desenvolver uma possível solução, foram utilizadas técnicas de engenharia de requisitos.

Do ponto de vista do processo de software, a engenharia de requisitos é uma ação importante da engenharia de software que começa durante a atividade de comunicação e continua durante a atividade de desenvolvimento (Rocha, 2022). Deve ser adaptado às necessidades do processo, do projeto, do produto e das pessoas que realizam o trabalho. Segundo Fred Brooks *“The hardest single part of building a software system is deciding what to build. No part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later.”*(Brooks, 1986). Sendo a engenharia de requisitos a primeira atividade do ciclo de desenvolvimento de software é importante que esta seja realizada com rigor.

A engenharia de requisitos permite identificar as necessidades do cliente, negociar a solução a implementar e perceber o retorno versus o custo da implementação da solução. Para tal existe um conjunto de quatro tarefas distintas que ajudam na execução deste processo, sendo elas (Bourque, Fairley, and IEEE Computer Society, 2014):

- **Elicitação:** esta fase engloba todas as atividades envolvidas na descoberta de requisitos, como entrevistas, workshops, análise de documentos, prototipagem e outros. Como resultados desta análise obtém-se a identificação dos principais stakeholders (sendo estes definidos como qualquer parte interessada no projeto, como por exemplo futuros utilizadores da aplicação e seus desenvolvedores), uma maior compreensão do modo de execução das tarefas pelo cliente, do ambiente em que o produto vai ser utilizado e perceber as necessidades, funcionalidades e expectativas de cada classe de utilizador.
- **Análise:** esta etapa envolve obter uma melhor compreensão de cada requisito e representar estes a partir de várias vistas. Isto envolve analisar as informações recebidas dos diferentes utilizadores para distinguir os seus objetivos de tarefas que o produto deve realizar, dos requisitos funcionais, expectativas, regras de negócio, sugestões de soluções e outras informações (identificando lacunas nos requisitos ou requisitos desnecessários no que se refere ao escopo definido, podendo ser necessário negociar prioridades de implementação).
- **Especificação:** esta envolve representar e armazenar o conhecimento de requisitos obtidos da fase anterior de uma maneira persistente e bem organizada. A principal atividade é então traduzir as necessidades do futuro utilizador em requisitos e diagramas adequados para a compreensão, revisão e uso do seu público-alvo.
- **Validação:** esta é a última fase que consiste em confirmar o conjunto de informações e de requisitos que irão permitir fazer a implementação da solução que satisfaça os objetivos de negócio pela equipa de desenvolvedores. Nesta fase os requisitos documentados devem ser revistos para corrigir quaisquer erros antes que a equipa de desenvolvimento comece a implementar. Podem também ser desenvolvidos testes e critérios de aceitação para confirmar se o produto baseado nos requisitos responde as necessidades do cliente.

Importante salientar que o objetivo do desenvolvimento de requisitos é obter um entendimento compartilhado destes e que seja bom o suficiente para permitir a construção de todo o projeto. Se o levantamento de requisitos não for bem realizado pode levar à necessidade de refazer trabalho que não estava previamente planejado.

As tarefas descritas anteriormente não têm de seguir necessariamente uma sequência linear de execução. Na prática, estas são entrelaçadas, iterativas e incrementais (tal como ilustra a Figura 2).

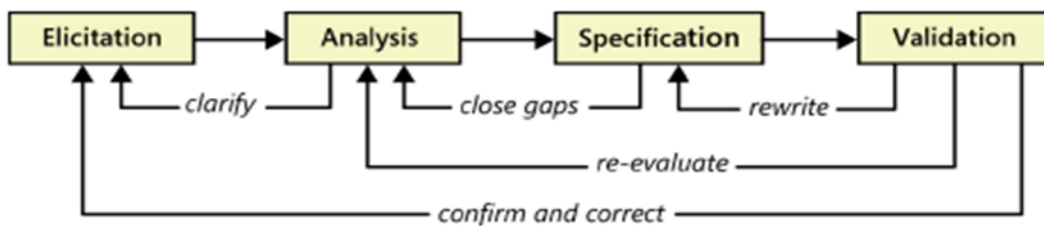


Figura 2 - Workflow da execução das diferentes tarefas no processo de levantamento de requisitos (Wieggers and Beatty, 2013)

Para o levantamento de requisitos realizado foi possível executar cada uma destas tarefas pelo menos uma vez, sendo o resultado obtido o documento de requisitos apresentado nas secções seguintes. As tarefas tiveram também em consideração algumas das boas praticas existentes para o levantamento de requisitos, expostas na Tabela 1.

Tabela 1 - Boas práticas da engenharia de requisitos (adaptado (Wieggers and Beatty, 2013))

<b>Elicitação</b>	<b>Análise</b>	<b>Especificação</b>	<b>Validação</b>
Identificar classes de utilizadores	Analisar a viabilidade dos requisitos	Adotar templates de documentos de requisitos	Rever os requisitos
Realizar entrevistas de Elicitação	Priorizar os requisitos principais	Identificar origem dos requisitos	Simular os requisitos
Identificar requisitos dos utilizadores	Realizar análise dos documentos	Identificar de forma única cada requisito	Testar requisitos
Identificar eventos e respostas do sistema	Identificar o ambiente de implementação do produto	Identificar regras de negócio	
Preparar documento de entrevista		Especificar requisitos não funcionais	
Reutilizar requisitos existentes			

### 2.3.1 Processo de elicitação

As sessões de um processo de elicitação envolvem várias etapas (desde a preparação, execução da elicitação e posterior acompanhamento) que não irão ser tratadas no presente documento visto sair do escopo deste. Para executar o processo de elicitação existem várias técnicas especificadas na literatura que ajudam a identificar as funcionalidades necessárias (por vezes não reconhecidas diretamente pelos futuros utilizadores finais). A Figura 3 ilustra qual a técnica de elicitação que deverá ser utilizada tendo em conta as características do projeto em causa.

	Interviews	Workshops	Focus groups	Observations	Questionnaires	System interface analysis	User interface analysis	Document analysis
Mass-market software	x		x		x			
Internal corporate software	x	x	x	x		x		x
Replacing existing system	x	x		x		x	x	x
Enhancing existing system	x	x				x	x	x
New application	x	x				x		
Packaged software implementation	x	x		x		x		x
Embedded systems	x	x				x		x
Geographically distributed stakeholders	x	x			x			

Figura 3 - Técnicas de elicitação sugeridas por característica do projeto (Wieggers and Beatty, 2013)

A técnica selecionada para realizar o processo de Elicitação foi a entrevista, dado tratar-se do desenvolvimento de uma nova aplicação informática, melhorar um sistema existente e ser o mecanismo mais ágil para obter respostas por parte dos utilizadores. De modo a estas serem mais eficazes na obtenção dos requisitos de desenvolvimento foram elaborados questionários para tornar a entrevista orientada. Estes questionários foram elaborados dando preferências a questões do tipo aberto (de modo a dar mais liberdade e desenvolvimento na resposta e obter uma maior quantidade de informação para posteriormente traduzir em requisitos) tendo em conta o perfil de cada um dos entrevistados. Foram consideradas ainda as boas práticas de desenvolvimento de um questionário, tais como as descritas em seguida (Dawson, 2007):

- Certificar que as perguntas não contêm algum tipo de preconceito de prestígio (selecionando as perguntas consoante o perfil do entrevistado);
- Evitar perguntas que induzam a resposta;
- Evitar perguntas com denotação negativa;
- Evitar palavras vagas como “frequentemente” e “às vezes”.

As respostas dos entrevistados foram registadas em anotações. Este método apesar de tornar difícil manter o foco nas respostas dadas e procurar mais informações, não necessita de nenhum equipamento para realizar a gravação e os entrevistados podem dar um feedback

imediatamente à medida que as notas estão a ser registadas. Em anexo encontram-se as cinco entrevistas realizadas (representativas dos principais utilizadores, com 3 perfis diferentes) e as respostas registadas durante o processo de elicitação desenvolvido. O resultado deste trabalho será a especificação de requisitos apresentado no capítulo 5 - Arquitetura e desenho da aplicação.

## **2.4 Sumário**

Neste capítulo foi apresentado o contexto em que o problema em causa se insere, uma apresentação detalhada do mesmo e qual a abordagem adotada para o tentar resolver.

Como proposto na introdução do capítulo, é possível agora responder a cada uma das questões expostas.

### **Qual o contexto em que o problema se insere?**

A instituição INEGI possui múltiplas salas com múltiplos equipamentos de diferentes áreas e com diferentes características. É também apresentado um cenário de crescimento quer a nível estrutural, quer a nível da necessidade de adquirir novos equipamentos (e até a nível de integração de novos colaboradores) que irão ter necessidade de uma melhor gestão e organização.

### **Qual o problema que se pretende resolver?**

O problema reside na falta de informação consistente, fiável e de rápida obtenção sobre o utilizador e a localização do equipamento requisitado. Há também a perceção que num futuro próximo o processo atualmente implementado irá tornar-se obsoleto e menos eficaz no registo da informação. A digitalização e automatização do processo atualmente implementado poderão ajudar na resolução deste problema e aumentar a sua eficiência.

### **Qual a abordagem adotada para interpretar o problema a resolver?**

De modo a interpretar melhor o problema foram adotadas técnicas da engenharia de requisitos (nomeadamente o recurso a entrevistas preparadas antecipadamente) para poder obter o maior conjunto de informação possível. Esta informação irá ser utilizada para construir os requisitos que serão a base do desenvolvimento da solução informática

## 3 Estado de arte

Tendo obtido uma melhor visão sobre o problema é necessário sintetizar o conhecimento existente para proceder à sua resolução. O objetivo deste capítulo é então responder ao *Outcome 2* – resumir o conhecimento científico e as boas práticas de engenharia informática para a resolução do problema. Para tal, neste capítulo é feita uma pequena introdução à digitalização dos processos e à Indústria 4.0 sendo em seguida analisadas tecnologias existentes que podem auxiliar na construção da solução.

No final deste capítulo procura-se responder às seguintes perguntas:

- Que tecnologias operacionais existem que podem auxiliar na localização de equipamentos?
- Como podem as tecnologias de informação ajudar no processo de localização dos equipamentos?

### 3.1 Digitalização dos processos

O processo de digitalização de uma tarefa consiste na conversão de uma atividade geradora de informação para o formato digital de modo que esta possa ser acedida e processada a partir de meios computacionais. As organizações podem ver esta transformação digital como uma forma de aumentar níveis de eficiência nas suas operações e produtos através da integração de novas tecnologias impulsionadas pelos novos desafios e necessidades emergentes levando à adaptação dos modelos de negócio (Ebert and Duarte, 2018). Esta mudança e adaptação pode ser mais ou menos profunda dependendo do modelo de negócio da empresa e do processo evolutivo que a mesma sofre com o passar dos anos.

A abordagem para implementar o processo de digitalização deve ter em conta o contexto da empresa e deve ser orientada aos desafios específicos desta. É por isso fundamental identificar os objetivos específicos a atingir no final do processo de transformação digital, reconhecendo quais as grandezas relevantes, qual o grau de maturidade da entidade relativamente à

digitalização dos processos e gestão da informação, determinar os recursos e competências necessárias. Após esta sequência é necessário fazer a implementação do ecossistema digital de forma horizontal (dando mais autonomia aos funcionários para tomar decisões) ou vertical (no qual as funções e cargos estão limitados e bem definidos) (*O que é gestão horizontal e vertical e suas principais características*, 2019) conforme o que melhor se adequar à organização.

Cabe assim à instituição verificar os meios tecnológicos que dispõe para implementar a estratégia de digitalização definida. Este processo de transformação digital e de utilização dos sistemas de informação para fazer a interligação das operações é referenciado como a quarta revolução industrial ou indústria 4.0

### 3.2 Indústria 4.0

Graças aos avanços tecnológicos que se tem observado nos últimos anos, foi possível desenvolver um conjunto de ferramentas que não só mudam a forma como se vive, como também permitem criar novos tipos de negócios e novas formas de realizar as tarefas do dia a dia (acelerando os ciclos de inovação, aumentando a produtividade e reduzindo riscos). A Figura 4 ilustra a estrutura hierárquica típica de uma fábrica na era da indústria 4.0.

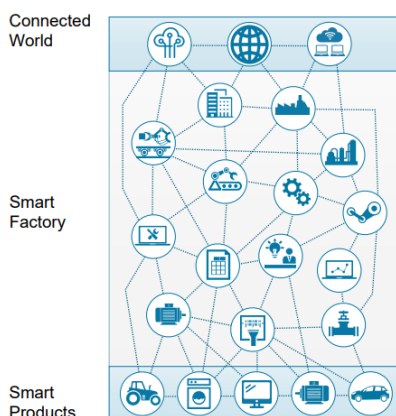


Figura 4 - Estrutura hierárquica de uma fábrica na era da Indústria 4.0 (Melzer, 2016)

A aplicação dos conceitos da indústria 4.0 trazem algumas alterações face aos conceitos anteriores, sendo eles:

- Sistemas e máquinas mais adaptáveis;
- Funções distribuídas por toda a rede;
- Maior comunicação e troca de dados com todos os níveis hierárquicos (permitindo tomadas de decisão mais rápidas).

O principal foco da indústria 4.0 é a melhoria contínua dos processos em termos de eficiência, segurança, produtividade das operações e especialmente no retorno do investimento. São várias as tecnologias e tendências facilitadoras disponíveis e que permitem criar conceitos

como a Internet das coisas (*IoT – Internet of Things*), ou os sistemas ciber físicos (*CPS – cyber physical systems*). Antes de abordar que tecnologias podem ser utilizadas no desenvolvimento do sistema, vai ser esclarecido o que se entende por IoT e como este se interliga com o projeto a desenvolver

### **3.2.1 Internet das coisas (IoT)**

A IoT é utilizada para reunir, sincronizar, organizar e classificar dados de diferentes fontes dentro de uma fábrica, organização ou empresa. Os dados fornecidos pela interação entre os diferentes objetos permitem que qualquer tipo de processo seja conduzido de uma forma mais controlada e recolher um maior número de dados. Daqui resultam análises mais detalhadas e tomadas decisão mais dinâmicas e eficazes.

Assim, a expressão “*Internet of Things*” refere-se a um sistema de identificação física e virtual que está ligado à internet. O termo teve origem no MIT (*Massachusetts Institute of Technology*), onde um grupo realizou um trabalho na área da identificação de objeto por radio frequência (*RFID - Radio Frequency Identification*) em 1999. A maior miniaturização e a redução dos custos dos componentes eletrónicos permitiram expandir a Internet para uma nova dimensão, no qual os objetos quotidianos são aprimorados pela adição de um pequeno dispositivo eletrónico que fornece inteligência e conectividade ao ciberespaço estabelecido pela Internet.

Um *smart object* será assim um sistema ciberfísico (com componentes físicos e de software) ou um sistema embutido (sistema de processamento do objeto onde reside o código para ser executado) constituído pela entidade física (o objeto) e um componente que armazene e/ou processe informação e que permita a comunicação desta com a internet (normalmente por uma comunicação sem fios) (Kopetz, 2011). Um exemplo de um *smart object* utilizado na indústria petroquímica (mais precisamente pelos trabalhadores na construção de estradas) pode ser observado na Figura 5. Este *smart object* é utilizado para monitorizar informações de saúde dos trabalhadores expostos a vibrações intensas e prolongadas.

Considerando a necessidade de obter informação e localização dos equipamentos a monitorizar e não possuindo estes quaisquer tipo de sistema embutido que os tornem “inteligentes”, é importante perceber que tecnologias operacionais (*OT – operational technologies*) e tecnologias de informação (*IT- information technologies*) podem ser utilizadas para permitir responder as necessidades do projeto. Na próxima secção vão ser abordados estes dois temas.



Figura 5 - Exemplo de um objeto inteligente utilizado para monitorizar a saúde dos trabalhadores de obras nas estradas (Kortuem *et al.*, 2010)

### 3.3 Tecnologias operacionais e tecnologias de informação

O aparecimento da quarta revolução industrial, juntamente com a transformação digital e o IoT levou as empresas de vários setores a repensar alguns dos seus modelos de negócio. Como consequência dessa análise, setores que se encontravam previamente isolados, como as IT e as OT começaram a convergir. A Figura 6 ilustra a convergência entre estas duas tecnologias e como a colaboração entre ambas é fundamental para monitorizar e regular processos de negócio. Nesta secção vão ser retratadas cada uma delas e apresentadas algumas tecnologias que podem ajudar no processo de requisição de equipamentos.

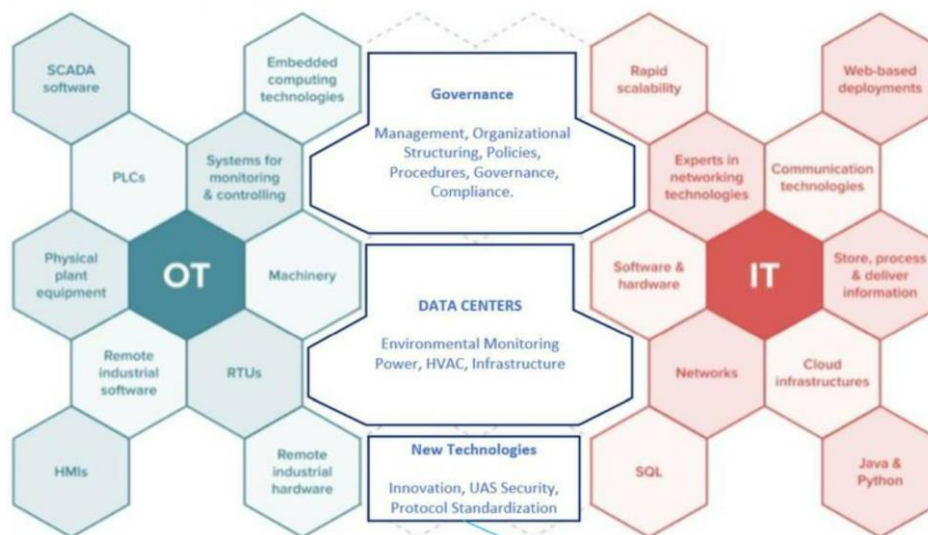


Figura 6 - Alinhamento das tecnologias de IT e de OT ((Price, 2020))

#### 3.3.1 Tecnologias operacionais

As tecnologias operacionais são hardware e software que detetam ou causam uma mudança, por meio da monitorização e/ou controlo direto dos ativos, equipamentos industriais, e processos (*Definition of Operational Technology (OT) - Gartner Information Technology Glossary, 2022*). As tecnologias operacionais permitem preparar os equipamentos físicos e fazer uma

melhor gestão dos dados provenientes destes. São normalmente caracterizados por trabalharem em ambiente fabril ou laboratorial, orientados para a engenharia e componentes de hardware com longos ciclos de vida útil, onde os requisitos do utilizador não são tão dinâmicos, havendo normalmente prioridade para a transmissão dos dados em tempo real.

Para identificar e localizar os equipamentos é necessário reconhecer que tecnologias operacionais podem ser utilizadas. Estas podem ser divididas em duas categorias: tecnologias de localização de equipamentos de curto alcance e de longo alcance (classificadas pelo alcance na Figura 7). Nas subsecções seguintes serão apresentadas sucintamente cada uma destas tecnologias.

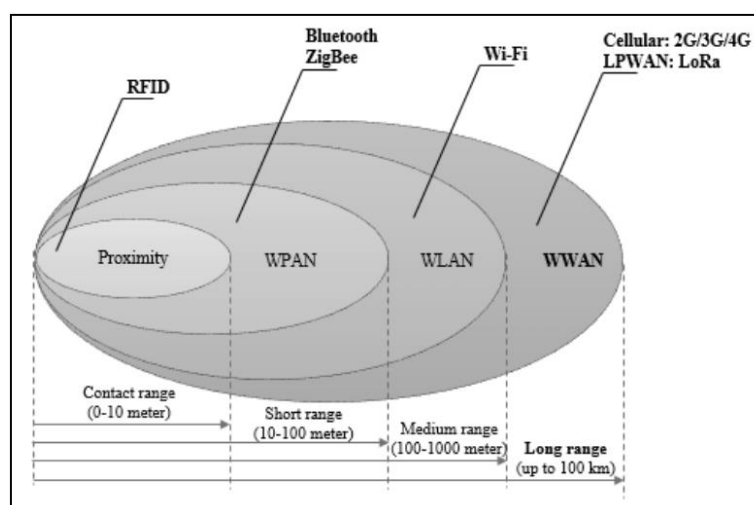


Figura 7 - Classificação de tecnologias sem fio com base no alcance (Ibrahim, 2019)

### 3.3.1.1 RFID

A tecnologia de RFID (*Radio Frequency-identification*) é um termo genérico para tecnologias que utilizam frequência de rádio para realizar comunicação. Esta tecnologia utiliza etiquetas eletrónicas (*tags de RFID*) que emitem sinais de rádio frequência que são captadas por leitores de RFID a curtas distâncias permitindo a leitura e/ou escrita de dados. O leitor de RFID não necessita de uma linha de visão direta para a *tag* de RFID. A *tag* de RFID armazena o código eletrónico de produto (EPC) exclusivo ao objeto anexado. A padronização da tecnologia de RFID pelo *International Standard Organization* (ISO) e a sua adoção em massa, permitiu reduzir significativamente o seu custo nos últimos anos (Kopetz, 2011). Os elementos necessários para a implementação de um sistema de RFID encontram-se representados na Figura 8 (etiquetas de RFID, antenas, leitores de RFID e um computador com software adequado).

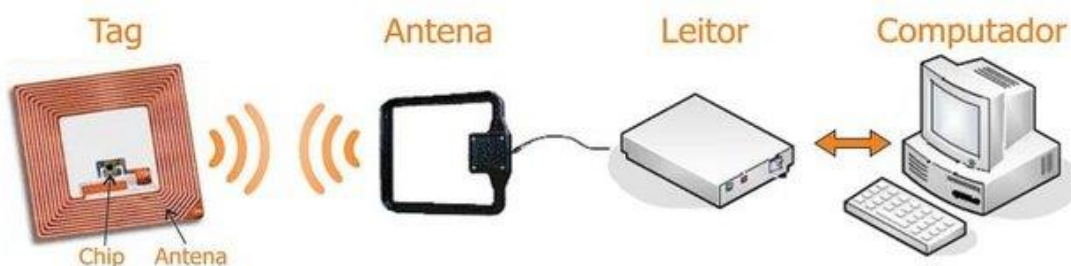


Figura 8 - Funcionamento do RFID ((Mendes, 2018))

Tendo em conta o sistema a desenvolver, a *tag* selecionada para o sistema poderá ser de leitura ou leitura e escrita. As *tags* de leitura apenas permitem acesso aos dados que já estão na *tag*, enquanto as *tags* de leitura e escrita permitem leitura e escrita de novos dados.

O tipo de etiquetas disponíveis tem vindo a ser padronizado e desenvolvido, podendo este cair em duas categorias: *tags* de RFID ativas e passivas.

As *tags* de RFID passivas não possuem uma fonte de alimentação própria. Elas obtêm a energia necessária para operação do campo elétrico criado pelo leitor de RFID (normalmente inferior a 30  $\mu\text{W}$ ).

As *tags* de RFID ativas têm a sua própria fonte de alimentação integrada, por exemplo, uma bateria, que lhes dá capacidade de suportar muito mais serviços do que as *tags* passivas. Normalmente este tipo de *tags* tem autonomia de 1 ano e podem receber e transmitir a uma distância maior do que *tags* passivas (normalmente na ordem das dezenas de metros), podem ter sensores para monitorizar o seu ambiente (temperatura, pressão) e podem suportar protocolos de comunicação padrão da Internet.

Os leitores de RFID são responsáveis por obter a informação da etiqueta e transmiti-la a um computador. Dependendo do tipo de aplicação as frequências de RFID estas podem ser divididas em 4 faixas de frequências diferentes (U.Farooq *et al.*, 2015):

- Baixa frequência (135 KHz ou menos);
- Alta frequência (13,6 MHz);
- Frequência ultra alta (862 MHz a 928 MHz);
- Frequências Micro-ondas (*Microwave frequency*) (2,4GHz a 5,80 GHz).

### 3.3.1.2 Bluetooth low energy (BLE)

O Bluetooth low energy (BLE) começou como parte da especificação Bluetooth 4.0 e foi apresentado como uma versão mais otimizada do Bluetooth (tendo o BLE sido projetado para servir como uma estrutura para troca de dados). O objetivo foi desenvolver um padrão de rádio com o menor consumo de energia possível, otimizado para baixo custo, baixa largura de banda, baixo consumo de energia, baixa complexidade e comunicação de curto alcance (é possível um

dispositivo BLE transmitir dados até 30 metros na linha de visão, mas um alcance típico de operação é normalmente de 2 a 5 metros).

Dispositivos BLE podem ter 2 papéis diferentes (Anne, 2019):

- Dispositivos centrais (ou mestres): geralmente computadores ou smartphones com maior poder de processamento;
- Dispositivos periféricos (ou escravos): sensores ou dispositivos de baixa frequência que se conectam ao dispositivo central.



Figura 9 - Papéis de diferentes dispositivos BLE

Os dispositivos com este tipo de tecnologia podem comunicar de duas formas: transmissão, no qual os envios de pequenas quantidades de informação em períodos fixos de tempo são enviadas para múltiplos dispositivos que estejam a “ouvir” (*one way direction*); ou conexões no qual o envio de informação é feita de forma periódica em pontos específicos de tempo apenas entre um dispositivo escravo e um dispositivo mestre (podendo haver transmissão de dados em ambas as direções). Este protocolo opera na banda de frequência dos 2.4GHz.

### 3.3.1.3 Wi-Fi

Wi-Fi é uma tecnologia de comunicação sem fio de baixa potência que usa ondas de rádio. Esta pode ser encontrada numa variedade de aparelhos modernos, incluindo telefones, computadores pessoais, entre outros. Os utilizadores só podem conectar-se a essas redes se estiverem próximos de um roteador. A tecnologia Wi-Fi transmite mais de 60% do tráfego mundial, tendo por isso uma grande difusão (*Here's How Wi-Fi Actually Works*, 2022). O fato da transferência de dados ser bidirecional é uma característica importante dessa tecnologia, que é muito popular em aplicações de rede que fornecem mobilidade sem fio.

A Wi-Fi enquadra-se na categoria das WLAN (Wireless Local Area Network), tendo um alcance de até 100 metros e é um protocolo de comunicação baseado na norma IEEE 802.11. Esta opera em frequências que não exigem licença (entre os 2,4 GHz e os 5GHz). Os dados enviados por ondas rádio são suscetíveis à interferência de outros sinais de Wi-Fi, obstruções como paredes ou ondas de rádio geradas por outros dispositivos. Quanto mais baixa for a frequência de transmissão maior será o alcance do sinal (as frequências de 2,4GHz terão maior alcance que as frequências de 5GHz).

Existem atualmente 6 tipos de tecnologias Wi-Fi (ordenadas cronologicamente): Wi-Fi-802.11a, Wi-Fi-802.11b, Wi-Fi-802.11g, Wi-Fi-802.11n, Wi-Fi-802.11ac e Wi-Fi-802.11ax (ou Wi-Fi 6). As principais diferenças entre cada uma das versões são o aumento gradual da largura de banda (de 11 Mbits/s na versão Wi-Fi-802.11a até 2400 Mbits/s na versão Wi-Fi-802.11ax) (*Different Wi-Fi Protocols and Data Rates*, 2021).

O sistema de posicionamento Wi-Fi (WiPS – *Wi-Fi Positioning system*) baseia-se na medição da intensidade do sinal recebido (RSS – *Received Signal Strength*) nos pontos de acesso e no método de *fingerprinting*. Parâmetros tipicamente utilizados para geolocalizar o ponto de acesso Wi-Fi ou ponto de acesso sem fio incluem o SSID (*Service Set Identifier*) e o endereço MAC (*Media Access Control address*) do ponto de acesso. A precisão depende do número de posições que foram inseridas na base de dados. As possíveis flutuações de sinal que ocorrem podem aumentar os erros e imprecisões na localização do utilizador.

#### 3.3.1.4 Ultra-wide-band

Assim como o Bluetooth e o Wi-Fi, a ultra-wide band (UWB) é um protocolo de comunicação desenvolvido pela IEEE (802.15.3) sem fio de curto alcance que opera por meio de ondas de rádio. Ao contrário das suas contrapartes, opera em frequências muito altas (dentro de um espectro entre 3.1 e 10.6 GHz, longe de bandas congestionadas em torno dos 2.4GHz, permitindo-lhe coexistir com outras tecnologias sem fios como o Wi-Fi, Bluetooth e NFC) e pode ser usado para capturar dados espaciais e direcionais altamente precisos (precisão essa que as tecnologias Wi-Fi e Bluetooth carecem).

De modo a conseguir ter uma precisão alta, o UWB usa uma largura de banda mais ampla (500 MHz) e pulsos breves (dois nanossegundos cada). Os movimentos do dispositivo são rastreados em tempo real através do procedimento de posicionamento UWB. Os dispositivos com capacidade de usar a UWB podem perceber movimento e localização relativa. Quando o dispositivo com capacidade de usar UWB (como um *smartphone* ou uma *tile*) se encontram próximos um do outro começam o processo de *ranging*. Este refere-se ao cálculo de tempo de voo entre os 2 dispositivos (no capítulo Características físicas dos sinais é explicado com mais pormenor este processo).

Um exemplo da aplicação deste tipo de tecnologia são as *AirTags* da *Apple* (Figura 10) sendo por isso uma tecnologia relativamente recente.

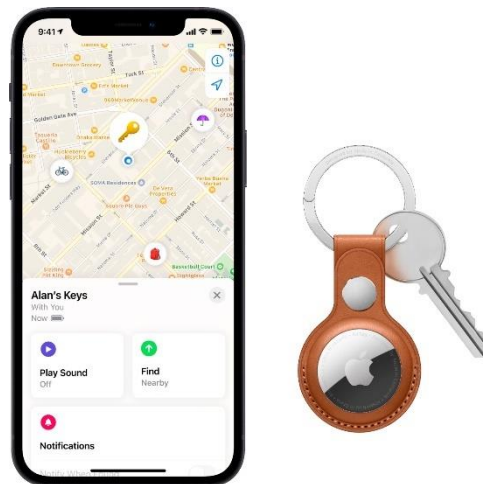


Figura 10 - Apple Airtags (Apple AirTag Silver MX532AM/A, 2021)

### 3.3.1.5 LPWAN (LoRa, LTE\_M, NB-IoT)

Uma rede de área ampla e baixa potência (LPWAN – Low-Power Wide-Area Network) é um tipo de rede de longa distância de telecomunicações sem fio que permite comunicações de longo alcance a uma taxa de bits baixa entre itens conectados, como sensores alimentados por uma bateria. Esta rede difere de uma WAN (Wide Area Network) sem fio, pois foi projetada para conectar utilizadores ou empresas utilizando baixa potência e a uma baixa taxa de bits/s, ao contrário das WAN que transportam mais dados e exigem mais energia. A taxa de dados de um canal LPWAN pode variar de 0,3 a 50 kbit/s (Neumann, Montavont and Noel, 2016). A Figura 11 ilustra tecnologias que utilizam este tipo de rede.

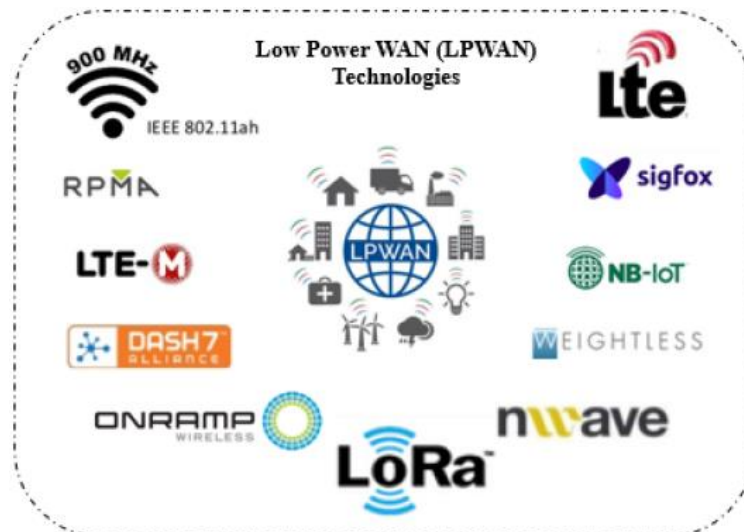


Figura 11 - Tecnologias de WAN de baixo consumo (LPWAN) (Ibrahim, 2019)

A tecnologia LPWAN pode ser dividida em duas categorias (Ibrahim, 2019): *Long Range* (LoRa) e *narrowband* (NB-IoT). LoRa não é simplesmente uma rede de longo alcance; também permite a transferência de dados a uma taxa baixa e com consumo mínimo de energia. Além disso, a rede de área sem fio LoRa (LoRaWAN) é uma tecnologia promissora desenvolvida para resolver

os problemas de aplicações de IoT. A implementação de uma rede LoRaWAN é baseada na topologia de rede em estrela, tendo esta como principal benefício preservar a vida útil da bateria e diminuir a complexidade da rede (cada nó apenas propaga e recebe informação relativa a ele próprio, gerida por um elemento central). A arquitetura LoRaWAN encontra-se representada na Figura 12.

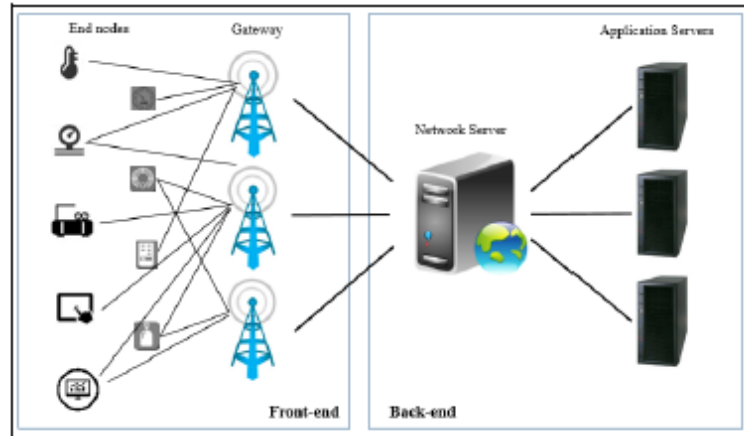


Figura 12 - Arquitetura LoRaWAN (Ibrahim, 2019)

A tecnologia LoRa, num estudo desenvolvido pela Universidade de North Carolina em 2019 (Islam *et al.*, 2019), foi comparada com a tecnologia Wi-Fi e BLE. Esta demonstrou ser mais permeável e resiliente a artefactos ambientais internos (como a presença de paredes, tetos e obstáculos móveis e imóveis) do que as outras duas tecnologias devido às inferiores bandas de frequências de operação (868 MHz na Europa (Master, 2022)). Para além disso, LoRa usa modulação *Chirp Spread Spectrum (CSS)* que diminui o efeito de *multipath* (devido a reflexões e refrações). Estas características tornam a tecnologia LoRa um bom candidato para localização interna.

### 3.3.1.6 GPS

O sistema de posicionamento global (GPS) é um sistema mundial de radionavegação por satélite constituído por uma constelação de 24 satélites e as suas estações terrestres. Este é utilizado para determinar a posição geodésica e altitude na superfície da terra (Wells, 1987). Cada satélite tem um conjunto de relógios atómicos (com precisão de nanossegundos) que são mantidos em sincronia com os relógios atómicos terrestres e um transmissor que envia o sinal para a terra. Este sinal é percecionado por um recetor de GPS que processa os sinais dos satélites (compara a hora de receção do sinal com a hora de emissão e calcula o tempo que o sinal levou para viajar – *Time of Flight*) para determinar a posição, velocidade e tempo do utilizador (sendo por isso um processo computacionalmente intensivo). Este processo para obter a localização na terra necessita de 3 satélites e está representado na Figura 13.

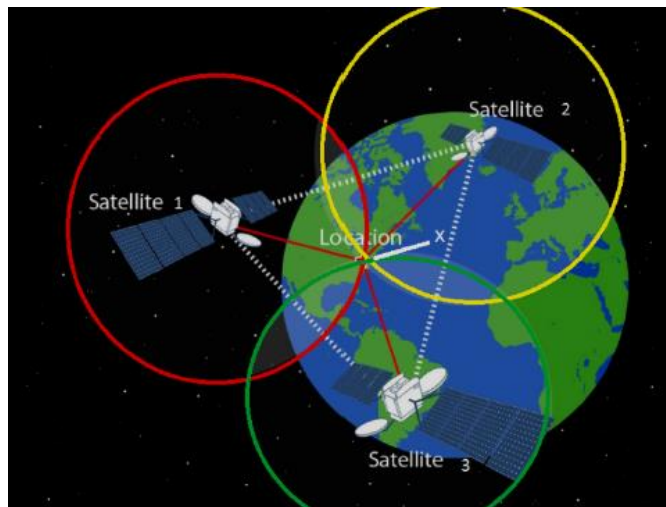


Figura 13 - Sistema de GPS (National Geographic Society, 2022)

Um dos problemas que este sistema levanta é que as velocidades das ondas de radio não são constantes (devido a obstruções, como edifícios, túneis que podem enfraquecer ou até bloquear o sinal) levando a imprecisões nas medições da posição. Outro problema está relacionado com o recetor receber 2 sinais do mesmo satélite (um sinal direto e um sinal refletido) devido a reflexões do sinal em prédios (*multipath*). Dado que os 2 sinais percorrem distâncias diferentes o recetor calcula distâncias diferentes para o mesmo satélite (podendo até haver cancelamentos de sinal). Podem também ocorrer erros associados a imprecisões no sincronismo entre os relógios atômicos e o relógio do utilizador (sendo utilizado um quarto satélite para diminuir este tipo de erros).

Normalmente este tipo de tecnologia é apenas utilizada em locais ao ar livre pois o sinal de GPS tem dificuldade em viajar através de edifícios.

### 3.3.1.7 Cellular Network

As redes móveis (*cellular networks*) habilitam dispositivos como *smartphones* da capacidade de ter uma rede de comunicação de voz e dados de alta velocidade e alta capacidade com recursos avançados de multimídia e roaming contínuo.

A utilização de telemóveis está muito difundida podendo estes serem utilizados para realizar o processo de localização. A primeira alternativa seria utilizar tecnologia GPS, mas esta nem sempre se encontra disponível em todos os dispositivos e o sinal pode ser bloqueado por obstruções físicas e consome muita energia. O recurso a redes moveis permite cobrir melhor estas situações, com maior eficiência energética e independente das capacidades dos dispositivos (visto que qualquer telemóvel tem rede móvel). Podem utilizar-se técnicas como o *Cell ID* onde a localização é estimada com base na longitude e latitude da célula de rede mais forte (as células são equipamentos responsáveis por transmitir as ondas radio numa área geográfica). A estimativa da localização resultante deste tipo de processo é pouco precisa. Outras abordagens como o *fingerprinting* fornecem uma maior precisão, mas exigem uma maior recolha de dados e por isso é mais demorado. Uma vez que a distância é inversamente proporcional à intensidade do sinal recebido (RSS) pode usar-se esta abordagem para localizar

o utilizador (quanto mais próximo da célula de rede, mais forte é o sinal). A combinação destas técnicas permite obter a localização do utilizador com recurso a rede móvel do telemóvel (Elbakly and Youssef, 2019).

#### 3.3.1.8 Características físicas dos sinais

Muitas das tecnologias aqui apresentadas utilizam características do sinal transmitido para estimar a localização de um terminal relativamente a outro. Existem três propriedades básicas que permitem a medição de distância (definida com a determinação do raio de um círculo ou de uma esfera) e localização (ponto no espaço com coordenadas definidas) a partir da análise de características físicas específicas de sinais de rádio:

- Intensidade do sinal recebido (RSS): A potência de uma onda eletromagnética é proporcional à potência transmitida e inversamente proporcional ao quadrado da distância até à fonte;
- Tempo de voo (TOF): distância entre um transmissor e um recetor é igual ao tempo de voo. A distância pode ser determinada a partir da medição do tempo de chegada (TOA – *Time of Arrival*) de um sinal ao recetor (quando o tempo de transmissão é conhecido);
- Ângulo de chegada (AOA – *angle of arrival*) ou direção de chegada (DOA – *direction of arrival*): Este método estima a localização do alvo a partir da interseção dos vários pares de linhas de direção do ângulo.

#### 3.3.1.9 Resumo e considerações sobre escolha de tecnologia

A Tabela 2 e Tabela 3 resume cada uma das tecnologias apresentadas.

Tabela 2 - Resumo das tecnologias de localização de curto alcance

Tecnologia operacional	Princípio de funcionamento	Pros	Contras
RFID	Tags de RFID emitem sinais de radio frequência que são captadas por leitores de RFID permitindo a leitura (identificação do equipamento) e/ou escrita de dados. Até 12 m de alcance com UHF.	As etiquetas podem conter muita informação; é um sistema robusto, e existem vários tipos de tags disponíveis.	Custo de infraestrutura específica
Bluetooth Low Energy	Dispositivo central (master) recebe sinais (permanente ou periódicos) de dispositivos periféricos que comunicam por <i>broadcasting</i> ou conexão direta. Alcance típico de operação de 5 m.	Baixo consumo energético; <i>smartphones</i> recentes já incorporam esta tecnologia	Mais caro que tags de RFID; baixo alcance; não compatível com Bluetooth
Wi-Fi	Medição da intensidade do sinal recebido (RSS) nos pontos de acesso e uso do método de <i>fingerprinting</i> para determinar a localização do dispositivo. Alcance de até 100 m com 2.4GHz e <i>line-of-sight</i> .	Muito disseminado a nível de infraestrutura e presente em muitos dispositivos móveis.	Flutuações de sinal aumentam imprecisões de localização; computacionalmente intensivo; problemas de privacidade.
Ultra-wide band (UWB)	UWB utiliza métodos como <i>Time of Flight (ToF)</i> , <i>Angle-of-arrival (AoA)</i> e <i>Received Signal Strngth Indicator (RSSI)</i> em simultâneo para determinar a posição do equipamento com precisão. Alcance até 100 metros (fazendo parte das WPANs).	Mais preciso que Wi-Fi e Bluetooth;	Tecnologia recente (poucos dispositivos conseguem usar esta tecnologia); preço equipamento alto comparativamente com as anteriores.

Tabela 3 - Resumo das tecnologias de localização de longo alcance

Tecnologia operacional	Princípio de funcionamento	Pros	Contras
LPWAN	Utiliza bandas de frequência baixas (868 MHz) e modulação <i>Chirp Spread Spectrum (CSS)</i> que diminui o efeito de <i>multipath</i> . Medindo parâmetros de sinal é possível obter localização do equipamento. Alcances registrados de 15km.	Baixo consumo de energia, baixa potência, alta permeabilidade e resiliência a fatores externos (como paredes)	Necessita de bateria no emissor, baixa taxa de bits/s.
GPS	Determina a posição geodésica e altitude na superfície da terra recorrendo a 3 ou 4 satélites. Recetor de GPS processa os sinais dos satélites determina o ToF para calcular distâncias e obter localização.	Amplamente utilizado e cada vez mais disseminado.	Computacionalmente intensivo, problemas de privacidade, baixa permeabilidade dentro de edifícios. Necessita equipamento com
Cellular Network	Combinação de técnicas como o <i>Cell ID</i> (onde a localização é estimada com base na longitude e latitude da célula de rede mais forte), RSS e <i>fingerprinting</i> para localizar dispositivo.	Utilização de telemóveis está muito difundida e presente nos dispositivos móveis.	Computacionalmente intensivo, processo complexo, precisão localização baixa.

A opção por uma destas tecnologias depende das necessidades do negócio e pode ser selecionada com base em 3 fatores:

- Mobilidade do equipamento: é a extensão de movimento que necessita de ser rastreado. Se o movimento estiver restrito a um edifício, uma sala ou um armazém, as tecnologias de localização de curto alcance serão a melhor opção. Se o equipamento é movido de um local para outro (muito afastados geograficamente), então tecnologias de longo alcance (como o GPS) serão melhor opções;
- Precisão da localização: se houver uma necessidade de determinar a posição do equipamento com precisão de até alguns centímetros, a banda ultra larga (*UWB – ultra wide band*) pode ser utilizada. Por outro lado, se for necessário localizar um equipamento com menor precisão (até alguns metros) ou dentro de uma determinada zona, a tecnologia GPS pode ser uma opção;
- Frequência de localização: Se for necessário saber a localização de um equipamento a cada minuto ou hora, as tecnologias BLE, GPS ou Wi-Fi são as mais indicadas. Nos casos em que não é preciso saber em tempo real a localização de um ativo (por exemplo, se

é apenas preciso saber se um equipamento está num laboratório bastará registar a sua entrada ou saída dessa sala) a tecnologia de RFID poderá ser uma boa opção.

Estes três fatores anteriormente mencionados são os mais relevantes podendo, contudo, ser tido em conta outros como o custo de implementação, a linha de visão, as restrições de localização, a atual infraestrutura do edifício, e a facilidade de implementação. O design da solução também influencia as características de cada tecnologia e deve ter em conta as necessidades do negócio. Por exemplo, a precisão de rastreamento do equipamento pode ser melhorada se houver um maior investimento em infraestruturas de suporte, como a aquisição de um maior número de *tags* e de leitores posicionados estrategicamente (no caso da tecnologia de RFID).

#### 3.3.1.10 Outras tecnologias

Existem outras tecnologias, como é o caso do código de barras e do QR code (representados na Figura 14), que não recorrem a sinais de rádio para obter informação do equipamento, mas interessa fazer uma breve menção a estas tecnologias dado que também podem ser utilizadas para localização e identificação de equipamentos.

O código de barras foi a primeira tecnologia usada no rastreamento de equipamentos e mercadorias. É muito económico, eficaz e fiável. O problema do código de barras é que necessita de estar alinhado corretamente com os *scanners* para que sejam digitalizados e as informações possam ser lidas. Se a etiqueta do código de barras for alterada, o *scanner* não consegue digitalizar e não recupera os dados associados.

Os QR code são uma versão avançada dos códigos de barras (versão 2D destes). Estes não foram apenas idealizados para o rastreamento de ativos, como também podem ser utilizados em publicidade ou marketing.

Entre estas duas tecnologias o código de barras apenas consegue armazenar até 25 caracteres, enquanto o QR code pode ir até 2000 caracteres e o código de barras é menos tolerante a danos e erros de impressão (e falta de redundância na codificação), enquanto o QR code tem capacidade de correção de erros se estiver sujo ou danificado (*What Makes QR Code Different from Barcode in Asset Tracking? - Asset Infinity, 2021*). A versatilidade do QR code também permite que este seja utilizado por dispositivos móveis com recurso a câmara.



Figura 14 - Exemplo de tags de código barras (esquerda) e QR code (direita)

### 3.3.2 Tecnologias de informação

As tecnologias de informação é o termo usado para todo o espectro de tecnologias de processamento de informação, incluindo software, hardware, tecnologias de comunicação e serviços relacionados. Em geral, as IT não incluem tecnologias incorporadas que não geram dados para uso corporativo (*Definition of Information Technology (IT) - Gartner Information Technology Glossary, 2022*). Estes tipos de tecnologias podem ser vocacionados para o desenvolvimento aplicacional (nos quais os requisitos têm um ciclo de vida médio ou curto e dinâmicos) cuja prioridade é direcionada para a eficiência e usabilidade dos recursos, integridade e disponibilidade dos dados.

De modo a filtrar os tipos de tecnologias de informação é necessário estudar e analisar o contexto em que o problema se insere. Pela interpretação do que foi descrito no capítulo 2, pretende-se obter uma solução que seja facilmente acessível, que agilize o processo de requisição do equipamento e que rapidamente ajude a consultar a informação em diferentes contextos (no laboratório ou no escritório). Assim sendo, vão ser abordados tipos de desenvolvimento de aplicações móveis dado que estas permitem dar resposta aos problemas.

#### 3.3.2.1 Desenvolvimento de aplicações móveis

A evolução dos telemóveis e de outros dispositivos móveis tornou-os capazes o suficiente de executar sistemas operacionais que suportam um conjunto de programas pré-instalados bem como uma colocação de aplicações de terceiros. Estas últimas são criadas recorrendo a interfaces de programação de aplicações (Xanthopoulos and Xinogalos, 2013). A adoção elevada desta tecnologia também acelerou o desenvolvimento de um maior conjunto de programas com as mais diversas funcionalidades que tiram proveito dos sensores com que os dispositivos móveis vêm equipados (como câmara, rede móvel, GPS entre outros).

O desenvolvimento de aplicações móveis pode ser classificado em três tipos principais, sendo eles o nativo, híbrido e *web*. Vão ser apresentadas cada uma destas abordagens bem como as suas vantagens e desvantagens.

#### **Desenvolvimento nativo**

Este tipo de desenvolvimento é utilizado para criar aplicações para uma plataforma específica. A linguagem de programação depende dessa plataforma alvo bem como do ambiente de desenvolvimento. Por exemplo, as aplicações desenvolvidas para a plataforma Android são acessíveis na Google Play Store e podem ser escritas na linguagem Java ou Kotlin (esta última é uma linguagem que compila na máquina virtual Java). Outro exemplo são as aplicações desenvolvidas para o sistema operativo iOS que podem ser acedidas a partir da Apple Store (a iStore) e são escritos na linguagem Objective-C ou Swift.

A principal vantagem deste tipo de aplicações em relação às restantes é que estas conseguem ter acesso direto aos componentes das plataformas alvo (como o sistema de notificações, GPS, entre outros). Como consequência, estas normalmente apresentam um melhor desempenho do que os restantes tipos de aplicações desenvolvidas de outra forma. Também são

caracterizadas por fornecerem uma interface intuitiva e mais simples. Outra vantagem é o facto de garantirem uma maior compatibilidade, qualidade e segurança com os dispositivos alvo ([Infographic] *A Guide to Mobile App Development: Web vs. Native vs. Hybrid*, 2019).

Contudo, este tipo de desenvolvimento requer a utilização de linguagens de programação específicas e precisam de desenvolvedores mais experientes, uma vez que se houver a necessidade de criar uma aplicação para diferentes plataformas, este terá de conhecer várias linguagens (tornando o seu desenvolvimento complexo e demorado). O desenvolvimento de código fonte nativo para uma plataforma não pode ser reutilizado noutras plataformas. Normalmente a opção por este tipo de desenvolvimento pode levar à necessidade de uma equipa específica para cada plataforma móvel (Xanthopoulos and Xinogalos, 2013). Porém, a plataforma Xamarin permite desenvolver código fonte em C# que pode ser posteriormente compilado para aplicações de código nativo (permitindo que estas sejam executadas em Android, iOS, Mac e Windows com apenas um código fonte) (Hermes, 2015).

### **Desenvolvimento web**

As aplicações *web* recorrem a tecnologias como JavaScript, HTML e CSS (conjunto que permite a criação de um *website*) e que pode ser acedido através de um browser no dispositivo.

O termo “*Mobile Web Development*” refere-se ao desenvolvimento de *websites* destinados a dispositivos móveis (podendo estes ser *smartphones*, tablets ou outros). Estes *websites* têm a sua interação adaptada de modo a tornar a sua utilização mais simples e eficiente. Estas aplicações também devem ser “responsivas”, ou seja, ter facilidade em adaptar-se ao tamanho do ecrã dos dispositivos móveis. Este tipo de desenvolvimento tem vindo a crescer exponencialmente devido ao grande aumento no número de utilizadores que recorrem a dispositivos móveis.

Como principais vantagens, as aplicações resultantes deste tipo de desenvolvimento têm uma grande compatibilidade entre dispositivos com dimensões de ecrãs diferentes (continuando a oferecer uma boa experiência); facilidade em chegar aos utilizadores finais (estes conseguem aceder ao *website* recorrendo apenas a um link e a um *web Browser*), a nível de manutenção é mais fácil para o desenvolvedor (este apenas tem de fazer a alteração uma vez, devido ao site possuir uma única versão para todos) e mais prático para o utilizador uma vez que não necessita de transferir as atualizações como acontece nas aplicações móveis.

Ao nível das desvantagens, estas aplicações embora sejam adaptadas para serem utilizadas em dispositivos móveis, nem todos conseguem usar as funcionalidades específicas destes (como o GPS, ou a câmara) de forma tão eficaz como se fosse uma aplicação desenvolvida nativamente. Ao nível da experiência para o utilizador a responsividade da aplicação pode ser diferente de dispositivo para dispositivo, uma vez que se pode aceder a estas com ecrãs de diferentes dimensões. Estas aplicações também requerem acesso constante à internet para serem utilizadas.

As aplicações resultantes deste tipo de desenvolvimento seguem normalmente um modelo cliente-servidor, no qual o cliente faz um pedido e espera até obter resposta do servidor. O protocolo usado para fazer a transmissão de informação (como o HTML anteriormente mencionado, ou *Javascript Object Notation (JSON)*) é o *Hypertext Transfer Protocol (HTTP)*. Este foi desenvolvido para comunicação entre navegadores e serviços web (podendo ser usado para outros propósitos) (MDN, 2022).

### **Desenvolvimento híbrido**

As aplicações provenientes do desenvolvimento híbrido resultam da utilização de tecnologias web para escrever o código fonte, sendo posteriormente encapsulado numa aplicação nativa e disponibilizado para os sistemas operativos Android e iOS nas lojas oficiais de cada plataforma. De modo a este tipo de aplicações conseguirem utilizar funcionalidades do dispositivo (como por exemplo o Bluetooth) é adicionado código nativo ou plug-ins específicos (*Ionic Article: What is Hybrid App Development?*, 2019).

O código de uma aplicação híbrida recorre a linguagens como HTML (Hypertext Markup Language), CSS (Cascading Style Sheets) e JavaScript para fazer o seu desenvolvimento, sendo este posteriormente executado dentro de uma aplicação nativa (um *wrapper*, como por exemplo Apache Cordova ou Ionic Capacitor) que permitem o acesso aos recursos nativos e que tem o seu próprio navegador embutido (que é invisível ao utilizador e que substitui o navegador para usar a aplicação. No caso de uma aplicação iOS esta usará o WKWebView, enquanto o Android utilizará o WebView) (*Ionic Article: What is Hybrid App Development?*, 2019).

A constituição de uma aplicação híbrida pode ser dividida em duas partes:

- Uma *Shell* nativa que é descarregada e utiliza um *wrapper* como o WebView para executar o código;
- Código fonte em HTML, CSS e JavaScript para a estrutura, apresentação e lógica da aplicação.

Como principais vantagens destacam-se o facto das aplicações híbridas não necessitarem de um navegador *web*, terem acesso às APIs internas e ao hardware do dispositivo e necessitam apenas de desenvolver um código fonte que funciona em diferentes plataformas.

Como desvantagens estes tipos de aplicações têm pior performance que aplicações nativas, estão dependentes de uma plataforma externa para implantar o *wrapper* da aplicação, e o esforço de desenvolvimento aumenta muito com a necessidade de introduzir capacidades mais complexas às aplicações a serem desenvolvidas (quanto mais o comportamento da aplicação híbrida se aproximar de uma aplicação nativa, mais complexa será a implementação).

### **Resumo**

Seguidamente é apresentada uma Tabela que resume as principais características de cada tipo de desenvolvimento para aplicações móveis.

Tabela 4 - Resumo das principais características do tipo de desenvolvimento de aplicações Móveis

<b>Característica</b>	<b>Desenvolvimento Nativo</b>	<b>Desenvolvimento Web</b>	<b>Desenvolvimento híbrido</b>
Acesso a APIs e funcionalidades nativas do dispositivo	Total	Limitado	Total (com plugins)
Performance	Alta	Alta	Média
Linguagem de desenvolvimento	Específica da plataforma	HTML, CSS, JavaScript	HTML, CSS, JavaScript (necessidade de desenvolver plugins)
Suporte multiplataforma	Não	Sim	Sim
Experiência para o utilizador	Muito boa	Boa	Boa
Reutilização do código	Não	Sim	Sim

### 3.3.2.2 Arquitetura de software

Para além do tipo de desenvolvimento aplicacional também é importante estudar que tipos de arquitetura de software existem que podem ser utilizados para realizar o desenvolvimento da aplicação. A definição da arquitetura de um software tem um papel crucial na sua capacidade de dimensionar e atender às demandas dos utilizadores ao longo do tempo.

Qualquer aplicação sem uma arquitetura formal torna-se fortemente acoplada, frágil, e de difícil manutenção. Isto leva a que seja difícil perceber quais são as características de desempenho da aplicação, o quão responsiva é a arquitetura, o quão difícil é realizar a sua manutenção, entre outros. A definição de uma arquitetura ajuda a perceber as características básicas e o funcionamento da aplicação. Por exemplo, algumas arquiteturas ajudam a estruturar aplicações que têm facilidade em escalonar enquanto outras permitem criar aplicações de forma ágil. É assim importante conhecer os pontos fortes e fracos de cada arquitetura de modo a escolher aquela que corresponda melhor às necessidades e objetivos do negócio (Richards, 2017). Vão ser apresentadas 3 arquiteturas que melhor se enquadram com o problema em estudo.

#### **Arquitetura em camadas**

Esta é a arquitetura mais comum no desenvolvimento de aplicações (também conhecida por arquitetura de n-camadas). Este padrão organiza os componentes em camadas horizontais, com cada uma desempenhando uma função distinta dentro do programa (por exemplo, lógica de apresentação ou de negócios). Apesar desta arquitetura não definir o número ou o tipo de

níveis que devem existir, a maioria inclui quatro camadas comuns: apresentação, negócio, persistência e base de dados (a Figura 15 representa estas camadas). Cada uma delas tem uma função e responsabilidade específica dentro da aplicação. Por exemplo, a camada de apresentação é responsável por lidar com a interface do utilizador e a lógica de comunicação dentro do navegador, enquanto a de negócio é responsável por executar regras específicas da matéria em questão associadas ao pedido. Isto garante separação de responsabilidades (*Separation of concerns*) entre os diversos componentes presentes nas camadas (nos quais cada componente lida apenas com a lógica que pertence a essa camada) (Richards, 2017).

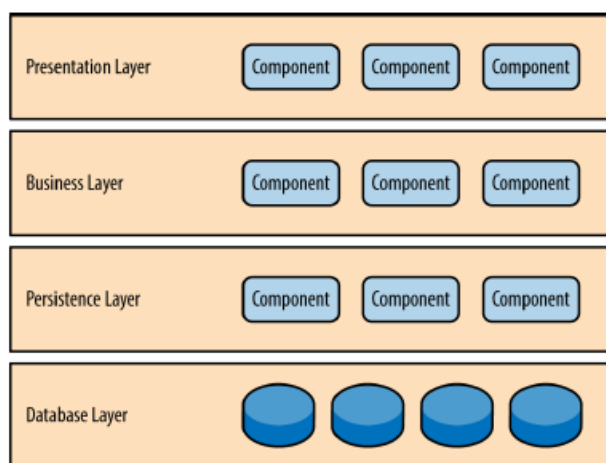


Figura 15 - Representação da arquitetura em camadas (Richards, 2017)

### **Arquitetura orientada a microsserviços**

Uma arquitetura orientada a microsserviços é vista como uma possível alternativa para aplicações monolíticas (que normalmente consistem em componentes fortemente acoplados que fazem parte de uma única unidade, sendo posteriormente difícil alterar, testar ou implantar a aplicação). Cada componente da arquitetura orientada aos microsserviços é desenvolvido como uma unidade separada, permitindo um processo de implantação mais rápido e fácil, maior estabilidade e um elevado grau de desacoplamento entre os diferentes componentes (podendo ser testados independentemente de qualquer outro componente ou serviço).

Um dos conceitos associados a este tipo de arquitetura é o de *service components* (representados na Figura 16). Estes contêm um ou mais módulos que representam uma função de único propósito (como por exemplo fornecer o clima para uma cidade) ou uma parte de lógica de negócio de uma grande aplicação. Outro conceito-chave associada a esta arquitetura é o de *distributed architecture*, no qual todos os componentes se encontram completamente desacoplados e comunicam a partir de um protocolo de acesso remoto (Richards, 2017).

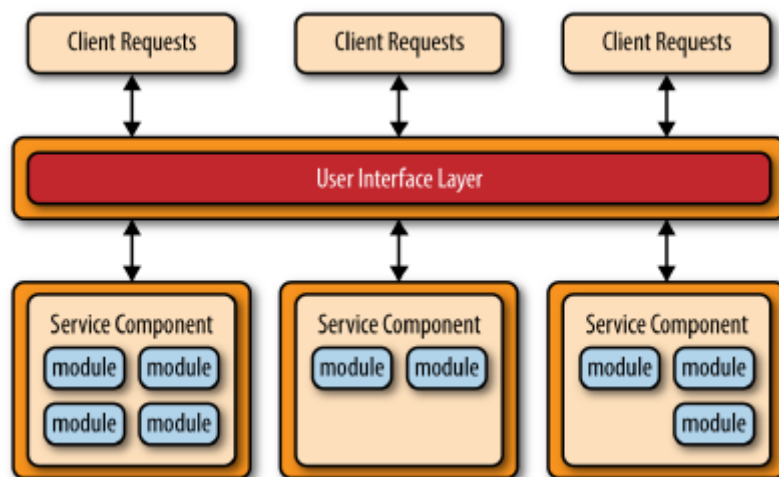


Figura 16 - Representação esquemática de uma arquitetura de microsserviços (Richards, 2017)

### Arquitetura orientada a eventos

A arquitetura orientada a eventos é um padrão de arquitetura assíncrona distribuída utilizado para produzir aplicações escalonáveis. É também altamente adaptável podendo ser implementado em aplicações de pequenas e grandes dimensões. Esta arquitetura é composta por componentes de processamento de eventos desacoplados, criados para um propósito bem definido que recebem e processam eventos de forma assíncrona. A Figura 17 representa esquematicamente uma arquitetura definida com base neste padrão.

Este padrão arquitetural tem duas topologias principais, o mediador e o *broker*. O mediador é utilizado quando é necessário orquestrar várias etapas de um evento por meio de um mediador central. Por outro lado, a topologia *broker* é utilizada quando há necessidade de encadear eventos sem o uso de um mediador central (Richards, 2017).

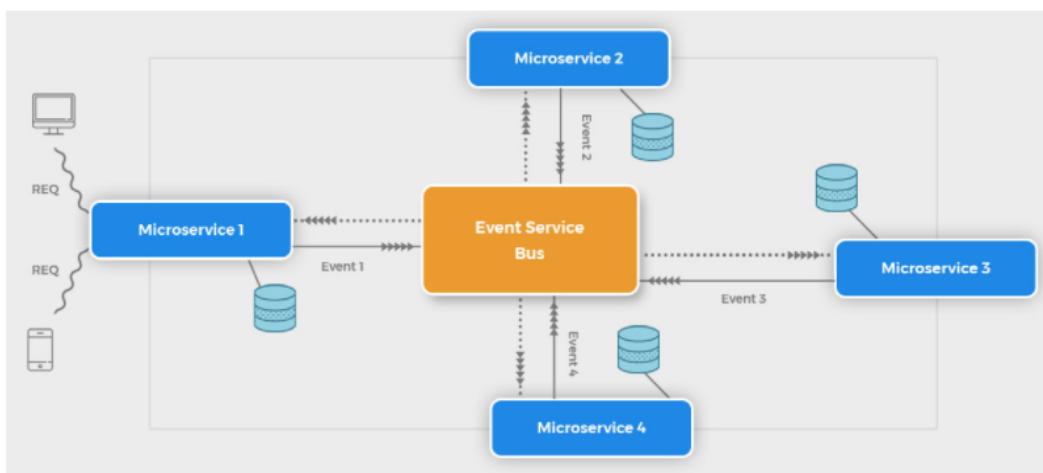


Figura 17 - Representação da arquitetura orientada a eventos (*A Quick Guide to Understand the Event-driven Architecture*, 2019)

## **Resumo**

A Tabela 5 resume as principais características dos 3 padrões arquiteturais apresentados. Estes vão ser classificados segundo a capacidade de rapidamente responder a um ambiente em constante mudança (agilidade em fazer alterações), à facilidade de fazer implantação (uma pequena alteração num componente pode requerer implantar em toda a aplicação), à testabilidade (facilidade em simular o comportamento de código ou objetos existentes de forma controlada), à performance (tempo que demora a executar pedidos e operações), à escalabilidade (relacionada com o acoplamento entre os diferentes módulos de software) e quanto à facilidade de desenvolvimento (relacionado com a quantidade e qualidade de exemplos e fontes de informação que implementam a arquitetura).

Tabela 5 - Principais características dos padrões arquiteturais

<b>Característica</b>	<b>Arquitetura em camadas</b>	<b>Arquitetura orientada a microsserviços</b>	<b>Arquitetura orientada a eventos</b>
Agilidade em fazer alterações	Baixa	Alta	Alta
Facilidade de implantação	Baixa	Alta	Alta
Testabilidade	Alta	Alta	Baixa
Performance	Baixa	Baixa	Alta
Escalabilidade	Baixa	Alta	Alta
Facilidade de desenvolvimento	Alta	Alta	Baixa

### 3.3.2.3 Exploração tecnológica

Com vista à implementação da solução de acordo com as arquiteturas estudadas anteriormente, foi também efetuada uma exploração de tecnologias que poderão ser alavancadas. O foco desta exploração recaiu sobre tecnologias de desenvolvimento *web* dada a abundância destas e por outro lado permitirem maior flexibilidade na implementação, serem as que mais se adequam aos requisitos levantados inicialmente e dado já existir algum nível de proficiência nas mesmas. Contudo a implementação e escolha da arquitetura para a solução final deverá ser independente do tipo de tecnologia. Para realizar a exploração tecnológica foram utilizados os seguintes critérios de seleção das tecnologias:

- A facilidade de implementação de acordo com as arquiteturas estudadas anteriormente;
- O facto de estarem bem documentadas e serem muito utilizadas;

- Boa integração com as tecnologias presentes nas restantes camadas.

A exploração tecnológica pode ser agrupada em 2 grupos (Fitzgibbons, 2022):

- **Tecnologias *Back-end***: responsáveis por implementar a funcionalidade central do sistema e encapsular as regras de negócio. Para além disso, também fornecem acesso a dados hospedados dentro dos limites do sistema ou expostos por outros sistemas de *back-end*.
- **Tecnologias *Front-end***: implementam a funcionalidade necessária para permitir ao utilizador interagir com a aplicação e ajudam a sincronizar e orquestrar as interações do utilizador.

As tecnologias vão ser apresentadas sucintamente dado não ser relevante fazer uma exploração exaustiva nem com grande detalhe destas na medida em que todas têm potencialidade para serem utilizadas na criação da solução final. O objetivo é agregar as que existem e podem ser usadas na implementação da solução em conformidade com a arquitetura apresentada no capítulo 5. Os critérios para a seleção destas tecnologias são descritos no capítulo 6.

### **Tecnologias de Back-end**

As tecnologias Back-end podem ser empregues para implementar a lógica de negócio e coordenação da execução de tarefas (podendo ser utilizadas na *Business Layer* numa arquitetura em camadas, no desenvolvimento de um serviço na arquitetura em microsserviços ou na implementação de um mediador central na arquitetura orientada a eventos).

Exemplos de tecnologias que podem ser utilizadas para desenvolver a aplicação de Back-end são (Dats, 2021):

- **NodeJS**: É um ambiente de execução JavaScript *open-source*. Este permite a utilização da linguagem de programação JavaScript para escrever scripts de execução da aplicação do lado do servidor. A utilização desta tecnologia permite “unificar” o desenvolvimento aplicacional *web*, visto poder utilizar esta do lado do servidor e do lado do cliente.
- **.NET**: Esta framework utilizada a linguagem de programação C# (projeta especificamente para ser utilizada nesta tecnologia) e pode ser utilizada para desenvolver aplicações *web*. Os desenvolvedores podem utilizar os recursos da API .NET para criar aplicações, contudo estas só podem ser executadas em plataformas que suportem o runtime .NET. Para que os programas sejam executados noutras plataformas o código deve ser compilado (numa ferramenta como a Microsoft .NET Native).

Também é importante ter em conta tecnologias de armazenamento de dados. Estas são transversais a qualquer tipo de arquitetura pelo que são necessárias para fazer a persistência e disponibilização dos dados. Exemplos de tecnologias de base dados são (Taylor, 2020):

- **MySQL:** é um sistema RDBMS (Relational Database Management Systems) amplamente utilizado e funcional no modelo de base dados relacionais. Isto torna a administração do banco de dados mais fácil. Contudo este tipo de tecnologia exige pré-definir o esquema do banco de dados com base nos requisitos e regras que ajudem a controlar os relacionamentos entre os campos e as tabelas.
- **MongoDB:** é um banco de dados NoSQL orientado a documentos e utilizado para armazenamento de um alto volume de dados. Este surgiu em meados dos anos 2000. Este tipo de DBMS (Database Management Systems) utiliza esquemas dinâmicos, permitindo criar registos sem definir uma estrutura inicial (os campos necessários e os tipos de valor). O MongoDB permite alterar a estrutura dos registos adicionando novos campos ou excluindo existentes.

### Tecnologias de Front-end

As tecnologias aqui apresentadas seguem o padrão de desenvolvimento *Single-page Application* (SPA), no qual a página *web* é reescrita dinamicamente com os novos dados do servidor (em vez do método padrão de recarregar as páginas completamente). Desta forma é possível obter transições mais rápidas que fazem o *website* parecer uma aplicação nativa. As atualizações completas da página nunca ocorrem e o código HTML, JavaScript e CSS é obtido pelo *web* Browser com apenas um carregamento da página (podendo também outros recursos ser carregados dinamicamente e adicionados conforme necessário, normalmente em resposta a ações do utilizador) (Flanagan, 2020).

Estas tecnologias são utilizadas para criar aplicações com as quais o utilizador interage. Serão assim alocadas na camada de apresentação na arquitetura em camadas e na arquitetura orientada a microsserviços (*User interface layer*). Exemplos de tecnologias que podem ser utilizadas para fazer o desenvolvimento da aplicação *web* e que permite a interação com o utilizador são (Krunal, 2022):

- **React:** também conhecido como React.js ou ReactJS, é uma biblioteca *open-source* front-end e utiliza JavaScript e JSX (uma extensão de sintaxe do JavaScript) para descrever como a interface do utilizador deve ser apresentada (no seu código fonte). É mantido pelo Facebook e uma comunidade de empresas e desenvolvedores. React pode ser utilizado para desenvolver SPAs ou aplicações mobile (React Native). O foco do React é a renderização de dados pelo que a criação de aplicações em React normalmente requer bibliotecas adicionais para gestão de estado e *routing*.
- **Angular:** é uma framework *open source* de aplicações *web* liderada pela equipa Angular da Google. Angular é a uma versão reescrita pela mesma equipa que criou o AngularJS (sua primeira versão). Esta utiliza TypeScript (um superconjunto de JavaScript) para fazer o desenvolvimento de aplicações SPA. Esta framework simplifica o desenvolvimento e teste de aplicações *web* proporcionando o desenvolvimento num padrão de arquitetura em 3 camadas denominada Model View Controller (MVC) do lado do cliente juntamente com componentes comumente utilizados.

## Outras tecnologias

As tecnologias apresentadas até agora são as basilares para fazer o desenvolvimento do sistema. Serão apresentadas aqui outras que dão suporte às anteriores, não sendo por isso fundamentais à construção da solução.

Para ajudar na construção da solução de Front-End pode optar-se por uma *framework* de CSS. Esta vem munida de componentes de interface de utilizador pré desenhados e com comportamentos pré configurados, o que agiliza o processo de desenvolvimento e torna o mesmo a nível de design mais apelativo e fácil de usar. Exemplos de *frameworks* de CSS que podem ser utilizadas são (Bacinger, 2022):

- **Material UI:** é uma *framework* de CSS que fornece componentes prontos para utilização e segue as diretivas do Material Design desenvolvido pela Google. Este possibilita o uso de diferentes componentes para criar uma interface gráfica para aplicações móveis e *web*. O Material Design inclui diretrizes para tipografia, espaçamento, cor, imagens, escalas, etc. garantindo que os utilizadores têm uma experiência consistente.
- **Bootstrap:** é um kit de ferramentas HTML, CSS e JavaScript para criar e construir páginas e aplicações *web*. É um projeto *open source* hospedado no GitHub e originalmente criado pelo Twitter. Como principais vantagens este tem um design responsivo, alta compatibilidade com navegadores e oferece design consistente e componentes reutilizáveis.

## 3.4 Sumário

Neste capítulo foram abordados os tipos de tecnologias operacionais e de informação existentes que podem ajudar no desenvolvimento da solução que vai dar resposta ao problema exposto no capítulo 2. Inicialmente fez-se uma pequena introdução à digitalização dos processos e a sua relação com a indústria 4.0 e como a IoT pode ser utilizada para reunir, sincronizar, organizar e classificar dados de diferentes fontes dentro de uma fábrica ou organização. Por fim foi feito um levantamento das tecnologias operacionais e tecnologias de informação que podem ser utilizadas para permitir responder às necessidades do projeto. A opção pelas tecnologias apresentadas e que poderão melhor responder ao problema será feita no capítulo de Seleção de Tecnologias.

Relativamente às perguntas colocadas na introdução do capítulo, é possível agora dar resposta.

### **Que tecnologias operacionais existem que podem auxiliar na localização de equipamentos?**

Foi elaborada uma pesquisa sobre as principais tecnologias que podem ser utilizadas para tornar os equipamentos operacionais e de modo a conseguir extrair a informação e a localização do equipamento em causa. Não se pretende utilizar todas as tecnologias, mas perceber se faz sentido a sua aplicação e qual o contributo para a resolução do problema.

Qualquer uma destas tecnologias necessitaria de aquisição de equipamento especializado e alterações na infraestrutura do edifício para poderem ser implementadas. No capítulo 4 vai ser realizada a geração de ideias e seleção da que se apresentar mais vantajosa na sua implementação. Entre as tecnologias identificadas existem as de curto alcance como o RFID, BLE, Wi-Fi e UWB (os códigos de barras e o QR code foram enquadradas noutras tecnologias uma vez que permitem a identificação dos equipamentos, mas necessitam de alguma lógica externa para determinar a sua posição) e as tecnologias de longo alcance como o GPS, a LPWAN e a *Cellular Network*.

### **Como podem as tecnologias de informação ajudar no processo de localização dos equipamentos?**

As tecnologias de informação auxiliam no processamento, armazenamento e gestão da informação e a sua prioridade é direcionada para a eficiência e usabilidade dos recursos, integridade e disponibilidade dos dados. Deste modo foram estudados métodos de desenvolvimento de aplicações móveis dado que estes permitem responder às necessidades levantadas pelo projeto (uma solução que seja facilmente acessível, que agilize o processo de requisição do equipamento, que rapidamente ajude a aceder e consultar a informação em diferentes contextos). Foram também estudadas arquiteturas de software que ajudem a desenvolver uma solução informática adequada (com características necessárias para dar resposta ao projeto) para realizar a requisição dos equipamentos (permitindo fazer a gestão destes).

## 4 Análise de valor

Este capítulo vai ser dedicado à análise de valor e a dar resposta ao *Outcome 3* – avaliar diferentes abordagens para a resolução do problema. Para tal vai ser descrito o modelo de *New Concept Development* (NCD) seguido da sua aplicação. Por fim vai ser apresentada uma proposta de valor para o produto a ser desenvolvido recorrendo ao *Value Proposition Canvas* de Osterwalder.

No final deste capítulo pretende-se dar resposta às seguintes perguntas:

- Que soluções existem para a resolução do problema?
- Qual a solução que melhor se enquadra às necessidades do projeto?

### 4.1 Definição da análise de valor

A análise de valor tem como objetivo aumentar o valor de um projeto, produto ou serviço fornecendo as funções necessárias para que este atinja o desempenho necessário ao menor custo possível.

A conceção de um novo produto implica um processo de inovação inicial que contém risco económico e empresarial. De modo a diminuir esse risco, o processo de inovação pode seguir as 3 fases apresentadas em seguida e ilustradas na Figura 18 (Koen *et al.*, 2002):

1. **Fuzzy Front End** (FFE) – é a fase inicial de projeto no qual há uma grande incerteza em termos de calendarização, de análise de ideias e de identificação das oportunidades;
2. **New Product Development** (NPD) – fase do processo caracterizado por ser mais disciplinado e existir um planeamento com foco nos objetivos pretendidos e no desenvolvimento do produto ou serviço;
3. **Comercialização** – fase final onde ocorre a divulgação e venda do produto ou serviço.

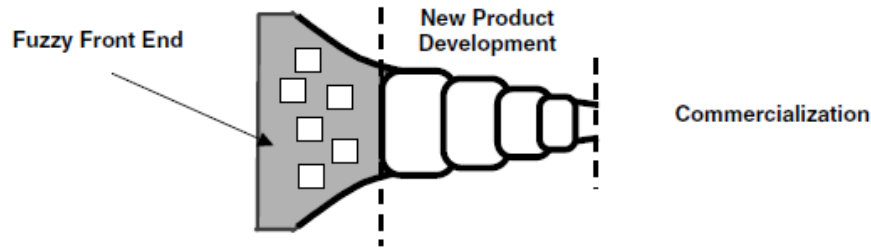


Figura 18 - O processo de inovação pode ser dividido em 3 partes: *Fuzzy Front End*, *New Product Development* e *Commercialization* (Koen et al., 2002)

Devido à falta de termos (linguagem e vocabulário comuns) e definições comuns para os elementos-chave da primeira fase do processo de inovação (FFE), o que dificultava o processo de transferência de conhecimento, levou à criação do modelo NCD (*New Concept Model*) que será em seguida descrito.

## 4.2 Modelo NCD

O modelo NCD, apresentado na Figura 19, é composto por três partes (Koen et al., 2002):

- **Engine:** representa a liderança, cultura e estratégia dos negócios da organização e é responsável por impulsionar os cinco elementos de atividade do FFE;
- **Os cinco elementos de atividade no interior do modelo:** sendo eles a identificação de oportunidade, a análise da oportunidade, a geração de ideias e a definição do conceito do FFE;
- **Fatores de influência:** Estes fatores afetam todo o processo de inovação até a comercialização e são relativamente incontroláveis pela corporação. Consistem nas capacidades organizacionais, no mundo exterior (canais de distribuição, lei, política governamental, clientes, concorrentes e clima político e económico), e nas ciências capacitadoras (internas e externas).

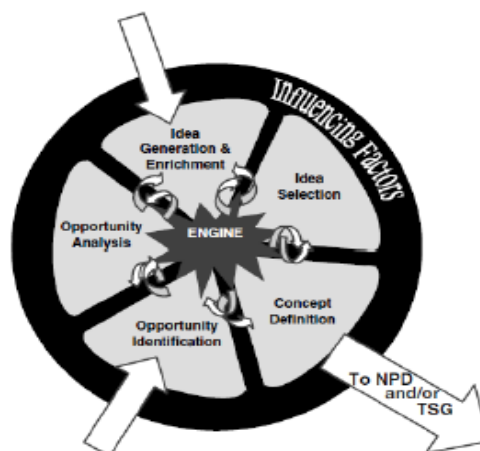


Figura 19 - Modelo NCD (Koen et al., 2002)

Cada um dos elementos de atividade no interior do modelo NCD vão ser em seguida explicados e aplicados à resolução do problema tendo em conta a descrição apresentado no capítulo 2 e o estado de arte levantado no capítulo 3.

#### **4.2.1 Identificação de oportunidade**

Este elemento do modelo NCD tem como objetivo identificar oportunidades de negócio e tecnologias que podem ser aplicadas ao processo atual. É normalmente impulsionado pelos objetivos de negócio com vista a uma oportunidade de investir em tecnologia que a empresa vê como vantajosa.

A oportunidade e a necessidade de desenvolver um sistema para a requisição de equipamentos de laboratório surgiu com a expansão do INEGI, quer a nível de pessoas quer a nível de equipamentos, de modo a tornar as operações mais eficientes (diminuindo o tempo de procura). Atualmente a requisição dos equipamentos é realizada de uma forma manual por inserção da informação numa folha de papel com uma estrutura pré-definida. Muitas vezes o equipamento que o operador requisitou não é devolvido no intervalo de tempo indicado, quer por esquecimento, quer por necessidade de prolongamento do tempo de utilização do mesmo, sendo que nestas situações é prática comum a folha de registo não ser atualizada. Para além disto, não é guardado um histórico das requisições do equipamento, não permitindo assim, de forma imediata, saber se há equipamentos mais requisitados que outros e eventual necessidade de aquisição de novos. O processo atual é, ainda, pouco amigável para operadores inexperientes que desconhecem o nome das ferramentas, podendo levar a nomeá-las de forma errónea.

Para além disto, o aparecimento do Covid-19 provocou mudanças sociais em todo o mundo e obrigou empresas e organizações a acelerarem a digitalização dos seus processos e interações. As empresas reconheceram que a introdução de tecnologia nos seus sistemas de trabalho poderia trazer bastantes benefícios. De acordo com pesquisa realizada pela Mckinsey, os entrevistados indicaram que suas empresas foram capazes de adotar mudanças digitais 20 a 25 vezes mais rápido do que o esperado (Mckinsey, 2020). As ideias e soluções da indústria 4.0 associadas á transformação digital provocou mudanças nos modelos de negócio e procedimentos de trabalho, acelerando-os e tornando-os mais eficientes (Savić, 2020).

#### **4.2.2 Análise de oportunidade**

Neste elemento do NCD é reunida informação adicional para traduzir as oportunidades identificadas em oportunidades específicas de negócio e tecnologia para a organização. Alguns métodos de análise da oportunidade existentes são a elaboração de estudos de mercado, experiências científicas, ou até criação de grupos focais.

De modo a obter uma perspetiva sobre o que poderia ser a solução final a ser implementada e desenvolvida foram estudados métodos de engenharia de levantamento de requisitos que

podem ser aplicados. Deste modo optou-se pela realização de várias entrevistas com os *stakeholders* e futuros utilizadores do sistema. Estas entrevistas foram preparadas tendo em conta o perfil dos *stakeholders* e as boas praticas no desenvolvimento de questionários descritas no capítulo de Processo de elicitação. O resultado esperado com estas entrevistas seria obter os requisitos do sistema que servirão de base ao desenvolvimento da solução final.

Para além disto é preciso perceber o que se entende por gestão de equipamentos e que possam contribuir para a idealização do produto final. Segundo a ISO 55000, a gestão de equipamentos é um conjunto de atividades, algumas delas enumeradas em seguida (Hastings, 2015):

- Identificar quais equipamentos necessários;
- Identificar a necessidade de adquirir novos equipamentos;
- Fornecer apoio na manutenção dos equipamentos;
- Renovação de equipamentos.

Para atingir o objetivo do negócio há necessidade de recorrer a equipamentos físicos que apoiem a execução das operações. Um sistema de gestão de equipamentos pode ajudar no planeamento, aquisição e manutenção destes.

#### **4.2.3 Geração de ideias**

Em seguida procede-se à geração de ideias. Neste elemento é idealizado o primeiro conceito para desenvolvimento da oportunidade até atingir o patamar de ideia concreta. É um processo iterativo, onde podem ser usados métodos como *brainstorming*, no qual a ideia vai sofrendo alterações durante as sessões. Aqui é importante haver trocas de opiniões, podendo as ideias ter origem num membro da organização ou utilizador final.

Tendo em conta as tecnologias identificadas que podem ajudar no processo de requisição e localização dos equipamentos vão ser propostas e abordadas duas soluções que utilizam conjuntos de tecnologias diferentes (poderiam ser desenvolvidas mais soluções com combinações de tecnologias diferentes, mas as duas aqui apresentadas são representativas do espectro geral).

##### **Solução 1**

A primeira solução passaria por selecionar tecnologias operacionais identificadas e integrá-las com uma aplicação informática criada com base nos métodos estudados na secção de tecnologias de informação. A Figura 20 ilustra o conceito idealizado.

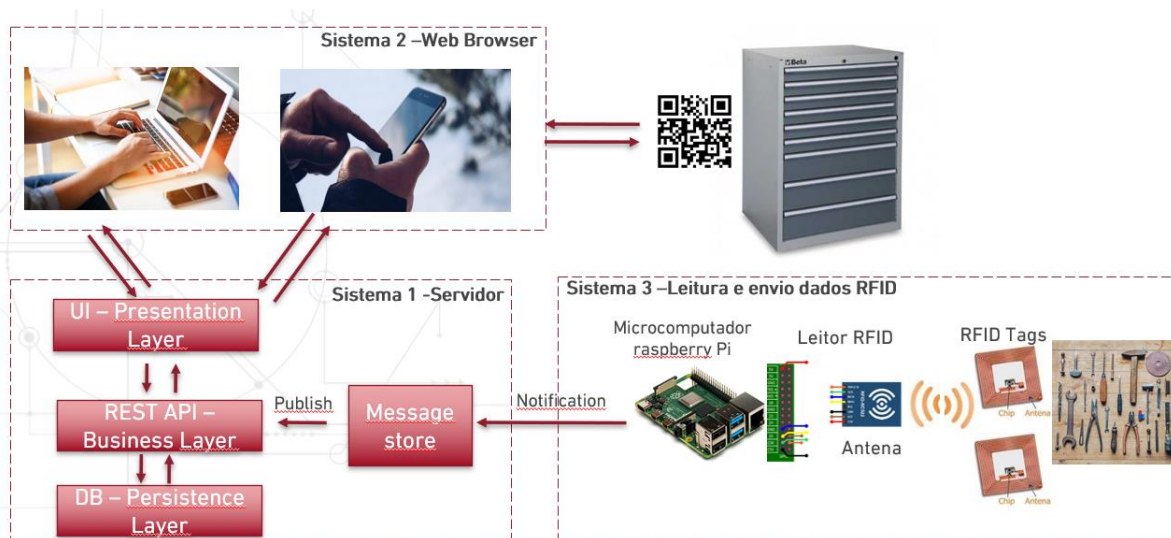


Figura 20 - Representação esquemática solução 1

Esta solução é constituída por 3 sistemas. O sistema 1 será a solução informática alojada num servidor da organização. Esta solução será criada de acordo com o desenvolvimento de aplicações móveis com recurso as tecnologias *web* seguindo uma arquitetura em camadas. Esta arquitetura torna o desenvolvimento da aplicação mais ágil, permite manter o código mais organizado, estruturado, com separação das responsabilidades adequada e facilita a testabilidade dos componentes. Considerando que não é previsto um grande número de utilizadores em simultâneo, a performance da aplicação não deverá ser afetada. O sistema 2 é o dispositivo por onde se acede à aplicação, podendo ser um *smartphone* ou um computador (necessita apenas de ter um *web browser* e a aplicação adaptar-se-á às diferentes dimensões de ecrã). Por fim, o sistema 3 terá um computador (ou um microcomputador como uma raspberry pi) que por protocolo SPI (*Serial peripheral interface*) comunica com um leitor de RFID para a identificação de tags instaladas em equipamentos. Este sistema também teria uma aplicação de leitura e de envio de dados para um *broker* alojado no sistema 1. A comunicação entre o sistema 1 e o sistema 3 seria implementada segundo uma arquitetura orientada a eventos, permitindo esta se realize de forma assíncrona entre a aplicação de gestão dos equipamentos (sistema 1) e a localização dos equipamentos (realizada pelo sistema 3).

Esta solução tornaria o sistema mais otimizado e automático na localização dos equipamentos, mas teria um custo de infraestrutura elevado (devido à aquisição de componentes como antenas e microcomputadores) e uma implementação mais complexa dado a interligação e lógica que seria necessário desenvolver entre os diferentes sistemas. De entre todas as tecnologias operacionais, a tecnologia de RFID é a que melhor se enquadra dado a possibilidade de integrar tags passivas nos equipamentos (suprimindo a necessidade de substituir ou recarregar baterias), que localizariam estes em posições estratégicas do edifício (não havendo uma monitorização constante da localização como aconteceria no recurso a tecnologia WiFi). Esta também evitaria problemas de privacidade por parte do utilizador que requisita o equipamento.

## Solução 2

A segunda solução utiliza também algumas tecnologias operacionais, identificadas no capítulo de Tecnologias de informação e integra-as numa aplicação informática criada com base nos métodos estudados na secção de tecnologias de informação. A Figura 21 ilustra o conceito idealizado.

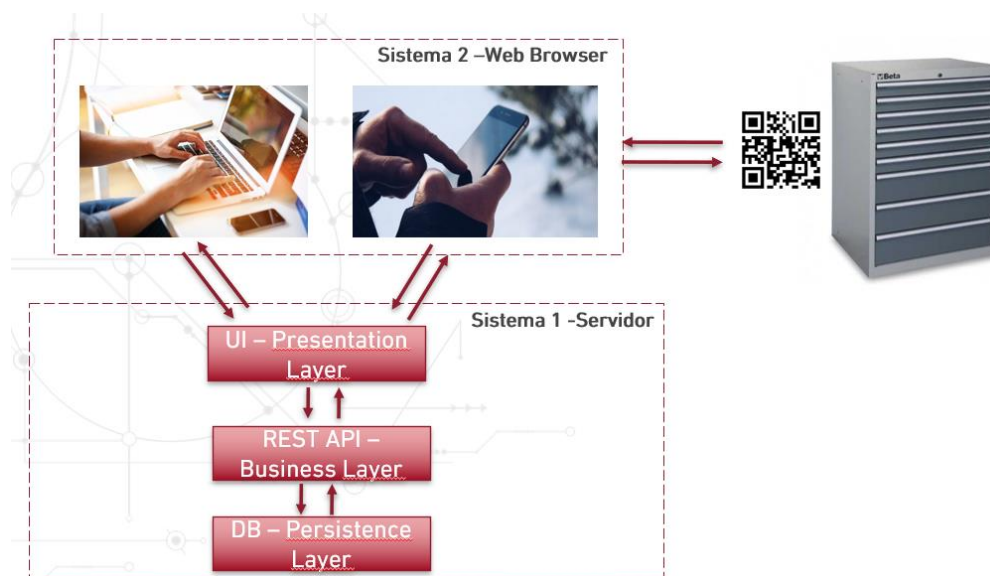


Figura 21 - Representação esquemática solução 2

Esta solução seria constituída por 2 sistemas. O sistema 1 (semelhante ao descrito na primeira solução) será a solução informática alojada num servidor da organização. Esta solução será criada de acordo com o desenvolvimento de aplicações móveis (dado que este poderá ser acedido por um dispositivo móvel como um *smartphone* ou um computador), com recurso a tecnologias *web* (opção por este tipo de tecnologias deve-se a estas apresentarem menor custo e maior facilidade para disponibilização quando comparadas com o desenvolvimento nativo ou híbrido que implicariam ter contas de desenvolvedor para distribuir a aplicação nas plataformas iOS e Android (*Should I Publish an iOS or Android App?*, 2015)) e seguindo uma arquitetura em camadas. Esta arquitetura torna o desenvolvimento da aplicação mais ágil, permite manter o código mais organizado, estruturado, com separação das responsabilidades adequada e facilita a testabilidade dos componentes. Considerando que não é previsto um grande número de utilizadores em simultâneo, a performance da aplicação não deverá ser afetada.

O sistema 2 é o dispositivo por onde se acede à aplicação, podendo ser um *smartphone* ou um computador. Nesta solução, o utilizador com recurso à câmara do telemóvel faz a leitura de um *QR code* colocado num armário de modo a filtrar as ferramentas que se encontraram disponíveis para requisitar e as já requisitadas ou em utilização (com alguma lógica seria possível saber quem requisitou o equipamento e dessa forma localizá-lo). Em seguida o utilizador seleccionaria a ferramenta a requisitar, introduzindo os dados necessários (como o período espectável de devolução) que serão posteriormente guardados na base de dados. Esta funcionalidade da requisição manual também estaria presente na solução 1.

#### 4.2.4 Seleção da ideia

Neste elemento do NCD é selecionada a ideia tendo em conta diversos fatores tais como os benefícios diretos, indiretos, requisitos a serem implementados e os custos associados a essa implementação. Dada a dificuldade do processo, é vulgar aplicar um método (formal ou não) de modo a averiguar qual a ideia que pode trazer o maior valor ao negócio (entende-se por valor o produto que atinge o desempenho necessário ao menor custo possível).

De forma a selecionar uma das soluções apresentadas na secção de geração de ideias, recorreu-se ao método AHP. O método de análise hierárquica (AHP) é um método de decisão multicritério criado pelo professor Thoma L. Saaty em 1980. Este método permite a utilização de critérios qualitativos e quantitativos no processo de avaliação. A principal ideia deste método é dividir o problema de decisão em níveis hierárquicos facilitando a compreensão e avaliação (Taherdoost, 2017). Este método é seccionado em 7 fases que serão descritas e postas em prática (foi também desenvolvida uma folha de cálculo em Excel para auxiliar na obtenção dos valores, sendo neste documento apresentados e explicados os resultados obtidos).

A **primeira fase** consiste na construção da árvore hierárquica de decisão (representada na Figura 22) na qual o problema é decomposto e estruturado num diagrama hierárquico (composto no mínimo por um objetivo, critérios e alternativas). Ao nível dos custos c de decisão definiram-se os seguintes:

- **Custo (C)**: associado à necessidade de adquirir equipamentos especializados para fazer a localização dos ativos;
- **Tempo de desenvolvimento (T)**: duração estimada para a finalização do projeto de tese e também relacionado com a complexidade de implementação da solução final;
- **Automatização do processo (A)**: capacidade da solução efetuar registos e gestão dos equipamentos sem necessidade de inputs por parte do utilizador.

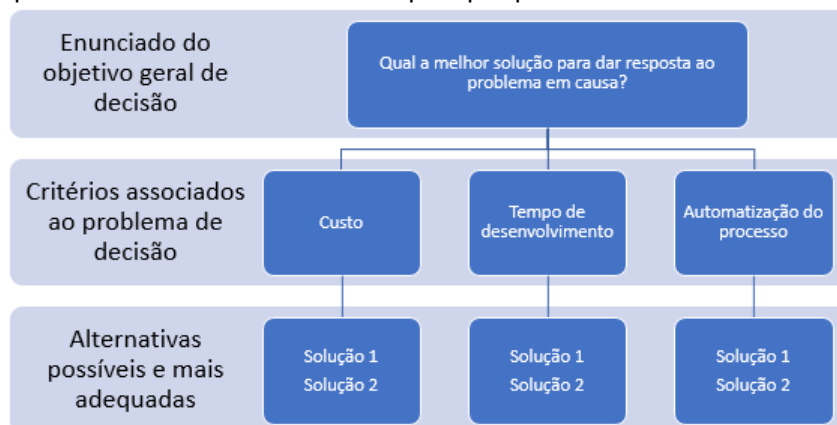


Figura 22 - Árvore hierárquica de decisão

Na **segunda fase** são estabelecidas prioridades entre os critérios definidos anteriormente por meio de uma matriz de comparação. O autor do modelo AHP definiu uma escala fundamental que consiste num conjunto de níveis de importância para comparar critérios. A Tabela 6 ilustra esta escala de níveis de importância, sua definição e explicação.

Tabela 6 - Escala fundamental - Níveis de importância de comparações (Saaty, 1990)

Nível de importância	Definição	Explicação
1	Igual importância	As duas atividades contribuem igualmente para o objetivo
3	Fraca importância	A experiência e o julgamento favorecem levemente uma atividade em relação à outra
5	Forte importância	A experiência e o julgamento favorecem fortemente uma atividade em relação à outra
7	Muito forte importância	Uma atividade é muito fortemente favorecida em relação a outra
9	Importância absoluta	A evidência favorece uma atividade em relação a outra com o mais alto grau de
2,4,6,8	Valores intermediários	Quando se procura uma condição de compromisso entre duas definições

Considerando os 3 critérios da estrutura hierárquica foi desenvolvida a matriz de comparação representada na Figura 23. Nesta o tempo de desenvolvimento tem 5 vezes mais importância que o custo e 4 vezes mais importância que a automatização do processo. Por outro lado, a automatização do processo tem prioridade 3 vezes superior ao custo (sendo necessário adquirir equipamentos especializados para fazer a localização).

$$MCC = \begin{matrix} & C & T & A \\ \begin{matrix} C \\ T \\ A \end{matrix} & \begin{bmatrix} 1 & 1/5 & 1/3 \\ 5 & 1 & 4 \\ 3 & 1/4 & 1 \end{bmatrix} \end{matrix}$$

Figura 23 - Matriz de comparação de critérios (MCC)

A **terceira fase** permite obter a prioridade relativa de cada critério. Para tal é necessário obter o vetor de prioridades, calculado com a base da média aritmética dos valores de cada linha da matriz normalizada (para isso cada valor da matriz da Figura 23 é dividido pelo total da sua respectiva coluna). O vetor próprio obtido está representado na Figura 24.

$$\begin{matrix} C \\ T \\ A \end{matrix} \begin{bmatrix} 0.104 \\ 0.665 \\ 0.231 \end{bmatrix}$$

Figura 24 - Vetor Próprio

A **quarta fase** consiste em calcular a razão de consistência (RC) que mede o quanto as suposições feitas na segunda etapa foram consistentes em relação a grandes amostras de juízos completamente aleatórios. Se o RC for superior a 0.1 as suposições não são confiáveis porque estão demasiado perto para o conforto de aleatoriedade e diz-se que os valores não são

consistentes. Para calcular a RC é necessário obter o valor  $\lambda_{max}$  que representa o maior valor próprio da matriz A obtido a partir da equação abaixo.

$$Ax = \lambda_{max}x \quad (4.1)$$

Após obter o valor de  $\lambda_{max}$  deve-se calcular o Índice de Consistência (IC) para em seguida obter a RC. O cálculo de IC é obtido recorrendo a fórmula apresentada abaixo.

$$IC = \frac{\lambda_{max} - n}{n-1} \quad (4.2)$$

Por fim, o cálculo da RC é obtido a partir da fórmula abaixo,

$$RC = \frac{IC}{IR} \quad (4.3)$$

em que o valor do Índice Aleatório (IR) é referente a um grande número de comparações par a par efetuadas e é obtido para matrizes quadradas de ordem  $n$  a partir da Tabela 7 - Valores de IR para matrizes quadradas de ordem  $n$ .

Tabela 7 - Valores de IR para matrizes quadradas de ordem  $n$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0.0	0.0	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.48	1.56	1.57	1.59

Tendo isto em conta, o valor obtido para  $\lambda_{max}$  foi de 3.086, permitindo obter um valor de IC de 0.043 para  $n=3$ , e obtendo assim um valor de RC de 0.075 (inferior a 0.1), podendo por isso concluir-se que os valores do vetor próprio (das prioridades relativas) estão consistentes.

A **quinta fase** consiste em contruir a matriz de comparação para cada critério e para cada alternativa que compõe a estrutura hierárquica do problema em questão (solução 1 – S1, solução 2 – S2) de modo a obter o vetor de prioridades de cada critério. A Figura 25, Figura 26, Figura 27 representam cada uma das matrizes de comparação para cada critério tendo em conta cada uma das soluções alternativas indicadas.

$$C = \begin{matrix} & S1 & S2 \\ S1 & [1 & 1/7] \\ S2 & [7 & 1] \end{matrix}$$

Figura 25 – Matriz de Comparação de custo

$$T = \begin{matrix} & S1 & S2 \\ S1 & [1 & 1/7] \\ S2 & [7 & 1] \end{matrix}$$

Figura 26 - Matriz de comparação de tempo de desenvolvimento

$$A = \begin{matrix} & S1 & S2 \\ S1 & [1 & 1/9] \\ S2 & [9 & 1] \end{matrix}$$

Figura 27 - Matriz de comparação de automatização do processo

Os resultados obtidos do cálculo dos vetores de prioridade calculados para cada critério são agrupados na matriz de prioridade representada na Figura 28.

$$MP = \begin{matrix} & C & T & A \\ S1 & [0.125 & 0.125 & 0.900] \\ S2 & [0.875 & 0.875 & 0.100] \end{matrix}$$

Figura 28 – Matriz de prioridade (MP)

Na **sexta fase** obtém-se as prioridades compostas das alternativas, multiplicando os valores da matriz de prioridade pelo vetor próprio dos critérios (que representa as prioridades dadas a cada critério). A Figura 29 representa o ranking das duas soluções propostas.

$$RS = \begin{matrix} S1 [0.304] \\ S2 [0.696] \end{matrix}$$

Figura 29 – Ranking das soluções

A **sétima fase** é a escolha da solução. Pode-se verificar que a solução 2 é que tem maior classificação, logo será esta a considerada mais indicada para desenvolver. A solução 1 apresenta maior prioridade relativa apenas no critério de automatização do processo, mas é mais custosa e implica mais tempo de desenvolvimento.

#### 4.2.5 Definição da conceção

Este é o último elemento a ser considerado do NCD e consiste no desenvolvimento do conceito de negócio. Este deve ter em conta algumas variáveis como as necessidades dos clientes, os requisitos de investimento, a incerteza do sucesso do projeto, entre outros.

Uma das metodologias utilizadas em desenvolvimento e conceção de produtos é o Quality Function Deployment (QFD). Este permite identificar e classificar quais são os requisitos mais importantes dos clientes que necessitam de ser implementados. O QFD começou a ser utilizado pela Mitsubishi em 1972 tendo sido refinado desde então para considerar funcionalidade do produto, pontos de falha e tecnologias a usar. A metodologia pode ser melhor compreendida definindo cada termo do QFD (manufacturing Group, 2010):

- *Quality*: Reunir os requisitos do cliente;
- *Function*: Focar a atenção no que deve ser feito;
- *Deployment*: Quem vai executar e quando;

Essa abordagem permite produtos de alta qualidade e lançamentos mais rápidos para o mercado a um custo reduzido, além de possibilitar construir produtos que atendam às

demandas específicas dos clientes e controlem o processo de desenvolvimento do produto (garantindo uma melhor qualidade final do produto) (manufacturing Group, 2010).

A House of Quality (HOQ) é uma matriz que pode ser utilizada para fazer a análise QFD. Esta representa as informações obtidas dos clientes a partir dos métodos aplicados de engenharia de requisitos. Não existe um formato standard para a HOQ, mas uma estrutura possível pode ser visualizada na Figura 30 que vai ser explicada e usada como referência para criar a HOQ para a solução proposta.

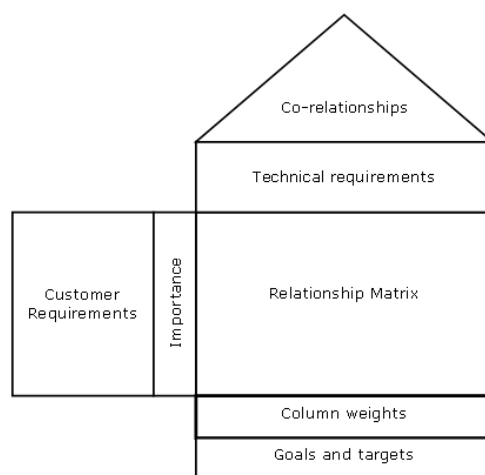


Figura 30 - Estrutura da matriz House of Quality (Corporation, 2008)

O preenchimento da HOQ começa pelos requisitos do cliente do lado esquerdo. Esta secção normalmente inclui também uma escala que reflete a importância dos requisitos levantados. Os requisitos técnicos são estabelecidos em resposta aos requisitos do cliente e colocados na secção *Technical requirements*. A área com o nome *Relationship Matrix* mostra a conexão entre os requisitos técnicos e os requisitos do cliente (estes têm uma pontuação de 0,1,3, ou 9, sendo que 0 significa que não existe relação entre a funcionalidade e a tecnologia, e 9 têm uma relação muito forte). A área de *column weights* é a soma dos valores na coluna correspondente a um dado requisito multiplicado pelo ajuste de peso de importância do desenvolvimento de um dado requisito (isto permite perceber quais são os requisitos técnicos mais importantes desenvolver para responder aos requisitos do cliente). A secção de *goals and targets* (que representa os valores alvo para os requisitos técnicos) não foi usada dado que a sua avaliação é qualitativa e será feita com recurso a uma metodologia própria para avaliar a implementação no próximo capítulo. A área das *co-relationships* mostra as correlações entre os requisitos técnicos (uma correlação positiva indica que ambos os requisitos técnicos podem ser melhorados ao mesmo tempo, enquanto uma correlação negativa indica que melhorar um requisito técnico irá piorar o outro). A Figura 31 apresenta a HOQ preenchida para a solução proposta.

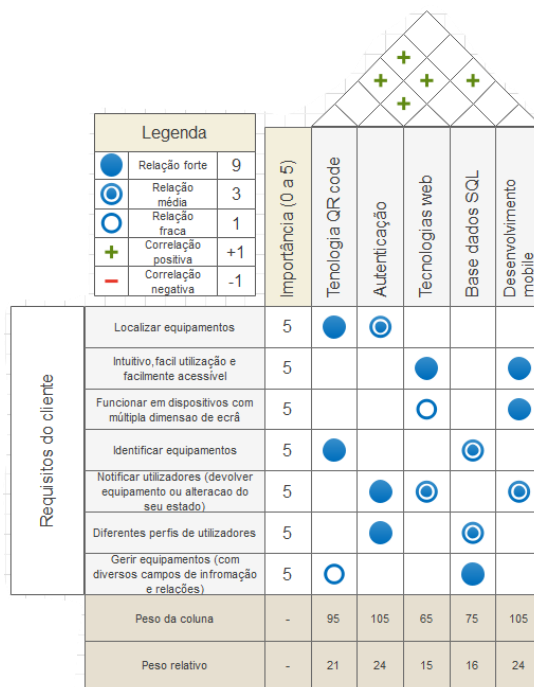


Figura 31 - House of Quality para a solução proposta desenvolver

Da análise da HOQ desenvolvida percebe-se que a autenticação (para identificar os utilizadores e restringir os acessos as funcionalidades do sistema) e o desenvolvimento de uma aplicação que funcione e se adapte a dispositivos móveis são os aspetos técnicos mais importantes a ter em conta quando se implementar a solução. A tecnologia de QR code irá permitir identificar os equipamentos pelo que também terá bastante importância. Existem também algumas correlações positivas entre os aspetos técnicos, como por exemplo a autenticação e a base dados SQL, dado que a forma como a informação é armazenada facilitará na obtenção da mesma de forma estruturada para fazer a autenticação do utilizador no sistema.

### 4.3 Valor

Tendo analisado qual a melhor solução a ser implementada é preciso também perceber que valor a implementação desta solução poderá trazer para o cliente. Segundo Woodall (Woodall, 2003), o valor para o cliente (*value for the customer*) corresponde a uma perceção pessoal (do lado da demanda) da associação deste com a oferta de uma organização. Esta pode ocorrer através da redução de sacrifício, presença de benefícios (percebidos como atributos e resultados) ou a combinação de ambos. A Tabela 8 mostra os benefícios (sendo os atributos a aplicação a desenvolver e os resultados o que se espera conseguir obter com esta solução) e sacrifícios que a implementação desta solução trará.

Tabela 8 – Relação benefícios e sacrifícios

Benefícios		Sacrifícios
Atributos	Resultados	
Solução desenvolvida (intuitiva, fácil utilização, disponível em vários tipos de dispositivos)	Diminuição do tempo de procura das ferramentas	Necessidade de formação e criação de hábitos na utilização da solução
	Saber necessidades de substituição de equipamento antes dele ser necessário	Atualizações frequentes do estado dos equipamentos são necessários para a solução ser eficaz
	Obter informação detalhada sobre a utilização do equipamento	Registo inicial da informação dos equipamentos no sistema pode ser <b>tedioso</b>
	Obter informação sobre a localização do equipamento	

A percepção de valor (*perceived value*) será o resultado da avaliação geral do consumidor sobre a utilidade do produto e será feito com base no que é recebido (os benefícios) contra aquilo que é dado (os sacrifícios) (Ravald and Grönroos, 1996).

#### 4.4 Proposta de Valor

A proposta de valor permite apresentar ao cliente o valor do produto (ou serviços) e ajudar a clarificar o valor inerente e exclusivo fornecido.

Para o produto em causa a proposta de valor consiste no desenvolvimento de uma solução para a equipa técnica da organização que permita de forma fácil, rápida e intuitiva saber onde um equipamento se encontra, perceber com antecedência a carência de substituir um ativo perdido ou estragado ou até se existe a necessidade de adquirir mais ferramentas (devido à sua elevada frequência requisição). Estas ideias encontram-se resumidas no modelo de proposta de valor de Osterwalder apresentado na Figura 32.

A proposta de valor também pode ser aplicada a uma organização inteira. O *Business Model of Canvas* descreve em nove blocos qual o modelo de negócio da aplicação na organização. Este modelo pode ser consultado na secção Anexos.

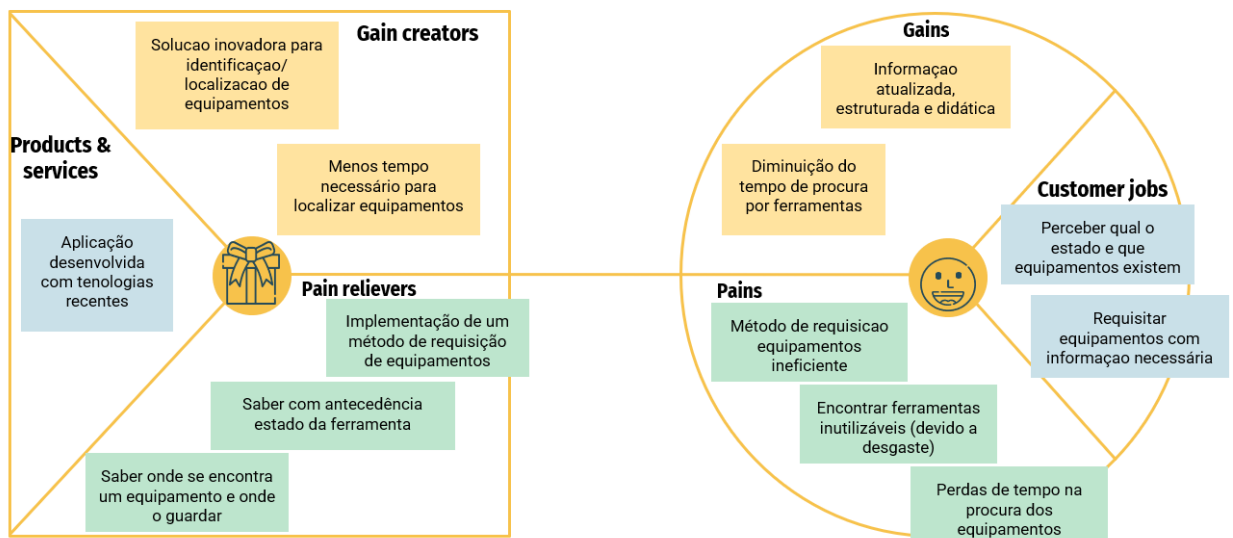


Figura 32 - Proposta de valor segundo modelo de *value proposition* de Osterwalder

## 4.5 Sumário

Neste capítulo foi apresentado o modelo de NCD e como este pode ser aplicado para desenvolver e avaliar diferentes abordagens para a resolução do problema. Deste modo foi feita a análise da oportunidade, a geração de ideias (onde foram propostas duas soluções com base no estado de arte desenvolvido no capítulo 3), a seleção da ideia recorrendo ao método AHP (de modo a perceber qual a solução que melhor responde às necessidades do negócio), e por fim a definição do conceito (recorrendo à metodologia QFD e à matriz HOQ). Por fim foi desenvolvida uma proposta de valor para a solução recorrendo ao *Value Proposition Canvas* de Osterwalder.

É assim possível responder as questões levantadas inicialmente:

### Que soluções existem para resolver o problema?

Foram criadas duas soluções tendo em conta as tecnologias identificadas no capítulo 3.3.2. Ambas combinam tecnologias OT e IT, sendo a solução 1 mais automatizada, mas também mais complexa e dispendiosa de implementar que a solução 2. Ambas as soluções permitem responder às necessidades do projeto.

### Qual a solução que melhor se enquadra às necessidades do projeto?

Dada a dificuldade deste processo foi aplicado o método formal AHP para averiguar qual a ideia que pode trazer o maior valor ao negócio. A solução 2 foi a que apresentou melhor classificação tendo em conta 3 critérios de decisão: custo de implementação, o tempo disponível para desenvolvimento e a automatização do processo.

# 5 Arquitetura e desenho da aplicação

Neste capítulo será apresentado com maior detalhe a arquitetura e desenho da aplicação que irá resolver o problema da gestão dos equipamentos de laboratório. Pretende-se desta forma dar resposta ao *Outcome 4* - desenhar uma solução para o problema adotando ciências e boas praticas de engenharia informática. O desenho e arquitetura apresentados neste capítulo têm por base a idealização da solução 2 seleccionada no capítulo 4 de acordo com os critérios nele definidos.

## 5.1 Descrição geral

Nesta secção vão ser detalhadas as funcionalidades bem como os requisitos funcionais que precisam ser implementados para dar resposta ao problema, os requisitos não funcionais, restrições de desenvolvimento, atributos de qualidade e estilo arquitetural.

### 5.1.1 Perspetiva do produto e funcionalidades

A solução de software prevista será composta apenas por uma plataforma que poderá ser acedida por diferentes utilizadores pertencentes apenas à organização (para entidades externas será necessário criar um perfil para poderem aceder com as respetivas restrições).

O sistema terá de dar resposta às necessidades dos utilizadores, sendo a localização, identificação, requisição e gestão dos equipamentos as mais importantes. A organização já possui um conjunto de equipamentos no laboratório que necessitaram de ser inicialmente guardados no sistema para posteriormente poder realizar a requisição dos mesmos.

Tendo em conta que não é expectável um elevado número de pedidos ao sistema em simultâneo (apenas alguns elementos da equipa técnica e esporadicamente elementos da unidade que necessitam de efetuar alguma atividade laboratorial) optou-se por desenvolver um sistema acessível através de qualquer navegador da Internet. Isto facilitará o seu acesso permitindo uma maior adesão ao mesmo. Assim a requisição dos equipamentos poderá ser

efetuada por computador ou por dispositivos móveis (como por exemplo *smartphones*) a partir de uma interface intuitiva e que rapidamente permita ao utilizador fazer a operação pretendida (requisitar ou apenas consultar um equipamento).

Optou-se pelo desenvolvimento desta solução de software dado que é possível responder às necessidades do projeto de uma forma ágil (limitada pelo tempo de desenvolvimento), com baixo custo (não há necessidade de aquisição de equipamentos especializados para realizar a localização dos equipamentos ou alterações nas infraestruturas do laboratório), sem comprometer questões de performance e manutenção. A solução desenvolvida seguirá também bons princípios de desenvolvimento de software (descritos neste capítulo), de modo a permitir extensão de novas funcionalidades do sistema (como a adição de um sistema externo que automatize a localização dos equipamentos) sem necessidade de fazer alterações ao código implementado (*Open/Close Principle – OCP*).

### **5.1.2 Restrições de implementação e design**

Foi realizada uma entrevista com o responsável da informática para perceber melhor quais seriam as restrições na implementação do sistema (as respostas obtidas na entrevista podem ser consultadas na secção Anexos).

Da análise dessa entrevista foi possível concluir que a solução tem de ter uma interface gráfica capaz de ser executada nos navegadores mais modernos e a capacidade de se adaptar a dispositivos móveis.

O design da solução e a sua implementação deve ter em conta a compatibilidade com o sistema operativo do servidor (Linux), tecnologias (tendo sido manifestada a preferência pela utilização de MySQL para a base de dados) e linguagens de programação que sejam utilizadas em outras aplicações existentes de modo a tornar a sua manutenibilidade mais fácil.

Relativamente a hardware (armazenamento e memória no servidor) não foram impostas restrições. O recurso a componentes externos de utilização não gratuita tem de ser devidamente justificado.

Quanto a questões de segurança, foi referido que as palavras-passe dos utilizadores deveriam ser encriptadas.

### **5.1.3 Diagramas de caso de uso**

Os diagramas de caso de uso descrevem uma sequência de interações entre um sistema e um ator externo tendo como resultado a capacidade do ator conseguir alcançar um dado resultado. Os nomes dos casos de uso são sempre escritos na forma de um verbo seguido por um objeto. Os diagramas de caso de uso fornecem uma representação visual de alto nível dos requisitos do cliente. A Figura 33 ilustra o diagrama elaborado para a solução a implementar sendo os atores descritos na subsecção seguinte. As linhas de cada ator conectam-se com os casos de

uso com os quais o ator interage. Estas também indicam que ele é o ator principal desse caso de uso. Cada caso de uso tem uma referência que o relaciona com a definição mais detalhada do requisito descrita em Anexos com o mesmo identificador.

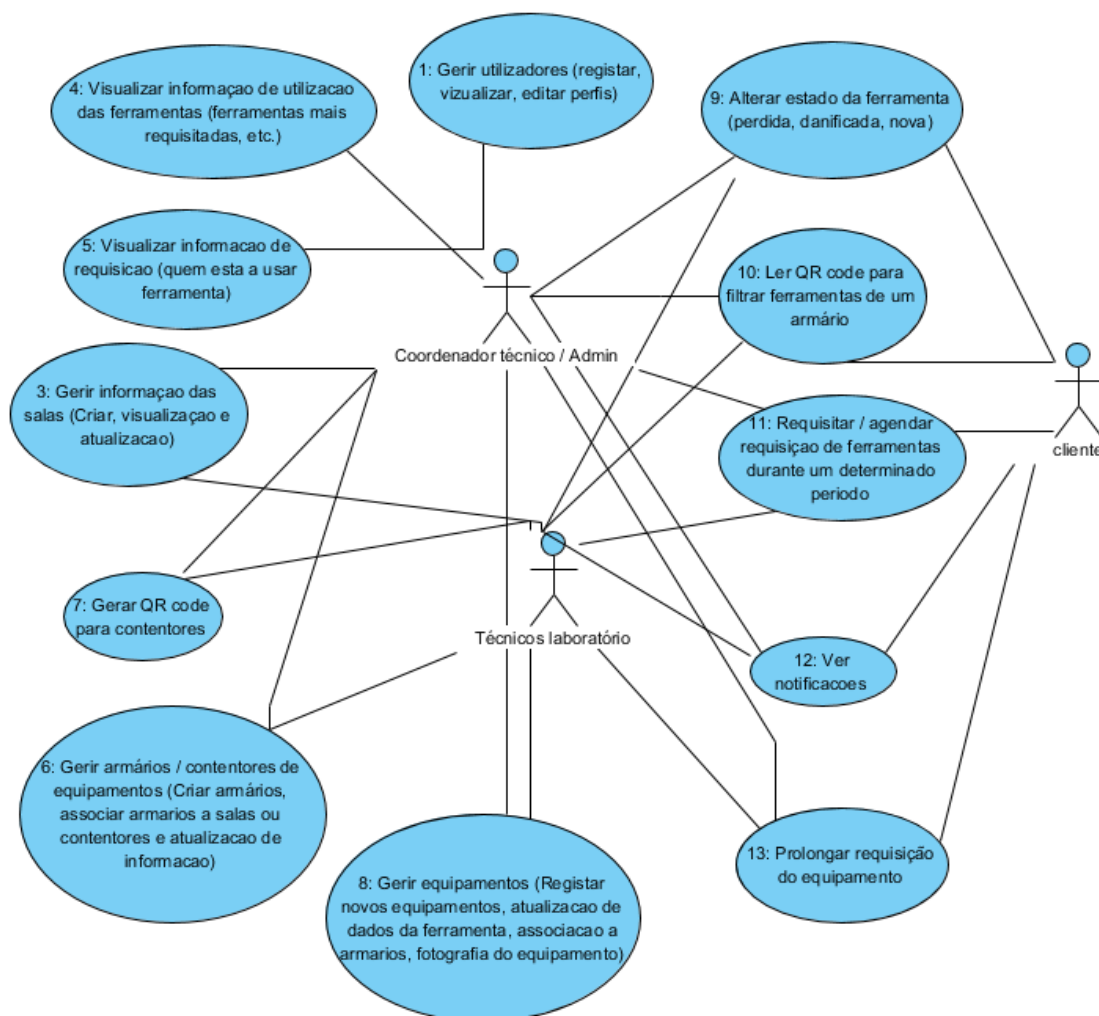


Figura 33 – Diagrama de casos de uso

#### 5.1.4 Stakeholders

O *stakeholder* é uma pessoa, grupo ou organização que está ativamente envolvida num projeto e é afetada ou pode influenciar o seu processo e resultado. As partes interessadas no desenvolvimento podem ser internas ou externas à equipa do projeto e à organização (Wieggers and Beatty, 2013). A Tabela 9 mostra os *stakeholders* identificados e cujos perfis precisam ser definidos na plataforma dado que cada um tem diferentes responsabilidades nesta.

Tabela 9 – Perfis dos utilizadores

<b>Cargo</b>	<b>Perfil</b>	<b>Funções</b>
<b>Coordenador técnico</b>	Responsável pela equipa de projetos	Este stakeholder tem acesso a todas as funcionalidades da plataforma e apenas ele pode visualizar informações mais sensíveis (como quem está a utilizar a ferramenta e algumas informações de gestão)
<b>Técnico laboratório</b>	Executantes das atividades dos	Responsável pela gestão dos equipamentos do laboratório e também faz requisição de equipamentos no sistema
<b>Cliente</b>	Elementos da equipa que pontualmente necessitam e ir ao laboratório	Apenas faz requisição dos equipamentos de laboratório e pode alterar os estados da ferramenta (perdida, estragada ou bom estado)

## 5.2 Requisitos do sistema

Nesta secção vão ser apresentados os requisitos funcionais identificados ao longo das entrevistas realizadas com os stakeholders. A análise mais detalhada de cada requisito encontra-se em Anexos e seguiram uma estrutura com os seguintes elementos essenciais (Wieggers and Beatty, 2013):

- Um identificador exclusivo e um nome sucinto que indica o objetivo do utilizador;
- Uma breve descrição textual com o propósito do requisito;
- Uma condição de *trigger* que inicia a execução do requisito;
- As pré-condições que devem ser satisfeitas antes de iniciar o caso de uso;
- Uma ou mais pós condições que descreve o estado do sistema depois de ter concluído o requisito com sucesso;
- Uma lista numerada de etapas que mostra a sequência de interações do ator com o sistema e que leva das pré-condições às pós-condições.

Os requisitos funcionais identificados tiveram por base os casos de uso identificados anteriormente. Estes têm uma importância elevada no desenvolvimento da aplicação (pois garantem a funcionalidade do sistema). Os requisitos funcionais são os seguintes:

- REQ-01: Gerir utilizadores;
- REQ-02: Autenticação de utilizadores;
- REQ-03: Gerir salas;
- REQ-04: Visualização de informação de utilização dos equipamentos;
- REQ-05: Visualização da informação de requisição;
- REQ-06: Gerir contentores de equipamentos;
- REQ-07: Gerar QR code para contentores;
- REQ-08: Gerir equipamentos;
- REQ-09: Alteração do estado da ferramenta;
- REQ-10: Ler QR code para filtrar equipamentos dentro de um contentor;

- REQ-11: Requisar equipamentos durante um determinado período de tempo;
- REQ-12: Ver notificações;
- REQ-13: Prolongar requisição do equipamento;

### 5.3 Processos de negócio

Embora muitos casos de uso possam ser descritos em texto simples, um modelo do processo de negócio é uma maneira útil de representar visualmente o fluxo lógico de um caso de uso complexo, conforme ilustrado na Figura 34 para o processo de requisição de equipamento. Este tipo de modelo tem como característica mostrar pontos de decisão e as condições que causam uma ramificação do fluxo normal para um fluxo alternativo com apenas uma representação visual.

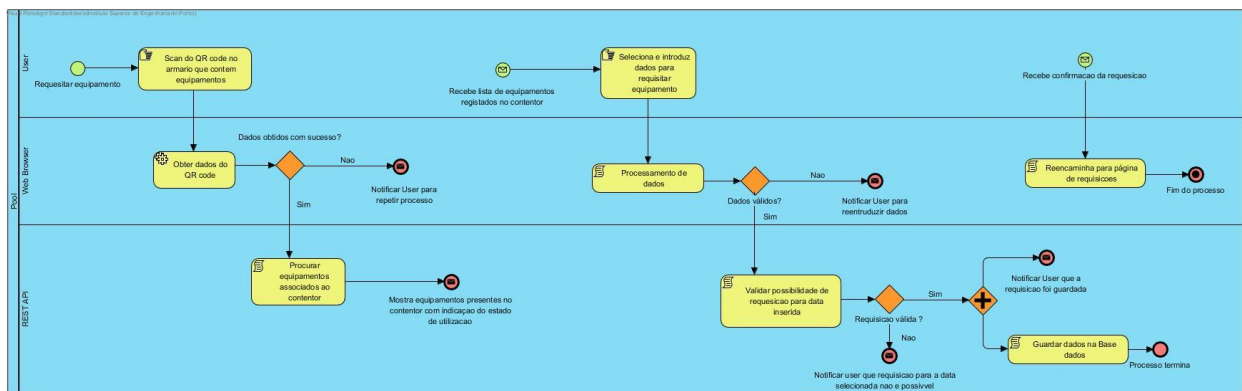


Figura 34 – Modelagem do processo da requisição de equipamento

Para este caso o processo inicia com utilizador a ler o QR code afixado num armário para filtrar os equipamentos disponíveis para requisição. Se o QR code for decodificado e dados válidos extraídos, é apresentada a lista dos equipamentos que o utilizador pode selecionar para requisitar. Após a inserção dos dados necessários estes ficam registados na base de dados e o utilizador é notificado do sucesso da operação.

### 5.4 Modelo de domínio

O modelo de domínio, como o próprio nome indica, é uma forma de conceitualizar o domínio e incorporar o comportamento dos dados (Sitefane, 2006). Nesta secção é apresentado o modelo domínio da aplicação (representado na Figura 35). A entidade representada com mais interligações é a ferramenta (tool) visto ser o elemento central para o qual há uma necessidade de gerir a informação. Esta tem uma cardinalidade de 1 para a entidade case que é responsável por armazenar vários equipamentos (podendo ser um estojos ou um armário). A entidade case pode conter outras dentro dela daí a cardinalidade de 0 para muitos para ela própria. A entidade room guarda várias cases e representa salas ou laboratórios. A entidade log requisition armazena logs (eventos computacionais) de ações feitas, como novas requisições, prolongar

tempos de requisição, entre outros. A entidade requisition regista todos as requisições que ocorreram. A entidade accessories permite relacionar diferentes tools (uma ferramenta pode ser acessória de outras).

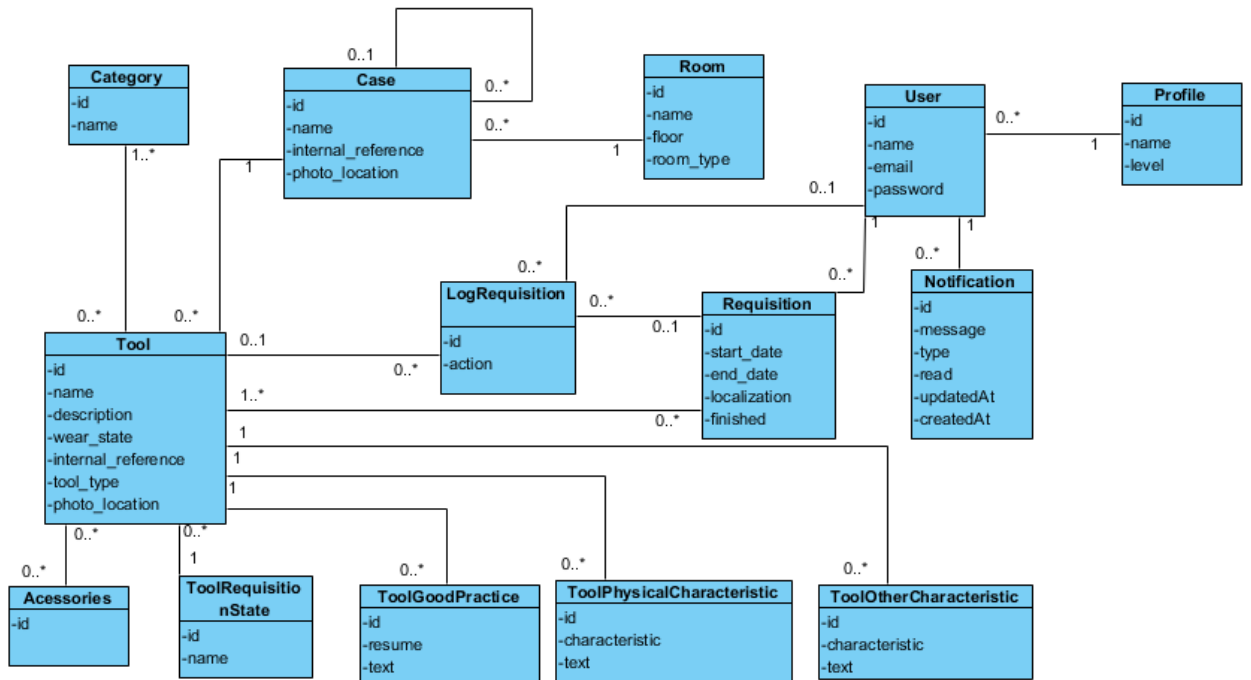


Figura 35 - Modelo de domínio da solução

## 5.5 Requisitos de interface externa

Esta seção fornece informações para garantir que o sistema comunica de forma adequada com os utilizadores e com elementos externos de hardware ou software.

### Interfaces com os utilizadores

Ao nível das interfaces gráficas foi afirmado que deveria seguir padrões semelhantes às aplicações já desenvolvidas na organização. Alguns destes são os esquemas de cores, grafismos do logótipo e avisos de direitos autorais e privacidade similares. Outro requisito foi relativo a dimensão dos ecrãs. A solução deverá ser capaz de funcionar em ecrãs de computadores e de smartphones.

### Interfaces de Software

Em termos de conexões deste produto com outros componentes de software foi identificado a necessidade de armazenar a informação numa base dados MySQL já existente num servidor. Outras ferramentas e componentes externos poderão ser utilizados, contudo se estes forem de utilização não gratuita têm de ser devidamente justificados.

A interação com o utilizador poderá ser através de uma plataforma *web*. Dado que a aplicação tem por base a utilização de tecnologias *web* (o que conduz a utilização do protocolo HTTP) a opção pelo padrão arquitetural cliente-servidor é a que fará mais sentido (uma vez que é baseado neste protocolo e permite compartimentar a aplicação). Este protocolo permite que pedidos HTTP padrão (como GETS e PUTS) sejam utilizados para consultar e modificar o estado do sistema (para além de permitir a utilização de outros padrões como JavaScript Object Notations - JSON) (Somasegar, 2008, p. 0).

### Interfaces de Hardware

A aplicação será implantada num servidor da organização já existente. Os utilizadores poderão aceder ao sistema a partir de computadores ou *smartphones*.

### Interfaces de comunicação

Como referido anteriormente, poderá ser utilizado um *web* browser para aceder ao sistema, utilizando protocolo HTTP para interação com o mesmo. É necessária a encriptação das palavras-passe dos utilizadores de modo que estas não possam ser tornadas públicas.

## 5.6 Requisitos não funcionais

Os atributos de qualidade são preocupações transversais (podendo ter impacto nas camadas ou *tiers* de toda a aplicação) e afetam o desempenho em tempo de execução e a experiência do utilizador. Estes atributos estão correlacionados com o design geral do sistema, mas também podem ser mais específicos, como por exemplo o tempo de design, o tempo de execução ou focados nas necessidades do utilizador. Os atributos de qualidade são importantes para a usabilidade geral, desempenho, confiabilidade e segurança das aplicações de software. Estes requisitos serão avaliados pela metodologia desenvolvida no capítulo 7.

Vão ser assim especificados alguns atributos de qualidade, também chamados de requisitos não funcionais, que a solução desenvolvida deverá ter:

- **Viabilidade:** define a proporção de tempo que o sistema está operacional. O sistema desenvolvido deve estar em funcionamento pelo menos durante o horário trabalho.
- **Desempenho:** capacidade de resposta para executar uma ação dentro de um determinado intervalo de tempo (menos de 5 segundos). Dado as tecnologias em causa (*web*) espera-se que a aplicação seja rápida na resposta havendo apenas alguma latência nas respostas do servidor.
- **Segurança:** um sistema seguro deverá proteger a informação e evitar modificações e alterações de perfis não autorizados (assegurado pelo login na aplicação e encriptação das palavras-passe na base de dados).
- **Usabilidade:** a interface do utilizador e o fluxo da utilização da aplicação precisam ser intuitivas e eficientes, para que os utilizadores tenham facilidade a realizar as tarefas e também ser capaz de se adaptar a diferentes dimensões de dispositivos.

- **Manutenibilidade:** deve ser fácil realizar alterações nos componentes de software.
- **Testabilidade:** facilidade em criar critérios de teste para o sistema.

Para além dos atributos de qualidade que a solução deverá ter, deverão ser tidos em conta alguns princípios chave e diretrizes de design que ajudem na criação da arquitetura e facilite a manutenibilidade da mesma. Estes são (Somasegar, 2008, p. 0):

- **Separation of concerns:** dividir a aplicação em funcionalidades distintas que se sobreponham o mínimo possível;
- **Single Responsibility Principle:** cada componente ou módulo deve ser responsável apenas por uma funcionalidade específica;
- **Don't Repeat Yourself (DRY):** uma funcionalidade específica deve ser fornecida por um componente apenas (não deve ser duplicada);
- **Avoid doing a big design upfront:** quando há requisitos que não estão claros ou há a possibilidade do design evoluir ao longo do tempo deve-se evitar fazer implementações prematuras.

## 5.7 Product Backlog

O *product backlog* de um projeto que segue uma metodologia de desenvolvimento ágil (com entregas contínuas e funcionalidades de software aos *stakeholders* para perceber se os requisitos estão a ser bem implementados), contém uma lista dos requisitos funcionais que são necessários implementar para que a solução se enquadre com as necessidades do cliente e permite determinar qual o esforço esperado no seu desenvolvimento (por exemplo, estimando quantas horas um determinado requisito pode demorar a implementar). A Tabela 10 ilustra o *product backlog* desenvolvido.

Tabela 10 – *Product backlog*

Ident. do requisito	Descrição	Estimativa horas
REQ-01	Gerir utilizadores	24h
REQ-02	Autenticação de utilizadores	15h
REQ-03	Gerir salas	10h
REQ-04	Visualização de informação de utilização dos equipamentos	15h
REQ-05	Visualização da informação de requisição	15h
REQ-06	Gerir contentores de equipamentos	15h
REQ-07	Gerar QR code para contentores	10h
REQ-08	Gerir equipamentos	15h
REQ-09	Alteração do estado da ferramenta	5h
REQ-10	Ler QR code para filtrar equipamentos dentro de um contentor	15h
REQ-11	Requisitar equipamentos durante um determinado período de tempo	24h
REQ-12	Ver notificações	10h
REQ-13	Prolongar requisição do equipamento	24h
Total		212h

## 5.8 Arquitetura sugerida

A arquitetura de uma aplicação cria a ligação entre os requisitos de negócio e os requisitos técnicos a partir da análise dos casos de uso seguida de um planeamento para fazer a implementação desses em software. O objetivo da arquitetura é identificar os requisitos que mais impacto têm na estrutura da aplicação. Uma boa arquitetura reduz os riscos de negócio associados à construção de uma solução técnica. Um bom design deve permitir flexibilidade suficiente para lidar com desvios que poderão ocorrer ao longo do tempo nas tecnologias de hardware e software ou até alterações nos casos de uso. Devem também ser tidos em conta os *tradeoffs* entre a implementação dos atributos de qualidade e os requisitos do utilizador e do negócio. Tendo isto em mente, uma boa arquitetura deve (Somasegar, 2008):

- Expor a estrutura do sistema, mas ocultar detalhes de implementação;
- Realizar todos os cenários dos casos de uso;
- Tentar resolver as dificuldades e preocupações de todos os stakeholders;
- Implementar os requisitos funcionais e de qualidade.

A arquitetura idealizada para este sistema encontra-se representada em seguida por diferentes vistas respeitantes ao modelo 4 + 1 e para diferentes níveis de granularidade. A Figura 36 representa a vista física (ilustra como se vão alojar os componentes da aplicação) e é constituída por 2 sistemas. O <<device>> *UserClient* é o dispositivo por onde se vai aceder à aplicação (a partir de um *web Browser*), podendo este ser um *smartphone* ou um computador. O <<device>> *Servidor* representa o servidor que vai alojar a solução desenvolvida. Esta aplicação é contruída seguindo uma arquitetura em camadas (*Presentation Layer, Business Layer e Data Layer*). A escolha por este tipo de arquitetura deve-se aos benefícios intrínsecos a ela, sendo eles:

- Desenvolvimento ágil da aplicação;
- Alta coesão (cada camada contém funcionalidade relacionada diretamente com as tarefas dela);
- Baixo acoplamento (a comunicação entre camadas é baseada na abstração que permite acoplamento baixo entre as camadas)
- Boa testabilidade (o facto de ter camadas bem definidas permite testar individualmente cada componente)
- Alto isolamento (permite isolar atualizações de tecnologia em determinadas camadas para reduzir o risco e o impacto no sistema geral)
- Independência (remove a necessidade de considerar problemas de hardware e implantação e futuras dependências com outras interfaces)

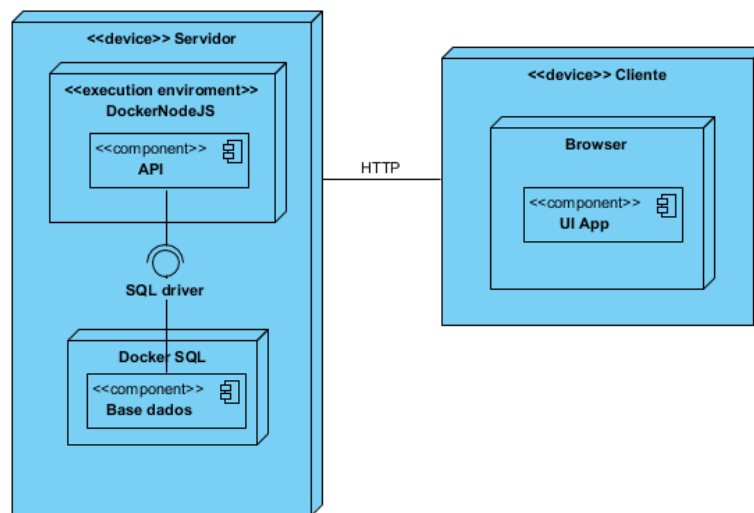


Figura 36 - Vista física nível 2

A solução desenvolvida representada esquematicamente na Figura 37 por uma vista lógica (ilustra como a informação é consumida pelos diferentes componentes) está dividida numa camada de apresentação (relacionada com a interface visual do utilizador e representada pela *UI App*) numa camada de negócios responsável por processar as regras de negócio

(representada pela API) e uma camada de dados (relacionada com o acesso aos dados e representada pela base dados).

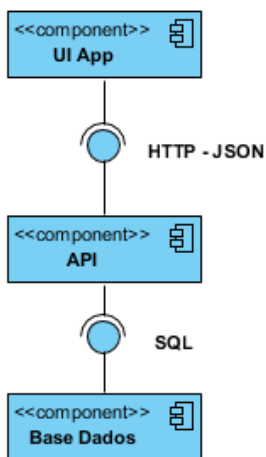


Figura 37 – Vista lógica de nível 2

A Figura 38 ilustra como a aplicação de *back-end* está idealizada, como vai receber os pedidos do front-end, fazer processamento e acesso à base de dados para depois devolver uma resposta.

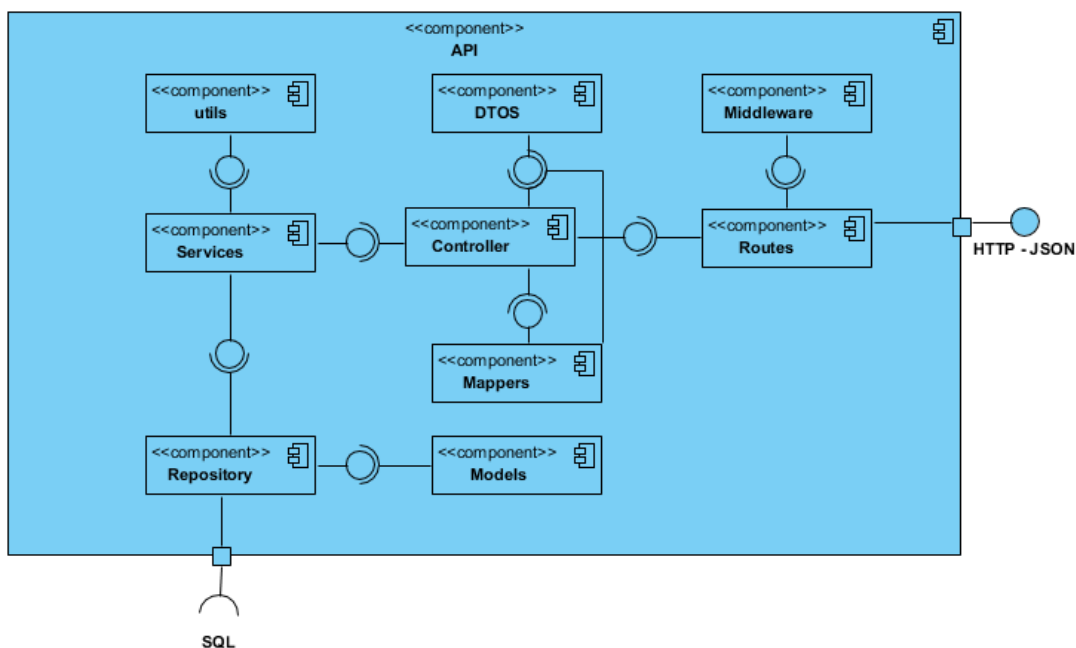


Figura 38 -Vista lógica de nível 3 – *Back-end*

O diagrama da Figura 38 também ilustra a interação entre os diferentes componentes. As responsabilidades de cada um deles são as seguintes:

- **Routes:** é a camada que recebe os pedidos exteriores e direciona os pedidos para o controller respetivo;

- **Middleware:** componente responsável por garantir a segurança do sistema. Protege e impede modificação não autorizada dos dados ou divulgação de informação com recurso a autenticação do utilizador no sistema;
- **Controller:** é a camada que vai receber os pedidos dos outros sistemas e reencaminha para os serviços. Faz validação da informação recebida do exterior e garante apenas a passagem de informação no estado válido para os restantes componentes (senão exceções são levantadas);
- **DTOs (Data Transfer Objects):** Um objeto que armazena os dados transportados entre os processos, reduzindo o número de chamadas de método necessárias;
- **Mapper:** faz o mapeamento entre os objetos e a estrutura da base dados (utilizado para mover dados de uma estrutura para a outra, mantendo as independentes);
- **Services:** contêm a lógica de negócio, fluxo de negócio e entidades e componentes do negócio;
- **Repository:** acessa os dados e comunica aos Services;
- **Models:** tem as classes de domínio que foram definidas no modelo de domínio
- **Utils:** funções que executam outras funcionalidades e que são partilhadas por outros serviços (gravação de uma fotografia, por exemplo).

No diagrama de vista lógica de nível 3 representado na figura Figura 39 é possível ver com maior pormenor a organização da aplicação UI (que poderá ser acedida a partir de um browser e realizará pedidos ao *back-end*).

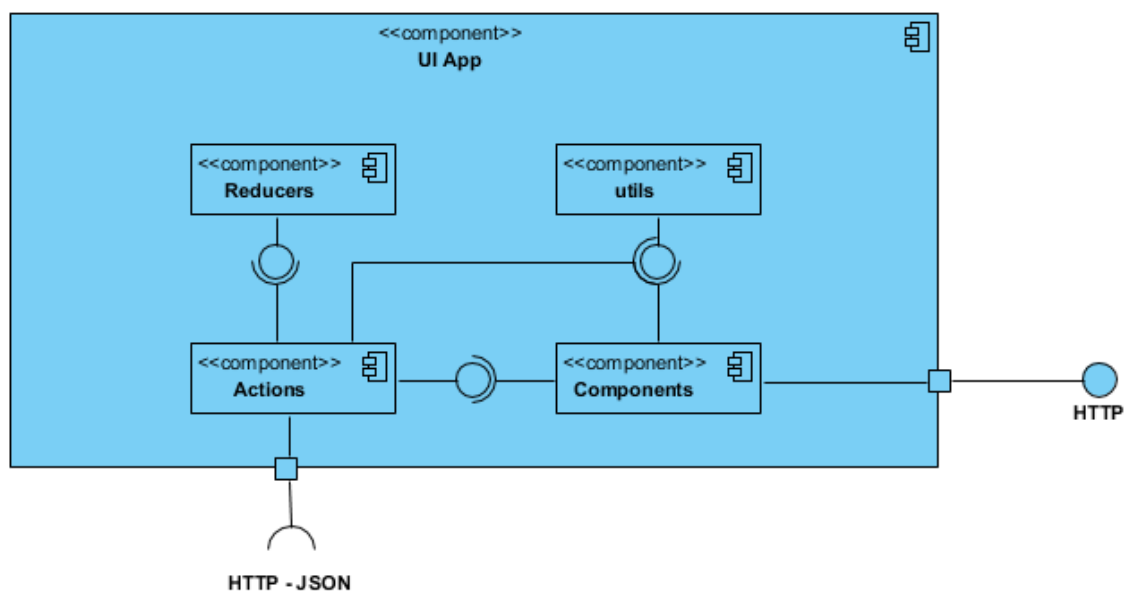


Figura 39 - Vista lógica de nível 3 – Front End

As responsabilidades de cada um dos componentes ilustrados é a seguinte:

- **Componentes:** Disponibilizam as interfaces gráficas para o utilizador poder visualizar a informação obtida das respostas do *back-end*. Pode conter alguma lógica para limitar

visualizações ou contruir a interface gráfica. Fazem pedidos ao *back-end* a partir das Actions.

- **Utils:** funcionalidades genéricas que podem ser partilhadas por diferentes componentes;
- **Actions:** Realiza pedidos ao back-end e prepara as respostas obtidas (podendo ou não libertar exceções).
- **Reducers:** armazena a informação obtida numa estrutura definida e disponibiliza a mesma a todos os componentes que necessitam de atualizar o seu estado.

A estruturação da aplicação em componentes específicos para realizar as diferentes operações (*Single Responsibility Principle*) permite não só ter o código mais organizado como facilita a sua manutenção, a adição de futuras funcionalidades e permite mais facilidade na testabilidade.

Os diagramas de sequência que ilustram a ordenação de operações entre os componentes serão apresentados no capítulo 6, para ajudar na explicação da implementação da solução. Os requisitos levantados e documentados nos anexos contêm estes diagramas.

## 5.9 Desenho da interface *front-end*

Previamente à implementação do sistema, foram também desenvolvidas *mockups* de *front-end* tendo em conta as funcionalidades anteriormente definidas. Estas *mockups* representam um design estático da página *web* e alguns elementos a serem considerados no design final. Não contêm a funcionalidade do *web site* e pretendem dar uma ideia de como será a configuração final da página (podendo conter alguns dados a server de *placeholder* para ter uma melhor representação) (HUFFORD, 2022).

Vão ser apresentadas algumas das *mockups* desenhadas e no capítulo 6 comparadas com o resultado final obtido. Todas as *mockups* criadas encontram-se em Anexos tendo algumas sido selecionadas para serem apresentadas nesta secção.

Para a criação destas foi utilizada uma abordagem *Mobile First*. Este termo foi utilizado pela primeira vez por Luke Wroblewski (diretor de produto da Google) em 2009 (SmartBear, 2022). *Mobile First Approach* refere-se à prática de projetar a aplicação primeiro para dispositivos móveis. Isto muda o fluxo de abordar primeiro os designs de *desktop* e começar a trabalhar primeiro na versão móvel. Isto permite aos desenvolvedores ter maior flexibilidade pois a tendência será de expandir em vez de reduzir (quando se faz o projeto para *desktop* há muito mais espaço para poder utilizar). Isto também está correlacionado com o facto de que a tecnologia não escala tão bem para dispositivos móveis, o que pode levar a uma experiência de utilização mais baixa.

Foram desenhados diversas *mockups* para as funcionalidades mais complexas, de modo também a compreender qual seria o fluxo de informação e as operações que o utilizador teria de executar para conseguir realizar uma dada operação.

A Figura 40 apresenta a página de login (primeira página que um utilizador não autenticado encontra) e a página de boas-vindas (página para a qual o utilizador é redirecionado após efetuar a autenticação na aplicação). Todas as páginas têm acesso á barra de navegação. Nesta é possível aceder ao menu lateral, fazer *logout* da aplicação, saber o perfil do utilizador autenticado e navegar para a página com os itens a requisitar (representado pela seta).

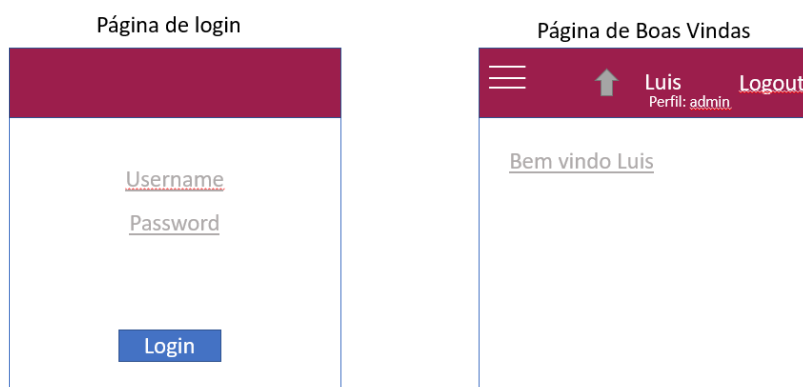


Figura 40 – Página de login e de boas-vindas

A Figura 41 representa o menu principal lateral (lado direito) que dará acesso a todas as funcionalidades da aplicação. Dependendo do perfil registado, estas funcionalidades podem ou não estar disponíveis (a Figura 41 tem uma legenda com as funcionalidades disponíveis para diferentes perfis de utilizador). A navegação entre as diferentes páginas da aplicação pode ser feita recorrendo a este menu (acessível a partir da barra da navegação persistente em todas as vistas). No lado esquerdo da imagem é mostrada a página das requisições de um utilizador autenticado e como a informação é apresentada. O botão de detalhes permite ver quais os equipamentos requisitados.

A Figura 42 representa a página de registo de um novo equipamento. Esta é constituída por um formulário para realiza a inserção de informação relativa ao equipamento (havendo campos de preenchimento obrigatório e outros não). Do lado direito da imagem é apresentada uma página pop-up que surge quando o utilizador pretende tirar fotografia ao equipamento. Nesta é também disponibilizada uma imagem com a pré-visualização da foto tirada.

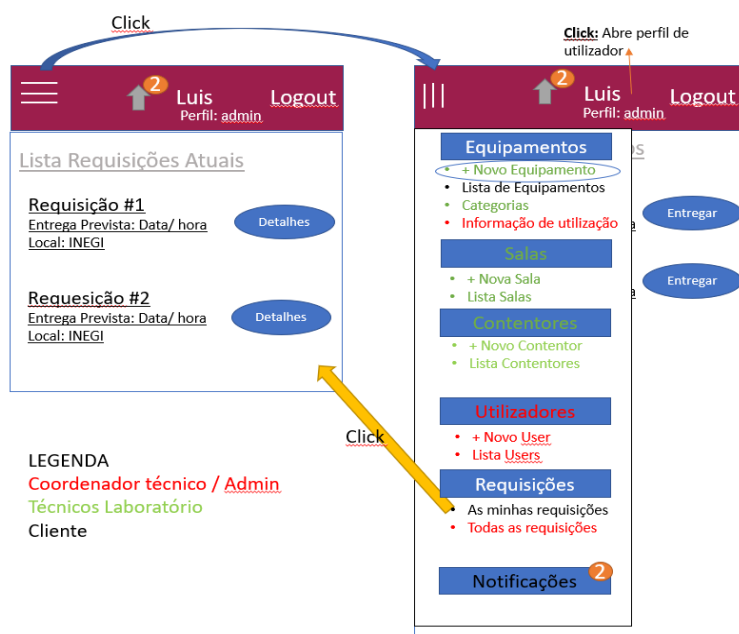


Figura 41 – Página de Requisições do utilizador (esquerda) e menu lateral (direita)

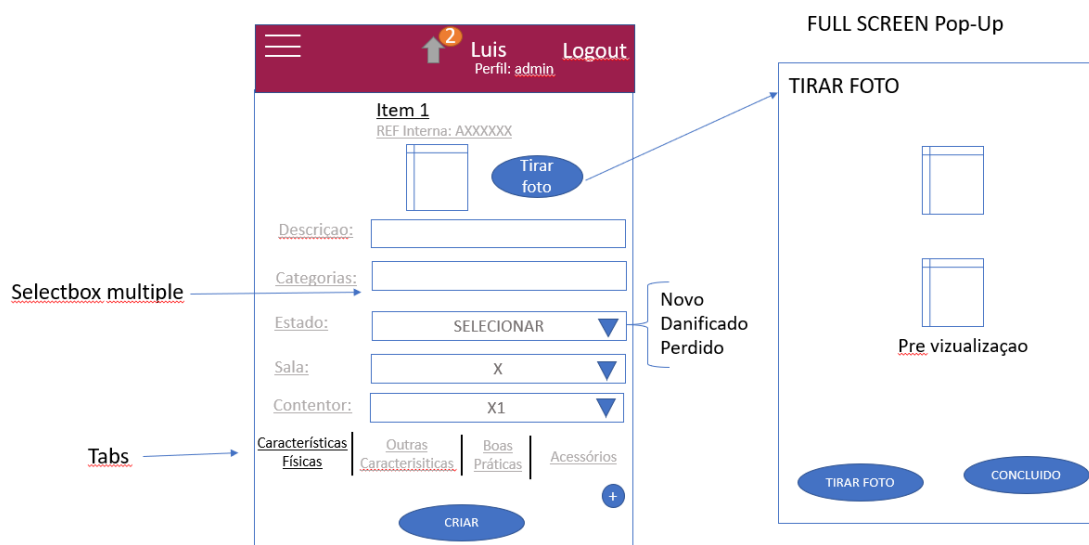


Figura 42 – Página de registo de novo equipamento

A Figura 43 representa a lista de equipamentos e o fluxo de ações para fazer a requisição. Ao seleccionar um equipamento para requisitar é apresentado um *pop up* com a lista de acessórios desse equipamento e que também podem ser adicionados à requisição. O ícone de requisição da barra de navegação mostra o número de itens seleccionados para a requisição. A página também tem vários tipos de filtros (incluindo leitor de QR code) para ajudar na procura do equipamento.

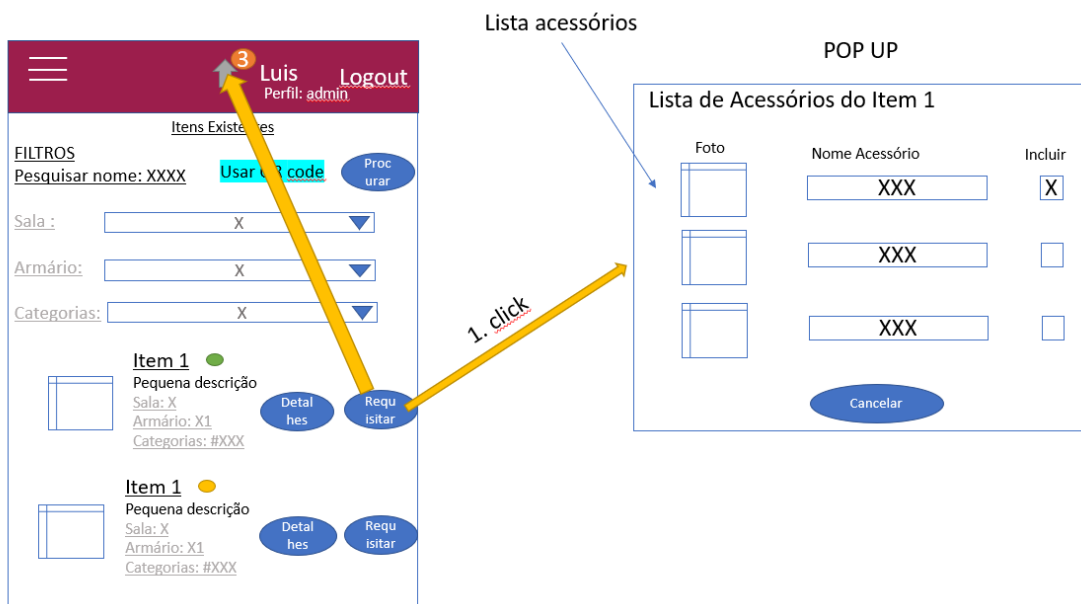


Figura 43 – Página da lista de equipamentos

Por fim, a Figura 44 ilustra o último passo da requisição de equipamentos. O utilizador introduz a data de início e de fim (podendo recorrer ao *datepicker* para ajudar no preenchimento) bem como a localização da utilização do equipamento. Caso não seja possível requisitar para as datas inseridas é despoletada uma mensagem de alerta a notificar da impossibilidade de requisitar o equipamento para os períodos seleccionados.

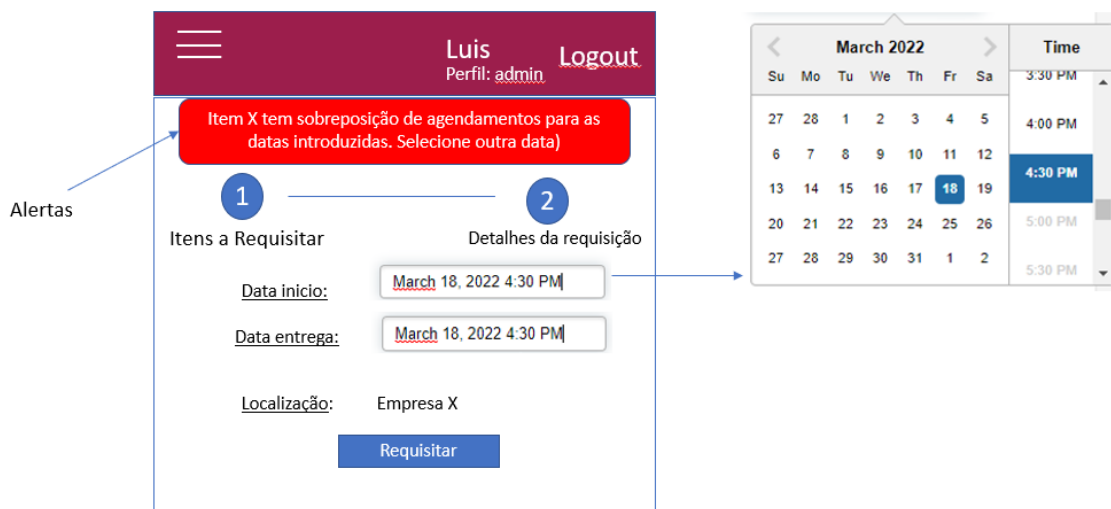


Figura 44 – Página de criação de requisição

## **5.10 Sumário**

Neste capítulo foi apresentado com maior detalhe a solução que irá dar resposta ao problema da organização. Foi feita uma descrição geral da solução, apresentados os requisitos funcionais e não funcionais, boas praticas de engenharia informática e design de software que podem ser utilizadas e uma arquitetura para realizar a implementação. Por último, foi apresentado uma possível solução para a interface do utilizador.

# 6 Desenvolvimento e construção da solução

Tendo sido estudadas as tecnologias que podem solucionar o problema e feita a respetiva especificação da solução, é chegada a fase de construção da aplicação de acordo com os requisitos e problemas levantados, e assim responder ao *Outcome 5* – construir a solução para o problema aplicando ciências e boas práticas de engenharia informática.

O presente capítulo focar-se-á na implementação da solução tendo em conta a arquitetura e o desenho já elaborados nos capítulos anteriores. Irá também ser comparada a solução final obtida com a especificação criada.

## 6.1 Seleção de Tecnologias

Um dos requisitos para a solução ter sucesso seria a possibilidade de aceder a esta a partir de dispositivos móveis e também de um computador. O conjunto de tecnologias que melhor se enquadra com este requisito são as tecnologias de desenvolvimento *web*, uma vez que:

- Com apenas um grupo de tecnologias e alguma lógica é possível implementar uma solução responsiva com capacidade de se adaptar a diferentes dimensões de ecrãs (não sendo necessário desenvolver uma solução específica para dispositivos iOS e Android visto que ambos podem aceder a aplicação através de um browser);
- A instituição possui algumas ferramentas desenvolvidas com base em tecnologias *web* (havendo mais foco nas tecnologias a utilizar, a manutenção das mesmas torna-se mais acessível);
- Existe grande quantidade de documentação disponível, soluções *open-source* e facilidade de acesso sem necessidade de instalação de qualquer software por parte dos utilizadores.

As tecnologias *web* estudadas no Capítulo 3, não têm nenhuma limitação ao nível das funcionalidades que precisam de ser implementadas ao ponto de preterir uma relativamente a outra. Desta forma a opção será feita com base em 3 fatores:

- Utilização de *stack* tecnológico já existente na empresa (*open-source*);
- Facilidade na implementação da solução com a arquitetura idealizada;
- Boa integração entre as tecnologias.

Desta forma, as tecnologias seleccionadas com as respetivas versões, divididas nas 3 camadas arquiteturais, são apresentadas na Tabela 11.

Tabela 11 – Tecnologias seleccionadas divididas por camadas

Camada	Tecnologia	Versão
Apresentação	React	17.0.1
Negócio	NodeJS	14.18.0
Dados	MySQL	8.0.28

## 6.2 Implementação dos requisitos

Uma vez seleccionadas as tecnologias basilares, vai ser agora apresentada a implementação ao nível do código (*back e front end*). Contudo, de modo a não tornar esta secção demasiado exaustiva e a obter-se uma visão geral da aplicação desenvolvida, serão apresentados os desenvolvimentos dos requisitos mais representativos para o bom funcionamento da aplicação e que se encontram representados nas *mockups* no capítulo 5 (funcionalidade de Login e autenticação, criação de um novo equipamento e visualização da lista de equipamentos existentes, e requisição). Os restantes requisitos seguiram a mesma lógica.

### 6.2.1 Autenticação de utilizadores

A primeira página da aplicação é dedicada à autenticação que permite validar o utilizador que tenta aceder ao sistema, (Figura 45).

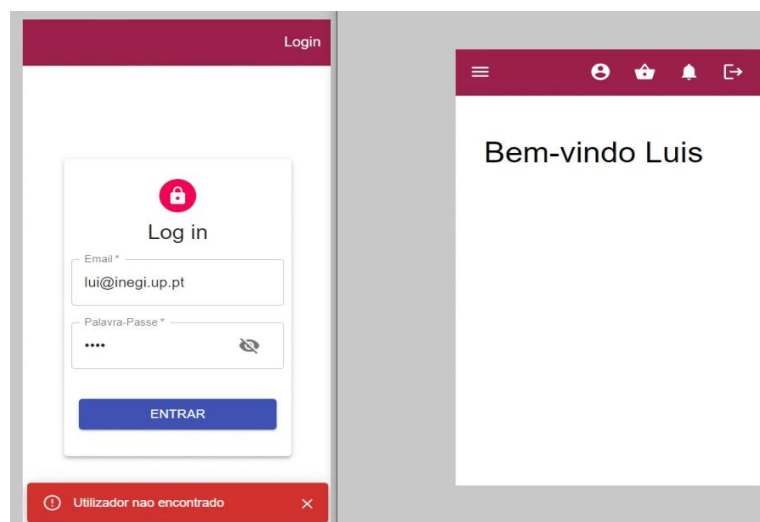


Figura 45 – Página de Login (esquerda) e de boas-vindas (direita) implementadas

Caso os dados inseridos não sejam válidos, é apresentada uma mensagem de erro com o motivo (na Figura 45 o utilizador não existe na base dados). Estas mensagens de aviso são um componente da biblioteca Material UI e são despoletadas pela aplicação React (consoante a resposta do servidor seja de sucesso ou erro). Três tipos de mensagens podem ser apresentadas e que se expõem na Figura 46.

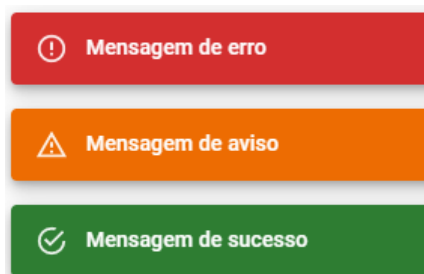


Figura 46 – Tipos de mensagens de aviso

Na aplicação NodeJS foi desenvolvida uma classe de erro específica para obter um maior controlo sobre o lançamento de erros em código síncrono e assíncrono (os erros lançados devem ser tratados de modo a recuperar o bom funcionamento da aplicação) e poder obter um maior grau de customização sobre a mensagem que aparece ao utilizador (evitando que mensagens sem significado apareçam na interface). O Excerto de código 1 representa essa classe a qual estende a classe de erro já existente em JavaScript. Estes erros são expectáveis e estão correlacionados com as ações do utilizador. O resultado do lançamento do erro varia entre uma mensagem para notificar o ocorrido até ao redireccionamento do utilizador para outra página da aplicação.

```

1. class HttpError extends Error {
2.   constructor(message, errorCode, errors) {
3.     super(message);
4.     this.code = errorCode;
5.     this.errors = errors;
6.   }
7. }

```

### Excerto de código 1 - Classe de erro desenvolvida

A perspetiva mais detalhada da sequência de operações que ocorre na aplicação NodeJS, quando o utilizador tenta fazer login, é apresentada a Figura 47, para ajudar nesta explicação.

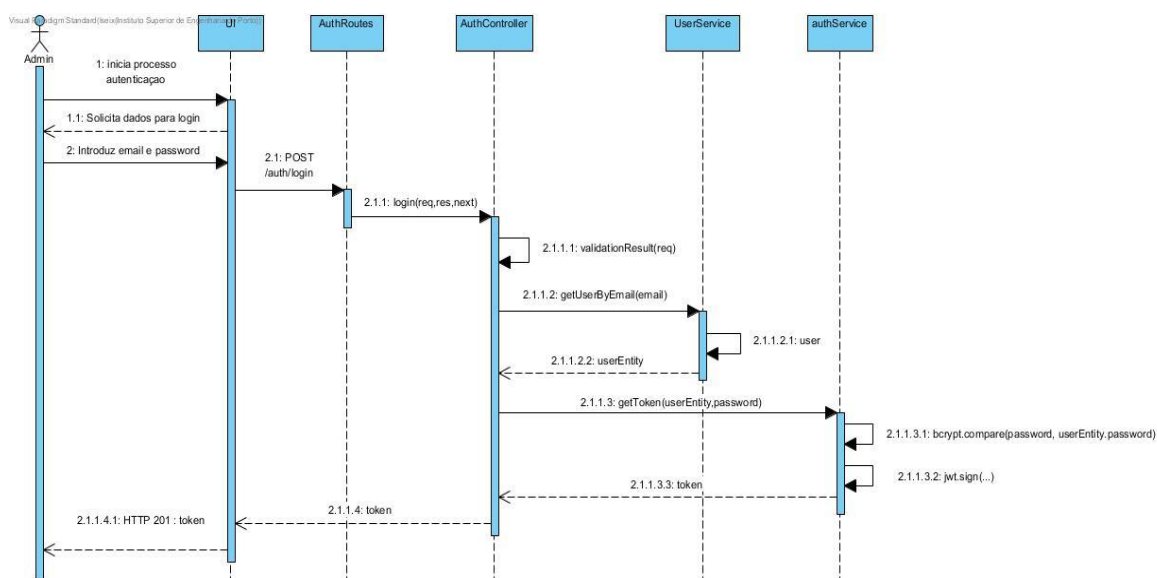


Figura 47 – Diagrama de sequência de autenticação

A descrição mais detalhada das classes representadas no diagrama de sequência da Figura 47 e seguintes pode ser consultada em Anexos (bem como o diagrama de classes de implementação). A função dos controllers, services e repositories (generalizada para as respetivas classes implementadas nos diagramas de sequência) foi apresentada no capítulo 5.

Após o envio do pedido pela aplicação React para o *endpoint*, responsável pela autenticação, este é redirecionado para o controller da autenticação (classe AuthController) que faz a validação dos parâmetros enviados, (o corpo de envio contém um parâmetro e-mail válido e um parâmetro password que não está vazio) e caso não haja nenhum erro prossegue para o UserService. Quando uma classe controller é instanciada ela recebe no seu construtor os serviços que necessita para prosseguir com a execução das operações. Este padrão de desenvolvimento de software é chamado de *Dependency Injection* (da classe A que utiliza alguns métodos da classe B) e permite manter baixo o nível de acoplamento entre diferentes módulos de um sistema. Deste modo a classe AuthController pede ao UserService para validar a existência de um utilizador (através do e-mail) na base dados. Apenas os serviços podem

interagir com o Repositório, que detém a responsabilidade de aceder aos dados. Este tipo de estruturação da aplicação vai ao encontro das diretrizes de design já apresentas no capítulo 5, nomeadamente a *Separation of concerns* (*Controllers, Services e Repositories* têm funções distintas e não sobrepostas) e a *Single Responsibility Principle* (cada módulo é responsável por uma funcionalidade específica).

Uma vez encontrado o utilizador na base dados, é gerado um *token* com o identificador deste que é enviado como resposta à aplicação React. Este *token* é guardado no armazenamento local do *browser* para, em futuros pedidos, fazer validação da autenticação e autorização do utilizador.

### 6.2.2 Criação de um novo equipamento

A criação de um novo equipamento foi uma das funcionalidades desenvolvidas para o requisito 8 - gerir equipamentos. Esta funcionalidade encontra-se apenas disponível para utilizadores com perfil de Administrador ou Técnico de Laboratório. A Figura 48 ilustra a opção visível no menu principal da aplicação React bem como a página de criação do equipamento.

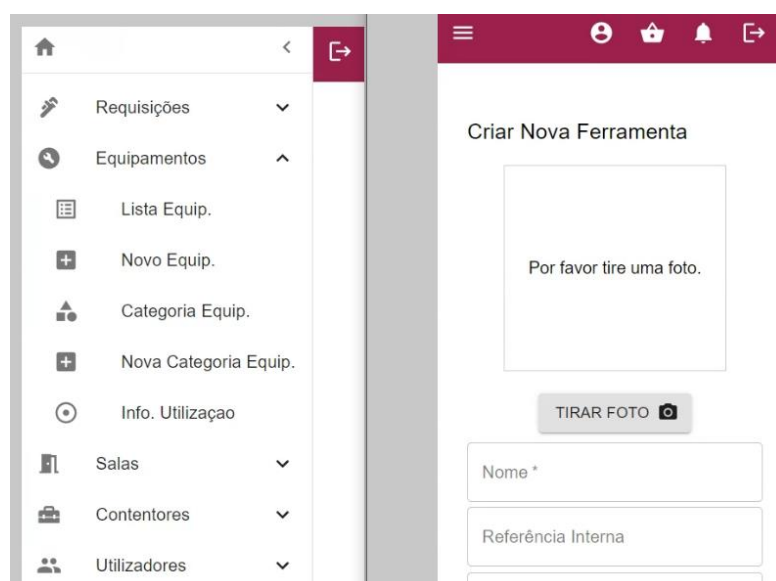


Figura 48 – Menu principal (esquerda) e página de criação de novo equipamento(direita)

De modo a evitar que utilizadores não autenticados tenham acesso às páginas das funcionalidades da aplicação, foi criado um componente que envolve todas as rotas que exigem a condição de utilizador autenticado. O código desse componente encontra-se representado no Excerto de código 2.

```

1. const PrivateRoute = ({ component: Component }) => {
2.   const { isAuthenticated, loading } = useSelector((state) =>
   state.auth);
3.
4.   if (loading) return <CircularProgress />;
5.   if (isAuthenticated) return <Component />;
6.
7.   return <Navigate to='/' />;
8. };

```

### Excerto de código 2 - Componente de validação de autenticação

A linha 4 retorna um componente de indicador de progresso que informa o utilizador sobre o estado do processo (valida o token e permite ao utilizador prosseguir para a página que solicitou). Caso o utilizador não se encontre autenticado este é redirecionado para a página de *login* (linha 7).

Relativamente ao acesso à câmara do utilizador existe a API (*Application Programming Interface*) JavaScript *MediaStream* que possibilita a utilização de áudio e vídeo capturados pelo dispositivo por meio de *streams* (se o utilizador consentir a sua utilização). Esta API está relacionada com a tecnologia *WebRTC* (*Web Real-Time Communication*) que permite que as aplicações *web* capturem e transmitam média (áudio e vídeo) entre browsers prescindindo de um intermediário (MDN, 2022). A utilização de um componente React (React *Webcam*) permitiu abstrair a implementação desta funcionalidade, tornando o desenvolvimento mais rápido. Este componente retorna imagens codificadas em base64 que podem ser utilizadas para transportar a informação da aplicação React para a API.

Relativamente à aplicação NodeJS, esta também faz validação de autenticação para garantir que o utilizador não acesse as operações recorrendo a outro cliente sem estar autenticado ou autorizado. A Figura 49 ilustra a sequência de operações (simplificada), que ocorrem no servidor.

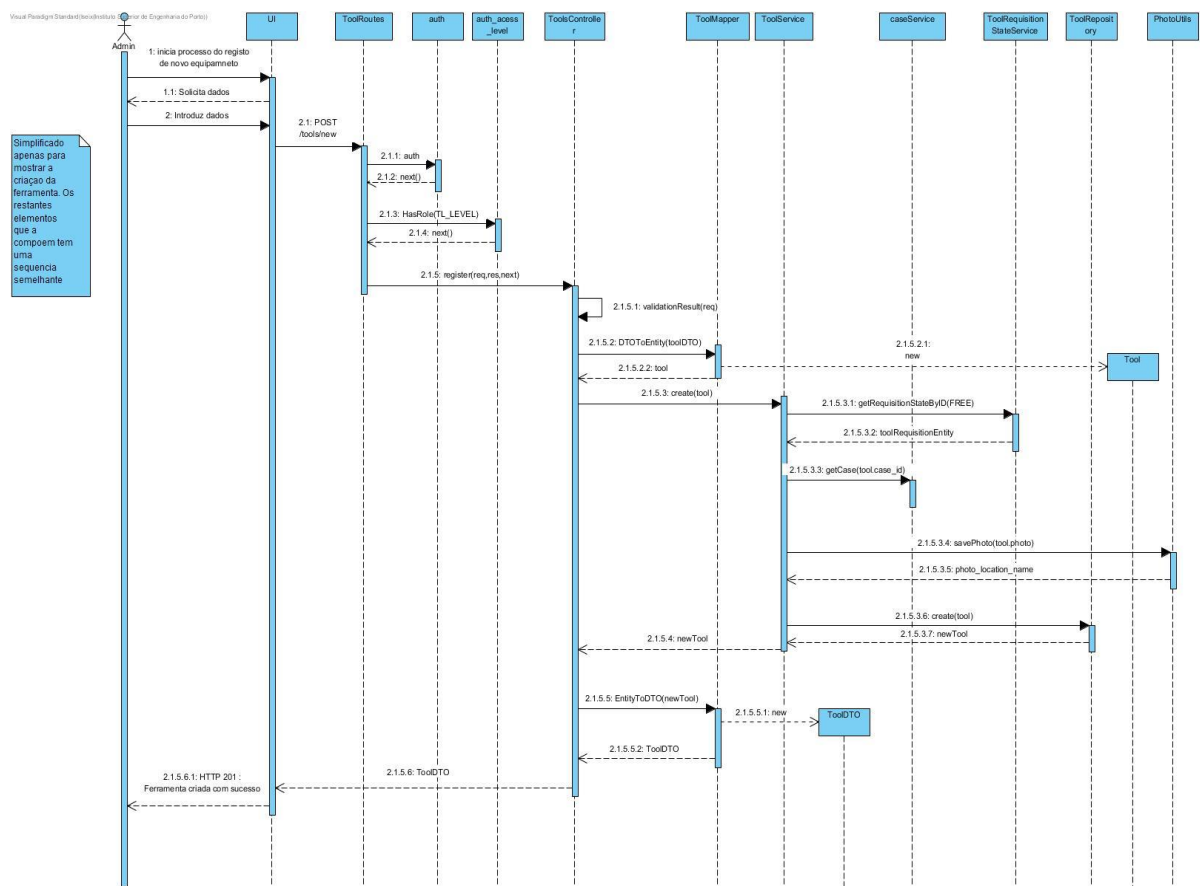


Figura 49 - Diagrama de sequência de criação de equipamento

Conforme ilustrado na Figura 49, existe inicialmente uma validação não só da autenticação do utilizador como também da autorização para aceder à funcionalidade (os restantes *endpoints* incluem também esta proteção).

Após o preenchimento do formulário pelo utilizador, com os dados do novo equipamento, os campos obrigatórios (como o nome, estado do equipamento, contentor onde se encontra), são validados para garantir o não envio de parâmetros vazios ou nulos. Esta validação também é efetuada pelo servidor. A fotografia é um dos dados facultativos a ser submetidos para a criação do equipamento. Esta é recebida pela aplicação NodeJS em formato base64 o que aumenta os requisitos de armazenamento (a informação codificada neste formato tem 33% mais dimensão que o ficheiro binário original (Cook, 2020)). Foi por isso criada uma função (*savePhoto*) apresentada no Excerto de código 3, que recebe uma *String* com a informação da fotografia e retorna a sua localização no sistema de armazenamento.

```

1. function savePhoto(photo) {
2.   let photo_location;
3.   if (photo) {
4.     if (photo.split('/').includes('uploads')) {
5.       return photo;
6.     }
7.     let base64Image = photo.split(';base64,').pop();
8.     let image_name = uuidv4();
9.     const filePath = __dirname +
    `../../uploads/images/${image_name}.png`;
10.
11.       let buff = new Buffer.from(base64Image, 'base64');
12.       fs.writeFileSync(filePath, buff);
13.
14.       req.filePath = filePath;
15.       photo_location = `uploads/images/${image_name}.png`;
16.     } else {
17.       photo_location = 'uploads/images/noImage.png';
18.     }
19.
20.     return photo_location;
21.   }

```

### Excerto de código 3 - Função para guardar imagens

Esta função faz uma validação inicial (linhas 3 a 6), para perceber se o parâmetro que receciona já foi guardado (dado que também é utilizado para a atualização da imagem) ou se é necessário processá-la e guardá-la (em formato PNG que é um formato de compactação de arquivo sem perdas de qualidade de imagem, contudo o tamanho do ficheiro também será maior (Adobe, 2022)). A função retorna uma *String* com a localização da gravação da fotografia associada ao equipamento e persistida na base dados.

### 6.2.3 Visualizar equipamentos existentes

A Figura 50 apresenta a página da lista de equipamentos registados na aplicação na versão *desktop*. Nesta imagem é possível constatar que a barra de navegação apresenta mais informação ao utilizador (nome do utilizador, o seu perfil, o ícone de notificações representado por um sino, ícone de requisição representado por uma cesta e ícone de saída do lado direito, nome da aplicação e ícone de acesso ao menu principal do lado esquerdo) dado o ecrã que acede há aplicação ser de maior dimensão. Isto deve-se ao facto do *front-end* ter sido criado para ser responsivo, ou seja, capaz de redimensionar, ocultar, reduzir ou ampliar automaticamente a apresentação da informação, de modo que esta seja disponibilizada de modo semelhante em todos os dispositivos (computadores e telemóveis).

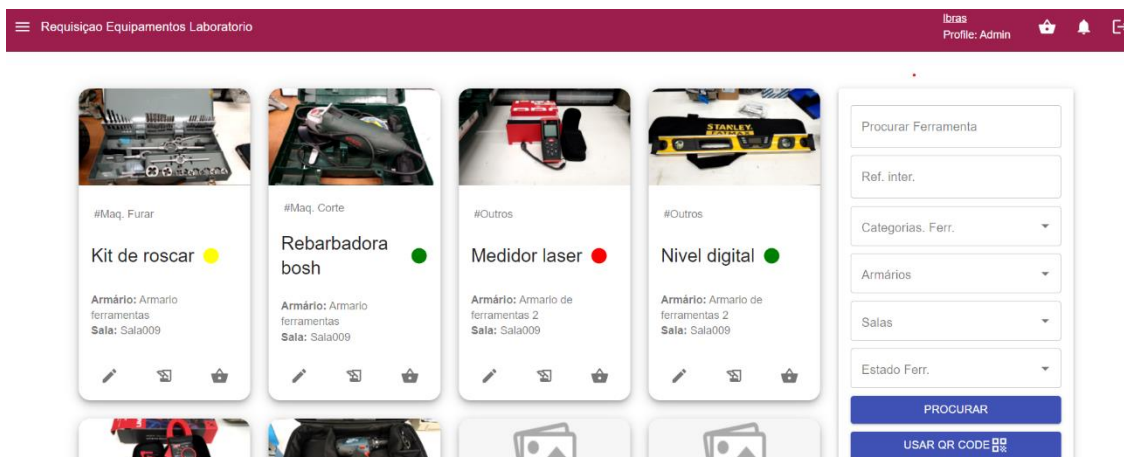


Figura 50 – Página de visualização de equipamentos (vista de *desktop*)

Esta página também dá a indicação do estado do equipamento através do seguinte código de cores:

- círculo verde - o equipamento não tem qualquer requisição agendada;
- círculo amarelo - o equipamento tem pelo menos uma requisição agendada;
- círculo vermelho - o equipamento está em utilização.

Do lado direito da página encontram-se os filtros para facilitar a procura do equipamento, cuja sua interseção permite obter a solução. Este filtro foi criado na aplicação React, tornando a procura por equipamentos local e por isso muito mais rápida do que fazer pedidos à base de dados a partir da API. Nesta página o utilizador também pode recorrer ao QR code para filtrar os equipamentos associados a um contentor (este QR code é gerado pela aplicação na página dos contentores).

#### 6.2.4 Criação de uma requisição

Por fim, para efetuar uma requisição o utilizador tem associado ao cartão do equipamento um ícone para o efeito. Todos os ícones disponibilizam uma mensagem para obtenção da explicação da sua utilização; este comportamento está ilustrado na Figura 51 do lado esquerdo. Após clicar nesse ícone é apresentada uma mensagem a notificar o sucesso da operação e simultaneamente uma lista de acessórios requisitáveis juntamente com a ferramenta desejada. O ícone de requisição na barra de navegação também é atualizado com o número de equipamentos a requisitar.

A Figura 52 apresenta a página de requisição, na qual o utilizador preenche a data de início e fim e o local de utilização. Caso o equipamento esteja livre nas datas pretendidas e após validação pelo *back-end* a requisição é criada e guardada na base dados e o utilizador é reencaminhado para a página das suas requisições (Figura 53). Se o utilizador não devolver os equipamentos no prazo estabelecido, recebe uma notificação para prolongar ou terminar esta.

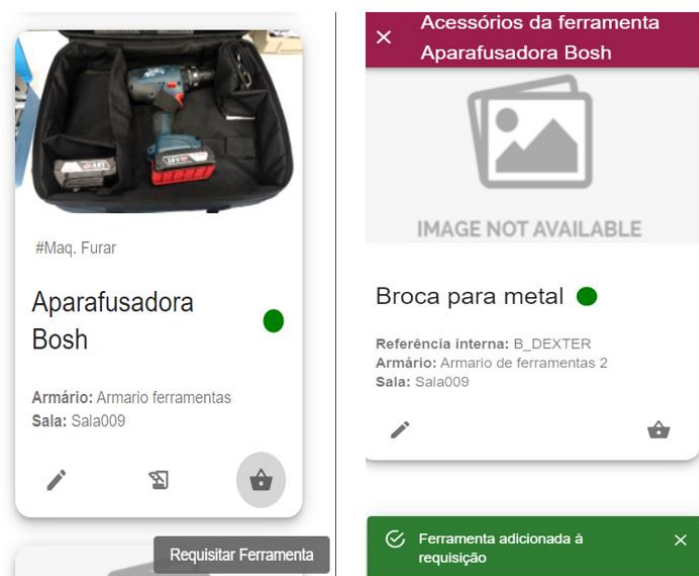


Figura 51 – Cartão do equipamento a requisitar (esquerda) e acessórios deste (direita)

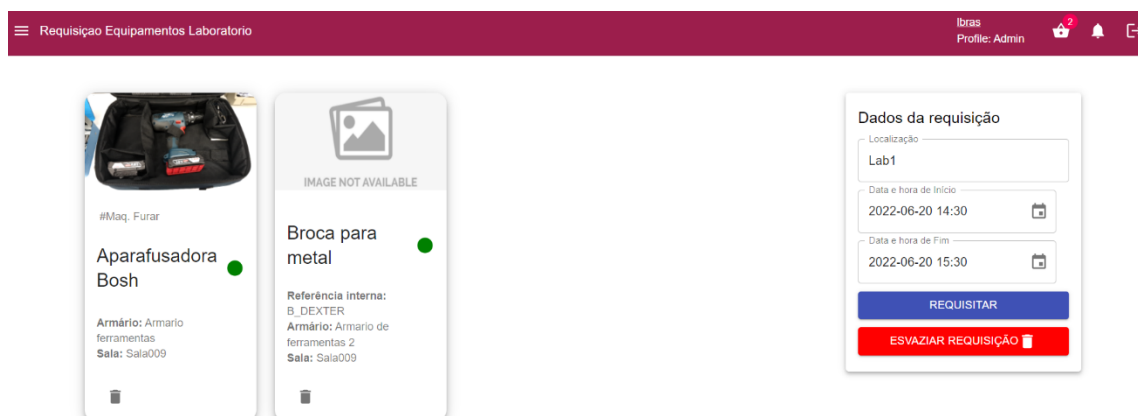


Figura 52 – Página de requisição de equipamentos (vista *desktop*)

Requisição Equipamentos Laboratorio

lbras  
Profile: Admin

Todas as Requisições

Número	Data Inicio	Data Fim	Localização	Estado	Ações	Detalhes
16	2022-06-20 14:30:00	2022-06-20 15:30:00	Lab1	Não Termi.		ABRIR

Figura 53 – Página com lista de requisições do utilizador (vista *desktop*)

A Figura 54 ilustra a sequência de operações levadas a cabo pela aplicação NodeJS aquando da criação de uma nova requisição.

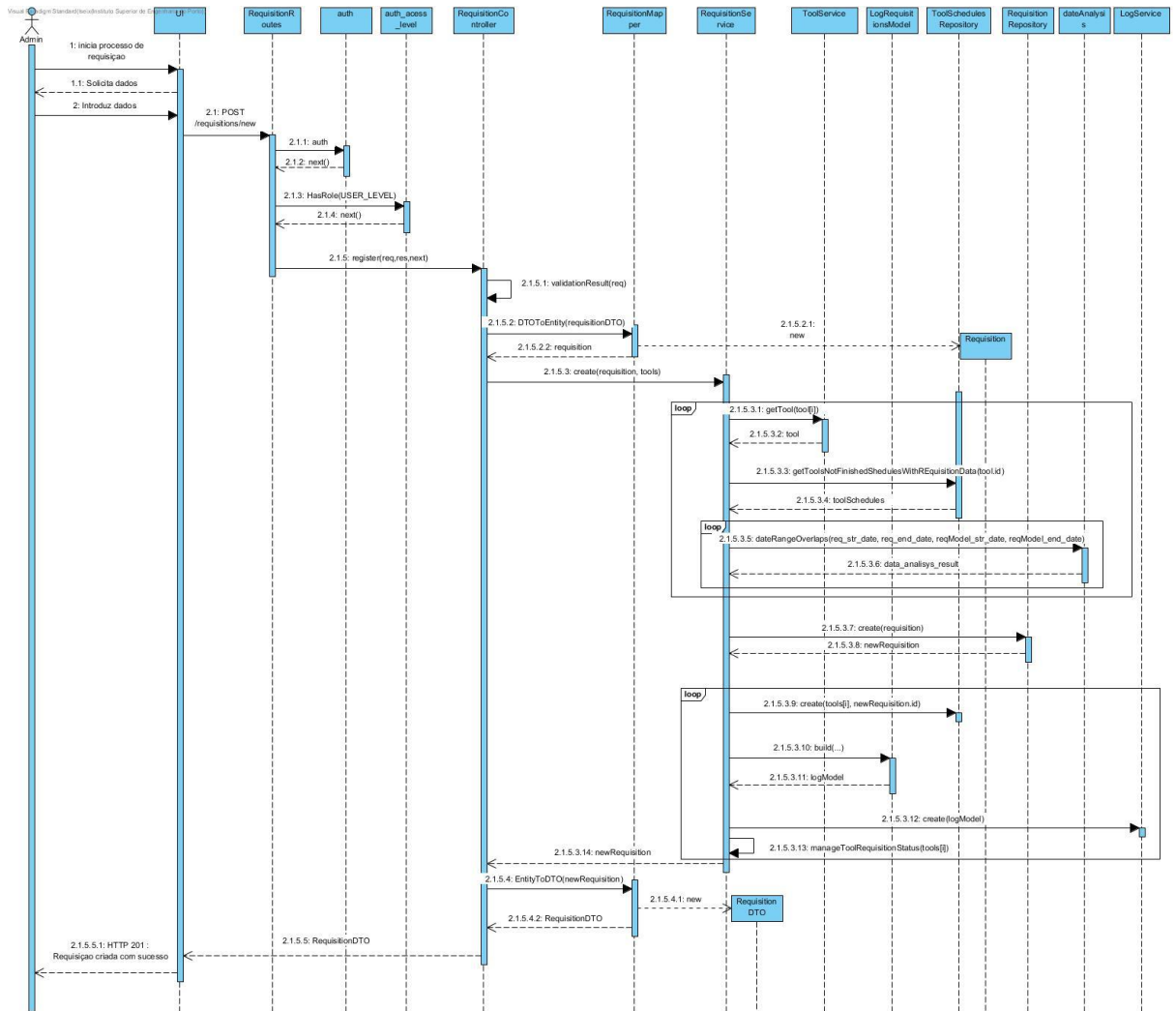


Figura 54 - Diagrama de sequência de criação de requisição

Este processo de validação vai desde a verificação da existência do equipamento na base dados até ao registo de *logs* da operação (que serão utilizados para calcular quais as ferramentas mais utilizadas). O algoritmo apresentado no Excerto de código 4 permite validar se existe alguma sobreposição entre dois intervalos de tempo fornecidos por parâmetro, retornando *false*, caso não haja.

```
1. function dateRangeOverlaps(a_start, a_end, b_start, b_end) {
2.   if (a_start <= b_start && b_start <= a_end) return true; // b
   starts in a
3.   if (a_start <= b_end && b_end <= a_end) return true; // b ends in a
4.   if (b_start < a_start && a_end < b_end) return true; // a in b
5.   return false;
6. }
```

Excerto de código 4 - Função de validação de sobreposição de intervalos temporais

## 6.3 Sumário

Neste capítulo foram selecionadas as tecnologias para desenvolver a solução e apresentado o resultado da implementação da solução *back* e *front end* juntamente como os princípios adotados, não só ao nível das boas práticas de desenvolvimento de software, como também alguns cuidados considerados ao nível da experiência do utilizador (*User Experience*) na aplicação React, de modo a torná-la intuitiva e apelativa ao nível do design. Relativamente à especificação inicialmente elaborada e apresentada nas *mockups*, foram introduzidas algumas alterações ao nível do design (nomeadamente na página de requisição de equipamentos), de modo a tornar a solução final mais coerente em todas as páginas e também de forma a agilizar o processo de desenvolvimento (condicionado pela duração da tese), mas sem comprometer nenhuma funcionalidade documentada nos requisitos do sistema. Desta forma foi possível criar uma solução bem estrutura aplicando bons princípios de engenharia informática e de software e capaz de responder aos problemas levantados inicialmente. Dada a extensão da aplicação desenvolvida, foram apenas selecionados 4 casos de uso para a apresentação dos detalhes técnicos (os mesmos que os das *mockups*).

# 7 Experimentação e avaliação da solução

Este capítulo visa fazer a experimentação e a avaliação da solução implementada de modo a perceber se ela conseguiu dar resposta aos problemas levantados e se tem a qualidade necessária. Para dar resposta ao último *Outcome* – Avaliar a solução desenhada e implementada, aplicando ciências e boas praticas de engenharia informática, foi definida uma abordagem a seguir para classificar a solução final e com base nesta concluir sobre a qualidade do trabalho desenvolvido.

## 7.1 Abordagem de avaliação

Após o desenvolvimento do sistema, é necessário especificar uma metodologia para avaliar a qualidade da solução. Esta pode ser definida como um conjunto de características que necessitam de ser implementadas e validadas de modo que a solução final vá ao encontro dos requisitos levantados pelo cliente (Rocha *et al.*, 2008).

Antes de especificar a metodologia de avaliação é enunciada a hipótese de investigação, ou seja, o que se pretende provar (o objetivo a alcançar com a solução), e definidos os indicadores e fontes de informação (os dados que permitam avaliar a aplicação).

### 7.1.1 Hipótese de investigação

O objetivo a atingir com o desenvolvimento desta aplicação informática é digitalizar um processo manual que já estava implementado na organização, mais concretamente o processo de requisição de equipamentos.

Foi definido como hipótese que o recurso à combinação de tecnologias OT e IT permite tornar os processos de requisição de equipamentos mais organizada, estruturada e eficiente, culminando com o desenvolvimento de uma aplicação informática que irá auxiliar nesta tarefa e trazer novas fontes de informação úteis à equipa.

### **7.1.2 Indicadores e Fontes de Informação**

Como fontes de informação para avaliar se o produto desenvolvido corresponde às necessidades da equipa e se é possível atingir a hipótese de investigação, é preciso analisar se os requisitos especificados e recolhidos a partir das entrevistas realizadas, estão implementados de acordo com as necessidades levantadas pelo cliente.

A solução implementada foi experimentada pelo cliente (testes de aceitação manuais) para em seguida, responder a um questionário de averiguação da qualidade do sistema desenvolvido. As respostas a este questionário encontram-se nos Anexos, tendo sido feitas questões relativas à interface do utilizador, ao conteúdo digital e ao desempenho da aplicação. Desta forma será possível avaliar quantitativamente o mesmo.

Para além destas fontes de informação, foram também desenvolvidos testes unitários, de integração e *end 2 end* para verificar e validar se a aplicação de software elaborada vai ao encontro dos requisitos técnicos do negócio, que definiram o desenho e implementação. Os resultados obtidos nestes testes permitiram também avaliar quantitativamente a solução.

### **7.1.3 Metodologias de avaliação**

Como metodologia de avaliação optou-se por desenvolver o cenário de qualidade, no qual foram definidos critérios de cumprimento para cada um dos requisitos (funcionais e não funcionais) e escalas de medição. Para medir a qualidade do sistema e obter os valores para as métricas é necessário ter em conta as respostas dos utilizadores ao questionário, que avalia a experiência. Por fim é aplicado o *Qualitative Evaluation Framework (QEF)*, modelo de avaliação de software (Rocha *et al.*, 2008) com o objetivo de avaliar e garantir a qualidade da aplicação desenvolvida.

Criou-se para esta tarefa uma folha de cálculo em Excel (em Anexos são apresentados excertos desta já preenchida com os resultados obtidos), a qual contém o cenário de qualidade estruturado em 3 níveis: as dimensões de interesse, os fatores agregados às dimensões e os requisitos com impacto sobre os fatores, sendo para cada um destas definidas métricas de avaliação e níveis de cumprimento que permitirão obter a pontuação de qualidade do sistema e perceber este corresponde ao sistema ideal.

## **7.2 Testes realizados ao sistema**

Sendo que a dimensão da manutenibilidade tem em conta o fator da testabilidade do sistema e as métricas definidas têm em conta os resultados obtidos nos testes, é necessário especificar como é que estes foram desenvolvidos e que componentes foram testados. O facto de a aplicação desenvolvida ter sido feita segundo uma arquitetura em camadas, simplificou o processo de testagem, uma vez que este tipo de arquitetura beneficia de possuir interfaces e camadas bem definidas, baixo acoplamento e alta coesão.

Vão ser descritos em seguida os tipos de testes desenvolvidos e os seus resultados.

### 7.2.1 Testes unitários

A função dos testes unitários é verificar se partes pequenas e isoladas do software funcionam de acordo com o especificado. Estes tipos de testes vão ser utilizados na aplicação *de back-end*, com foco nos serviços implementados, para averiguar se a lógica e as regras de negócio estão de acordo com o especificado. O objetivo não é testar toda a aplicação *de back-end* mas sim os serviços mais críticos para garantir o bom funcionamento da aplicação. O quociente entre o número de linhas testadas e o número total de linhas de um ficheiro é uma das métricas comumente utilizadas para limitar o código de teste da aplicação e também garantir que este tem uma boa cobertura de testes.

$$\text{Cobertura de código} = \frac{\text{Número de linhas percorridas no ficheiro}}{\text{Número de linhas total no ficheiro}} \times 100 \quad (7.1)$$

O objetivo não é maximizar o resultado da equação acima, uma vez que atingir 100% de cobertura de código é muito dispendioso a nível de tempo e não é representativo do bom funcionamento da aplicação. A cobertura de código permite perceber que uma linha foi executada, mas não permite:

- Perceber se o código funcionará de acordo com os requisitos de negócio;
- Saber se funciona bem com outras partes do código;
- Inferir se testar esta parte do código é mais importante que as restantes.

Deste modo é importante ter em conta não só este indicador, mas também que os testes criados vão ao encontro dos casos de uso que o código implementa. Uma cobertura de código de 70% a 80% é uma meta razoável para testes de sistema na maioria dos projetos.(Cornett, 2013)

Os testes desenvolvidos tiveram por base a tecnologia Mocha (uma *framework* de testes JavaScript executada em NodeJS). Visto que os serviços têm dependências com o repositório e não é objetivo testar este último, (não se pretende testar a interação com a base dados, isso será efetuado com outro tipo de teste) mas sim a lógica implementada no serviço, foram utilizados *Stubs* que são componentes que simulam respostas a dependências externas e efeitos secundários.

A Figura 55 ilustra o exemplo de um teste unitário criado seguindo esta lógica.

```

1.   it('should create room', async () => {
2.     const roomRepoStub = sinon
3.       .stub(roomRepo, 'create')
4.       .returns(roomEntityStubValue);
5.
6.     const roomCreated = await
roomService.create(roomEntityStubValue);
7.
8.     expect(roomRepoStub.calledOnce).to.be.true;
9.     expect(roomCreated.id).to.equal(roomEntityStubValue.id);
10.    ...
11.    ...
12.    roomRepoStub.restore();
13.  });

```

Figura 55 – Teste unitário criado para o *Room service* (criação de sala)

Por fim, foram criados testes unitários para todos os serviços com uma cobertura superior a 80% para todos eles. A Figura 56 mostra os resultados obtidos (a coluna *Lines* representa a cobertura dos testes criados).



Figura 56 – Resultados obtidos com testes unitários

### 7.2.2 Testes de integração

Os testes de integração verificam se várias unidades (*controllers*, *services*, *repository*) conseguem trabalhar em conjunto para alcançar o objetivo pretendido. Este tipo de testes é usado para averiguar o comportamento do sistema como um todo e por isso utilizam todas as funcionalidades nele presentes.

Para criar os testes de integração foi utilizado o Postman (plataforma para testar APIs). Deste modo foi possível perceber a resposta da aplicação *back-end* a diferentes tipos de dados e se armazenam estes no formato pretendido. Os testes criados focaram-se nos requisitos funcionais especificados e cujo bom funcionamento é crítico para o sucesso da aplicação. A Figura 57 mostra a divisão dos testes por requisito funcional.

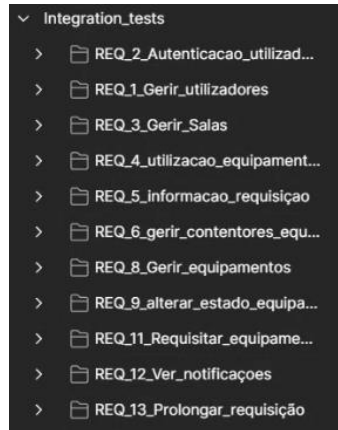


Figura 57 – Estrutura dos testes de integração

Uma vez que estes testes fazem inserção de valores na base dados e de modo a não introduzir dados derivados na produção, estes foram realizados com a base de dados de desenvolvimento. A Figura 58 mostra os resultados obtidos nos testes de integração

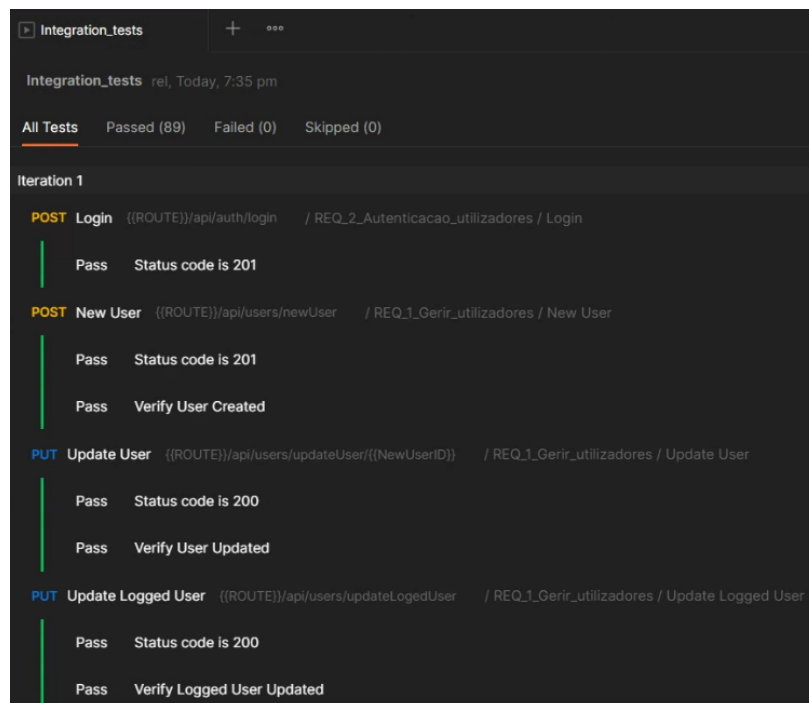


Figura 58 – Execução Teste de integração

Foram criados 89 testes de integração, tendo todos resultados positivos. Estes validam não só o código de resposta como também o seu conteúdo (de modo a concluir se os dados foram guardados com o formato pretendido).

### 7.2.3 Testes *end 2 end*

Os testes *end 2 end* (também chamados de testes funcionais) abrangem todos os sistemas (aplicações *back* e *front end*) e avaliam-nos como um todo. Estes testes automatizam o navegador e tentam reproduzir o fluxo típico de um utilizador quando este interage com a aplicação.

Uma baixa quantidade deste tipo de testes (relativamente aos anteriores) é capaz de cobrir uma grande de funcionalidade do software. Como ferramenta foi optada pelo Cypress uma vez que, este adota uma sintaxe para a escrita dos testes semelhante à do Mocha (utilizados nos testes unitários). Como estrutura de testes optou-se por criar 1 ficheiro de teste para cada um dos casos de uso (Figura 59).

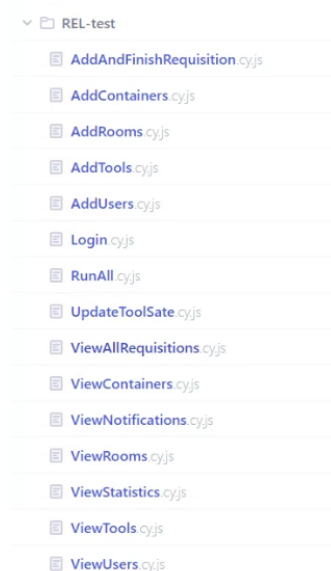


Figura 59 – Estrutura de testes *End 2 End*

Como estes testes validam o comportamento do *front end* e das respostas do *back end*, são os que demoram mais tempo a executar. Foi mais uma vez utilizada a base dados de desenvolvimento de modo a não influenciar os dados armazenados em produção. Os testes criados avaliam não só o procedimento desejado, mas também erros que o utilizador pode cometer ao introduzir os dados, o que permite perceber se o sistema tem a resposta esperada. A Figura 60 mostra o resultado positivo da execução dos 35 testes.

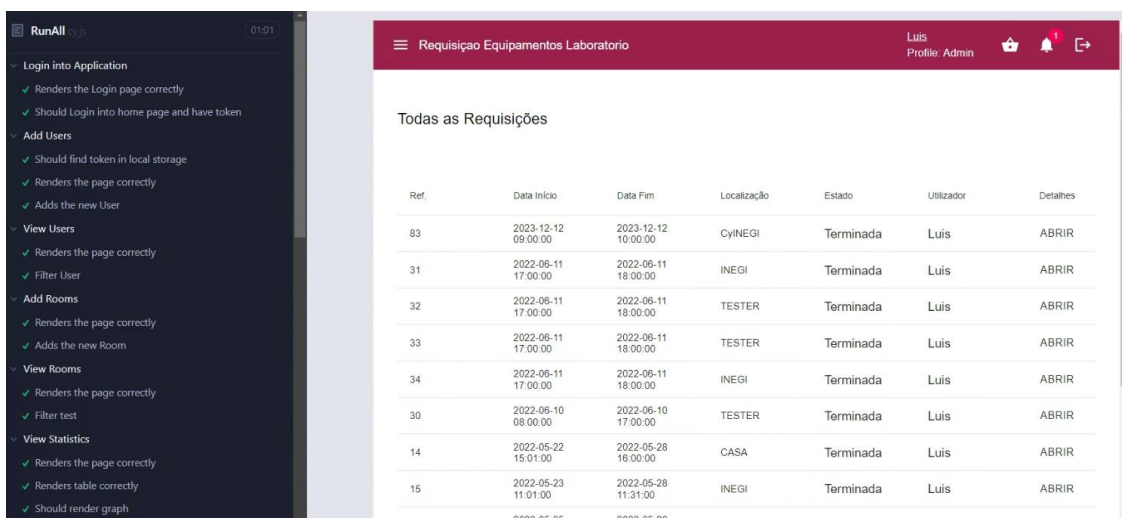


Figura 60 - Dashboard com resultado da execução dos testes end 2 end

### 7.3 Análise do questionário e resultados obtidos

Para além dos testes automatizados, que executam etapas previstas e definidas previamente e permitem validar que funcionalidades críticas do sistema, continuam operacionais, depois de realizar ajustes, foi pedido a doze utilizadores que utilizassem a aplicação, de modo a poder detetar erros que não foram previstos pelos testes. Foi desenvolvido um questionário para recolher o feedback dos utilizadores, cujas respostas encontram-se em Anexo. Este feedback servirá para fazer a avaliação da solução de acordo com as métricas e níveis de cumprimento definidos. A Tabela 13 relaciona as questões com o requisito e o seu fator de cumprimento (que pode ir de 0 a 100).

Da análise dos resultados às questões é possível constatar que os filtros de pesquisa e as mensagens de erros, são os que tiveram pior pontuação e que poderão ter de sofrer alguns ajustes numa versão posterior da aplicação. Do questionário elaborado também constou um campo aberto e facultativo de preenchimento para sugestões de melhoria. A Tabela 13 resume essas sugestões que visam pequenas melhorias na criação de algumas entidades e novas funcionalidades.

Tabela 12 – Sugestões de melhoria

	Sugestão
1	Possibilidade de adicionar quantidades a equipamentos.
2	Permitir a criação de salas sem especificação da localização.
3	Compatibilização com um sistema de RFID (tags autocolantes para ferramentas e cartões dos utilizadores, fechaduras eletrónicas e <i>rfid readers</i> nos armários).
4	Notificação do utilizador, por e-mail, para alertar da expiração do prazo de requisição.
5	Possibilidade de criar equipamentos sem necessidade de uma referência interna.

Tabela 13 – Relação entre questões, requisitos e fator de cumprimento obtido

Requisito	Questão	%Fator de cumprimento (wf <sub>k</sub> )
FIU01	A aplicação é intuitiva?	100
FIU02	O design é semelhante em toda a aplicação?	100
FIU03	A interação com a interface é semelhante em toda a aplicação?	100
FIU07	As funcionalidades do sistema estão presentes em diferentes dimensões de ecrã (telemóvel, computador)?	100
FIU06	As principais funções são rapidamente acedidas (poucas ações)?	100
FIU04	A experiência de navegação é semelhante em toda a aplicação?	100
FIU09	Os filtros de pesquisa de informação são intuitivos e de fácil utilização?	50
FCD01	As informações dos equipamentos estão bem	100
FCD02	Os textos são lexicalmente corretos?	100
FCD03	As mensagens apresentam códigos ilegíveis ou símbolos indecifráveis?	100
FCD05	A aplicação notifica o utilizador sobre ações necessárias?	100
DQV01	As imagens são nítidas e têm boa qualidade?	100
DN02	A interface do utilizador é rápida, havendo pouca latência nas respostas?	100
DN03	Quando ocorrem erros, estes são apresentados ao utilizador com uma mensagem elucidativa do mesmo?	50

Com o sistema desenvolvido e implementado, as métricas obtidas da execução dos diferentes tipos de testes e a análise das respostas ao questionário, fica-se em condições de preencher o modelo de avaliação de software (QEF) para averiguar a qualidade da aplicação desenvolvida. A Tabela 14 resume os valores calculados.

Tendo em conta os resultados obtidos e as métricas definidas, o sistema apresenta uma qualidade de 94% face ao pretendido atingir e uma distância ao sistema ideal de 0.14 (quanto menor o valor mais próximo se fica do sistema ideal). Para os fatores e pesos definidos, a dimensão da usabilidade, na qual está refletida a implementação dos requisitos funcionais, é a que apresenta melhor qualidade (95.26%), seguido do desempenho e da manutenibilidade.

Tabela 14 – Valores obtidos no cálculo da qualidade do sistema desenvolvido

q(%)	D	Q <sub>i</sub> (%)	Dimensão	Q <sub>j</sub> (%)	Fator
94	0.14	95.26	Usabilidade	100	Requisitos Funcionais
				87.5	Interface do utilizador
				100	Conteúdo digital
		90	Manutenibilidade	100	Testabilidade
				75	Manutenção
		91.23	Desempenho	100	Qualidade visual
				100	Segurança
				86.84	Navegação

## 7.4 Sumário

Neste capítulo foi feita a especificação de uma metodologia de avaliação para analisar a qualidade do sistema. Para tal foram definidas métricas concretas que tiveram por base as respostas a um questionário de averiguação da experiência dos utilizadores, sobre a implementação dos requisitos funcionais e os resultados obtidos nos testes automatizados. Por fim foi possível calcular uma qualidade geral do sistema de 94% e uma distância ao sistema ideal de 0.14. Isto demonstra a qualidade do sistema desenvolvido e a capacidade de responder as necessidades dos utilizadores.

## 8 Conclusão

Este capítulo apresenta as conclusões finais de todo o trabalho desenvolvido, com referência ao contexto e problema; a abordagem e solução para este e os objetivos realizados. Igualmente é dado a conhecer o trabalho que no futuro espera vir a realizar-se.

### 8.1 Contexto e problema

Nos últimos anos o INEGI tem participado e consolidado a sua posição como parceiro da indústria nacional e internacional em diversas áreas de interesse, entre elas, o desenvolvimento de produtos e sistemas, a engenharia industrial e a gestão de projetos de I&I. Neste âmbito, dispõe de várias salas laboratoriais com armazenamento de equipamentos que são essenciais para a realização de diversas atividades no âmbito das Engenharias Mecânica, Eletrónica, entre outras.

Com a expansão do INEGI, que tem vindo a verificar-se nos últimos 2 anos, quer a nível de pessoas, quer a nível de equipamentos, decorre a necessidade de uma melhor gestão da requisição destes, de modo a tornar as operações mais eficientes e eficazes. Aquando do início do desenvolvimento do trabalho, a requisição dos equipamentos era realizada de uma forma manual por inserção da informação numa folha de papel com uma estrutura de informação pré-definida. Muitas vezes o equipamento que o operador requisitou não era devolvido no intervalo de tempo indicado, quer por esquecimento, quer por necessidade de prolongamento do tempo de utilização do mesmo, sendo que, nestas situações era prática comum a folha de registo não ser atualizada gerando dificuldades ao nível da sua localização. Para além disto, não era guardado um histórico das requisições do equipamento, não permitindo assim, de forma imediata, determinar se há equipamentos mais requisitados que outros e eventual necessidade de manutenção e/ou aquisição de novos. O processo era também pouco amigável para operadores inexperientes e que desconhecem o nome das ferramentas, podendo levar a nomear estas de forma errónea.

## 8.2 Abordagem e solução

O trabalho teve início com a interpretação do problema a resolver, tendo sido realizadas entrevistas com os elementos da empresa, futuros utilizadores da nova aplicação. Para a preparação das entrevistas foram adotadas técnicas de engenharia de requisitos. Em seguida analisaram-se as respostas de modo a extrair informação relevante à interpretação correta do problema e à idealização das primeiras soluções.

Tendo interpretado o problema, iniciou-se a pesquisa dos principais conceitos e sintetizado o conhecimento existente que podia contribuir para o desenvolvimento da solução. Para tal foi estudado o conceito de digitalização dos processos e quais as tecnologias existentes para auxiliar na construção da solução. As tecnologias encontradas foram divididas em dois grupos:

- as OT que permitem operacionalizar os equipamentos físicos;
- as IT mais vocacionadas para o desenvolvimento aplicacional.

Em seguida fez-se a avaliação das diferentes abordagens para a resolução do problema com recurso ao NCD e ao AHP para selecionar a solução que melhor se adequa à resolução do problema. Foi também apresentada a proposta de valor deste produto recorrendo ao *Value Proposition Canvas* de Osterwalder.

Consequentemente foi delineada a arquitetura e desenho de uma possível solução tendo em conta os requisitos funcionais e algumas das tecnologias OT e IT estudadas.

Uma vez estabelecida a arquitetura, os requisitos funcionais e não funcionais e desenhada a interface do utilizador, procedeu-se à sua implementação. Foram selecionadas as tecnologias estudadas que melhor se enquadravam para dar resposta aos problemas identificados.

## 8.3 Objetivos realizados

Os objetivos do trabalho desenvolvido passavam por criar um sistema para gestão da requisição dos equipamentos de laboratório. O sistema desenvolvido é composto por um serviço *web* e uma aplicação móvel acessível ao utilizador através de um navegador.

Para averiguar se a solução implementada conseguiu dar resposta aos problemas levantados e com a qualidade necessária, foi efetuada a experimentação e avaliação da solução implementada pelos utilizadores. De modo a quantificar a qualidade do sistema desenvolvido foi feita a especificação de uma metodologia de avaliação do software, através da qual foi possível demonstrar que o sistema criado era capaz de responder às necessidades dos utilizadores e estava muito próximo da qualidade do sistema ideal.

A experimentação pelos utilizadores do sistema criado, permitiu dar resposta às suas necessidades de acesso a um sistema que lhes permite identificar, localizar e requisitar os

equipamentos de forma eficaz por parte dos colaboradores através da rede interna da Organização.

## **8.4 Trabalhos futuros**

Futuramente será necessário continuar a utilizar o sistema desenvolvido de modo a alimentá-lo com mais dados e perceber quais os pontos a serem melhorados para tornar mais fácil o processo de requisição de equipamentos laboratoriais. Para além da implementação das sugestões de melhoria obtidas através das respostas aos questionários, seria interessante alargar a utilização da aplicação por forma a que ela inclua a gestão de inventário (ou seja, produtos consumíveis). Esta funcionalidade possibilitaria a notificação dos utilizadores responsáveis sobre uma iminente rutura de stock, permitindo que sejam tomadas medidas preventivas para evitar a ocorrência. Também poderia ser interessante associar aos equipamentos ficheiros de certificação das ferramentas, (visto que algumas necessitam de passar por uma inspeção antes de serem reutilizadas).

Outro módulo a ser adicionado está relacionado com a utilização de outras tecnologias estudadas (para além do QR code), quanto à localização dos equipamentos de forma mais automática (como as tags de RFID). Contudo, estas poderiam trazer custos adicionais associados a aquisição de equipamentos especializados, pelo que seria necessário fazer previamente uma análise de custos para perceber se o investimento faria sentido.



# Referências

*A Quick Guide to Understand the Event-driven Architecture* (2019) Softobiz Technologies India. Available at: <https://softobiz.com/understanding-the-event-driven-architecture/> (Accessed: 5 February 2022).

Adobe, inc (2022) *JPEG versus PNG: Qual é melhor?* | Adobe. Available at: <https://www.adobe.com/pt/creativecloud/file-types/image/comparison/jpeg-vs-png.html> (Accessed: 14 June 2022).

Anne, M. (2019) *What is the difference between the master and slave mode of BLE?* Available at: <https://www.linkedin.com/pulse/what-difference-between-master-slave-mode-ble-ma-bluetooth-module/> (Accessed: 25 June 2022).

*Apple AirTag Silver MX532AM/A* (2021) Best Buy. Available at: <https://www.bestbuy.com/site/apple-airtag-silver/6461348.p?skuld=6461348> (Accessed: 2 February 2022).

Bacinger, T. (2022) *What is Bootstrap? A Short Bootstrap Tutorial on the What, Why, and How*, *Toptal Engineering Blog*. Available at: <https://www.toptal.com/front-end/what-is-bootstrap-a-short-tutorial-on-the-what-why-and-how> (Accessed: 13 June 2022).

Bourque, P., Fairley, R.E., and IEEE Computer Society (2014) *Guide to the software engineering body of knowledge*.

Brooks, F. (1986) 'No Silver Bullet – Essence and Accident in Software Engineering', p. 16.

Cook, S. (2020) *Managing image uploads in your Browser: Base64 vs ObjectURLs*, *Medium*. Available at: <https://levelup.gitconnected.com/managing-image-uploads-in-your-browser-base64-vs-objecturls-part-1-cec5ee9af7be> (Accessed: 14 June 2022).

Cornett, S. (2013) *Minimum Acceptable Code Coverage*. Available at: <https://www.bullseye.com/minimum.html> (Accessed: 16 June 2022).

Corporation, R. (2008) *Interaction Between the House of Quality (within Design for Six Sigma) and the FMEA (within Design for Reliability)*. Available at: <https://www.weibull.com/hotwire/issue93/hottopics93.htm> (Accessed: 10 February 2022).

Dats, O. (2021) *Node.js vs. .NET: What to Choose in 2022 - TechMagic, Blog* | TechMagic. Available at: <https://www.techmagic.co/blog/node-js-vs-net-what-to-choose/> (Accessed: 13 June 2022).

Dawson, C. (2007) *A practical guide to research methods: a user-friendly manual for mastering research techniques and projects*. Oxford: How To Books. Available at: <http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=211035> (Accessed: 1 May 2021).

*Definition of Information Technology (IT) - Gartner Information Technology Glossary (2022)* Gartner. Available at: <https://www.gartner.com/en/information-technology/glossary/it-information-technology> (Accessed: 3 February 2022).

*Definition of Operational Technology (OT) - Gartner Information Technology Glossary (2022)* Gartner. Available at: <https://www.gartner.com/en/information-technology/glossary/operational-technology-ot> (Accessed: 31 January 2022).

*Different Wi-Fi Protocols and Data Rates (2021)* Intel. Available at: <https://www.intel.com/content/www/us/en/support/articles/000005725/wireless/legacy-intel-wireless-products.html> (Accessed: 1 February 2022).

Ebert, C. and Duarte, C.H.C. (2018) 'Digital Transformation', *IEEE Software*, 35(4), pp. 16–21. doi:10.1109/MS.2018.2801537.

Elbakly, R. and Youssef, M. (2019) 'Crescendo: An Infrastructure-free Ubiquitous Cellular Network-based Localization System', in *2019 IEEE Wireless Communications and Networking Conference (WCNC). 2019 IEEE Wireless Communications and Networking Conference (WCNC)*, Marrakesh, Morocco: IEEE, pp. 1–6. doi:10.1109/WCNC.2019.8885420.

Firtman (firt.dev), M. (2020) 'Progressive Web Apps on iOS are here 🚀', *Medium*, 18 August. Available at: <https://medium.com/@firt/progressive-web-apps-on-ios-are-here-d00430dee3a7> (Accessed: 4 February 2022).

Fitzgibbons, L. (2022) *What are front end and back end? Definition from WhatIs.com, WhatIs.com*. Available at: <https://www.techtarget.com/whatis/definition/front-end> (Accessed: 13 June 2022).

Flanagan, D. (2020) 'JavaScript: The Definitive Guide', p. 707.

manufacturing Group, W. (2010) *Quality Function Deployment*.

Hastings, N.A.J. (2015) *Physical Asset Management*. Cham: Springer International Publishing. doi:10.1007/978-3-319-14777-2.

*Here's How Wi-Fi Actually Works (2022)* Time. Available at: <https://time.com/3834259/wifi-how-works/> (Accessed: 2 February 2022).

Hermes, D. (2015) *Xamarin mobile application development: cross-platform C# and Xamarin.Forms fundamentals*. New York, NY: Apress (The expert's voice in mobile application development).

HUFFORD, B. (2022) *What is a Mockup? (+How to Create a Mockup in 2022)*, *Clique Studios*. Available at: <https://cliquestudios.com/mockups/> (Accessed: 13 June 2022).

Ibrahim, D.M. (2019) 'Internet of Things Technology based on LoRaWAN Revolution', in *2019 10th International Conference on Information and Communication Systems (ICICS). 2019 10th International Conference on Information and Communication Systems (ICICS)*, Irbid, Jordan: IEEE, pp. 234–237. doi:10.1109/IACS.2019.8809176.

[Infographic] *A Guide to Mobile App Development: Web vs. Native vs. Hybrid* (2019) Clearbridge Mobile. Available at: <https://clearbridgemobile.com/mobile-app-development-native-vs-web-vs-hybrid/> (Accessed: 26 October 2020).

Ionic Article: *What is Hybrid App Development?* (2019). Available at: <https://ionic.io/resources/articles/what-is-hybrid-app-development> (Accessed: 4 February 2022).

Islam, B. *et al.* (2019) 'LoRaIn: Making a Case for LoRa in Indoor Localization', in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Kyoto, Japan: IEEE, pp. 423–426. doi:10.1109/PERCOMW.2019.8730767.

Koen, P.A. *et al.* (2002) 'Effective Methods, Tools, and Techniques', *The PDMA ToolBook for New Product Development*, p. 32.

Kopetz, H. (2011) *Real-Time Systems*. Boston, MA: Springer US (Real-Time Systems Series). doi:10.1007/978-1-4419-8237-7.

Kortuem, G. *et al.* (2010) 'Smart objects as building blocks for the Internet of things', *IEEE Internet Computing*, 14(1), pp. 44–51. doi:10.1109/MIC.2009.143.

Krunal (2022) 'Angular vs React Comparison in 2022: Which is Better and Why', *AppDividend*, 1 February. Available at: <https://appdividend.com/2022/02/02/angular-vs-react/> (Accessed: 13 June 2022).

Master, W. (2022) *LoRaWAN*. Available at: <https://ecsxtal.com/lora-timing> (Accessed: 3 February 2022).

Mckinsey (2020) 'How COVID-19 has pushed companies over the technology tipping point—and transformed business forever', p. 9.

MDN (2022) *HTTP | MDN*. Available at: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP> (Accessed: 26 June 2022).

MDN, W.Api. (2022a) *Media Capture and Streams API (Media Stream) - Web APIs | MDN*. Available at: [https://developer.mozilla.org/en-US/docs/Web/API/Media\\_Streams\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Media_Streams_API) (Accessed: 14 June 2022).

MDN, W.Api. (2022b) *WebRTC API - Web APIs | MDN*. Available at: [https://developer.mozilla.org/en-US/docs/Web/API/WebRTC\\_API](https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API) (Accessed: 14 June 2022).

Melzer, B. (2016) 'Reference Architectural Model Industrie 4.0 (RAMI 4.0)', p. 15.

Mendes, V. (2018) *Aplicação de tecnologias livres na gestão de pessoas: proposta de um sistema de identificação e localização de professores em sala de aula*. ResearchGate.

National Geographic Society (2022) *Triangulation | National Geographic Society*. Available at: <https://education.nationalgeographic.org/resource/triangulation-sized> (Accessed: 25 June 2022).

Neumann, P., Montavont, J. and Noel, T. (2016) 'Indoor deployment of low-power wide area networks (LPWAN): A LoRaWAN case study', in *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, New York, NY: IEEE, pp. 1–8. doi:10.1109/WiMOB.2016.7763213.

*O que é gestão horizontal e vertical e suas principais características* (2019) Portal. Available at: <https://www.ibccoaching.com.br/portal/o-que-e-gestao-horizontal-e-vertical-e-suas-principais-caracteristicas/> (Accessed: 29 March 2022).

Price, T.M., Cybersecurity (2020) *Cybersecurity / Regulatory Overview - ppt download*. Available at: <https://slideplayer.com/slide/16122887/> (Accessed: 31 January 2022).

Ravald, A. and Grönroos, C. (1996) 'The value concept and relationship marketing', *European Journal of Marketing*, 30(2), pp. 19–30. doi:10.1108/03090569610106626.

Richards, M. (2017) 'Software Architecture Patterns', p. 54.

Rocha, Á. *et al.* (2008) 'Conselho Editorial / Consejo Editorial', p. 83.

Rocha, A. (2022) 'Requirements Engineering -LABDSOFT'. *LABDSOFT: Laboratório de Desenvolvimento de Software*, ISEP, 28 January.

Saaty, T.L. (1990) 'How to make a decision: The analytic hierarchy process', *Decision making by the analytic hierarchy process: Theory and applications*, 48(1), pp. 9–26. doi:10.1016/0377-2217(90)90057-I.

Savić, D. (2020) 'COVID-19 and Work from Home: Digital Transformation of the Workforce', *Grey Journal*, 16, pp. 101–104.

Seara.com (2021) *INEGI reforça infraestrutura laboratorial e experimental com expansão das instalações*, INEGI. Available at: <http://www.inegi.pt/pt/noticias/inegi-reforca-infraestrutura-laboratorial-e-experimental-com-expansao-das-instalacoes/> (Accessed: 15 January 2022).

*Should I Publish an iOS or Android App?* (2015) *GoodBarber*. Available at: <https://www.goodbarber.com/blog/should-i-publish-an-ios-or-android-app-a612/> (Accessed: 29 March 2022).

Sitefane, J. (2006) 'Desenvolvimento de Software Centrado no Domínio', p. 10.

SmartBear, bear (2022) *What is Mobile First?*, *smartbear.com*. Available at: <https://smartbear.com/learn/performance-monitoring/what-is-mobile-first/> (Accessed: 13 June 2022).

Somasegar, S. (2008) 'Application Architecture Guide 2.0', p. 381.

Taherdoost, H. (2017) 'Decision Making Using the Analytic Hierarchy Process (AHP); A Step by Step Approach', 2, p. 4.

Taylor, D. (2020) *MongoDB vs. MySQL: What's the difference?* Available at: <https://www.guru99.com/mongodb-vs-mysql.html> (Accessed: 13 June 2022).

U.Farooq, M. et al. (2015) 'A Review on Internet of Things (IoT)', *International Journal of Computer Applications*, 113(1), pp. 1–7. doi:10.5120/19787-1571.

Wells, D. (ed.) (1987) *Guide to GPS positioning*. 2. print., with corr. Fredericton, New Brunswick, Canada: Canadian GPS Associates.

*What Makes QR Code Different from Barcode in Asset Tracking? - Asset Infinity* (2021). Available at: <https://www.assetinfinity.com/blog/barcode-qr-code-difference-in-asset-tracking> (Accessed: 4 February 2022).

Wieggers, K.E. and Beatty, J. (2013) *Software requirements*. Third edition. Redmond, Washington: Microsoft Press, s division of Microsoft Corporation.

Woodall, T. (2003) 'Conceptualising "Value for the Customer": An Attributional, Structural and Dispositional Analysis', p. 44.

Xanthopoulos, S. and Xinogalos, S. (2013) 'A comparative analysis of cross-platform development approaches for mobile applications', in *Proceedings of the 6th Balkan Conference in Informatics on - BCI '13. the 6th Balkan Conference in Informatics*, Thessaloniki, Greece: ACM Press, p. 213. doi:10.1145/2490257.2490292.

# Anexos

Nesta secção vão ser apresentadas as perguntas desenvolvidas e respostas obtidas no processo de licitação, as entrevistas (no qual fora realizadas diretamente questões aos utilizadores preparadas de antemão).

## Reunião Coordenador técnico nº1

Data	29/11/2021
Local	INEGI
Horário	11h
Participantes	Coordenador técnico Mestrando

## Questões

- **Coordenador técnico**
  - **Q1 – Qual a solução atual para a gestão dos equipamentos de laboratório?**

Neste momento existe uma folha A4 com um template. Contudo esta não esta dedicada a todos os componentes do laboratório. É apenas utilizada para requisitar equipamentos de dois armários (que cotem os equipamentos mais dispendiosos, como o berbequim, entre outros). Idealmente a nova aplicação seria estendida a outros armários e laboratórios existentes.

- **Q2 – Quem vai usar a solução?**

Todo o pessoal técnico poderá vir a utilizar esta solução. Contudo será necessário definir perfis para limitar quem tem acesso a determinadas funcionalidades da aplicação (como adicionar equipamentos, e remover).

Há pessoas que vão muito raramente ao laboratório. O ideal seria tornar o processo mais automático para permitir uma mais fácil adesão ao mesmo. Muitas vezes o pessoal não está habituado à logica de trabalhar em oficina e precisam de aceder a equipamentos para realizar uma dada operação para um projeto. A aplicação também visa uniformizar o processo de requisição e permitir que mais pessoas o utilizem de forma facilitada.

- **Q3 – Quais as funcionalidades espectáveis para a aplicação?**

Há duas funcionalidades principais. A localização das ferramentas é a mais importante (permitir saber onde elas estão). Outra coisa muito importante é controlar quem as está a utilizar e tempos de requisição das mesmas (de forma a fazer uma melhor gestão dos mesmos).

É também importante fazer a inventariação dos equipamentos. Esperamos que esta aplicação possa ajudar com isto.

- **Q4 – Alguma restrição relativamente ao tipo de aplicação? (Mobile, browser)**

Gostaria que a aplicação fosse facilmente acessível. Por exemplo a partir do telemóvel, mas também gostaria que a aplicação pudesse ser acedida a partir de um computador. A disponibilidade nos telemóveis é importante. Mas no futuro podemos prever que venha a ter um tablet no laboratório para poder aceder a aplicação de requisição dos equipamentos. Isto porque pode suceder haver pessoas que vão ao laboratório e não levem o telemóvel e precisem de fazer a requisição do equipamento (temos de garantir as condições para que as pessoas usem a aplicação).

- **Q5 – Que benefício poderá trazer esta aplicação?**

Menos tempo perdido na procura de ferramentas, menor número de ferramentas perdidas / espalhadas pelos laboratórios, registo da utilização das ferramentas, identificar necessidade de inventário de novas ferramentas, identificar necessidade de aquisição de ferramentas que estejam a ser muito requisitadas.

- **Q6 – Qual o tipo de informação a ser guardada em cada processo?**

No registo de equipamento quero mínimo de inputs possível para garantir a máxima adesão ao processo (facilitar o mesmo). A pessoa pede a ferramenta por um período de tempo. Mas pode ser necessário definir um limite máximo de requisição dos equipamentos. Pode fazer sentido colocar uma caixa de opções. Por exemplo: até ao fim do dia, até ao fim da manhã, próximos 2 dias, excepcional e neste caso a pessoa introduz as horas que pretende requisitar o equipamento. Há também situações em que pode ser necessário levar a caixa de ferramentas para um cliente e pode ser necessário requisitar por períodos mais longos (estas são situações excecionais com períodos muito longos, de 1 ou mais meses). Garantir que uma pessoa recebe um ticket caso não respeite o período de devolução.

O caso anterior é para a situação de utilização.

Para o caso de equipamentos (registo destes). É importante ter informação que permita pessoal novo e que normalmente não está habituado a trabalhar com ferramentas também aprendam o que são os diferentes equipamentos que existem no laboratório. Para equipamentos mais técnicos como multímetro, berbequim e rebarbadora, pode ser bom ter alguma informação sobre que cuidados e as normas de utilização estas possam ter no seu manuseamento. Campo de boas praticas ou manuais de utilização também podem ser interessantes.

Gostava também de ter um campo de acessórios ou que outros equipamentos possam estar correlacionados uns com os outros, mas sem entrar em demasiado pormenor de modo a não tornar a ferramenta demasiado exaustiva na sua utilização. Perceber que equipamentos se podem utilizar com outros equipamentos.

Pode haver casos em que há várias ferramentas que utilizam os mesmos acessórios.

Quando introduzo uma ferramenta nova no sistema quero colocá-la dentro de um campo categorias. Por exemplo uma categoria “chave para parafusos” faz sentido. Pode haver uma categoria “outros” para meter a ferramenta. A dimensão de uma ferramenta pode ser importante também. Uma foto também faz sentido.

O processo é – tenho um parafuso que quero apertar e procuro na ferramenta por categoria e depois por tamanho. O tamanho são dimensões standard, mas a forma não. Seleciono “chaves de porcas” e depois ter um descritivo associado à ferramenta que permita perceber para que serve a ferramenta. Os tamanhos podem estar numa sublista associada a ferramenta. E eu posso selecionar vários tamanhos de ferramenta para poder levar.

- **Q7 – Que informações gostaria de saber, ponto de vista de gestão?**

Ferramentas com mais procura. Tempos de uso das ferramentas e saber quem esta a utilizar uma dada ferramenta. Era bom ter uma dashboard que pudesse aceder para consultar esta informação. Uma lista das ferramentas utilizadas aparecerem associadas com o nome da pessoa que a esta a utilizar e um top com as ferramentas mais utilizadas.

- **Q8 – Tem alguma preferência pela forma como a informação é apresentada?**

Uma interface acessível que toda a gente possa consultar.

- **Q9 – Em termos de usabilidade o que pretende?**

O mais intuitiva possível e com o mínimo de *clicks* possíveis ou ações para fazer a requisição de um equipamento.

- **Q10 – Que cuidados são necessários com os dados?**

Quem pode ver quem esta com a ferramenta. Ainda não se sabe se pode ser algo limitado a um determinado perfil ou publico, mas preferencialmente limitado a unidade (perfis e login são necessários). Temos informação sobre o que a pessoa está a fazer no momento e isso deve ser tratado com respeito e cuidado. Por exemplo, não poder usar utilizar a informação contra a pessoa em casos legais (de modo a não ver a ferramenta como uma forma de traçabilidade excessiva da pessoa que a esta utilizar). Pode fazer sentido não saber quem tem a ferramenta, mas que alguém a tem. Não conseguir identificar a pessoa ou o registo de quem teve a ferramenta ser apenas acedido a determinados perfis e não algo publico. Um utilizador comum deve saber que a ferramenta esta a ser utilizada, mas não saber quem a esta utilizar.

- **Q11 – Quais são os perfis de acesso?**

-**User** que vai requisitar equipamentos e que não pode adicionar novas ferramentas (só consultar e requisitar).

-Um perfil para adicionar e gerir as ferramentas (pessoal técnico) e que também pode requisitar equipamentos.

- **Admin** (por exemplo o Diretor de unidade, coordenador técnico, pessoas com responsabilidade sobre o laboratório) que pode fazer tudo o que os outros fazem mais

a visualização de informação sensível (descrita no ponto Q10). O perfil Admin pode adicionar novos *users* a aplicação.

- **Q12 –Existe alguma funcionalidade específica que gostaria de ver implementada na aplicação?**

Tudo o descrito anteriormente. A funcionalidade de QR *code* pode ser interessante para filtrar as ferramentas associadas a um armário. Ver quais as ferramentas desse armário que estão requisitadas e que existem para requisitar.

- **Q13 –Gostaria de acrescentar mais alguma coisa?**

Não tenho nada a acrescentar.

## Reunião Pessoal técnico nº1

Data	06/12/2021
Local	INEGI – laboratório 007
Horário	10h45min
Participantes	Técnico de oficina Mestrando

### Questões

- **Pessoal técnico**

- **Q1 – Pode aprofundar o processo de requisição de equipamentos atualmente implementado?**

Existe uma folha A4 no laboratório 007 que usamos para escrever 7 campos. Temos de ter em atenção a data de saída e a data de entrega do equipamento. Tiramos a folha da mica e preenchemos os campos lá indicados.

- **Q2 – Que dados sobre o processo de requisição de equipamentos pretende guardar no sistema?**

Queremos manter a informação que já é guardada atualmente. As datas e o responsável do equipamento. A que não me parece ser tao importante e que está a ser guardada é a descrição do projeto a que o equipamento está associado. O restante parece ser tudo relevante. Nós queremos e que o pessoal fique responsável pelo conjunto todo. Por exemplo se o equipamento tem acessórios a pessoa tem de levar os acessórios também, porque não podemos utilizar o equipamento sem o resto dos acessórios. A pessoa que leva o equipamento também leva os acessórios e ficaria responsável por eles. Mas acontece por vezes que levam só determinados acessórios para utilizar com outras ferramentas. Alguns equipamentos têm uma referência, mas nem todos. Era uma ideia pôr em todos os equipamentos essa referência.

- **Q3 – Que informações gostaria de saber, ponto de vista de utilização?**

Podia haver um campo opcional para localizar o equipamento. Por vezes as pessoas levam o equipamento para empresas e não sabemos onde estão. Dentro do INEGI é fácil localizar a pessoa com o equipamento, mas quando esse equipamento é levado para fora já sabemos que nem há necessidade de falar com essa pessoa porque o equipamento não está nas instalações.

- **Q4 – Quais as principais funcionalidades que espera que estejam presentes, bem como os dados apresentados?**

Queremos uma funcionalidade para requisitar o equipamento de laboratório. Reservar um equipamento pode ser interessante. Por exemplo saber que o

equipamento já está reservado para uma data no futuro mesmo que esteja atualmente disponível.

○ **Q5 – Tem alguma preferência pela forma como a informação é apresentada?**

Uma foto do equipamento com algumas das suas características e informações deste. Era interessante poder ver os detalhes do equipamento (várias fotos de vários pontos de vista, e com o estojo aberto ou fechado caso tenha um estojo). Queríamos ver se dentro da bolsa de um equipamento tem todos os acessórios. Para perceber se falta alguma coisa naquela bolsa ou se o equipamento tem alguma coisa desaparecida. Uma vista interessante seria ver um horário do equipamento para ver quando é que alguém o requisita.

○ **Q6 – Em termos de usabilidade o que pretende?**

Ter alguma forma de encontrar o equipamento rapidamente na aplicação. Filtrar por salas onde o equipamento se encontra. Gostava de garantir que as pessoas usassem os acessórios de uma ferramenta com essa ferramenta e não utilizassem os acessórios de outro laboratório. Lista de algumas categorias: equipamentos elétricos e não elétricos, rebarbadora, pistola de ar quente, chaves mecânicas, conjuntos. Períodos longos de utilização são os mais importantes de serem registados na aplicação. Por vezes as pessoas têm necessidade de requisitar um carrinho completo com todos os equipamentos. Esses equipamentos desse carrinho devem ficar todos automaticamente bloqueados para essa utilização.

○ **Q7 – Existe alguma funcionalidade específica que gostaria de ver implementada na aplicação?**

Os carrinhos podem sair do sítio e serem requisitados (uma vez até foi para a Alemanha). Por vezes o equipamento (uma chave por exemplo) é perdido ou partido. E passado algum tempo esse equipamento pode voltar a aparecer. Gostava de perceber qual é o estado dessa ferramenta. Se esta requisitada ou perdida ou partiu e precisa de ser substituída. (a pessoa que requisitou esse equipamento há de saber do paredeiro desta).

○ **Q8 – Gostaria de acrescentar mais alguma coisa?**

Queria que a aplicação fosse muito fácil de utilizar. Ou seja, sem grande esforço as pessoas conseguirem introduzir a informação direito e útil. O problema do papel e mesmo esse. E demorado, tem de pegar na caneta e preencher com a informação. As vezes as pessoas esquecem se que estão com o equipamento com elas e podem se esquecer de voltar a colocar no sítio. Gostava de ter acesso a aplicação no telemóvel e no computador.

## Reunião Pessoal técnico nº2

Data	10/12/2021
Local	INEGI - Laboratório 007
Horário	15h45min
Participantes	Técnico Mestrando

### Questões

- **Pessoal técnico**

- **Q1 – Pode aprofundar o processo de requisição de equipamentos atualmente implementado?**

Atualmente é utilizada uma folha de papel A4 muito pouco utilizada pelo pessoal. Maioritariamente e utilizada pelo pessoal técnico. Muitas vezes não sabemos onde estão as ferramentas e temos de perder tempo para a localizar.

- **Q2 – Que dados sobre o processo de requisição de equipamentos pretende guardar no sistema?**

Quem, onde e quando. O onde não tem de ser muito preciso. Pode ser apenas saber que esta ca dentro do INEGI ou está a ser utilizado la fora. Datas de requisição e data prevista de entrega da ferramenta.

- **Q3 – Que informações gostaria de saber, ponto de vista de utilização?**

Características gerais das ferramentas, gama, pequena descrição da ferramenta. Uma coisa que poderia ser útil e saber que gamas de força a ferramenta consegue fazer (por exemplo, nas ferramentas dinamométricas e preciso saber que forças ela consegue apertar. Pode ser útil para saber se temos alguma ferramenta no INEGI que consiga fazer esse aperto). Saber a localização da ferramenta (onde normalmente e guardada também pode ser útil). Dimensões das ferramentas (chave de bocas número 20 / 23; as chaves inglesas afináveis têm uma gama de abertura) (há sempre uma característica associada ao uso delas que deveria estar incluída na definição da ferramenta). Saber as características da ferramenta e a localização delas (porque pode haver umas que são iguais, mas estão em carrinhos diferentes e são de marcas diferentes). O estado da ferramenta também e importante. Nas brocas e ferramentas mais frágeis e importante saber que a ferramenta esta partida. Porque é preciso saber se a fermenta esta partida ou se está perdida (e dar a baixa da ferramenta) (ou em bom estado). E notificar se é preciso comprar uma peça de substituição. (Uma ideia sugerida para automatizar o processo consistiria em utilizar os cartões que já possuímos do INEGI como leitores de RFID para identificar a pessoa e saber que ferramenta ela levou, cada ferramenta teria também de ter uma tag de RFID, seriam também necessários pórticos/ leitores de RFID).

- **Q4 – Quais as principais funcionalidades que espera que estejam presentes, bem como os dados apresentados?**

Wiki da ferramenta. Imagens, características, e instruções de como utilizar a ferramenta, como armazenar, boas praticas. Outras informações como EPIs – equipamentos de proteção individuais, que tem de ser utilizados com a ferramenta (por exemplo luvas, óculos de proteção, botas, tampões para os ouvidos, e se possível saber onde estão os EPIs a utilizar com a ferramenta e advertências no uso da ferramenta). Um exemplo de boas práticas em chaves dinamométricas seria a necessidade de serem descarregadas antes de ser guardadas, isto porque a molas destas quando em carga deforma a mola e descarrega o equipamento. Isto estraga a ferramenta.

Uma coisa importante é garantir que as pessoas guardam as coisas no mesmo sítio onde as encontraram porque senão não sabemos onde vão ser guardadas as ferramentas.

- **Q5 – Tem alguma preferência pela forma como a informação é apresentada?**

Informação em lista seria interessante. O estilo semelhante ao que existe na **EletoDoc**. Dividir em subcategorias e a partir daí a pessoa ia acedendo às informações da ferramenta. Por exemplo – categoria principal elétricas. Gostava de saber que tipo de parafuso a ferramenta aperta (Torx interior). Não sabemos se é melhor categorizar pela característica da ferramenta ou pela categoria do que vai apertar (da funcionalidade que vai realizar). Eu preciso de apertar X. O que tenho cá que possa fazer isso.

As pesquisas podem ser do género:

1. Copo de chave de caixa. Copo de roquette.
2. Extensão de quadra de 1/8 para 1/8.
3. Ferramenta para sextavados exterior ou para parafuso Philips ou para parafuso sextavado interior.

Mas depende de ferramenta para ferramenta... há algumas ferramentas que podemos aplicar a pesquisa por funcionalidade. Outras em que faz só sentido pesquisar pelo nome. Preciso avaliar melhor.

- **Q6 – Em termos de usabilidade o que pretende?**

Fácil usabilidade. Pesquisa rápida de ferramentas por funcionalidade ou por nome. Para requisitar um equipamento interessa guardar o tempo de requisição (data de início e de fim). O menor número de clicks possível para requisitar o equipamento (se der muito trabalho a requisitar o equipamento ou for preciso muitos clicks pode fazer com que haja menos adesão à ferramenta).

- **Q7 – Existe alguma funcionalidade específica que gostaria de ver implementada na aplicação?**

Consulta muito esporádica. O registo da ferramenta e a localização da ferramenta (saber onde ela está ou com quem está) é a funcionalidade que vejo a ser mais utilizada. Visto que vai ser mais utilizada por pessoal técnico, o mais útil é saber a localização da ferramenta. Para pessoal mais recente pode ser interessante ter em atenção as outras funcionalidades.

- **Q8 – Gostaria de acrescentar mais alguma coisa?**

Notificações de devolução. É recorrente as pessoas esquecerem-se de devolver o equipamento. Poder notificar as pessoas que requisitaram o equipamento quando o tempo de requisição estiver a acabar.

Gostava de poder também aceder a aplicação no computador e não só no telemóvel.

## Reunião informático nº1

Data	07/12/2021
Local	INEGI
Horário	16h10min
Participantes	Mestrando Informático

### Questões

- **Informático**

- **Q1 – Qual a solução atual para a gestão dos equipamentos de laboratório?**

Neste momento existe uma solução em papel. Não existe nada em formato digital. O papel que existe afixado não cobre todos os equipamentos atualmente existentes (apenas os do laboratório, estando este afixado ao armário). Uma solução que cobrisse todos os equipamentos de forma centralizada era importante.

- **Q2 – Quem vai usar a solução?**

As equipas que requisitam atualmente os equipamentos. A equipa dos técnicos mecânicos mais concretamente os responsáveis pelos 2 laboratórios e pelas bancas de maquinaria. As equipas que requisitam os equipamentos (pelo menos são eles que irão utilizar mais vezes isto). Esporadicamente pode haver pessoas interessadas que vão realizar uma operação ao laboratório e que precisam do equipamento (também é interessante que a ferramenta seja *user friendly* para eles).

- **Q3 – Quais as funcionalidades espectáveis para a aplicação?**

Gostava que a ferramenta fizesse a gestão da disponibilidade de equipamentos (criação, edição de equipamentos e gestão da sua disponibilidade. Entenda-se esta última como perceber o estado do equipamento, disponível para requisitar ou não). Gestão de utilizadores pode ser necessário (perfis de acesso). Listas de espera para requisição de equipamentos. Notificações de disponibilidade do equipamento. Este é o core da aplicação. Mas também poderia ser interessante gestão de stocks.

- **Q4 – Alguma restrição relativamente ao tipo de aplicação? (mobile, browser, responsiva)**

A aplicação deve ser orientada ao browser para permitir facilmente acesso a partir de qualquer ponto da rede. É também importante que esta seja responsiva porque é antevisto que uma forma rápida de acesso seja o *smart phone*. Por isso o *front-end* deve se adaptar à janela de visualização.

Grande parte das tecnologias utilizadas devem ser facilmente integradas no nosso sistema. Não deve haver tecnologias que imponham limites. De preferência que sejam em boa parte tecnologias já utilizadas cá, por questões de manutenção futuras (não

tem de ser 100%, mas é algo a ter em conta). É expectável que a aplicação seja disponibilizada na rede interna (não vai estar disponível exteriormente).

- **Q5 – Que benefício poderá trazer esta aplicação?**

Controlo sobre o estado dos equipamentos. Queremos tirar uma preocupação dos técnicos sobre os equipamentos. Neste momento os técnicos têm de ir sempre verificar ao laboratório se o equipamento esta disponível. A responsabilidade de saber o estado do equipamento passa a estar do lado do sistema e não do lado das pessoas (libertando-as assim para outras tarefas). Queremos tornar a utilização desta aplicação um procedimento interno. Ou seja. A requisição dos equipamentos deve ser feita recorrendo a esta aplicação.

- **Q6 – Qual o tipo de informação a ser guardada em cada processo?**

A identificação exata do equipamento (nome, modelo e fabricante... pelo menos antevejo estes tipos de informação). Dimensões do equipamento. O estado (ocupado, disponível). Data de requisição e a data em que o equipamento ficou efetivamente ocupado com a requisição (com aquele utilizador). Data em que o equipamento foi libertado (ou data final de ocupação / fim de trabalho). Informação do utilizador pode ser o *username* (informação que ajude a identificar o utilizador). Quem, quando e o quê.

- **Q7 – Que informações gostaria de saber, ponto de vista de gestão?**

Alem do referido anteriormente da informação operacional, também pode ser interessante agregados mensais (ou outros períodos de tempo) sobre que equipamentos foram utilizados mais vezes e por quem. Outra que também pode ser interessante é perceber que equipamentos foram menos utilizados. Perceber se temos de adquirir novos equipamentos ou se temos equipamentos a mais.

- **Q8 – Tem alguma preferência pela forma como a informação é apresentada?**

Quero que o que for apresentado seja feito de forma consistente. Se houver 3 formulários de registo de equipamento, todos eles devem ser coerentes e ter o mesmo aspeto. Que a nível visual como a nível de aspeto. Ser consistentes em todos os momentos e em toda a aplicação. Mesmo a nível de títulos devem ser parecidos. Depende do tipo de informação e do objetivo, mas deve ser sempre o mais coerente possível.

- **Q9 – Em termos de usabilidade o que pretende?**

Em termos de usabilidade geral, o menos é mais. Pouca informação, mas a suficiente para perceber o que é suposto fazer. O menor número de *clicks* para aceder a funcionalidade. Aspeto *clean* com poucas cores.

- **Q10 – Que cuidados são necessários com os dados?**

Considerando que os dados serão sempre armazenados em base de dados nas nossas instalações, gostava de garantir que as palavras-passe não ficassem em texto legível. E em termos da aplicação não antevejo limitações de visualização da informação. Esta aplicação deve partilhar a informação com todos os *users* da aplicação.

Não correr o risco da utilização do equipamento não esta disponível para todos pode ser interessante. Ou seja. A pessoa não poder dirigir se a quem esta a utilizar o equipamento e pedir “emprestado por 5 min” o equipamento. Isto poderia comprometer a utilização do equipamento porque pessoas poderiam “emprestar o equipamento” e depois não se saberia exatamente quem teria o mesmo. Poderia ser interessante quando a pessoa entra na aplicação ter logo uma vista dos pedidos de requisição que existem para as ferramentas que o user esta a utilizar.

- **Q11 – Quais são os perfis de acesso?**

O administrador será algo como um informático (uma pessoa mais técnica). Há também a questão da periodicidade de introdução de novas ferramentas no sistema. Se é no início meter todas as ferramentas no sistema ou se é uma nova por semana. Depende da dinâmica com que a introdução de novas ferramentas acontece. Se forem poucas de longe a longe o perfil admin pode ser responsável por tal. Se for muito mais dinâmico pode ser necessário ter um perfil dedicado para isto.

- **Q12 – Existe alguma funcionalidade especifica que gostaria de ver implementada na aplicação?**

Controlo de equipamentos por visão. Pode partir de uma simples prova de conceito com uma diversidade curta de equipamentos, que através de uma imagem gerada por uma camara seja possível identificar características do equipamento e dessa forma saber qual é (isso levava a aplicação para outro nível). Outra prova de conceito seria localização do equipamento dentro do edifício. Esta poderia estar limitada a apenas aos equipamentos mais dispendiosos. Apenas localizar o equipamento dentro do edifício e não fora. Ter sempre em atenção o nível de privacidade das pessoas... por isso pode ser algo sensível localizar um equipamento (porque estará com uma pessoa).

- **Q13 –Gostaria de acrescentar mais alguma coisa?**

Não quero acrescentar mais nada.

## Reunião informático nº2

Data	07/12/2021
Local	INEGI
Horário	17h
Participantes	Mestrando Informático

### Questões

- **Q1 - Existem algumas restrições relacionadas com a *User Interface*? Se sim quais?**

Sendo uma aplicação web esta deve correr nos browsers mais modernos e ter a capacidade de se adaptar a dispositivos móveis.

- **Q2 - Existem algumas restrições relacionadas com o hardware? Se sim quais?**

Aquelas que garantem a execução da aplicação no sistema. Base dados, *Back-end*, e clientes do *front-end*. (só vendo os requisitos mínimos dos serviços para correr). Armazenamento, memória.

- **Q3 - Existem algumas restrições relacionadas com o software? Se sim quais?**

Restrições técnicas ao nível dos browsers (devem ser capazes de executar a aplicação). Aplicação deve ser otimizada para as versões mais recentes dos browsers utilizados. Devem ser utilizadas tecnologias que funcionem nos nossos sistemas (no servidor Linux), e preferencialmente tecnologias que já existem ca para questões de manutenções futuras.

- **Q4 - Existem políticas de segurança já definidas? Se sim quais?**

A ferramenta só vai ser utilizada na rede interna. Palavras-passe não devem estar em texto legível (*plain text*) (proteção desta informação dos utilizadores que estão a utilizar os equipamentos)

- **Q5 - Será necessário aceder a aplicações externas? Se sim quais?**

Não será necessário aceder a aplicações externas.

- **Q6 - Existem restrições com a linguagem utilizada para o desenvolvimento do sistema? Se sim quais?**

Devem ser linguagens que permitam a execução nos browsers mais recentes e que encaixem nos nossos sistemas (e que permitam a fácil manutenção).

- **Q7 - Existem restrições com a base de dados ou outras tecnologias e ferramentas a serem utilizadas? Se sim quais?**

Devem encaixar nos nossos sistemas. Utilização de MySQL é preferencial porque já temos instâncias instaladas localmente.

- **Q8 - Existem restrições com o ambiente do sistema? Se sim quais?**

Capacidade de espaço de armazenamento do servidor (em princípio não haverá problema, mas temos de ter isso em conta).

- **Q9 - É possível o uso de componentes externos?**

Sim, preferencialmente de utilização gratuita. Caso não seja de utilização gratuita deve ser justificado a necessidade de recorrer a esses componentes.

## **Requisitos do sistema**

### REQ-01: Gerir utilizadores

#### **Descrição**

O coordenador técnico / Admin tem a responsabilidade de fazer a gestão de utilizadores do sistema. Esta função inclui criar novos utilizadores, remover e atualizar a informação bem como fazer a associação de perfis.

#### **Simulação/ sequência de Respostas**

##### Pré-condições:

- O coordenador técnico / Admin tem de ter uma conta criada no sistema com as devidas permissões para gerir utilizadores e os respetivos perfis.

##### Fluxo de Eventos:

1. O coordenador técnico / Admin inicia sessão no sistema;
2. O coordenador técnico / Admin acede ao separador de gestão de utilizadores;
3. O coordenador técnico / Admin cria, modifica ou remove um utilizador com o respetivo perfil;

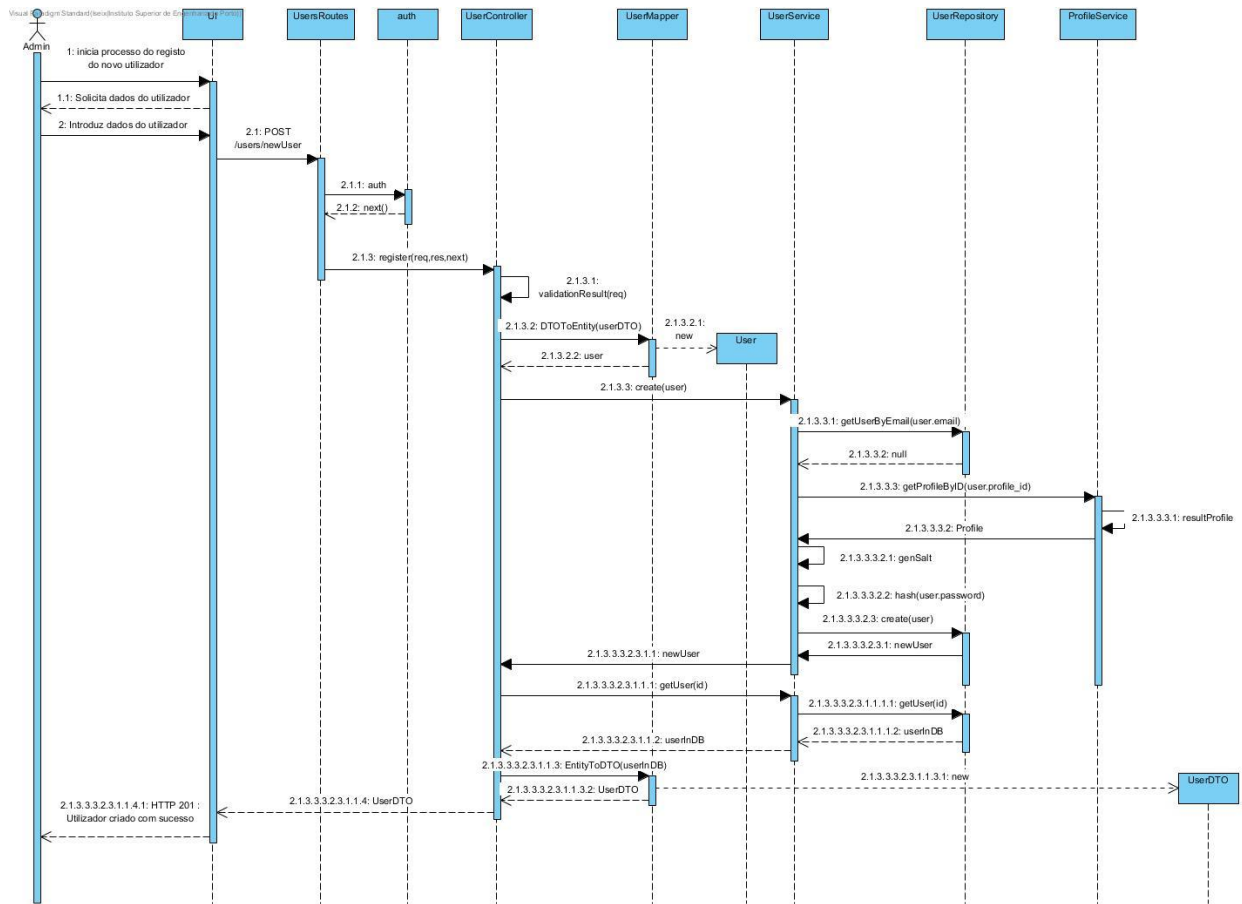
##### Pós-condições:

- Utilizador é criado no sistema com palavra-passe encriptada.

##### Validações:

- Não podem existir dois utilizadores com o mesmo e-mail no sistema.

# Diagrama de Sequência



## REQ-02: Autenticação de utilizadores

### **Descrição**

Qualquer dos stakeholders que pretendam aceder ao sistema tem de se encontrar registados no mesmo. Desta forma não só é garantido o acesso apenas a pessoal autorizado como também permite fazer a identificação do tipo de perfil que esta a aceder ao sistema e dessa forma apresentar apenas as funcionalidades que lhe competem.

### **Simulação/ sequência de Respostas**

#### Pré-condições:

- O utilizador tem de estar registado no sistema (o registo é realizado pelo informático aquando da configuração do sistema);

#### Fluxo de Eventos:

1. O utilizador introduz o seu e-mail de utilizador;
2. O utilizador introduz a palavra-passe;
3. Após verificação do registo no sistema e autenticado o utilizador é redirecionado para a visualização competente das suas funções;

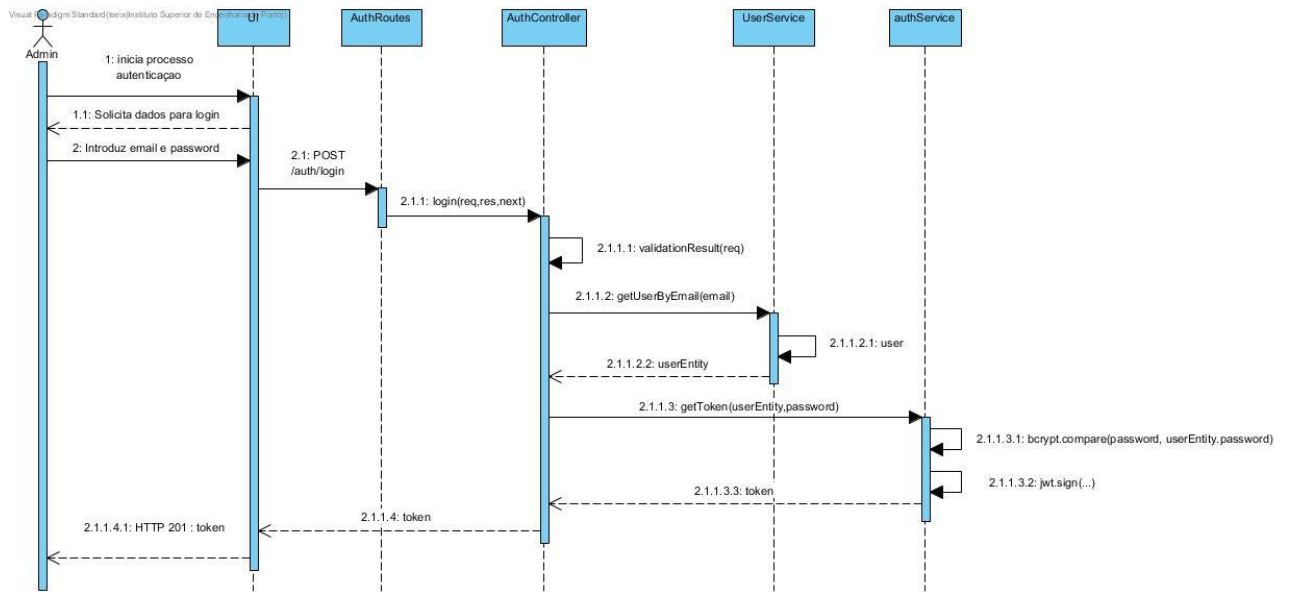
#### Pós-condições:

- O utilizador permanece com uma sessão aberta durante um período fixo de tempo ou até decidir encerrar a sessão.

#### Validações:

- Os campos do nome de utilizador e de palavra-passe não podem estar vazios quando o utilizador fizer o login;
- Apenas utilizadores registados no sistema podem aceder ao mesmo (verificação de registo);

# Diagrama de Sequência



## REQ-03: Gerir salas

### **Descrição**

O coordenador técnico ou os técnicos de laboratório acedem ao sistema e introduzem os dados necessários, nomeadamente o nome, andar e tipo de sala. Estes dados são registados no sistema.

### **Simulação/ sequência de Respostas**

#### Pré-condições:

- O coordenador técnico ou os técnicos de laboratório tem de ter uma conta criada no sistema com as devidas permissões para gerir salas.

#### Fluxo de Eventos:

1. O coordenador técnico ou o técnico de laboratório inicia sessão no sistema;
2. coordenador técnico ou o técnico de laboratório acede ao separador de gestão de salas;
3. O coordenador técnico ou o técnico de laboratório cria ou modifica uma sala;

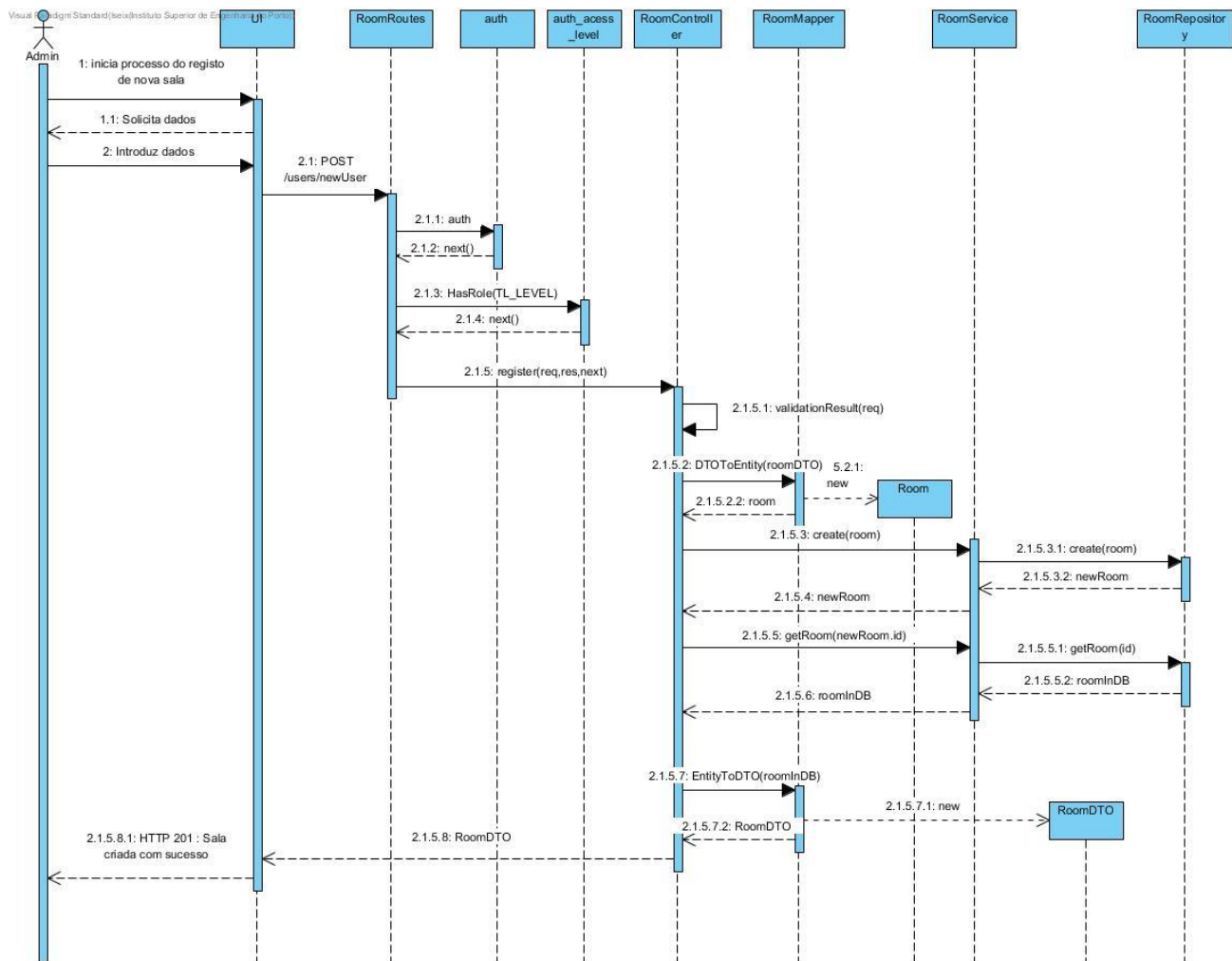
#### Pós-condições:

- Sala é criada no sistema.

#### Validações:

- Não podem existir duas salas com o mesmo identificador (id gerado pelo sistema).

# Diagrama de Sequência



## REQ-04: Visualização de informação de utilização dos equipamentos

### **Descrição**

O coordenador técnico acede ao sistema e consulta as estatísticas referentes à utilização dos equipamentos (ranking de equipamentos mais utilizados, número de equipamentos perdidos ou estragados e que necessitam de substituição).

### **Simulação/ sequência de Respostas**

#### Pré-condições:

- O coordenador técnico tem permissões para aceder a estas informações do sistema.

#### Fluxo de Eventos:

1. O coordenador técnico inicia sessão no sistema;
2. O coordenador técnico acede à página de consulta de informação da utilização dos equipamentos;
3. O coordenador técnico consulta as informações de que necessita;

#### Pós-condições:

- Sem nada a registar.

#### Validações:

- Sem nada a registar.



## REQ-05: Visualização informação de requisição

### **Descrição**

Apenas o coordenador técnico pode visualizar informação detalhada de uma requisição. Este tipo de funcionalidade estará associado ao perfil de utilizador.

### **Simulação/ sequência de Respostas**

#### Pré-condições:

- O utilizador tem de estar registado no sistema e associado a um determinado perfil (correspondente as suas funções neste caso de coordenador técnico).
- Tem de decorrer pelo menos uma requisição para poder ver a sua informação (utilizador que requisita o equipamento).

#### Fluxo de Eventos:

1. Utilizador com perfil de coordenador técnico inicia sessão no sistema.
2. O utilizador quer visualizar a informação de requisição.
3. Validação do perfil do utilizador para realização da operação.
4. Informação é obtida da base dados e apresentada ao utilizador.

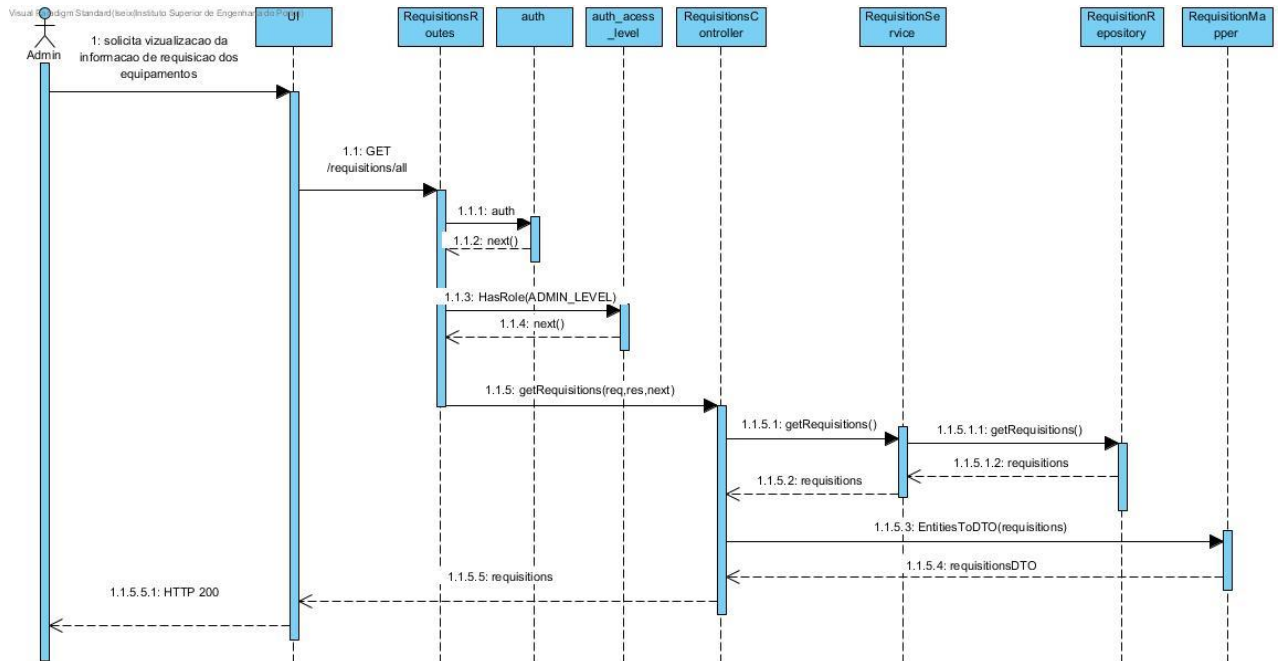
#### Pós-condições:

- O utilizador permanece com uma sessão aberta durante um período fixo de tempo ou até decidir encerrar a sessão (sempre associado ao seu perfil)
- O utilizador consegue obter as informações de quem requisita um equipamento.

#### Validações:

- Após validação de autenticação no sistema, e verificado qual o perfil do utilizador, é possível obter os dados de quem requisita o equipamento.

# Diagrama de Sequência



## REQ-06: Gerir contentores de equipamentos

### **Descrição**

O coordenador técnico ou os técnicos de laboratório acedem ao sistema e introduzem os dados necessários, nomeadamente o nome, a sala em que se encontra o contentor e caso necessário alguma observação que achem pertinente. Os contentores também podem representar armários ou estojos que podem ser contidos noutros contentores.

### **Simulação/ sequência de Respostas**

#### Pré-condições:

- O coordenador técnico ou os técnicos de laboratório tem de ter uma conta criada no sistema com as devidas permissões para gerir contentores de equipamentos.

#### Fluxo de Eventos:

1. O coordenador técnico ou o técnico de laboratório inicia sessão no sistema;
2. O coordenador técnico ou o técnico de laboratório acede ao separador de gestão de armários/contentores de equipamentos;
3. O coordenador técnico ou o técnico de laboratório cria ou modifica um contentor;

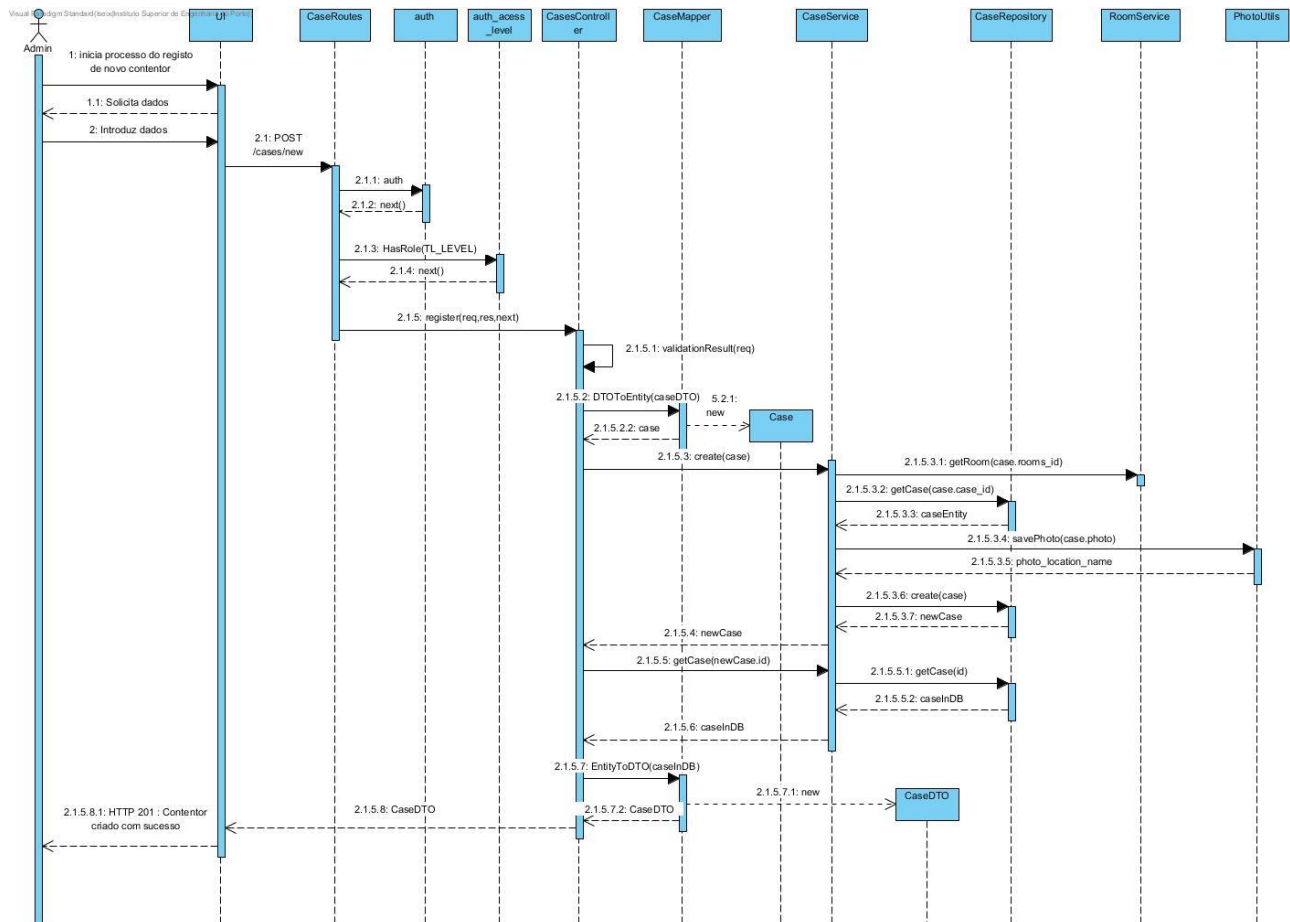
#### Pós-condições:

- Contentor é criado no sistema.

#### Validações:

- Não podem existir dois contentores com o mesmo identificador (id gerado pelo sistema).

# Diagrama de Sequência



## REQ-07: Gerar QR code para contentores

### **Descrição**

O coordenador técnico ou os técnicos de laboratório acedem ao sistema e pedem para gerar um QR code para os contentores. Podem posteriormente fazer download do QR code gerado para imprimir e colar junto ao contentor.

### **Simulação/ sequência de Respostas**

#### Pré-condições:

- O coordenador técnico ou os técnicos de laboratório tem de ter uma conta criada no sistema com as devidas permissões para gerir salas.
- Tem de estar registado no sistema um contentor para gerar o QR code para ele.

#### Fluxo de Eventos:

1. O coordenador técnico ou o técnico de laboratório inicia sessão no sistema;
2. coordenador técnico ou o técnico de laboratório acede ao separador de gestão de contentores;
3. O coordenador técnico ou o técnico de laboratório pedem para gerar um QR code para um dado contentor;

#### Pós-condições:

- O QR code é gerado e apresentado ao utilizador.

#### Validações:

- Sem nada a registar.

## REQ-08: Gerir equipamentos

### **Descrição**

O coordenador técnico ou os técnicos de laboratório acedem ao sistema e introduzem os dados necessários, nomeadamente o nome, o contentor em que se encontra o equipamento, caso necessário alguma observação que achem pertinente, boas práticas da utilização do equipamento, características físicas, outras características, fotografias do equipamento, estado de desgaste, a referência interna e tipo de ferramenta (principal ou acessório). Para além disso, o coordenador técnico ou os técnicos de laboratório podem definir que acessórios podem ser utilizados com esse equipamento (consoante a granularidade que quiserem definir).

### **Simulação/ sequência de Respostas**

#### Pré-condições:

- O coordenador técnico ou os técnicos de laboratório tem de ter uma conta criada no sistema com as devidas permissões para gerir equipamentos.

#### Fluxo de Eventos:

1. O coordenador técnico ou o técnico de laboratório inicia sessão no sistema;
2. O coordenador técnico ou o técnico de laboratório acede ao separador de gestão de equipamentos;
3. O coordenador técnico ou o técnico de laboratório cria ou modifica um equipamento;

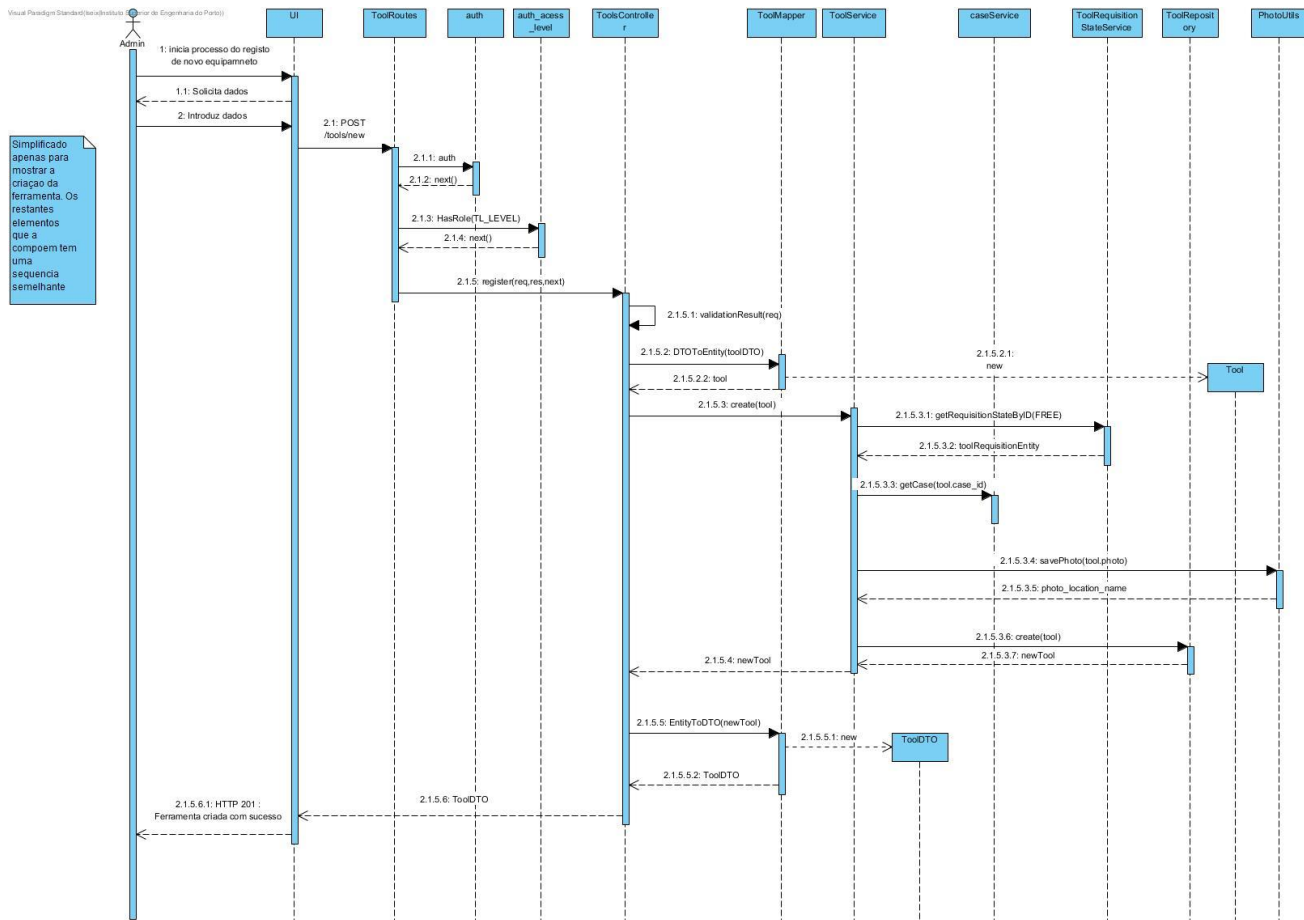
#### Pós-condições:

- Equipamento é criado no sistema.

#### Validações:

- Não podem existir dois equipamentos com o mesmo identificador (id gerado pelo sistema) e referência interna (introduzida pelo utilizador).

# Diagrama de Sequência



## REQ-09: Alteração do estado da ferramenta

### **Descrição**

O coordenador técnico ou os técnicos de laboratório ou os clientes acedem ao sistema e podem alterar o estado de um equipamento (perdido, estragado, novo).

### **Simulação/ sequência de Respostas**

#### Pré-condições:

- O coordenador técnico ou os técnicos de laboratório ou os clientes tem de ter uma conta criada no sistema com as devidas permissões para alterar o estado do equipamento.
- É necessário que o equipamento já se encontre registado no sistema.

#### Fluxo de Eventos:

1. O coordenador técnico ou o técnico de laboratório ou os clientes iniciam sessão no sistema;
2. O coordenador técnico ou o técnico de laboratório ou os clientes acedem ao separador de equipamentos;
3. O coordenador técnico ou o técnico de laboratório ou os clientes modifica o estado de um equipamento;

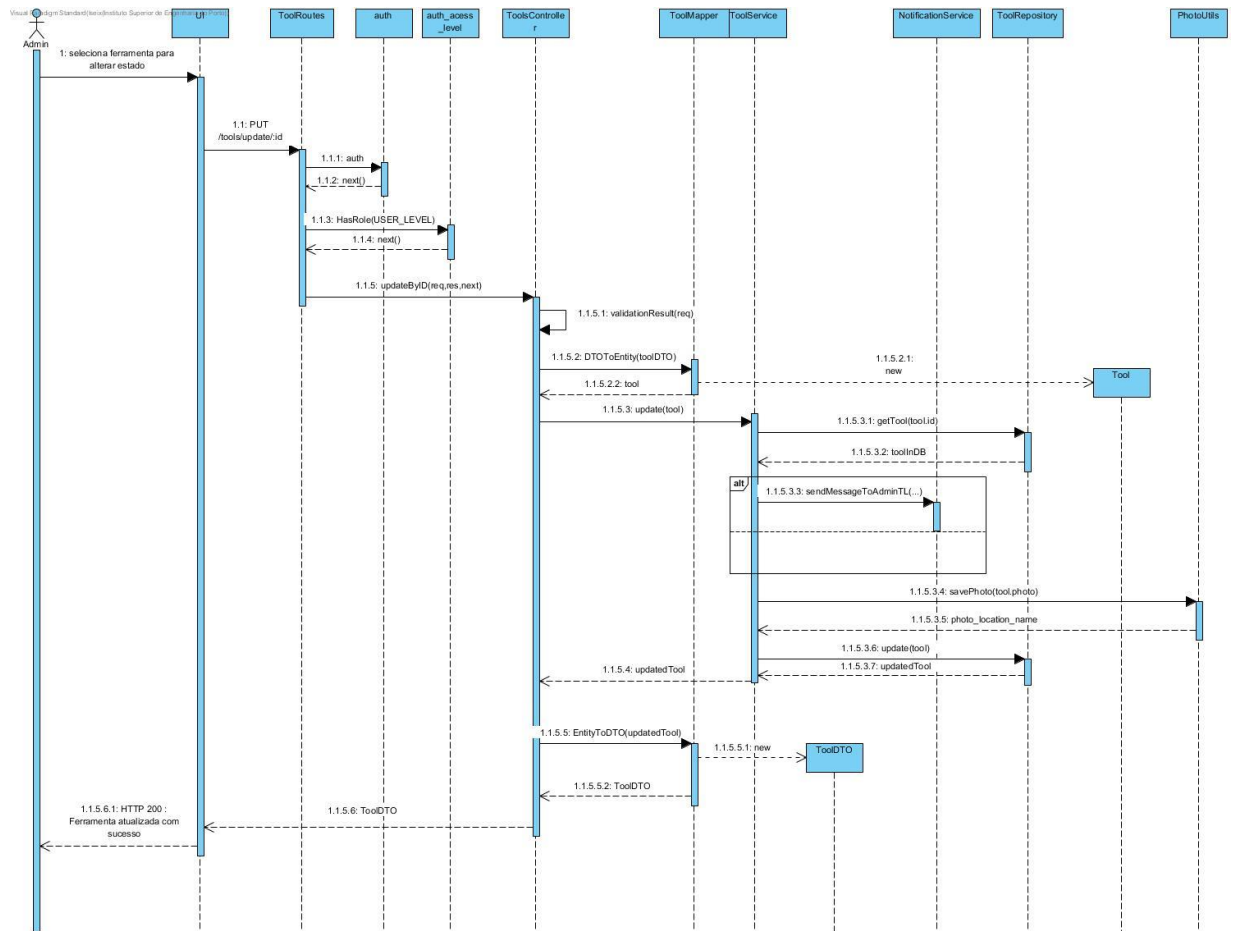
#### Pós-condições:

- O estado de um equipamento é alterado no sistema.
- Coordenador técnico ou técnicos de laboratório recebem uma notificação da alteração do estado de um equipamento se for perdido ou estragado.

#### Validações:

- Sem nada a registar.

# Diagrama de Sequência



## REQ-10: Ler QR code para filtrar equipamentos dentro de um contentor

### **Descrição**

O coordenador técnico ou os técnicos de laboratório ou os clientes acedem ao sistema e recorrendo à camara do dispositivo móvel fazem um scan do QR code afixado num contentor para obter a lista de equipamentos nele contidos (mostrando se estão em estado requisitado ou não).

### **Simulação/ sequencia de Respostas**

#### Pré-condições:

- O coordenador técnico ou os técnicos de laboratório ou os clientes têm de ter uma conta criada no sistema com as devidas permissões para ler o QR code afixado no contentor.
- É necessário ter um QR code gerado e válido afixado ao contentor do equipamento.

#### Fluxo de Eventos:

1. O coordenador técnico ou o técnico de laboratório ou os clientes iniciam sessão no sistema;
2. O coordenador técnico ou o técnico de laboratório ou os clientes acedem ao separador de ler QR code;
3. O coordenador técnico ou o técnico de laboratório ou os clientes obtêm a lista de equipamentos guardados dentro desse contentor.

#### Pós-condições:

- É apresentada uma lista com os equipamentos contidos nesse contentor.

#### Validações:

- Os dados do QR code tem de estar em conformidade com os dados necessários obter para posteriormente procurar equipamentos disponíveis nesse contentor.

## REQ-11: Requisitar equipamentos durante um determinado período de tempo

### **Descrição**

O coordenador técnico ou os técnicos de laboratório ou os clientes acedem ao sistema e selecionam um equipamento para requisitar introduzindo a data de início de requisição, a perspectiva de fim da requisição, se necessário a localização prevista do equipamento (dentro ou fora da organização) e os respetivos acessórios.

### **Simulação/ sequência de Respostas**

#### Pré-condições:

- O coordenador técnico ou os técnicos de laboratório ou os clientes têm de ter uma conta criada no sistema com as devidas permissões para requisitar equipamentos.
- É necessário ter pelo menos um equipamento registado no sistema e não estar em estado de requisitado.

#### Fluxo de Eventos:

1. O coordenador técnico ou o técnico de laboratório ou os clientes iniciam sessão no sistema;
2. O coordenador técnico ou o técnico de laboratório ou os clientes acedem ao separador de equipamentos;
3. O coordenador técnico ou o técnico de laboratório ou os clientes obtêm a lista de equipamentos e o selecionam um em estado não requisitado para poder requisitar o equipamento.
4. O coordenador técnico ou o técnico de laboratório ou os clientes introduzem os dados e selecionam os acessórios necessários.

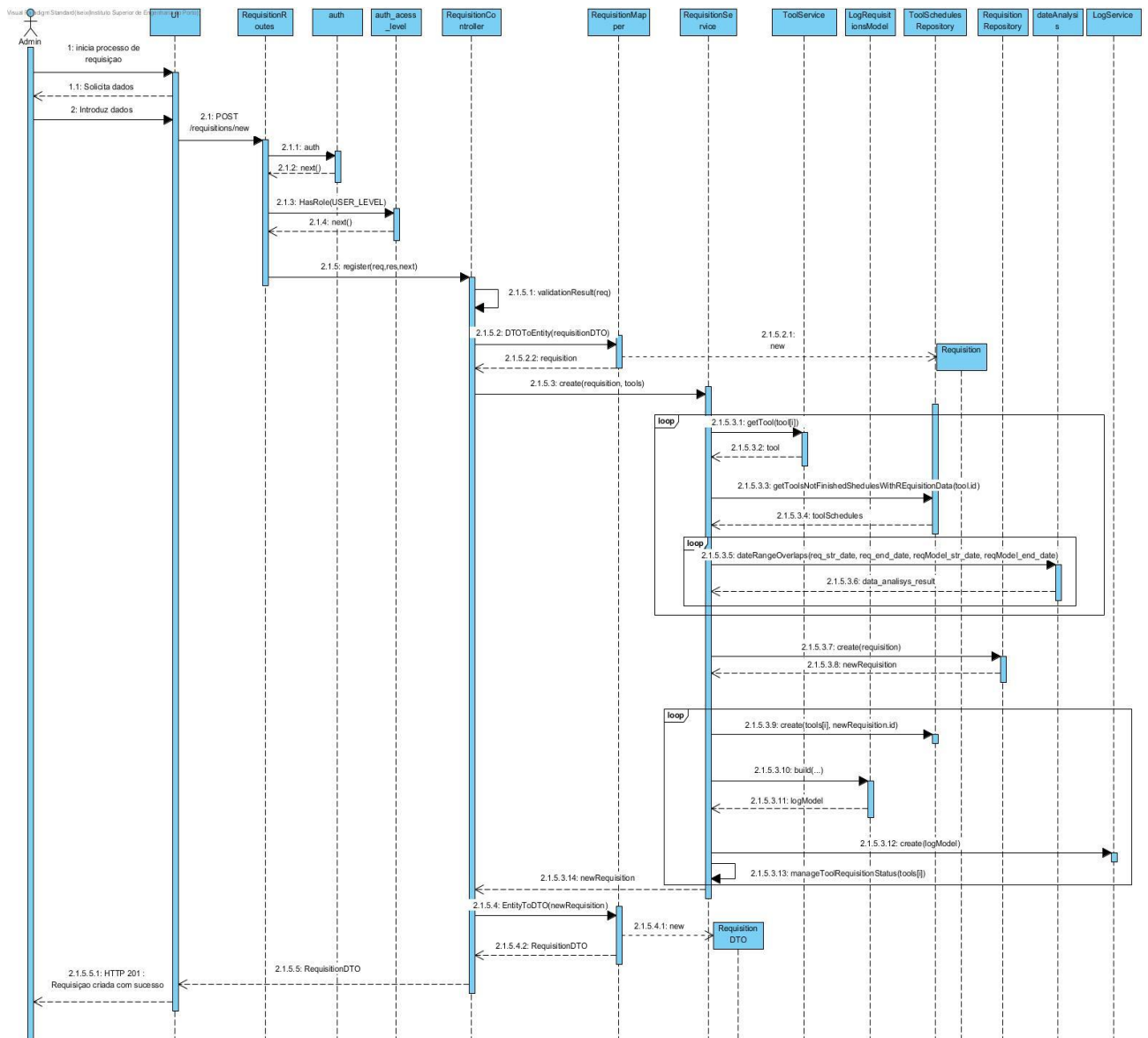
#### Pós-condições:

- O estado do equipamento e respetivos acessórios passam para requisitado.
- Caso haja sobreposição de requisições o sistema pode sugerir a data mais próxima válida (e com a mesma duração temporal inicialmente introduzida) para fazer a requisição.

Validações:

- Datas de requisição (início e perspectiva de fim) tem de ser posteriores a atual.
- Apenas podem ser requisitados equipamentos em estado não requisitado.
- Verificar sobreposições de datas de requisição de equipamentos.

**Diagrama de Sequência**



## REQ-12: Ver notificações

### **Descrição**

O coordenador técnico ou os técnicos de laboratório ou os clientes acedem ao sistema e tem um aviso que tem notificações por ler. Estas podem ser relativas a alteração do estado de desgaste de um equipamento (que precisa ser substituído) ou avisos do sistema para devolver equipamento (quando ultrapassada a data de requisição) ou prolongar a sua utilização.

### **Simulação/ sequência de Respostas**

#### Pré-condições:

- O coordenador técnico ou os técnicos de laboratório ou os clientes têm de ter uma conta criada no sistema para visualizar notificações.

#### Fluxo de Eventos:

1. O coordenador técnico ou o técnico de laboratório ou os clientes iniciam sessão no sistema;
2. O coordenador técnico ou o técnico de laboratório ou os clientes são notificados;
3. O coordenador técnico ou o técnico de laboratório ou os clientes acedem ao separador de notificações para consultar as suas notificações.
4. O coordenador técnico ou o técnico de laboratório ou os clientes podem eliminar a notificação.

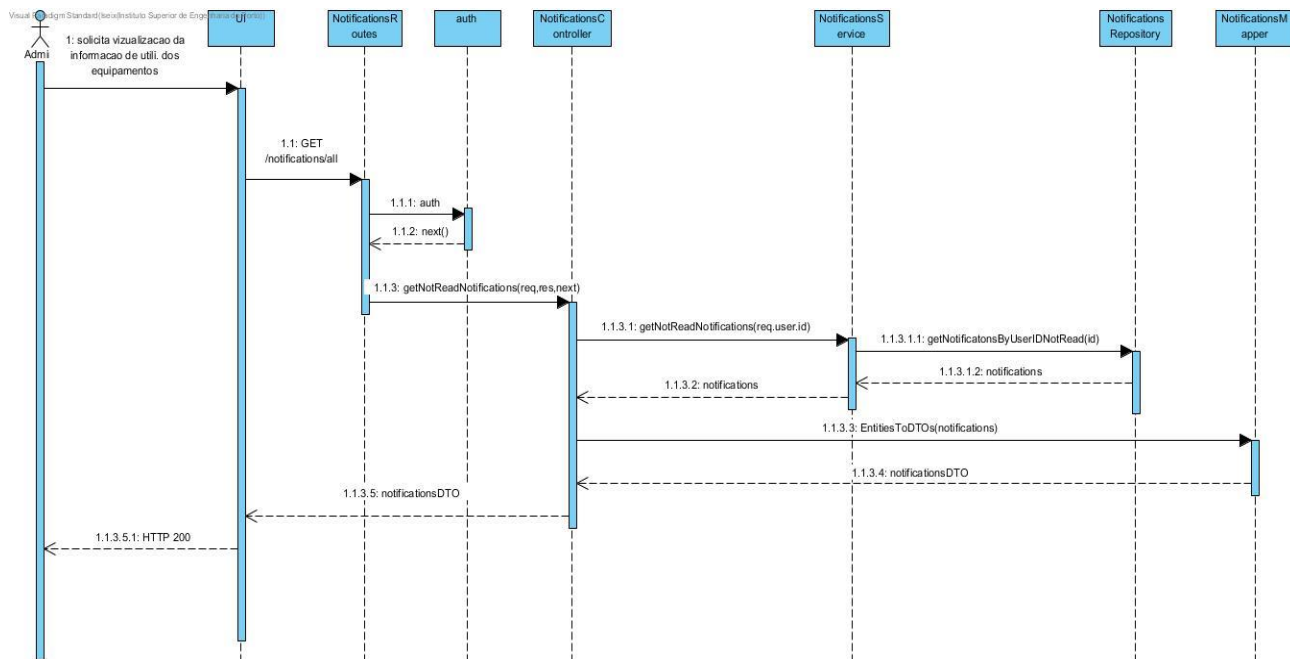
#### Pós-condições:

- É apresentada uma lista de notificações pendentes.
- Consoante a notificação, o tempo de requisição do equipamento pode ser prolongado ou o estado do equipamento alterado.
- A notificação pode ser eliminada.

#### Validações:

- Verificar sobreposição de datas de requisição de equipamentos.

# Diagrama de Sequência



## REQ-13: Prolongar requisição do equipamento

### **Descrição**

O coordenador técnico ou os técnicos de laboratório ou os clientes acedem ao sistema e querem prolongar a data de fim de requisição do equipamento.

### **Simulação/ sequência de Respostas**

#### Pré-condições:

- O coordenador técnico ou os técnicos de laboratório ou os clientes têm de ter uma conta criada no sistema para visualizar notificações.
- O coordenador técnico ou os técnicos de laboratório ou os clientes tem de ter um equipamento requisitado.

#### Fluxo de Eventos:

1. O coordenador técnico ou o técnico de laboratório ou os clientes iniciam sessão no sistema;
2. O coordenador técnico ou o técnico de laboratório ou os clientes acedem ao separador de equipamentos requisitados por ele;
3. O coordenador técnico ou o técnico de laboratório ou os clientes selecionam um equipamento para prolongar a sua utilização.

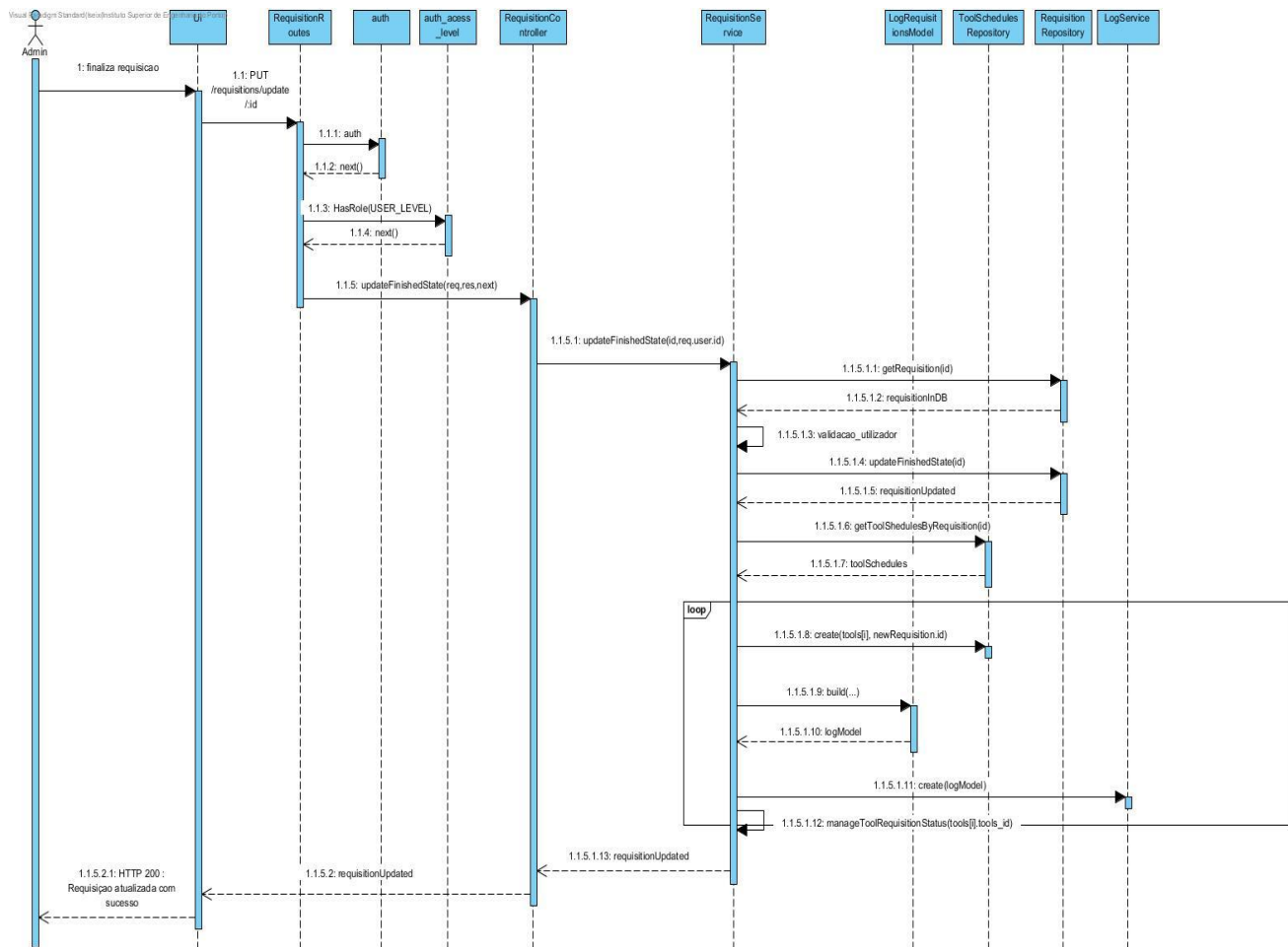
#### Pós-condições:

- A data de devolução da requisição do equipamento é alterada.

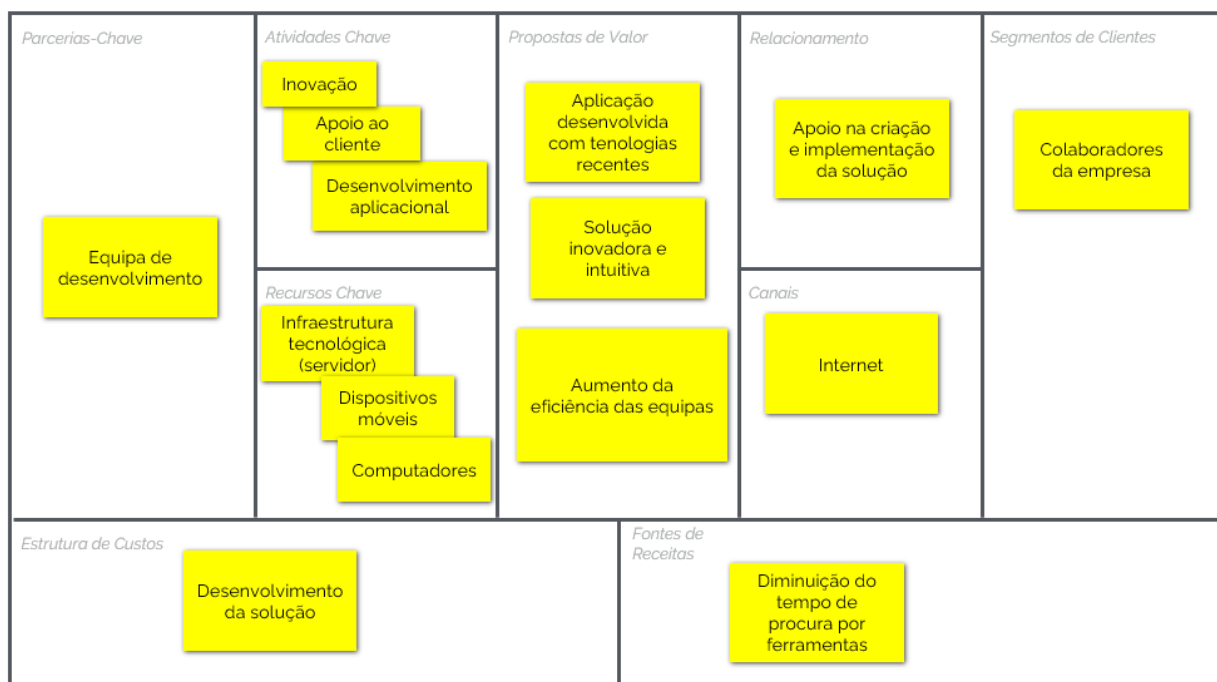
#### Validações:

- Verificar sobreposição de datas de requisição de equipamentos.

# Diagrama de Sequência



## Business model of Canvas



# QEF – Qualitative Evaluation Framework

q	D	Q <sub>i</sub>	Dimensão	Q <sub>j</sub>	W <sub>ij</sub> (Peso do fator j na dimensão i) [0,1]	Factor	r <sub>wk</sub> (peso do requisito k no Factor j) {2, 4, 6, 8, 10}	Requisito	wf <sub>k</sub> % do cumprimento do requisito k [0,100]		
94%	0.14	95.26	Usabilidade	100	0.45	Requisitos funcionais	10	REQ-01: Gerir utilizadores	100		
							10	REQ-02: Autenticação de utilizadores	100		
							10	REQ-03: Gerir salas	100		
							8	REQ-04: Visualização de informação de utilização dos equipamentos	100		
							10	REQ-05: Visualização da informação de requisição	100		
							10	REQ-06: Gerir contentores de equipamentos	100		
							4	REQ-07: Gerar QR code para contentores	100		
							10	REQ-08: Gerir equipamentos	100		
							8	REQ-09: Alteração do estado da ferramenta	100		
							8	REQ-10: Ler QR code para filtrar equipamentos dentro de um contentor	100		
							10	REQ-11: Requisitar equipamentos durante um determinado período de tempo	100		
							10	REQ-12: Ver notificações	100		
							8	REQ-13: Prolongar requisição do equipamento	100		
							10	FIU01 - A aplicação é intuitiva	100		
							8	FIU02 - A experiência de design é semelhante em toda a aplicação	100		
		8	FIU03 - A interação com a interface é semelhante em toda a aplicação	100							
		6	FIU04 - A experiência de navegação é igual em toda a aplicação	100							
		10	FIU05 - Permissões e opções específicas disponíveis consoante perfil logado	100							
		10	FIU06 - As principais funções são rapidamente acedidas	100							
		10	FIU07 - A interface consegue adaptar-se a diferentes dimensões de ecrã	100							
		2	FIU08 - A aplicação suporta o modo offline	0							
		10	FIU09 - Existem controlos de pesquisa de informação	50							
		8	FIU10 - A aplicação permite capturar fotografias	50							
		6	FIU11 - As informações estatísticas dos equipamentos estão bem organizadas e apresentam informação relevante	100							
		10	FCD01 - A informação sobre o equipamento está bem organizada	100							
		10	FCD02 - Os textos são lexicamente corretos	100							
		10	FCD03 - As mensagens são fáceis de entender	100							
		10	FCD04 - Todo o conteúdo está bem relacionado	100							
		10	FCD05 - A aplicação notifica o utilizador sobre ações necessárias ou atualizações ocorridas	100							
		90		100	Conteúdo digital	100	0.17		10	FCD01 - A informação sobre o equipamento está bem organizada	100
									10	FCD02 - Os textos são lexicamente corretos	100
									10	FCD03 - As mensagens são fáceis de entender	100
									10	FCD04 - Todo o conteúdo está bem relacionado	100
									10	FCD05 - A aplicação notifica o utilizador sobre ações necessárias ou atualizações ocorridas	100
				100	Manutenibilidade	100	0.60	Testabilidade	8	MT01 - Foram desenvolvidos testes para garantir a qualidade do sistema	100
6	MT02 - A aplicação tem uma boa cobertura de testes			100							
10	MT03 - A aplicação passa com sucesso na maioria dos testes			100							
75	Manutenção			75	0.40		8	MM01 - A aplicação segue diretrizes e padrões de design de software	100		
8							MM02 - A arquitetura da aplicação permite acrescentar com facilidade novas funcionalidades	50			
91.23		100	Desempenho	100	0.17		8	DQV01 - As imagens são nítidas e têm boa qualidade	100		
							10	DS01 - As palavras chave estão encriptadas na base de dados	100		
		88.842		88.842	0.67	Navegação	10	DN01 - Produto tem uma boa estrutura e permite aos utilizadores aceder aos conteúdos e às principais funções de forma intuitiva	100		
							8	DN02 - A interface de utilizador é rápida na resposta havendo pouca latência nas respostas com servidor	100		
							10	DN03 - O tempo de execução da aplicação não possui erros, e erros inesperados devem ser tratados	50		
							10	DN04 - Segurança de login e acesso	100		

Dimensão	Usabilidade			
Factor	Requisitos funcionais			
Requisito	Métrica de avaliação	Wfk - Cumprimento (%)		
		0	50	100
REQ-01: Gerir utilizadores	Esta função inclui criar novos utilizadores, remover e atualizar a informação bem como fazer a associação de perfil	Funcionalidade não desenvolvida	Parcialmente implementada	Funcionalidade totalmente implementada
REQ-02: Autenticação de utilizadores	Qualquer dos stakeholders que pretendam aceder ao sistema tem de se encontrar registados no mesmo	Funcionalidade não desenvolvida	-	Funcionalidade totalmente implementada
REQ-03: Gerir salas	Registar e atualizar a informação das salas	Funcionalidade não desenvolvida	Parcialmente implementada	Funcionalidade totalmente implementada
REQ-04: Visualização de informação de utilização dos equipamentos	Consultar estatísticas referentes à utilização dos equipamentos (ranking de equipamentos mais utilizados, número de equipamentos perdidos ou estragados e que necessitam de substituição)	Nenhuma informação estatística implementada	Apenas pode ver algumas informações estatísticas	Pode ver todas as informações estatísticas
REQ-05: Visualização da informação de requisição	Apenas o coordenador técnico pode visualizar informação detalhada de uma requisição	Funcionalidade não desenvolvida	Parcialmente implementada	Funcionalidade totalmente implementada
REQ-06: Gerir contentores de equipamentos	Registar e atualizar a informação dos contentores	Funcionalidade não desenvolvida	Parcialmente implementada	Funcionalidade totalmente implementada
REQ-07: Gerar QR code para contentores	Gerar um QR code para os contentores. Podem posteriormente fazer download do QR code gerado para imprimir e colar junto ao contentor	Funcionalidade não desenvolvida	Gera QR code mas não permite download	Gera QR code e permite fazer download
REQ-08: Gerir equipamentos	Criar, atualizar equipamento e definir acessórios	Funcionalidade não desenvolvida	Consegue criar equipamento mas não permite adicionar fotografias ou associar acessórios	Funcionalidade totalmente implementada
REQ-09: Alteração do estado da ferramenta	Qualquer utilizador pode alterar o estado de um equipamento (perdido, estragado, novo)	Funcionalidade não desenvolvida	-	Funcionalidade totalmente implementada
REQ-10: Ler QR code para filtrar equipamentos dentro de um contentor	Um utilizador acede ao sistema e recorrendo à câmara do dispositivo móvel faz um scan do QR code aliado num contentor para obter a lista de equipamentos nele contidos	Funcionalidade não desenvolvida	-	Funcionalidade totalmente implementada
REQ-11: Requisitar equipamentos durante um determinado período de tempo	O utilizador acede ao sistema e seleciona um equipamento para requisitar introduzindo os dados necessários	Funcionalidade não desenvolvida	Não são implementadas validações das datas de requisição	Funcionalidade totalmente implementada
REQ-12: Ver notificações	O utilizador acede ao sistema e tem um aviso que tem notificações por ler	Funcionalidade não desenvolvida	Recebe notificação mas não permite acesso rápido a funcionalidades	Funcionalidade totalmente implementada
REQ-13: Prolongar requisição do equipamento	O utilizador acede ao sistema e quer prolongar a data de fim de requisição do equipamento	Funcionalidade não desenvolvida	Não são implementadas validações das datas de requisição	Funcionalidade totalmente implementada

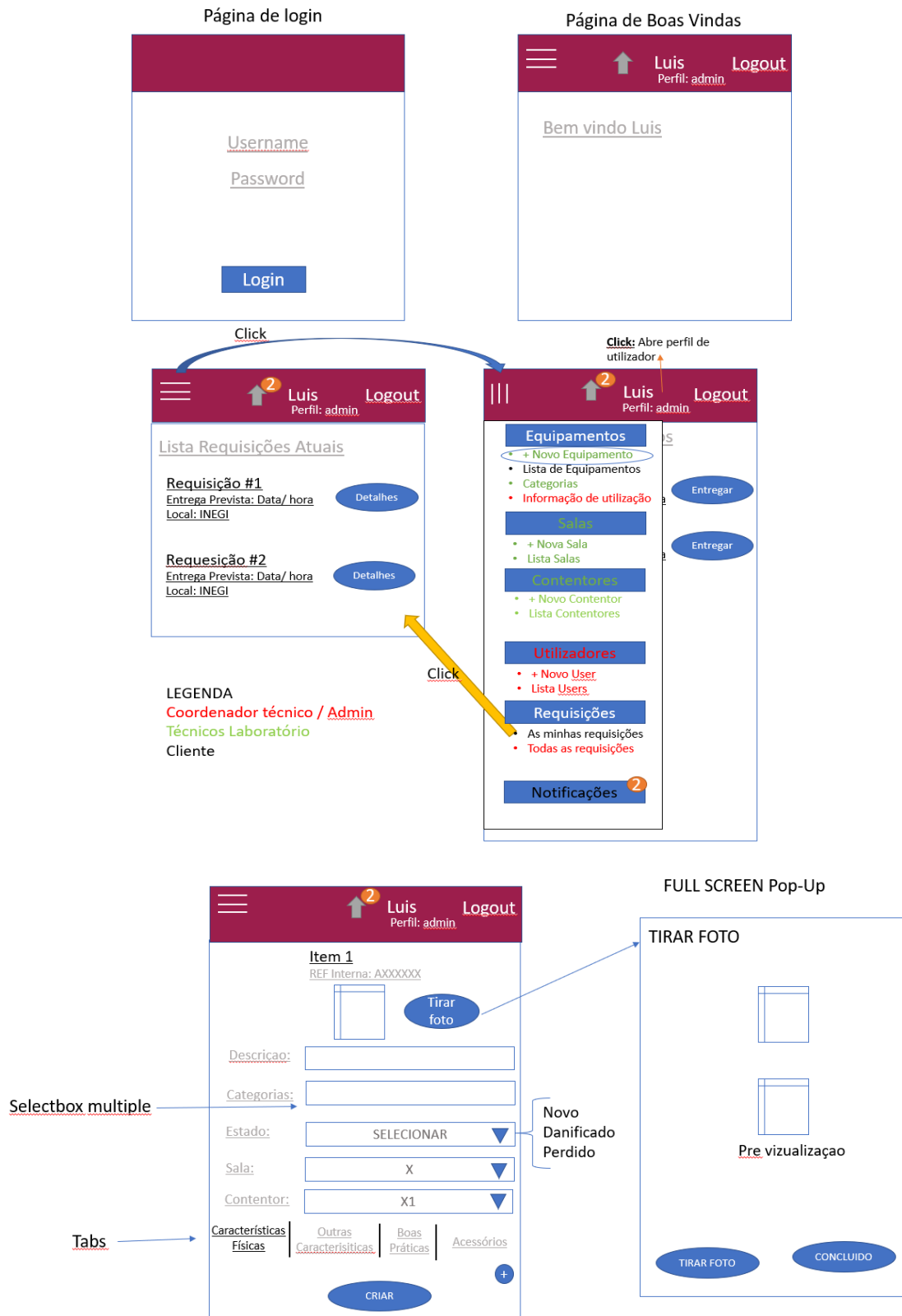
Dimensão	Usabilidade			
Factor	Interface do utilizador			
Requisito	Métrica de avaliação	Wfk - Cumprimento (%)		
		0	50	100
FIU01 - A aplicação é intuitiva	Utilizador sem qualquer formação sobre o sistema consegue interagir facilmente com este e aceder as funcionalidades que pretende	menos de 60% dos questionários com mais de 4 ou 5 pontos	entre 61% a 84% dos questionários com mais de 4 ou 5 pontos	mais de 85% dos questionários com mais de 4 ou 5 pontos
FIU02 - A experiência de design é semelhante em toda a aplicação	A experiência de design deve ser semelhante em toda a aplicação. Cores, design de botões, tipos de fonte, logótipos e ícones devem ser semelhantes.	menos de 60% dos questionários com mais de 4 em 5 pontos	mais de 60% dos questionários com mais de 4 em 5 pontos	mais de 85% dos questionários com mais de 4 ou 5 pontos
FIU03 - A interação com a interface é semelhante em toda a aplicação	Todos os ecrãs têm atalhos para aceder aos outros menus	menos de 60% dos questionários com mais de 4 ou 5 pontos	entre 61% a 84% dos questionários com mais de 4 ou 5 pontos	mais de 85% dos questionários com mais de 4 ou 5 pontos
FIU04 - A experiência de navegação é igual em toda a aplicação	A navegação entre telas apresenta a mesma experiência. Botões para menu, retroceder, confirmar e cancelar sempre nas mesmas posições	menos de 60% dos questionários com mais de 4 ou 5 pontos	entre 61% a 84% dos questionários com mais de 4 ou 5 pontos	mais de 85% dos questionários com mais de 4 ou 5 pontos
FIU05 - Permissões e opções específicas disponíveis consoante perfil logado	As funcionalidades devem estar disponíveis apenas para os utilizadores com perfil apropriado.	Uma ou mais funcionalidades disponíveis para tipos de utilizadores sem esta permissão	-	Funcionalidades disponíveis apenas para determinados tipos de utilizadores
FIU06 - As principais funções são rapidamente acessadas	A execução de uma funcionalidade não deve levar mais de 3 ações (cliques, seleções...)	menos de 60% dos questionários com resposta positiva	entre 61% a 84% dos questionários com mais de 4 ou 5 pontos	mais de 85% dos questionários com mais de 4 ou 5 pontos
FIU07 - A interface consegue adaptar-se a diferentes dimensões de ecrã	Consoante o dispositivo em que o utilizador acede, as posições dos elementos das páginas adaptam-se	menos de 60% dos questionários com resposta positiva	entre 61% a 84% dos questionários com mais de 4 ou 5 pontos	mais de 85% dos questionários com resposta positiva
FIU08 - A aplicação suporta o modo offline	Algumas funcionalidades ficam disponíveis quando não há rede (por exemplo consulta de equipamentos e outras funcionalidades de consulta)	Não implementado	Consulta parcial da informação	Consulta total da informação
FIU09 - Existem controlos de pesquisa de informação	A aplicação apresenta um mecanismo de pesquisa. Os filtros de pesquisa de informação são intuitivos e de fácil utilização	menos de 60% dos questionários com mais de 4 ou 5 pontos	entre 61% a 84% dos questionários com mais de 4 ou 5 pontos	mais de 85% dos questionários com mais de 4 ou 5 pontos
FIU10 - A aplicação permite capturar fotografias	A aplicação tem acesso a câmara do dispositivo e permite tirar fotografias e guardar no sistema e associar à ferramenta	Não	-	Sim
FIU11 - As informações estatísticas dos equipamentos estão bem organizadas e apresentam informação relevantes	As estatísticas devem ser apresentadas numa tabela ou gráficos com informação relevante para tirar conclusões e definir ações	Funcionalidade não desenvolvida	Apenas alguns parâmetros estatísticos apresentados	Funcionalidade totalmente implementada

Dimensão	Usabilidade			
Factor	Conteúdo digital			
Requisito	Métrica de avaliação	Wfk - Cumprimento (%)		
		0	50	100
FC001 - A informação sobre o equipamento está bem organizada	O equipamento tem toda a sua informação estrutura e facilmente acessível para a consulta (por exemplo na página do equipamento)	menos de 60% dos questionários com mais de 4 ou 5 pontos	entre 61% a 84% dos questionários com mais de 4 ou 5 pontos	mais de 85% dos questionários com mais de 4 ou 5 pontos
FC002 - Os textos são lexicalmente corretos	As frases são curtas e cumprem os princípios gramaticais elementares.	menos de 60% dos questionários com resposta positiva	-	mais de 61% dos questionários com resposta positiva
FC003 - As mensagens são fáceis de entender	As mensagens não apresentam códigos ilegíveis ou símbolos indecifráveis	menos de 50% dos questionários com resposta "sem mensagens ilegíveis"	entre 50% a 90% dos questionários com resposta "sem mensagens ilegíveis"	mais de 91% dos questionários com resposta "sem mensagens ilegíveis"
FC004 - Todo o conteúdo está bem relacionado	O equipamento está associado a um contentor que por sua vez esta guardado numa sala. Todo o conteúdo não deve apresentar dados fictícios ou irrelevantes	Não	-	Sim
FC005 - A aplicação notifica o utilizador sobre ações necessárias ou atualizações ocorridas	Notificações são apresentadas aos utilizadores sobre qualquer ação relevante realizada.	menos de 60% dos questionários com resposta positiva	-	mais de 61% dos questionários com resposta positiva

Dimensão	Manutenibilidade			
Factor	Testabilidade, Manutenção			
Requisito	Métrica de avaliação	Wfk - Cumprimento (%)		
		0	50	100
MT01 - Foram desenvolvidos testes para garantir a qualidade do sistema	Foram desenvolvidos diferentes tipos de testes para garantir a boa qualidade do software (por exemplo testes unitários, testes de integração, testes end2end, testes manuais de aceitação)	Não	Pelo menos 1 ou 2 tipos de testes	Mais do que 2 tipos de testes
MT02 - A aplicação tem uma boa cobertura de testes	Uma cobertura de código de 70-80% é uma meta razoável para a maioria dos sistemas. Uma cobertura mínima de código para teste unitários pode ser entre 10-20%	0% cobertura	10% a 40% cobertura	40 a 80% cobertura
MT03 - A aplicação passa com sucesso na maioria dos testes	Os testes a que a aplicação é submetida cumpre com os thresholds definidos	0 testes passados	Entre 40% a 60% dos testes passados	Mais de 90% dos testes passados
MM01 - A aplicação segue diretrizes e padrões de design de software	O código implementa as diretrizes e os padrões de design apresentados no documento de tese de mestrado	Não	-	Sim
MM02 - A arquitetura da aplicação permite acrescentar com facilidade novas funcionalidades	As funcionalidades do sistema podem ser entendidas com relativa facilidade (como por exemplo adição e facilidade de comunicação com um sistema dedicado à localização)	Não	Com alguma dificuldade	Sim

Dimension	Desempenho			
Factor	Qualidade visual, Segurança e Navegação			
Requisito	Métrica de avaliação	Wk - Cumprimento (%)		
		0	50	100
DQV01 - As imagens são nítidas e têm boa qualidade	As imagens tiradas com a câmara do telemóvel devem meter informação e ser nítidas o suficiente para visualmente permitir a identificação do equipamento	menos de 60% dos questionários com mais de 4 ou 5 pontos	entre 61% a 84% dos questionários com mais de 4 ou 5 pontos	mais de 85% dos questionários com mais de 4 ou 5 pontos
DS01 - As palavras chave estão encriptadas na base de dados	As palavras-chave são encriptadas e guardadas dessa forma na base de dados	Não	-	Sim
DN01 - Produto tem uma boa estrutura e permite aos utilizadores aceder aos conteúdos e às principais funções de forma intuitiva	A aplicação deve ter um menu principal e apenas um submenu para cada elemento do menu	Não	-	Sim
DN02 - A interface de utilizador é rápida na resposta havendo pouca latência nas respostas com servidor	As funcionalidades implementadas na solução devem demorar menos de 5 segundos a serem executadas	menos de 60% dos questionários com mais de 4 ou 5 pontos	entre 61% a 84% dos questionários com mais de 4 ou 5 pontos	mais de 85% dos questionários com mais de 4 ou 5 pontos
DN03 - O tempo de execução da aplicação não possui erros, e erros inesperados devem ser tratados	Se erros inesperados ocorrerem, estes devem ser apresentados ao utilizador com uma mensagem ácidaativa do mesmo	menos de 50% dos questionários com resposta "sempre"	entre 50% a 90% dos questionários com resposta "sempre"	mais de 91% dos questionários com resposta "sempre"
DN04 - Segurança de login e acesso	Exceções de login devem ser apresentadas ao utilizador com uma mensagem indicativa da causa da falha. Somente utilizadores registados podem fazer login.	Não	-	Sim

# Mockups front-end



**Alertas**

Item X tem sobreposição de agendamentos para as datas introduzidas. (Selecione outra data)

1 Itens a Requiritar      2 Detalhes da requisição

Data início: March 18, 2022 4:30 PM

Data entrega: March 18, 2022 4:30 PM

Localização: Empresa X

Requiritar

**Logout**

Notificações

- O item X com referencia interna XX sofreu alteração do seu estado para perdido.
- Requisição com o numero X terminou. Deseja Prolongar?

Prolongar

**Apenas Admins e técnicos de laboratório recebem esta notificação**

**Lista Notificações**

❖ Lista de equipamentos

LEGENDA ESTADO ITEM

- Requisitado (Red dot)
- Livre (Green dot)
- Agendado (Yellow dot)

Pesquisa por nome ou referencia interna

Auto complete

Icon

Badge

Usar QR code

Procurar

Items Existentes

FILTROS

Pesquisar : XXXX

Sala : X

Armário : X

Categorias : X

Item 1 (Green dot)

Pequena descrição

Sala: X

Armário: X1

Categorias: #XXX

Item 1 (Yellow dot)

Pequena descrição

Sala: X

Armário: X1

Categorias: #XXX

Detalhes

Requiritar

click

Item 1

REF Interna: AXXXXXX

Tirar foto

Pequena descrição

Categorias: #1 #2 #3 #4

Estado: Novo

Sala: X

Contentor: X1

Agendamentos

Características Físicas

Outras Características

Boas Práticas

Accessórios

Peso (em Kg): 20.2

Comprimento (em m): 20.2

Editar

Editar

ATUALIZAR

POP UP

Nova característica física

Nome Característica

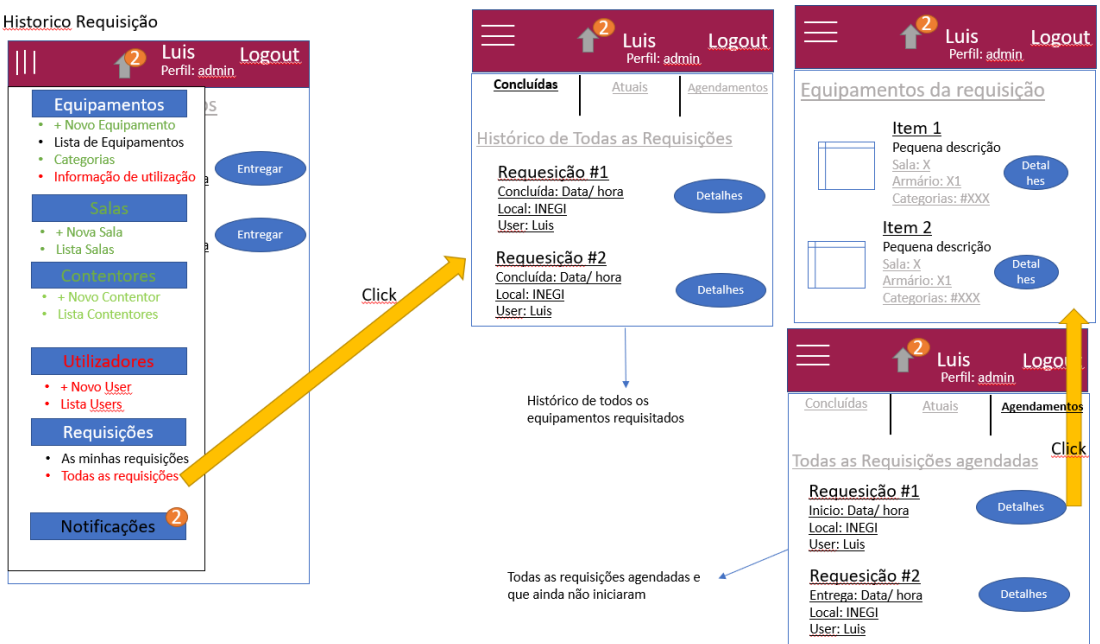
Valor

Novo Danificado Perdido

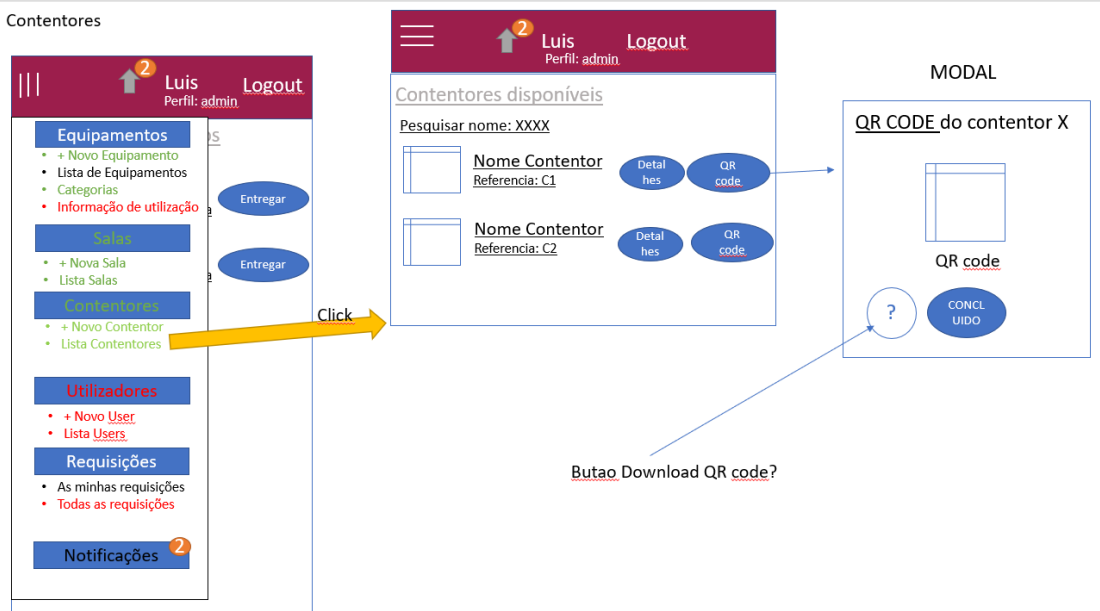
TABS

POP UP (mas preenchido)

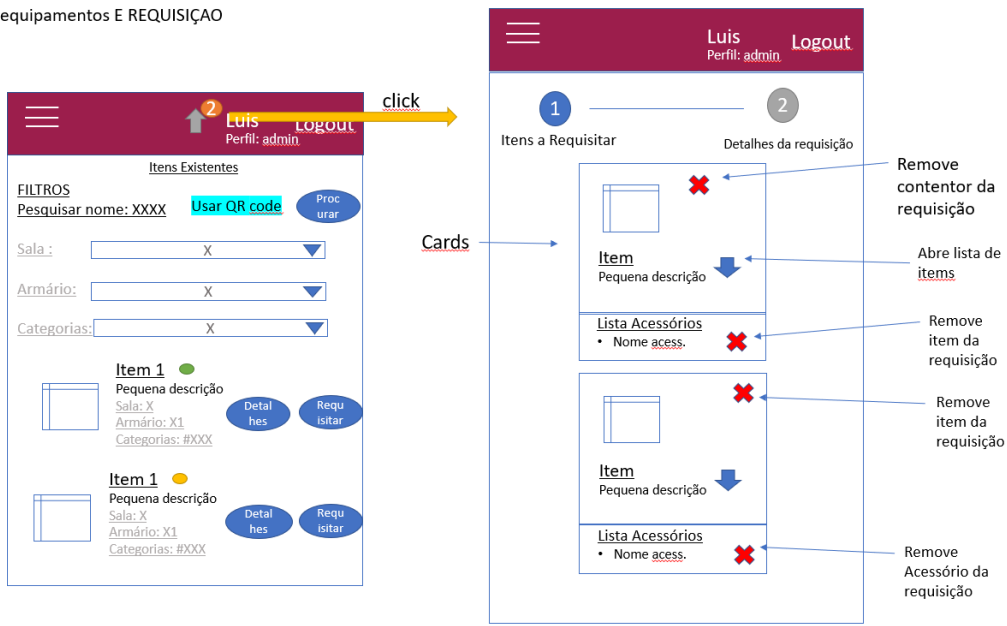
❖ Historico Requisição



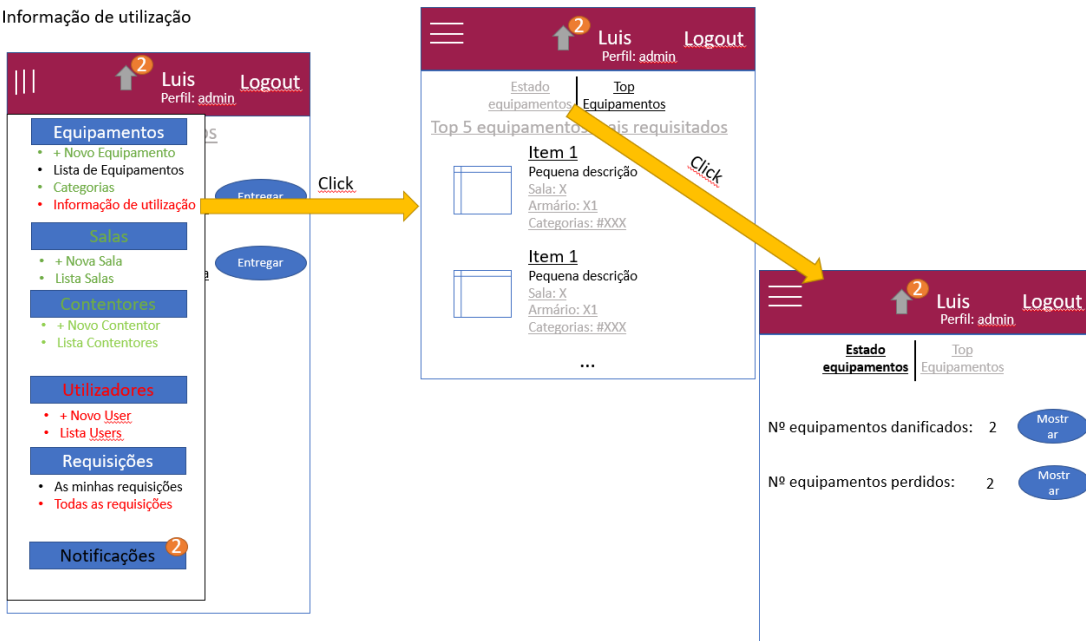
❖ Contentores



❖ Lista de equipamentos E REQUISIÇÃO



❖ Informação de utilização



## Repostas obtidas ao questionário de avaliação de qualidade da aplicação desenvolvida

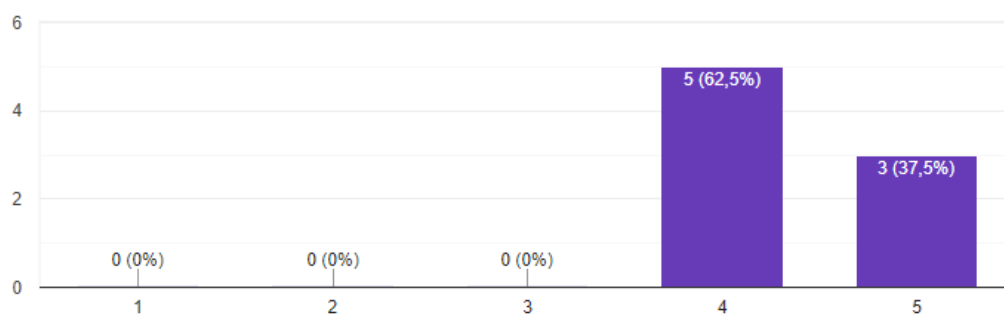
### Usabilidade

As seguintes questões são relativas à usabilidade da aplicação.

A aplicação é intuitiva?

 Copiar

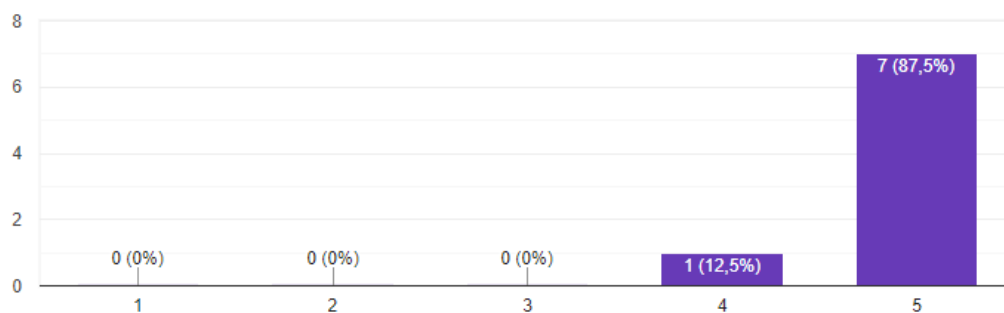
8 respostas



O design é semelhante em toda a aplicação?

 Copiar

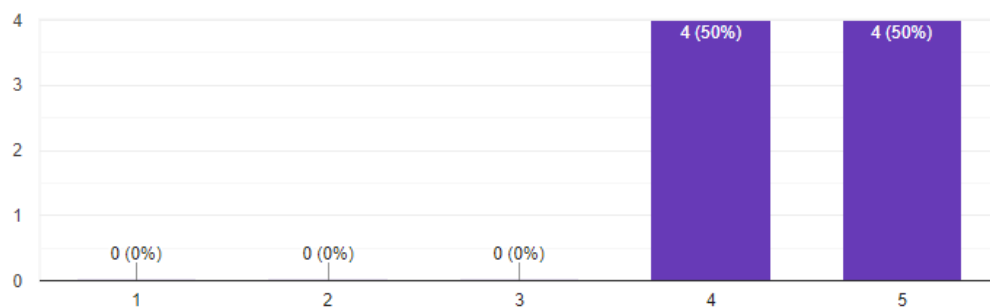
8 respostas



A interação com a interface é semelhante em toda a aplicação?

 Copiar

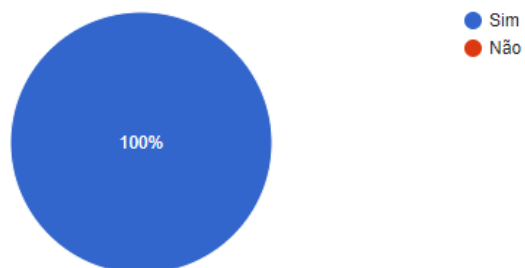
8 respostas



As funcionalidades do sistemas estão presentes em diferentes dimensões de ecrã (telemóvel, computador)?

 Copiar

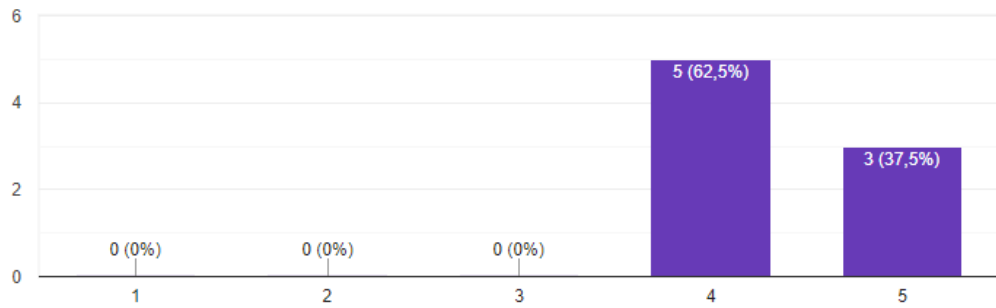
8 respostas



As principais funções são rapidamente acedidas (poucas ações)?

 Copiar

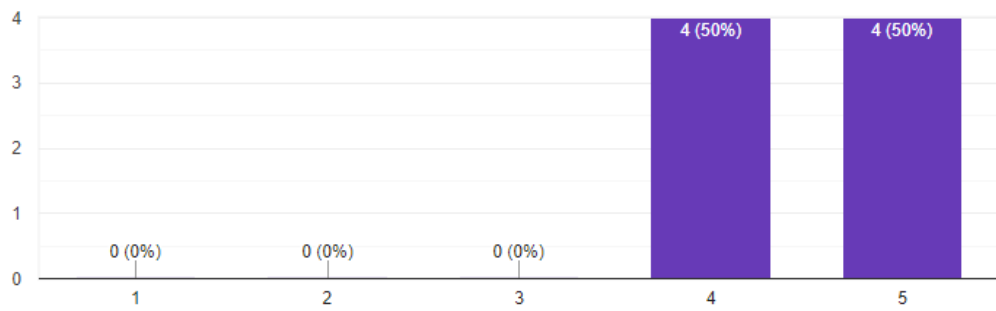
8 respostas



A experiência de navegação é semelhante em toda a aplicação?

 Copiar

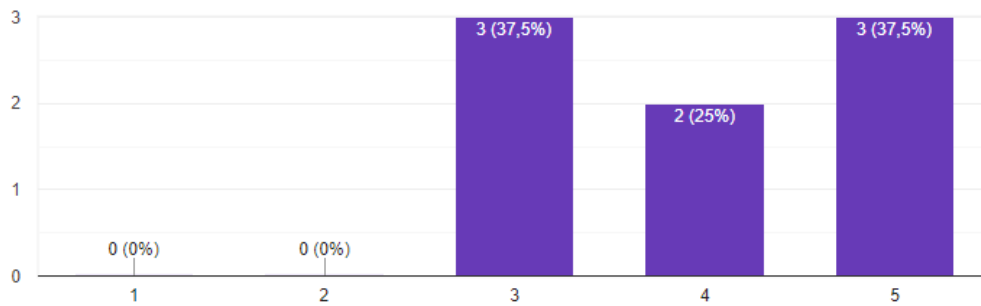
8 respostas



Os filtros de pesquisa de informação são intuitivos e de fácil utilização?

 Copiar

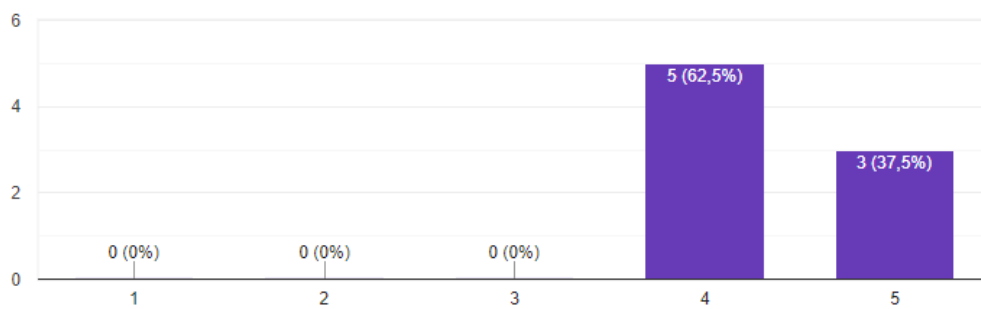
8 respostas



As informações dos equipamentos estão bem organizadas?

 Copiar

8 respostas



Os textos são lexicalmente corretos?

 Copiar

8 respostas



- Sim
- Não

As mensagens apresentam códigos ilegíveis ou símbolos indecifráveis?

 Copiar

8 respostas



- Nenhuma mensagem ilegível
- 1 a 2 Mensagens ilegíveis
- 3 ou mais Mensagens ilegíveis

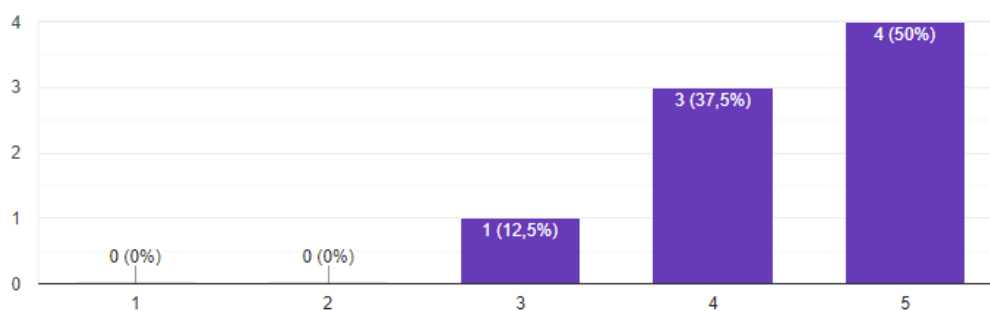
## Desempenho

As seguintes questões são relativas ao desempenho da aplicação

As imagens são nítidas e têm boa qualidade?

 Copiar

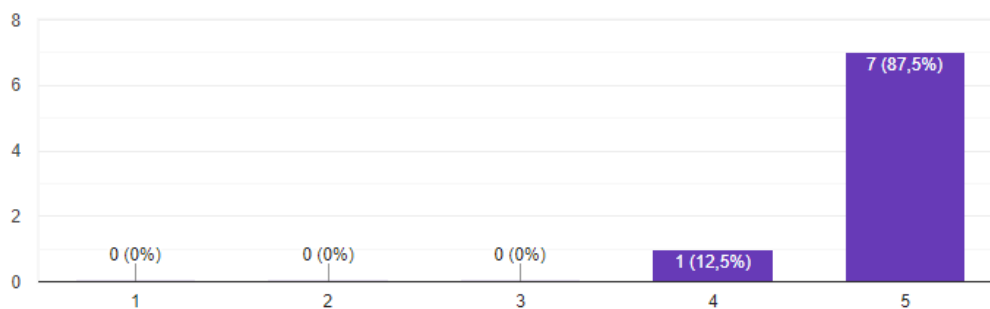
8 respostas



A interface do utilizador é rápida, havendo pouca latência nas respostas?

 Copiar

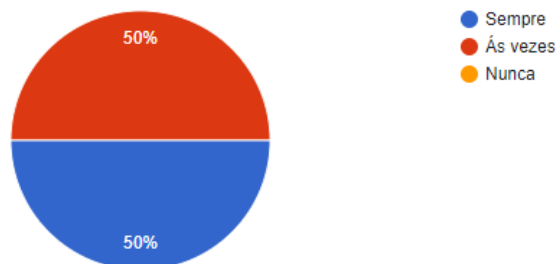
8 respostas



Quando ocorrem erros, estes são apresentados ao utilizador com uma mensagem elucidativa do mesmo?

 Copiar

8 respostas



## Classes software apresentadas e suas responsabilidades

### Controllers

<b>Classe software</b>	<b>Responsabilidade / Descrição</b>
AuthController	Recebe os pedidos dos outros sistemas relativos à autenticação do utilizador, faz validação da informação recebida e reencaminha para os serviços de autenticação.
UserController	Recebe os pedidos dos outros sistemas relativos à gestão de utilizadores (adição, atualização visualização), faz validação da informação recebida e reencaminha para os serviços de gestão de utilizadores
NotificationsController	Recebe os pedidos dos outros sistemas relativos às visualização e atualização das notificações. Faz validação da informação recebida e reencaminha para os serviços de notificação.
RequisitionsController	Recebe os pedidos dos outros sistemas relativos à gestão de requisições de equipamentos (criação, atualização visualização). Faz validação da informação recebida e reencaminha para os serviços de requisição.
CasesController	Recebe os pedidos dos outros sistemas relativos à gestão de contentores de equipamentos (criação, atualização visualização). Faz validação da informação recebida e reencaminha para os serviços de gestão de contentores.
RoomController	Recebe os pedidos dos outros sistemas relativos à gestão de salas (criação, atualização visualização). Faz validação da informação recebida e reencaminha para os serviços de gestão de salas.
ToolsController	Recebe os pedidos dos outros sistemas relativos à gestão de equipamentos (criação, atualização visualização). Faz validação da informação recebida e reencaminha para os serviços de gestão de equipamentos.

## Services

<b>Classe software</b>	<b>Responsabilidade / Descrição</b>
AuthService	Contêm a lógica de negócio relativa a autenticação do utilizador, nomeadamente a geração de tokens de autenticação.
UserService	Contêm a lógica de negócio, fluxo de negócio e entidades do negócio relativas a criação, obtenção de informação e atualização do utilizador.
ProfileService	Contêm a lógica de negócio e faz parte do fluxo de negócio na medida em que valida a existência de um perfil enviado para o servidor (obtenção de informação do perfil do utilizador).
NotificationService	Contêm a lógica de negócio, fluxo de negócio e entidades do negócio relativas a criação, obtenção e atualização do estado de leitura das notificações.
RequisitionService	Contêm a lógica de negócio, fluxo de negócio e entidades do negócio relativas a criação, obtenção e atualização do estado da requisição.
CasesService	Contêm a lógica de negócio, fluxo de negócio e entidades do negócio relativas a criação, obtenção e atualização do contentor.
RoomService	Contêm a lógica de negócio, fluxo de negócio e entidades do negócio relativas a criação, obtenção e atualização da sala
ToolService	Contêm a lógica de negócio, fluxo de negócio e entidades do negócio relativas a criação, obtenção e atualização da ferramenta.
ToolRequisitionStateService	Contêm a lógica de negócio e faz parte do fluxo de negócio na medida em que valida a existência de um estado da ferramenta enviado para o servidor (obtenção de informação do estado da ferramenta - livre, agendado ou ocupado).
LogService	Contêm a lógica de negócio, fluxo de negócio e entidades do negócio relativas a criação e obtenção dos Logs da requisição da ferramenta.

## Repositories

<b>Classe software</b>	<b>Responsabilidade / Descrição</b>
ToolRepository	Acessa os dados da ferramenta e comunica aos Services.
CaseRepository	Acessa os dados dos contentores e comunica aos Services
RequisitionRepository	Acessa os dados da Requisição e comunica aos Services
ToolSchedulesRepository	Acessa os dados dos agendamentos da ferramenta e comunica aos Services.
RoomRepository	Acessa os dados da sala e comunica aos Services
UserRepository	Acessa os dados do utilizador e comunica aos Services
NotificationsRepository	Acessa os dados das notificações e comunica aos Services
ProfileRepository	Acessa os dados do perfil do utilizador e comunica aos Services
ToolRequisitionStateRepository	Acessa os dados do estado de requisição da ferramenta e comunica aos Services

## Mappers

<b>Classe software</b>	<b>Responsabilidade / Descrição</b>
ToolMapper	Faz o mapeamento entre os objetos de domínio e os objetos que armazenam os dados de transporte entre processos da ferramenta.
CaseMapper	Faz o mapeamento entre os objetos de domínio e os objetos que armazenam os dados de transporte entre processos do contentor.
RequisitionMapper	Faz o mapeamento entre os objetos de domínio e os objetos que armazenam os dados de transporte entre processos da requisição.
RoomMapper	Faz o mapeamento entre os objetos de domínio e os objetos que armazenam os dados de transporte entre processos da sala.
UserMapper	Faz o mapeamento entre os objetos de domínio e os objetos que armazenam os dados de transporte entre processos do utilizador
NotificationsMapper	Faz o mapeamento entre os objetos de domínio e os objetos que armazenam os dados de transporte entre processos da notificação.

## DTOs

<b>Classe software</b>	<b>Responsabilidade / Descrição</b>
ToolDTO	Objeto que armazena os dados transportados entre os processos da ferramenta.
CaseDTO	Objeto que armazena os dados transportados entre os processos do contentor.
RequisitionDTO	Objeto que armazena os dados transportados entre os processos da requisição.
RoomDTO	Objeto que armazena os dados transportados entre os processos da sala.
UserDTO	. Objeto que armazena os dados transportados entre os processos do utilizador.

## Utils

<b>Objeto software</b>	<b>Responsabilidade / Descrição</b>
PhotoUtils	Objeto responsável pela conversão da fotografia de base64 para PNG e seu armazenamento no sistema.
auth	Objeto responsável pela autenticação do utilizador no sistema (se existir).
Auth_acess_level	Objeto responsável pelo controlo de acessos dos perfis dos utilizadores que acedem ao sistema.
DateAnalysis	Objeto responsável pela análise de sobreposição de intervalos de datas.

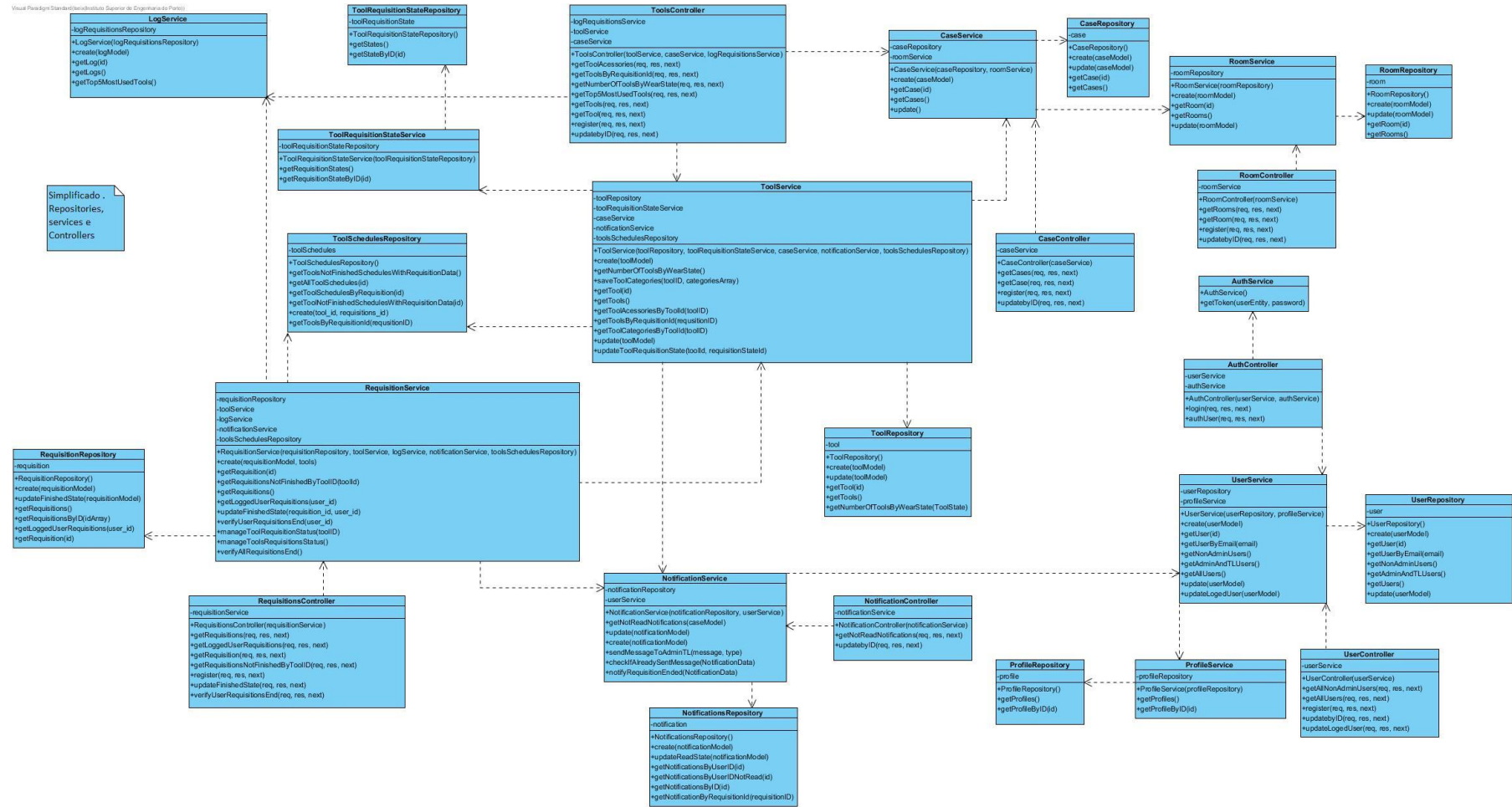
## Routes

<b>Classe software</b>	<b>Responsabilidade / Descrição</b>
ToolRoutes	Recebe os pedidos exteriores e direciona os pedidos para o controller das ferramentas.
AuthRoutes	Recebe os pedidos exteriores e direciona os pedidos para o controller da autenticação.
CaseRoutes	Recebe os pedidos exteriores e direciona os pedidos para o controller do contentor.
RequisitionRoutes	Recebe os pedidos exteriores e direciona os pedidos para o controller da requisição.
RoomRoutes	Recebe os pedidos exteriores e direciona os pedidos para o controller da sala.
UserRoutes	Recebe os pedidos exteriores e direciona os pedidos para o controller do utilizador.
NotificationRoutes	Recebe os pedidos exteriores e direciona os pedidos para o controller da notificação.

## Models

<b>Classe software</b>	<b>Responsabilidade / Descrição</b>
Tool	Classe de domínio da ferramenta.
Case	Classe de domínio do contentor.
Requisition	Classe de domínio da requisição.
Room	Classe de domínio da sala.
User	Classe de domínio do utilizador.

# Diagrama de Classes de implementação



Simplificado .  
Repositories,  
services e  
Controllers