



# **Plataforma de Audiologia *Evollu***

Mestrado em Engenharia Informática – Computação Móvel

Ricardo da Silva Domingues

Leiria, setembro de 2019



# **Plataforma de Audiologia *Evollu***

Mestrado em Engenharia Informática – Computação Móvel

Ricardo da Silva Domingues

Estágio de Mestrado realizado sob a orientação da Doutora Eugénia Moreira Bernardino, Professora da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria e coorientação da Doutora Anabela Moreira Bernardino, Professora da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria.

Leiria, setembro de 2019

# **Originalidade e Direitos de Autor**

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

Reproduções parciais deste documento serão autorizadas na condição de que seja mencionado o Autor e feita referência ao ciclo de estudos no âmbito do qual o mesmo foi realizado, a saber, Curso de mestrado em engenharia informática – computação móvel, no ano letivo 2018/2019 da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria, Portugal, e, bem assim, à data das provas públicas que visaram a avaliação destes trabalhos.

# Agradecimentos

Em primeiro lugar gostaria de agradecer a todas as pessoas e colegas que me acompanharam durante o meu percurso académico e contribuíram, direta ou indiretamente, para a conclusão de mais uma etapa da minha formação.

Agradeço à empresa *Sensing Evolution* pela oportunidade que me deu de estagiar na empresa e por me ter proporcionado todas as condições necessárias para a concretização deste estágio. Ao meu supervisor de estágio, Luís Marcelino, pela orientação e confiança que depositou em mim. A toda a restante equipa deixo um agradecimento, uma vez que sempre me proporcionaram o melhor ambiente de trabalho possível e que agora posso afirmar que os considero amigos próximos.

Quero também agradecer às orientadoras de estágio, Professora Doutora Anabela Bernardino e Professora Doutora Eugénia Bernardino, que foram incansáveis ao longo deste período de estágio, não só pela orientação que me deram, mas também pela disponibilidade, por toda a ajuda, apoio e motivação que sempre me transmitiram.

Agradeço ao Instituto Politécnico de Leiria por me ter acolhido e a todos os docentes por todo o conhecimento que me transmitiram, tanto a nível da licenciatura como do mestrado, sem dúvida que contribuíram para a minha formação.

Por último, e mais importante um agradecimento muito especial à minha família, à minha namorada e aos meus amigos, por todo o apoio e força que sempre me deram, sem eles nada disto seria possível.

# Resumo

O presente relatório descreve o trabalho desenvolvido no âmbito da unidade curricular de Estágio do Mestrado em Engenharia Informática – Computação Móvel, da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria.

O objetivo deste relatório é o de descrever todo o processo envolvido ao longo do estágio curricular de doze meses. O estágio foi realizado na empresa *Sensing Evolution*, uma empresa de desenvolvimento de software na área da saúde com foco na audiologia através da marca *Evollu* e, que proporciona soluções empresariais a empresas que atuem nesta área. Neste relatório é descrita a empresa onde o estágio curricular foi realizado, as tecnologias utilizadas, a metodologia seguida, o trabalho desenvolvido e as conclusões após o término do mesmo.

Este estágio, tinha como objetivo inicial, o estudo da plataforma atual, das funcionalidades a manter e a alterar na mesma, assim como, o desenvolvimento de novas funcionalidades e serviços na área da saúde, de modo a que todos estes se complementassem. Durante o desenvolvimento da plataforma *Evollu*, surgiu igualmente a necessidade de colaborar no processo de correção de falhas de um serviço já existente na empresa, designada como o serviço de *Leads*. Este serviço permite a angariação de *leads* a partir de quiosques disponíveis em farmácias.

Como resultado deste estágio, foi possível adquirir experiência e desenvolver competências a nível profissional através dos vários projetos realizados em ambiente empresarial. O desenvolvimento de software numa arquitetura em micro serviços foi bastante desafiante, dado que foi o primeiro contato feito com este tipo de arquitetura. A correção de falhas existentes e o desenvolvimento de novas funcionalidades na plataforma da *Evollu* auxiliou na conclusão do projeto, como também garantiu a satisfação de todos os *stakeholders*.

**Palavras-chave:** *Leads*, *Stakeholders*, micro serviços

# Abstract

This report describes the work developed in the Internship of the Master's degree in Computer Engineering – Mobile Computing from the school of Technology and Management of the Polytechnic Institute of Leiria.

The aim of this report is to describe the whole process involved throughout the entire internship of twelve months. The internship was conducted on *Sensing Evolution*, a company of software development mainly focused on bringing audiology solutions to other companies that operate in this field, through the brand *Evollu*. This report describes where the curricular internship was held, all the technologies used, the software development methodology used, the work made and the conclusions after the end of the internship.

This internship had the initial purpose the study of the current platform, the features to maintain, as well as the development of new features in order to bring optimized solutions to the users. During the development of the internship it was also refactored the initial service provided by *Evollu* in which it was possible to collect leads from kiosks available in local pharmacies.

As a result of this internship, it was possible to gain experience and develop skills at a professional level through contact with multiple technologies used in the whole platform in a business environment. The development of a platform based on microservices at an architecture level was quite challenging because it was the first contact done with this architecture. The development of new features and its maintenance helped to complete the project and led to a good final product in which the stakeholders were very pleased.

# Índice

<b>Originalidade e Direitos de Autor .....</b>	<b>iii</b>
<b>Agradecimentos .....</b>	<b>iv</b>
<b>Resumo .....</b>	<b>v</b>
<b>Abstract .....</b>	<b>vi</b>
<b>Lista de Figuras .....</b>	<b>ix</b>
<b>Lista de Tabelas .....</b>	<b>xi</b>
<b>Lista de Siglas e Acrónimos .....</b>	<b>xii</b>
<b>1. Introdução .....</b>	<b>1</b>
<b>1.1. Motivação e Objetivos .....</b>	<b>1</b>
<b>1.2. Cronograma do estágio .....</b>	<b>1</b>
<b>1.3. Estrutura do documento .....</b>	<b>3</b>
<b>2. Enquadramento .....</b>	<b>4</b>
<b>2.1. Entidade Acolhedora .....</b>	<b>4</b>
2.1.1. Caracterização .....	4
2.1.2. Equipa e condições de trabalho .....	4
2.1.3. Modelo de Negócio <i>Evollu</i> .....	5
<b>2.2. Audiologia em Portugal.....</b>	<b>6</b>
<b>2.3. Desenvolvimento de aplicações Web .....</b>	<b>7</b>
<b>2.4. Importância de Arquitetura no Desenvolvimento de Software.....</b>	<b>9</b>
2.4.1. Arquitetura Monolítica.....	10
2.4.2. Arquitetura em Micro Serviços .....	11
2.4.3. Arquitetura <i>Serverless</i> .....	13
<b>2.5. Plataforma Inicial .....</b>	<b>15</b>
<b>3. Metodologia de Desenvolvimento.....</b>	<b>17</b>
<b>3.1. Scrum.....</b>	<b>17</b>
<b>4. Desenvolvimento .....</b>	<b>20</b>
<b>4.1. Plataforma <i>Evollu</i> .....</b>	<b>20</b>
<b>4.2. Levantamento de Requisitos .....</b>	<b>21</b>
4.2.1. Perfis de Utilizador .....	21
4.2.2. Lista de Requisitos .....	23
<b>4.3. Arquitetura da Plataforma <i>Evollu</i> .....</b>	<b>29</b>

<b>4.4. Principais Tecnologias Utilizadas .....</b>	<b>34</b>
4.4.1. Servidor.....	34
4.4.2. Cliente.....	36
<b>4.5. Atividades desenvolvidas .....</b>	<b>37</b>
4.5.1. Serviço de Autenticação .....	38
4.5.1.1. Processo de Escolha de Tecnologias .....	38
4.5.1.2. <i>OAUTH 2</i> .....	39
4.5.1.3. Diagrama de Classes - Autenticação .....	42
4.5.1.1. Processo de Criação de Entidades.....	43
4.5.1.2. Processo de Autenticação .....	46
4.5.2. Serviço de <i>Leads</i> .....	50
4.5.2.1. Sistema de Notificações .....	51
4.5.2.2. Critérios de Notificações .....	51
4.5.2.3. Resultado Teste Auditivo.....	52
4.5.3. Serviço de Agendamento .....	53
4.5.3.1. Processo de Agendamento.....	54
4.5.3.2. Agenda.....	58
4.5.3.3. Eventos Recorrentes.....	60
4.5.3.4. Rastreios Auditivos .....	63
4.5.3.5. Identificação de Sessões de Rastreios Auditivos.....	66
4.5.3.6. Exportação de eventos para calendários externos .....	67
4.5.3.7. Métricas e Estatísticas de agendamentos .....	69
4.5.4. Serviço de Registos Clínicos de Audiologia.....	71
<b>4.6. Autorização – Permissões e <i>Roles</i> de utilizadores .....</b>	<b>73</b>
4.6.1. <i>Super User</i> .....	77
4.6.2. Gestão e Atribuição de Permissões.....	77
<b>4.7. Notificações Automatizadas.....</b>	<b>79</b>
<b>5. Processo de Desenvolvimento de Testes .....</b>	<b>83</b>
<b>6. Conclusão .....</b>	<b>85</b>
<b>Bibliografia .....</b>	<b>87</b>
<b>Anexos .....</b>	<b>91</b>
<b>Anexo I – Tarefas Realizadas .....</b>	<b>1</b>

# Lista de Figuras

Figura 1 - Cronograma do estágio .....	2
Figura 2 - Quiosque Físico.....	5
Figura 3 – Quiosque Portátil .....	5
Figura 4 - Dificuldades auditivas agrupadas por faixa etária, em Portugal no ano 2018 [6].....	7
Figura 5 - Arquitetura Monolítica [14] .....	10
Figura 6 - Arquitetura em micro serviços [16] .....	12
Figura 7 - Arquitetura <i>serverless</i> .....	14
Figura 8 - Plataforma corrente - página de estatísticas .....	15
Figura 9 - Plataforma corrente - página de <i>leads</i> .....	16
Figura 10 - Processo de Scrum [26].....	18
Figura 11 - Esquema geral plataforma <i>Evollu</i> .....	21
Figura 12 - Arquitetura inicial plataforma <i>Evollu</i> .....	29
Figura 13 - Arquitetura final ecossistema <i>Evollu</i> .....	30
Figura 14 - Organização dos micro serviços no processo de Audiologia.....	32
Figura 15 - Linguagens e tecnologias utilizadas nos micro serviços.....	35
Figura 16 - Linguagens e tecnologias utilizadas no <i>front-end</i> .....	36
Figura 17 - <i>Bundle</i> de dependências utilizando <i>Webpack</i> [35].....	37
Figura 18 - Constituição de uma chave JWT.....	41
Figura 19 - Diagrama das classes principais servidor autenticação.....	42
Figura 20 - Hierarchy de responsabilidades <i>Glantt</i> .....	44
Figura 21 - Formulário de criação de entidade .....	45
Figura 22 - <i>Workflow</i> na criação de uma entidade.....	45
Figura 23 - Processo autenticação plataforma <i>Evollu</i> .....	46
Figura 24 - Resposta do servidor de autenticação a um <i>login</i> válido .....	47
Figura 25 - Resultado final do menu a partir de JSON.....	48
Figura 26 - Utilização de meta components na <i>sidebar</i> .....	49
Figura 27 - Listagem de critérios de notificação de uma entidade. ....	51
Figura 28 - Formulário de criação de novos critérios de notificação .....	52

Figura 29 - Subscrição de notificações a quiosques .....	52
Figura 30 - Antes e depois do resultado do teste auditivo .....	53
Figura 31 - Listagem de instalações e respetivos detalhes .....	55
Figura 32 - Formulário de criação de disponibilidade de farmácia .....	56
Figura 33 - Gestão de atividades associadas a um utilizador .....	57
Figura 34 - Formulário de Atribuição de Profissional de Saúde .....	57
Figura 35 - Formulário de criação de consulta .....	58
Figura 36 - Agenda vista diária .....	60
Figura 37 - Agenda vista mensal .....	60
Figura 38 - Interface evento recorrente semanal .....	62
Figura 39 - Interface evento recorrente mensal .....	63
Figura 40 - Estrutura de componentes numa página Web .....	64
Figura 41 - Interface implementada para a página dos Rastreios Auditivos .....	65
Figura 42 - Listagem de rastreios em ecrãs de dimensões inferiores .....	66
Figura 43 - Exemplo email com <i>QR Code</i> .....	67
Figura 44 - Diagrama de fluxo da exportação de eventos .....	68
Figura 45 - Subscrição calendário a partir de <i>url</i> no <i>Google Calendar</i> .....	69
Figura 46 - Criação de <i>ICal</i> no perfil de utilizador. ....	69
Figura 47 - <i>Interface</i> de visualização de <i>ICal</i> gerado.....	69
Figura 48 - Métricas e filtros gerais dos rastreios .....	70
Figura 49 - Top Instalações .....	70
Figura 50 - Top Promotores .....	71
Figura 51 - Listagem de pacientes - audiologia.....	72
Figura 52 – Gestão de informação clínica de paciente.....	72
Figura 53 - Diagrama de classes intervenientes no processo de autorização .....	74
Figura 54 - Diagrama de classes - sistema de <i>roles</i> .....	76
Figura 55 - Vista permissões agrupadas por permissão .....	78
Figura 56 - Vista permissões agrupadas por entidade .....	78

# Lista de Tabelas

Tabela 1 - Listagem de requisitos principais da plataforma <i>Evollu</i> .....	23
Tabela 2 - Análise comparativa do nº linhas entre permissões e <i>roles</i> .....	75
Tabela 3 - Análise comparativa de várias API de envio de SMS .....	80
Tabela 4 - Listagem de notificações automatizadas .....	80

# Lista de Siglas e Acrónimos

API	<i>Application Programming Interface</i>
AWS	<i>Amazon Web Services</i>
BAAS	<i>Backend as a Service</i>
BDD	<i>Behaviour Driven Development</i>
B2B	<i>Business to Business</i>
CEO	<i>Chief Executive Officer</i>
CPU	<i>Central Processing Unit</i>
CRUD	<i>Create Read Update Delete</i>
CSS	<i>Cascading Style Sheets</i>
CTO	<i>Chief Technology Officer</i>
DAS	<i>Desenvolvimento Ágil Software</i>
ESTG	<i>Escola Superior de Tecnologia e Gestão</i>
GCP	<i>Google Cloud Platform</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
HTTPS	<i>Hypertext Transfer Protocol Secure</i>
IPL	<i>Instituto Politécnico de Leiria</i>
JWT	<i>Json Web Tokens</i>
KPI	<i>Key Performance Indicator</i>
ORM	<i>Object Relational Mapping</i>
PDF	<i>Portable Document Format</i>
PHP	<i>PHP:Hypertext Preprocessor</i>
QR	<i>Quick Response</i>
REST	<i>Representational State Transfer</i>
SAML	<i>Security Assertion Markup Language</i>
SMS	<i>Short Message Service</i>
SQL	<i>Structured Query Language</i>
UI	<i>User Interface</i>
URL	<i>Uniform Resource Locator</i>
UX	<i>User-Experience</i>

# 1. Introdução

O presente relatório descreve o trabalho desenvolvido ao longo do estágio inserido na unidade curricular de Estágio do Mestrado em Engenharia Informática – Computação Móvel, lecionado na Escola Superior de Tecnologia e Gestão (ESTG), nomeadamente, a metodologia de desenvolvimento, as tecnologias utilizadas e a descrição das tarefas desenvolvidas no estágio. Na Secção 1.1 deste capítulo, é descrita a motivação e objetivos do estágio; na Secção 1.2 é apresentado o cronograma de estágio com as principais tarefas desenvolvidas; e por fim, na Secção 1.3 é apresentada a estrutura do documento.

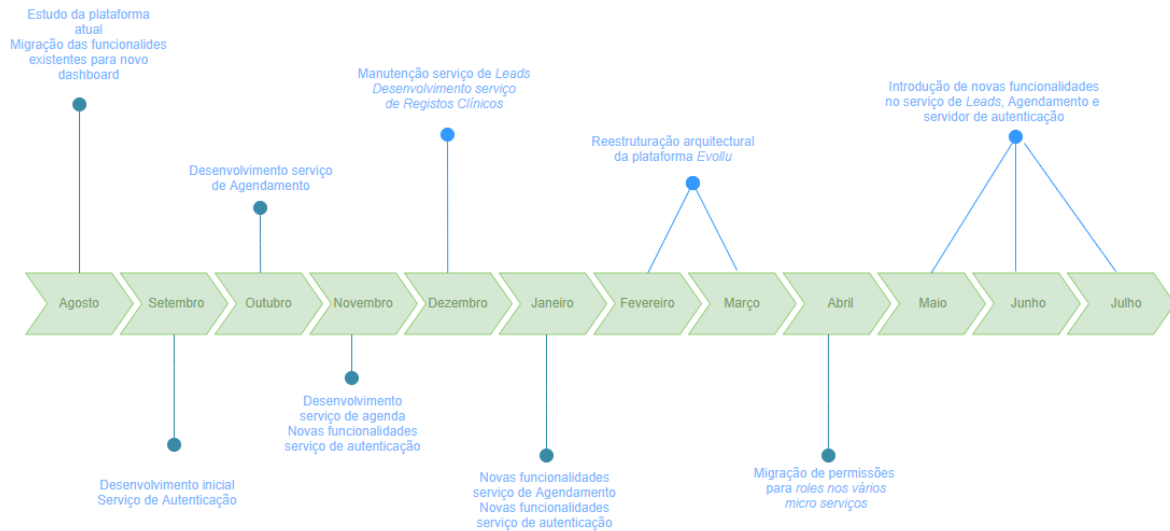
## 1.1. Motivação e Objetivos

O motivo que levou à decisão da realização do estágio curricular, no âmbito do mestrado, na empresa *Sensing Evolution*, deveu-se, principalmente, ao facto das tecnologias utilizadas pela mesma irem de encontro às tecnologias em que pretendia alargar os meus conhecimentos, e também, por ser considerada uma *startup* em que, por norma, por se denominar uma empresa em crescimento e ainda numa fase inicial, é possível ter um acréscimo de responsabilidades.

A *Sensing Evolution* através da marca *Evollu* é uma empresa de desenvolvimento de software que atua na área da saúde com foco particular na audiologia. A *Evollu* possui uma aplicação móvel onde é possível realizar um rastreio auditivo gratuito. Caso se verifique que a pessoa que realizou o rastreio auditivo possui dificuldades auditivas substanciais, é possível obter um acompanhamento personalizado. Para além da aplicação móvel, a *Evollu* conta, no momento, com vários quiosques instalados não só em farmácias em Portugal, como também em Inglaterra e Espanha.

## 1.2. Cronograma do estágio

O estágio, com a duração de 12 meses, teve início a 1 de agosto de 2018 e a sua conclusão a 31 de julho de 2018. Na Figura 1 é apresentado um cronograma das tarefas realizadas semanalmente.



**Figura 1 - Cronograma do estágio**

Uma vez que o estágio decorreu em ambiente empresarial, foi necessário um período inicial de adaptação, que consistiu no estudo da plataforma atual, análise dos módulos a desenvolver e adaptação às ferramentas de desenvolvimento da mesma.

O projeto iniciou-se com uma análise funcional do módulo já existente, o estudo das funcionalidades a manter e quais as funcionalidades a introduzir. Foram realizadas várias tarefas complexas no estágio, sendo uma delas a implementação do serviço referente ao agendamento, que consiste na criação de disponibilidades por parte de farmácias, assim como as consultas e respetivo acompanhamento por profissionais na área de saúde. Esta funcionalidade exigiu um período de desenvolvimento mais longo, devido à sua complexidade, como também pelas constantes mudanças nos requisitos da mesma, e pelas novas funcionalidades que foram sendo solicitadas ao longo de todo o processo para complementar o agendamento de consultas.

O serviço responsável pela autenticação de utilizadores e de gerir a informação relativa aos mesmos e às inúmeras entidades existentes foi bastante desafiante, uma vez que foi possível aprofundar conhecimentos em tecnologias que ainda não tinham sido empregues com frequência durante o percurso académico.

Durante o desenvolvimento de todos os micro serviços realizados foram solicitadas novas alterações e melhorias relativamente ao que estava planeado inicialmente, o que, de forma previsível, estendeu o tempo das mesmas.

### **1.3. Estrutura do documento**

O presente relatório está estruturado em seis capítulos, sendo que o Capítulo 2 (Enquadramento) apresenta uma contextualização do estágio, onde é descrita a empresa de acolhimento e as condições de desenvolvimento oferecidas pela mesma. Descreve ainda uma breve introdução ao desenvolvimento Web, às diferentes arquiteturas Web existentes e respetivas vantagens e desvantagens e, por fim, ao modelo de negócio corrente da *Evollu*.

No Capítulo 3 (Metodologia de desenvolvimento) é apresentada a metodologia de desenvolvimento utilizada e seguida ao longo do estágio.

No Capítulo 4 (Desenvolvimento) é apresentada a fase de implementação, que se encontra estruturada tendo em consideração cada projeto intervencionado. Tem início com um levantamento dos requisitos principais, de seguida são descritas as principais tecnologias utilizadas no decorrer do estágio, e posteriormente é apresentado o processo que foi tido em conta na transição da arquitetura inicial, sendo esta monolítica para uma arquitetura baseada em micro serviços. Por fim, são descritas as funcionalidades implementadas nos vários micro serviços realizados, assim como o modo de como foi desenvolvido todo o sistema de permissões que afeta todos estes micro serviços.

No Capítulo 5 são descritas as ferramentas utilizadas na elaboração dos testes implementados, assim como o processo envolvido na realização de testes *ad-hoc*.

Por fim, no Capítulo 6 (Conclusão) é feito um balanço geral do trabalho desenvolvido, referindo os resultados obtidos e as dificuldades que surgiram ao longo da realização do estágio.

## 2. Enquadramento

Neste capítulo é apresentado um enquadramento aos principais conceitos e temas relacionados com o trabalho desenvolvido. Inicialmente é descrita a entidade de estágio (Secção 2.1), e apresentada uma breve análise do estado da audiologia em Portugal (Secção 2.2). De seguida é realizado um enquadramento ao desenvolvimento de aplicações Web (Secção 2.3) e aos diversos tipos de arquitetura existentes de aplicações Web (Secção 2.4), e por fim, é feito uma avaliação à plataforma atual, que representa o projeto *Leads* que constitui o único serviço existente à data de início do estágio (Secção 2.5).

### 2.1. Entidade Acolhedora

Neste capítulo é feita uma apresentação da empresa onde decorreu o estágio, dando a conhecer a equipa e as condições de trabalho no decorrer do mesmo.

#### 2.1.1. Caracterização

A *Sensing Evolution* é uma *startup* fundada em 2015 que opera na área da saúde e bem-estar, oferecendo um conjunto de serviços inovadores suportados pela marca *Evollu* [1].

A *Sensing Evolution* tem o objetivo de conseguir dispor ao maior número de pessoas possíveis soluções auditivas inteligentes, discretas e capazes de melhorar e providenciar uma melhor saúde auditiva, possibilitando uma maior qualidade de vida, tornando-se assim num líder mundial nesta área [2].

#### 2.1.2. Equipa e condições de trabalho

Na data inicial do estágio, a equipa da *Sensing Evolution* era constituída apenas por 5 elementos, dos quais exclusivamente 3 realizavam desenvolvimento de software, sendo que os restantes faziam parte da direção.

Na data atual, a equipa de desenvolvimento é composta por 6 elementos, e a empresa contabiliza no total 11 colaboradores pertencentes a diversos cargos. A equipa de desenvolvimento encontra-se dividida nos vários projetos que se encontram a ser desenvolvidos. Desde o início do estágio, foi possível verificar a existência de um bom ambiente de trabalho entre todos os colaboradores, transmitindo sempre uma grande motivação e disponibilidade para esclarecer qualquer dúvida.

A empresa disponibilizou, em primeira mão, o material necessário para desempenhar as funções pretendidas. O material incluiu um computador portátil e um monitor secundário.

### 2.1.3. Modelo de Negócio *Evollu*

A *Sensing Evolution* é uma empresa que atua na área da saúde, no auxílio e fornecimento de serviços na área da saúde, focados na audiologia, a outras empresas.

De forma sucinta, a *Evollu* possui vários quiosques disponíveis para realizar avaliações auditivas a qualquer utilizador. Os quiosques são comercializados de duas maneiras distintas. Ambos correspondem a um *tablet android*, no entanto, estes distinguem-se entre portáteis e físicos. Os *tablets* portáteis, como o nome indica têm portabilidade, sendo que se encontram associados a promotores específicos para a realização de rastreios auditivos em várias farmácias (ver Figura 3), enquanto que os *tablets* físicos encontram-se imóveis num suporte físico e não podem ser movidos (ver Figura 2).



Figura 2 - Quiosque Físico



Figura 3 – Quiosque Portátil

Em parcerias com farmácias e outras entidades, maioritariamente, de origem Portuguesa, os quiosques são comercializados e instalados, onde qualquer pessoa pode realizar o teste auditivo e submeter os seus dados na plataforma para que possa ser contactado posteriormente, caso apresente problemas a este nível. Todos os utilizadores que submetam o teste auditivo correspondem a uma *lead*, isto é, num contexto de vendas e marketing uma

*lead* corresponde a um potencial cliente, que neste caso poderá vir a adquirir aparelhos auditivos.

É necessário ter em consideração que o maior foco da *Evollu*, no momento, não é realizar o acompanhamento destas *leads*, mas sim, possibilitar a outras empresas na área que o façam, de modo a facilitar a gestão de todo o processo. Assim, à data atual, a *Evollu* possui o papel de intermediário entre o consumidor final e outras empresas, sendo que toda a plataforma corresponde a um produto direcionado a B2B (*Business to Business*) e, onde existem vários pacotes que podem ser adquiridos por empresas, ajustados às suas necessidades [3].

Estas empresas podem consultar as suas *leads* na plataforma *Evollu*, e realizar todo o acompanhamento das mesmas nas suas plataformas internas. Caso se verifique que uma *lead* possui perda auditiva é paga uma comissão à *Evollu*, consequentemente a farmácia de onde a *lead* foi originada recebe uma parte deste valor.

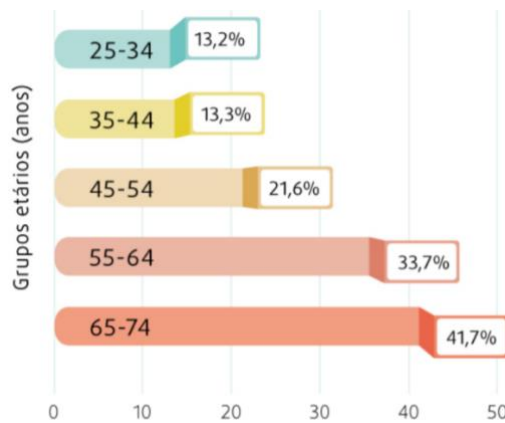
## **2.2. Audiologia em Portugal**

A saúde é um dos pilares do bem-estar do ser humano, porém o conceito de saúde está dividido em inúmeras especialidades médicas específicas. A audiologia refere-se à ciência responsável por diagnosticar possíveis patologias ou dificuldades auditivas, dedicando-se assim ao estudo do ouvido, debruçando-se sobre todas as suas patologias [4]. Para além disso, é responsável também por aconselhar os cuidados a ter de forma a proporcionar às pessoas um melhor bem-estar.

De acordo com os resultados do Inquérito Nacional de Saúde realizado pelo Instituto Nacional de Saúde Dr. Ricardo Jorge em 2018 (ver Figura 4), estima-se que um em cada quatro portugueses possuam dificuldades auditivas, sendo mais predominante em indivíduos em faixas etárias superiores (65-74 anos), uma vez que a maior parte das pessoas vai perdendo a audição de forma gradual à medida que vai envelhecendo, explicando assim uma maior perda auditiva na população sénior [5].

Em média, o tempo entre o aparecimento dos primeiros sintomas e a decisão de procurar ajuda médica especializada demora cerca de 7 anos, visto que as pessoas têm tendência a omitir problemas, devido aos sintomas serem tão ténues em fases iniciais. Alguns destes sintomas passam por, solicitar às pessoas para se repetirem múltiplas vezes, não conseguir dialogar com várias pessoas em simultâneo em ambientes com algum ruído ou até mesmo prestar demasiada atenção aos lábios do interlocutor enquanto se mantém uma conversa [7].

Assim, algumas das recomendações gerais de prevenção de deficiências auditivas são reduzir o número de aparelhos sonoros a emitir som em simultâneo, não ouvir música com auriculares durante longos períodos de tempo, manter o volume abaixo do limite máximo recomendado em 60% em dispositivos móveis [8].



**Figura 4 - Dificuldades auditivas agrupadas por faixa etária, em Portugal no ano 2018 [6]**

Apesar da predominância de dificuldades auditivas na população sénior, esta pode manifestar-se em qualquer faixa etária. Em Portugal, tem-se verificado um crescimento do número de crianças com perda auditiva. Três em cada mil crianças nascem já com problemas auditivos com origem genética, números estes que indicam a necessidade da existência de rastreios, para que se possa identificar de forma prévia, possíveis dificuldades ou deficiências, de modo a existir uma rápida intervenção [7].

Desta forma, a *Evollu* disponibiliza em primeira mão, tanto uma aplicação móvel como inúmeros quiosques instalados em farmácias, onde é possível realizar o teste auditivo, com o objetivo de efetuar um diagnóstico precoce de dificuldades auditivas para que a mesma possa ser tratada o mais rapidamente possível.

### **2.3. Desenvolvimento de aplicações Web**

A maior parte do trabalho realizado na empresa compreendeu o desenvolvimento de aplicações Web. O desenvolvimento de aplicações Web refere-se a todo o processo envolvido na criação de um *website*, podendo o mesmo variar desde uma página simples com informação apresentada de forma estática, até um portal completo com diversas páginas em que grande parte dos conteúdos é conseguido de forma dinâmica [9].

O desenvolvimento Web tem inúmeras etapas subjacentes, nomeadamente: *Web design*, criação de conteúdos, programação, criação de *scripts* do lado do cliente ou servidor, configurações de segurança, de *deployment*, entre outras tarefas [10].

No desenvolvimento de aplicações Web é necessário compreender os princípios atuais do desenvolvimento Web. Estes princípios ajudam os programadores a criar aplicações de sucesso de forma a satisfazer as necessidades atuais e futuras dos utilizadores e a adaptarem-se às tendências tecnológicas que estão em constante evolução [9].

Apesar do desenvolvimento Web não ser um conceito meramente recente, podendo-se afirmar que acabou por se torna numa área bastante madura apesar das constantes mudanças desde as tecnologias utilizadas, e às normas empregues. No entanto, existem princípios que devem ser seguidos sendo um dos principais o papel do utilizador. Este princípio enfatiza a importância do utilizador, e dessa forma a Web deve ser acessível a todas as pessoas e dispositivos, de forma a melhorar a usabilidade e a experiência de utilização [11].

Outro dos princípios fundamentais, que não se aplica apenas ao desenvolvimento Web, mas em todas as áreas da informática é a segurança da informação. A maior parte das aplicações Web sofrem de vulnerabilidades e estão sujeitas a inúmeros ataques, que devem ser evitados a todo o custo, através da implementação de boas práticas de segurança, validação dos dados, entre outras. Os programadores devem implementar mecanismos não só de autenticação, como também de autorização para legitimar qual o utilizador que se encontra a realizar uma ação e, se o mesmo tem permissões para tal [11].

Com o crescimento de aplicações Web, o desenvolvimento Web torna-se cada vez mais complexo, uma vez que acaba por exigir a quem trabalhe na área, capacidade para adquirir um vasto conjunto de conhecimentos para complementar as exigências necessárias de entregar um produto com qualidade. Assim, um outro princípio é o da reutilização, isto é, com a enorme quantidade de ferramentas, bibliotecas e *frameworks* testadas, estas podem auxiliar o programador a realizar as suas tarefas de forma mais fácil e eficaz [11].

Por fim, temos a própria arquitetura do sistema que apesar de ser algo que é transparente para o utilizador final, esta poderá ter impacto significativo no próprio sistema. A arquitetura retrata a escalabilidade, flexibilidade, segurança, manutenção, entre outros. A utilização de boas práticas de arquitetura de software poderá causar um atraso no desenvolvimento do

mesmo numa fase inicial, mas que a longo termo pode trazer inúmeros benefícios[11]. Na Secção 2.4 irá ser abordada, com mais detalhe, a importância da arquitetura utilizada.

Ao longo do decorrer do estágio, foi tido o cuidado de utilizar todos os princípios mencionados nesta secção.

## **2.4. Importância de Arquitetura no Desenvolvimento de Software**

O utilizador comum consegue aceder a uma página Web específica na Internet, através de uma série de interações entre vários componentes de uma aplicação. Este aglomerado de componentes que constituem todo o sistema constitui a arquitetura do mesmo.

A arquitetura de uma plataforma Web diz respeito à forma como todos os componentes referentes à aplicação interagem entre si. Estes componentes são, essencialmente, todos os elementos que constituem a plataforma. De uma forma abrangente é uma aplicação cliente-servidor, que possui base de dados para armazenar a informação, *middlewares*, e interfaces para os utilizadores, onde estes podem analisar e atuar sobre todo o sistema. O servidor é responsável por validar a informação que é submetida pelos utilizadores finais, armazenar a informação nas unidades de armazenamento e tornar toda esta informação disponível quando solicitada pelos utilizadores [12].

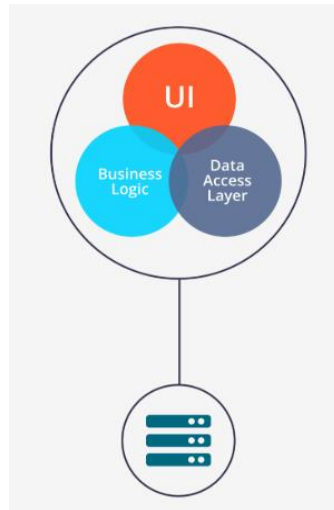
Existem vários padrões arquiteturais que permitem obter diferentes objetivos e resultados consoante os requisitos dos utilizadores, dos próprios programadores e dos responsáveis pelo produto. Os requisitos dos utilizadores prendem-se geralmente à usabilidade do sistema, isto é, à maneira como o sistema interage com os mesmos, ao tempo que leva a realizar uma ação específica, capacidade de guardar opções feitas em sessões futuras, facilidade de transição entre várias páginas, entre outros. Os requisitos dos programadores envolvem requisitos não funcionais, como é o caso da performance de um sistema; a sua escalabilidade, que se refere à capacidade de como a aplicação responde com um grande número de utilizadores em simultâneo; à introdução de novas funcionalidades no sistema e o tempo que leva a introduzir as mesmas; ao tempo de execução dos pedidos, em que estes são compostos pelo processamento dos mesmos; aos pedidos de informação às bases de dados e a respetiva resposta. Por fim, os requisitos dos responsáveis de projeto dizem respeito à forma como o sistema opera, e quais as exigências a nível de recursos como hardware, software, manutenções, infraestrutura da rede e segurança, que são necessários para que o mesmo possa tirar o melhor partido com o menor custo possível [13].

Assim, pode-se afirmar que não existe uma melhor arquitetura ou até mesmo a arquitetura ideal, existem sim arquiteturas de sistemas que se comportam de maneira mais eficiente, tendo em conta os requisitos e exigências impostos pelos utilizadores envolvidos.

Deste modo, a arquitetura da aplicação é algo extremamente relevante para que a mesma tenha êxito e, decisões inadequadas podem levar ao insucesso do projeto, assim é bastante importante a existência de uma reflexão sobre quais os objetivos da plataforma, para que haja o menor número possível de impedimentos a afetar a mesma.

#### 2.4.1. Arquitetura Monolítica

Uma arquitetura monolítica é uma arquitetura de software onde todo o código é composto num só fragmento. Se um dos requisitos necessitar de alterações pode influenciar todo o sistema ou outras funcionalidades, uma vez que todas fazem parte do mesmo processo. Assim, uma aplicação monolítica funciona como uma unidade única e autónoma, em que as combinações de diversos componentes formam a plataforma no seu todo, componentes estes como a autenticação e autorização, a camada de apresentação, a lógica de negócio, a camada de armazenamento, entre outros (ver Figura 5) [14].



**Figura 5 - Arquitetura Monolítica** [14]

Devido às características deste tipo de arquitetura é possível enumerar algumas desvantagens acerca da utilização desta abordagem, sendo elas as seguintes:

- Manutenção de funcionalidades, uma vez que a adição de novas funcionalidades neste tipo de sistemas poder-se-á tornar uma tarefa árdua, uma vez que pode afetar todo o sistema e requerer assim, uma maior garantia de qualidade através da

realização de testes ao software e a utilização de ferramentas que auxiliem neste tipo de tarefas.

- Curva de aprendizagem, isto é, num sistema já existente poderá tornar-se difícil a entrada de novos colaboradores, uma vez que apesar de neste tipo de arquitetura o desenvolvimento da mesma ser mais simples, poderá tornar-se complexo para novos membros integrarem o projeto.
- Disponibilidade, dado que por vezes um erro bastante simples pode afetar todo o sistema e tornar o mesmo indisponível para todos os utilizadores.
- Escalabilidade vertical, pois a tarefa de escalar partes únicas do sistema é extremamente difícil, visto que diferentes componentes de um sistema podem necessitar de diferentes tipos de recursos como um maior processamento de CPU (*Central Processing Unit*) ou até maior capacidade de memória disponível.

Ainda assim, existem inúmeras vantagens na utilização deste tipo de arquitetura, devido à sua natureza acoplada, tais como:

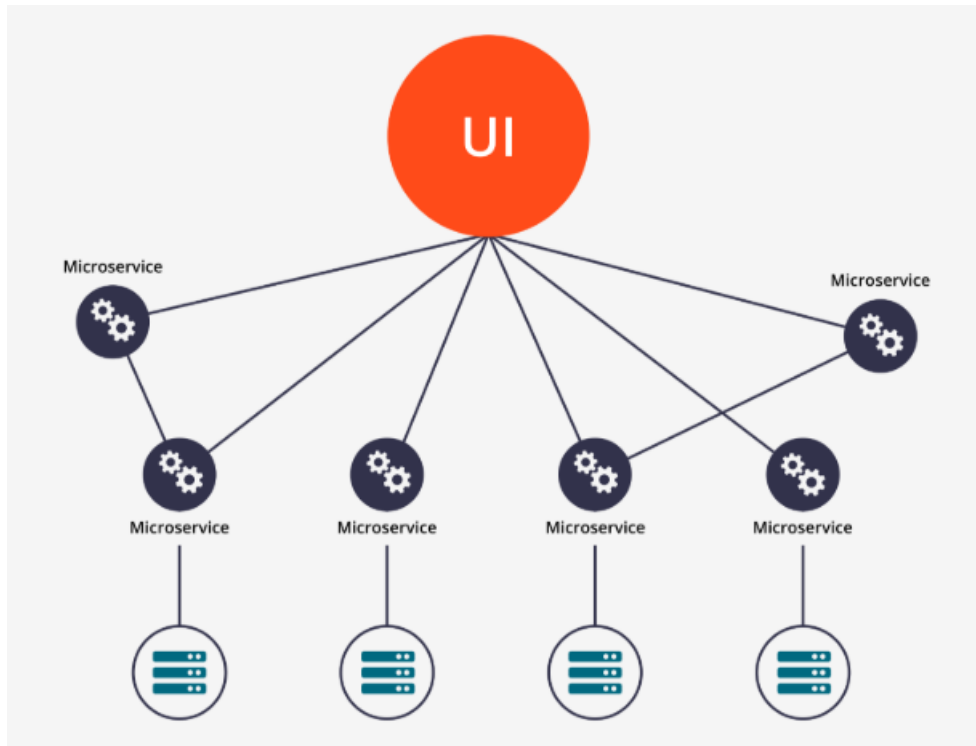
- Processo de desenvolvimento simples, uma vez que como todo o código se encontra no mesmo sítio leva a que numa fase inicial, a introdução de funcionalidades seja efetuada de forma rápida e eficaz.
- Qualidade de produto a nível de testes, devido ao facto de ser um sistema acoplado, a realização de testes desde unitários à integração ou até mesmo testes automatizados *selenium* torna-se uma tarefa relativamente simples.
- Escalabilidade horizontal, uma vez que apenas é necessário realizar uma cópia da aplicação e colocar um *load balancer* a redirecionar o tráfego entre as diversas cópias do sistema [15].

Deste modo, a utilização desta arquitetura poderá ser útil em alguns dos casos, mas de forma a tentar suprimir as desvantagens introduzidas pela mesma, uma nova arquitetura denominada arquitetura em micro serviços, foi introduzida.

#### 2.4.2. Arquitetura em Micro Serviços

A arquitetura em micro serviços foi uma técnica introduzida recentemente que tem como intuito reduzir toda a carga que uma solução monolítica provoca nos servidores, e em simultâneo, fornecer uma variedade de serviços que se encontram associados a grandes aplicações de negócio. Isto é adquirido através da separação entre responsabilidade da lógica

de negócio em inúmeros serviços menores, que são combinados para fornecer todos os recursos necessários aos utilizadores finais. Uma arquitetura com este padrão de desenvolvimento acaba por se tornar mais difícil de projetar e com uma curva de aprendizagem maior, no entanto, cada componente acaba por se tornar bastante simples e isolado (ver Figura 6) [16].



**Figura 6 - Arquitetura em micro serviços** [16]

Assim, uma arquitetura baseada em micro serviços garante que múltiplos serviços que se encontrem pouco acoplados funcionem em conjunto, em que cada serviço possui o seu propósito e, respetivamente, a informação necessária para conceber o mesmo.

Algumas das vantagens num sistema concebido em micro serviços são as seguintes:

- Melhor isolamento de erro, uma vez que a aplicação poderá continuar a funcionar isenta de qualquer problema, apesar de um dos módulos que a constitui não estar a funcionar.
- Adotar novas tecnologias, visto que existe a flexibilidade de utilização de novas tecnologias que revelam ser melhores em certas situações, apenas num serviço, sem danificar as dependências que existem para os outros.
- Escalabilidade, dado que é possível escolher as tecnologias mais apropriadas para cada módulo específico, e poder-se-á colocar o mesmo serviço em diferentes

servidores, de modo a garantir melhores resultados a nível de performance sem causar qualquer impacto nos restantes.

- Facilidade de interpretação, pois cada serviço possui apenas um único propósito.
- Qualidade, dado que os benefícios de construir uma solução em pequenos e vários módulos resulta numa melhoria da reutilização e manutenção do código, e a própria qualidade do mesmo a nível de testes [15].

Porventura, adotar este tipo de arquitetura não é de todo uma tarefa simples, como referido acima, a curva de aprendizagem é maior e, pode acabar por afetar a produtividade do produto final. Este tipo de soluções resolve alguns problemas, mas acaba por gerar outros, tais como:

- Micro serviços mal designados acabam por causar mais danos do que benefícios.
- A adoção de várias linguagens para diferentes micro serviços acaba por aumentar a complexidade da solução, uma vez que são necessárias equipas com vastos conhecimentos das linguagens utilizadas, e também, porque não permite uma total reutilização de código entre os mesmos.
- Interdependências entre micro serviços, ou seja, por vezes a falta de experiência no desenvolvimento de micro serviços tem como consequência as fronteiras entre os mesmos não serem bem definidas, o que leva a alguns serviços serem bastante dependentes de outros, o que vai contra os princípios de desenvolvimento em micro serviços [17].

#### 2.4.3. Arquitetura *Serverless*

Por último, existe a arquitetura *serverless*. Neste tipo de arquitetura a aplicação depende de *third-party services*, referidos como *BaaS – Backend as a Service*. O nome *serverless* refere-se ao facto de o utilizador não necessitar de adquirir, atualizar ou realizar manutenções nos seus servidores, uma vez que todo o código processado é realizado nos servidores do *cloud provider*. De uma forma bastante geral, neste modelo são implementadas várias funções, em que o código definido por elas irá ser executado na chamada de um *endpoint* [18].



Figura 7 - Arquitetura *serverless*

Algumas das vantagens deste tipo de arquitetura encontram-se descritas nos seguintes pontos:

- Escalabilidade, pois é possível que várias funções sejam executadas em paralelo, e pode ser concedido mais poder de processamento a certas funções que assim o necessitem.
- Não gerir os próprios servidores, esta tarefa apesar de muitas vezes não ser uma preocupação constante, a verdade é que com este tipo de modelo os receios sobre instalação ou atualização de software, ou caso o servidor seja reiniciado ou até ficar *down* já não existem.
- O preço é uma das principais vantagens, uma vez que o modelo de custo neste tipo de arquitetura é baseado no tempo de execução de cada função e da memória necessária para correr a mesma, ao invés do custo de provisionamento de servidores em arquiteturas tradicionais.

Quanto às desvantagens, pode-se afirmar que as principais são:

- Controlo reduzido, uma vez que o utilizador não possui qualquer tipo de infraestrutura.
- Dependente da *third party service* escolhida, o que exige que haja bastante confiança sobre a mesma.
- Testar funções que se encontram na *cloud* poderá tornar-se uma tarefa bastante complexa [18][19].

Mais à frente na Secção 4.3 iremos descrever tanto a arquitetura original à data do início do estágio como a arquitetura final da plataforma *Evollu*, assim como todo o processo que desencadeou a respetiva mudança.

## 2.5. Plataforma Inicial

A plataforma inicial contava apenas com 4 ecrãs distintos, como mostra a Figura 8 e a Figura 9, sendo eles:

- **Statistics** onde os dados que são avaliados correspondem às *leads*, que podem ser provenientes dos balcões das farmácias, ou dos quiosques onde foram realizados os rastreios. Neste ecrã também é possível visualizar a quantidade de pessoas que foram contactadas e, destas, quantas compareceram em consultas;
- **References** que corresponde a uma listagem de todas as *leads* submetidas para a plataforma procedentes dos quiosques e dos balcões;
- **Leads** que representa uma listagem de evolução e de históricos de todos os estados nas quais as *leads* transitaram;
- **Kiosks** onde é possível consultar toda a atividade dos mesmos agrupados por dia.

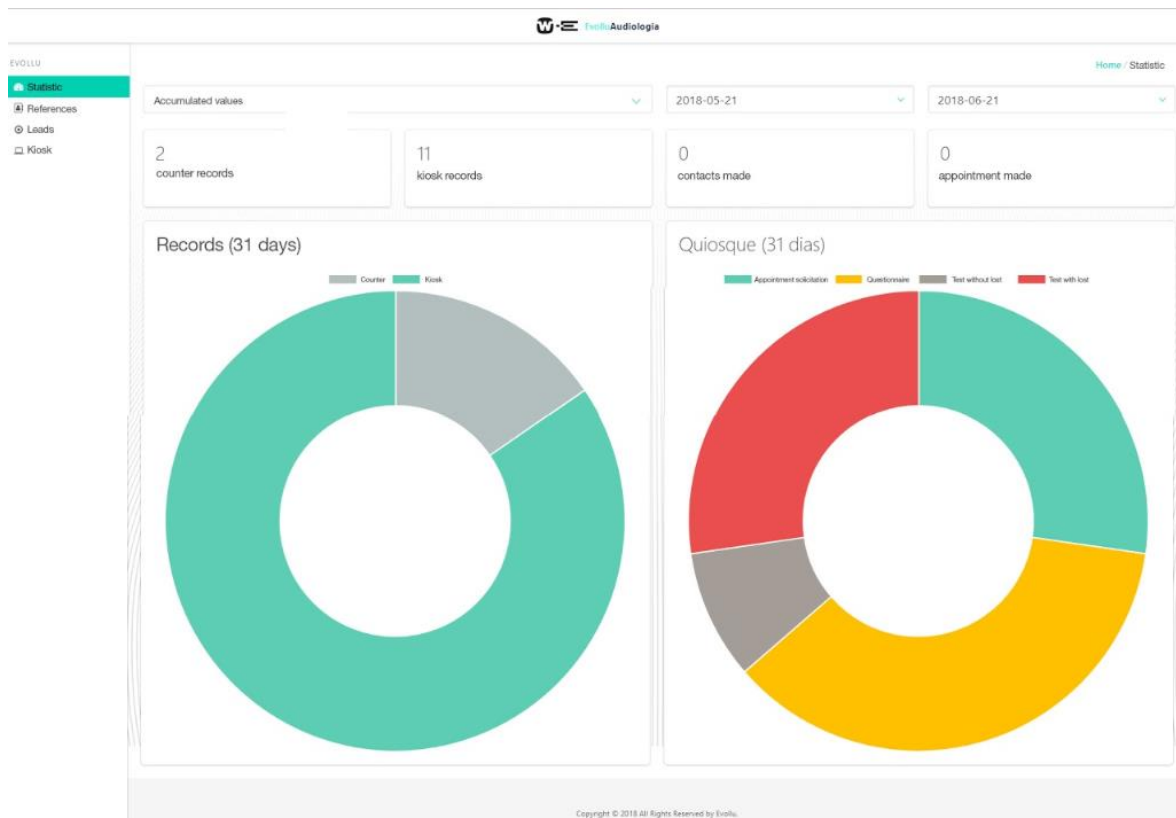


Figura 8 - Plataforma corrente - página de estatísticas

The screenshot displays the 'References' page of the Evollu Audiologia platform. On the left, a sidebar contains navigation icons for 'Statistic', 'References' (highlighted), 'Leads', and 'Kiosk'. The main content area is divided into two sections: a table of existing references and an 'Add' form for new entries.

Name	Phone	E-mail	Entry	State	Info
Maria Isabel Matos	967679407	isabelspeed@hotmail.com	2018-06-17 17:41:30	Kiosk	[Info] [Edit] [Delete]
Beatriz Patricio	910267497		2018-06-18 09:47:26	Kiosk	[Info] [Edit] [Delete]
viktor teste iberia	211111111		2018-06-14 15:03:23	Kiosk	[Info] [Edit] [Delete]
dulcelina	969696939		2018-06-12 23:48:05	Kiosk	[Info] [Edit] [Delete]
Tablet Vitor	214585236		2018-06-11 15:23:52	Kiosk	[Info] [Edit] [Delete]
Hearbase	296325888	mds@hearbase.com	2018-06-08 14:28:26	Kiosk	[Info] [Edit] [Delete]
Frank	232521248		2018-06-06 12:23:15	Kiosk	[Info] [Edit] [Delete]
manuel eanes	931013856	manuel.eanes@gmail.com	2018-05-30 21:56:54	Kiosk	[Info] [Edit] [Delete]
Tiago	913123123		2018-05-30 16:03:04	Counter	[Info] [Edit] [Delete]
showroom Glintt	912304639		2018-05-30 13:31:59	Kiosk	[Info] [Edit] [Delete]
Vitor Jesus	913326831	vitor.jesus@evollu.com	2018-05-30 12:20:25	Kiosk	[Info] [Edit] [Delete]
Sergio Paulo	912304639		2018-05-30 09:43:25	Counter	[Info] [Edit] [Delete]
farmacia colombo	213456789		2018-05-29 15:13:23	Kiosk	[Info] [Edit] [Delete]

The 'Add' form on the right includes the following fields: Name (text input), Phone (text input with a dropdown arrow), E-mail (text input with a dropdown arrow), Pharmacy (text input), Best contact period (text input with a dropdown arrow), and Comments (text area). At the bottom of the form are 'Save' and 'Clean' buttons.

**Figura 9 - Plataforma corrente - página de leads**

Depois de uma análise, conclui-se que a plataforma à data inicial do estágio possuía todas as funcionalidades básicas necessárias de modo a executar as tarefas relativas ao modelo de negócio correspondente referido na secção 2.1.3. É necessário realçar que a plataforma *Evollu* encontrava-se, na altura, numa fase piloto, em que apenas operava em 10 farmácias, todas situadas na região de Lisboa.

Pode-se verificar que a plataforma tinha algumas limitações em alguns aspetos, como a nível de UI (*User Interface*) e apresentação de informação, na medida em que a mesma não evidenciava um *design* intuitivo, o que não permitia uma utilização de forma simples e eficaz.

Uma das tarefas iniciais requeridas foi a migração de todas as funcionalidades existentes com algumas alterações para um *design* mais elaborado. Foi pedido que fosse utilizado um *template* base que permitisse assegurar uma maior consistência a nível de *design*.

Pretendia-se também, que no futuro a plataforma fosse capaz de gerir a angariação de *leads*, e ainda ser autónoma ao ponto de ser possível realizar todo o acompanhamento de utentes através do agendamento de consultas, o respetivo acompanhamento, assim como questões relativas a faturação e licenças.

## 3. Metodologia de Desenvolvimento

Neste capítulo será descrita a metodologia de desenvolvimento utilizada no decorrer da elaboração da plataforma de audiologia na *Evollu*.

### 3.1. *Scrum*

Nas últimas décadas verificou-se uma diminuição da utilização dos métodos tradicionais de gestão da elaboração de projetos, privilegiando desta forma as metodologias ágeis.

O Desenvolvimento Ágil de Software (DAS) foi introduzido formalmente na comunidade de engenharia do software em 2001. A rápida capacidade de adaptação a requisitos voláteis é a base das metodologias ágeis de desenvolvimento de software. Esta agilidade é um aspeto de forte contraste com as abordagens orientadas por um plano prescrito, segundo modelos de desenvolvimento de software mais tradicionais, como é o *waterfall-style*. Outras características distintivas do DAS incluem: (1) o foco nos aspetos humanos e sociais da engenharia de software, nomeadamente a satisfação do cliente; (2) a maior colaboração entre os clientes e a equipa; e (3) o enfoque na frequente entrega de resultados [20].

A metodologia de suporte ao desenvolvimento desta plataforma é uma metodologia baseada em *Scrum*. O *Scrum* caracteriza-se por uma metodologia ágil usada para gestão e desenvolvimento de projetos que promovem a flexibilidade, visibilidade e produtividade. É uma metodologia interativa e incremental o que permite, caso seja necessário, rever o planeamento, mudar de direção ou fazer adaptações nas interações seguintes, e ainda tem como característica atentar às prioridades das funcionalidades para o cliente [21].

Na *Evollu*, as *sprints* correspondem a um determinado intervalo de tempo (uma semana), dentro do qual um conjunto de tarefas deve ser realizado. Normalmente as *sprints* devem possuir a mesma duração ao longo da realização de projetos, para que haja sempre uma data de início e fim bem definidos.

Existem três funções distintas no *Scrum* que correspondem à *Scrum Team* sendo elas (1) *Product Owner*; (2) *Scrum Master*; e (3) equipa de desenvolvimento. O *Product Owner* é o responsável por identificar as funcionalidades a implementar, assim como a prioridade com que as mesmas devem ser implementadas em cada uma das *sprints*, em conjunto com todas as pessoas envolvidas no projeto. O *Scrum Master* é o responsável por assegurar que o

processo de *Scrum* é realizado, e por garantir ou minimizar fatores externos que possam impedir a equipa de desenvolvimento de não atingirem os seus objetivos em cada um dos ciclos. Por fim, a equipa de desenvolvimento é responsável por implementar as funcionalidades definidas pelo *Product Owner* em cada uma das *sprints* [22].

Numa fase inicial do projeto, o *Product Owner* define uma lista de funcionalidades ordenadas por prioridade, designada de *Product Backlog*. No início de cada *sprint* é realizado um *Sprint Planning Meeting*, que corresponde a uma reunião na qual toda a equipa está presente, em que um *Sprint Backlog* é definido, ou seja, todas as tarefas que irão ser implementadas ao longo da *sprint* atual [23].

Em cada dia da *sprint* a equipa de desenvolvimento em conjunto com o *Scrum Master* realiza uma reunião diária, denominada de *Daily Scrum*. Esta tem como objetivo disseminar conhecimento sobre o que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho a ser realizado no dia que se inicia [24].

No final de cada *sprint* é realizada uma reunião designada de *Sprint Review Meeting* com toda a equipa e clientes, em que o trabalho efetuado ao longo da *sprint* é validado por todos os intervenientes [25].

Todas as etapas pertencentes ao *Scrum* mencionadas anteriormente podem ser observadas através da Figura 10.

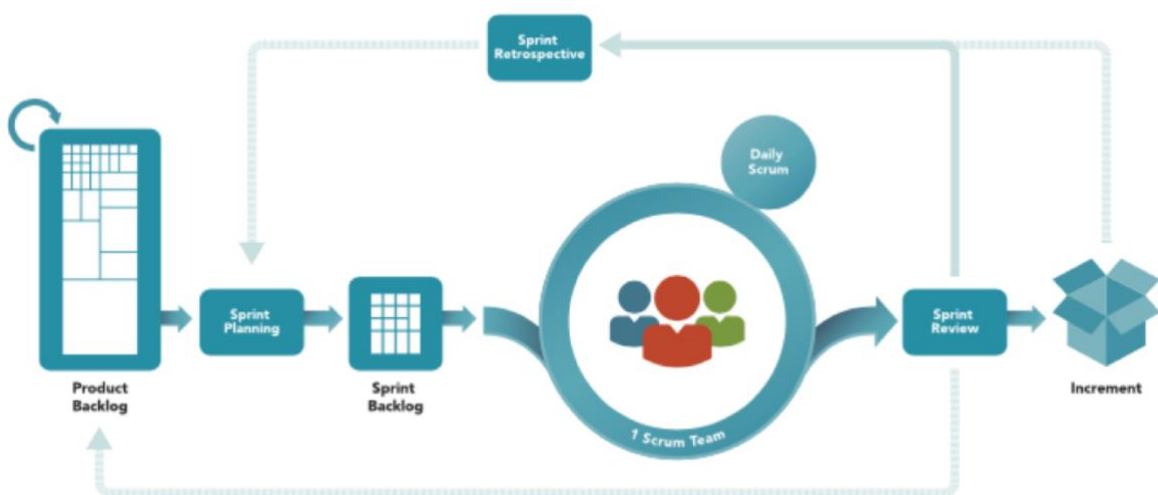


Figura 10 - Processo de Scrum [26]

Na realização da plataforma de audiologia foram realizadas reuniões semanais às segundas-feiras com toda a equipa de desenvolvimento em conjunto com o *Scrum Master*. O papel de *Scrum Master* foi desempenhado pelo CTO (*Chief Technology Officer*) da *Evollu* e supervisor de estágio, Luís Marcelino. O principal objetivo destas reuniões foi verificar o trabalho realizado durante a *sprint* anterior, se todos os requisitos planeados foram cumpridos e definir novas tarefas para realizar na *sprint* seguinte. Cada uma das tarefas possui um custo previsto em horas de esforço, custo este que era definido entre o *Scrum Master* e a pessoa que iria implementar a tarefa correspondente, usualmente eram atribuídas tarefas até perfazer o custo de 35h semanais de trabalho. O papel de *Product Owner* foi realizado pelo CEO (*Chief Executive Officer*) da *Evollu*, Sérgio Paulo, com o qual também existiram inúmeras reuniões presenciais ao longo do desenvolvimento da plataforma. Foram realizadas ainda reuniões sempre que surgisse a necessidade, para o esclarecimento de eventuais dúvidas ao longo do desenvolvimento da plataforma.

Ao longo do estágio curricular foram experimentadas várias plataformas para gestão de projeto, de modo a adotar a que se adaptasse melhor à metodologia de desenvolvimento utilizada na *Evollu* e, a que fornecesse mais funcionalidades úteis, como por exemplo, a possibilidade de quantificar semanalmente todo o trabalho desenvolvido através de relatórios e cronogramas.

As plataformas experimentadas foram o *Pódio* e o *Asana*, sendo que a última foi escolhida devido a um maior número de funcionalidades disponíveis associadas a um menor preço por utilizador, sendo que possui um plano gratuito para equipas até 15 membros, o que se encontra dentro das características da *Evollu*.

O *Asana*, de forma simples, permite realizar toda a gestão de projetos incluindo o *tracking* de tarefas, convertendo objetivos em planos realizáveis e possibilitando a criação de relatórios onde é possível perceber se os prazos impostos vão ser cumpridos.

## 4. Desenvolvimento

Este capítulo descreve o desenvolvimento dos projetos ao longo do estágio. Começa por descrever os objetivos pretendidos para a plataforma *Evollu* (Secção 4.1), de seguida é descrito o levantamento de requisitos (Secção 4.2), de seguida é descrita a arquitetura atual de toda a plataforma (Secção 4.3), as tecnologias utilizadas no desenvolvimento da mesma (Secção 4.4), o processo e as decisões tidas em conta nas funcionalidades implementadas (Secção 4.5). Estas atividades envolveram: criação de um servidor de autenticação, (Secção 4.5.1), a correção de falhas e o desenvolvimento de novas funcionalidades na plataforma de *Leads* (Secção 4.5.2), a realização do serviço de Agendamento (Secção 4.5.3) e, por fim, de Registos Clínicos (Secção 4.5.4).

### 4.1. Plataforma *Evollu*

Desde uma fase inicial, a *Evollu* pretendia alargar as funcionalidades expostas na plataforma existente, na medida em que possibilitasse, tanto à *Evollu* como a outras empresas utilizadoras da plataforma usufruírem das mesmas para uma gestão de processos internos. Essas funcionalidades passavam por realizar o angariamento de *leads* de inúmeros canais distintos, permitir o agendamento de consultas em determinadas instalações pertencentes às empresas que providenciassem serviços de audiologia, e por fim, de realizar o acompanhamento de todos os seus utentes através da realização de consultas nesta área de especialização.

Deste modo, a Figura 11 representa um esquema da plataforma a implementar, onde é possível identificar os micro serviços a desenvolver, de maneira a tornar a plataforma capaz de realizar todos os pontos indicados anteriormente.

Ao observar a Figura 11, é possível identificar que a implementação de uma arquitetura baseada em micro serviços adequa-se aos objetivos da plataforma, uma vez que é possível distinguir vários serviços de um modo imediato. O serviço de *leads* que já se encontra parcialmente implementado, um serviço de agendamento responsável pela marcação de consultas em farmácias, e o serviço de registos clínicos de audiologia capaz de gerir e armazenar toda a informação necessária acerca dos dados clínicos dos utentes. Por fim, mas não menos importante, um serviço de autenticação capaz de autenticar os diferentes utilizadores e interligar todos estes serviços.

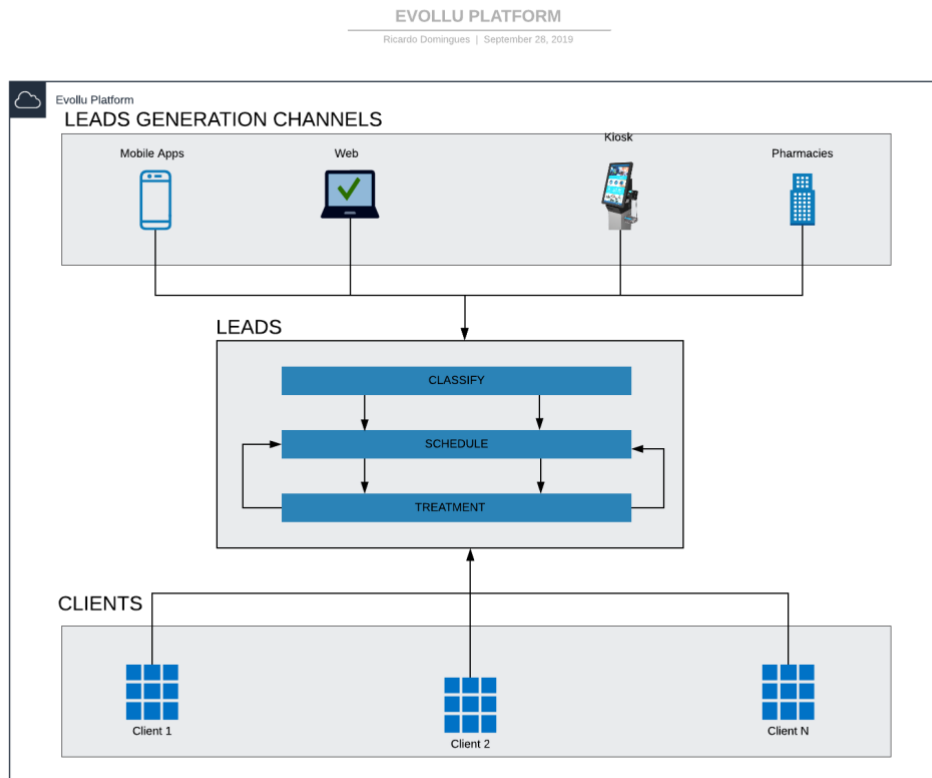


Figura 11 - Esquema geral plataforma *Evollu*

## 4.2. Levantamento de Requisitos

Após um estudo da plataforma atual do modelo de negócio em que a *Evollu* se enquadra, foi realizado um estudo dos perfis de utilizador e dos requisitos dos mesmos na plataforma. Assim, esta secção começa por especificar os inúmeros perfis de utilizador da plataforma (Secção 4.2.1) e, por fim, é detalhado uma listagem de requisitos funcionais e não-funcionais relativo a estes utilizadores (Secção 4.2.2).

### 4.2.1. Perfis de Utilizador

Após uma análise dos principais objetivos que a plataforma *Evollu* pretende solucionar, foi possível extrair os seguintes perfis de utilizador: administrador, responsável da farmácia, farmácia, *lead*, responsável por promotores, promotor, *audiology provider*, audiologista e utilizador comum que representa qualquer utilizador que se consiga autenticar na plataforma.

**Administrador**

Um utilizador “Administrador”, como o próprio nome indica, poderá administrar toda a plataforma. Este perfil está diretamente associado a membros da *Evollu*, onde a principal função dos mesmos é dar suporte a todos os outros utilizadores existentes na plataforma.

**Responsável Farmácia**

Um utilizador “Responsável Farmácia” deverá, por norma, conseguir aceder a todas as informações das suas farmácias como as *leads* geradas, requisitar rastreios auditivos e realizar toda a gestão de utilizadores da respetiva farmácia.

**Farmácia**

Um utilizador “Farmácia” é um membro pertencente a uma farmácia. Este deverá, por norma, visualizar toda a informação associada à sua farmácia, assim como gerir todo o processo de agendamento de rastreios, que conta com a marcação de utentes para os mesmos.

**Lead**

Um utilizador do tipo “Lead” tem apenas autorização para utilizar as várias aplicações móveis fornecidas pela *Evollu*. Este poderá gerir toda a informação do seu perfil de utilizador, assim como ter acesso a uma listagem de rastreios auditivos próximos da sua zona de residência, de onde poderá marcar-se como disponível para os mesmos. Todas as marcações realizadas por estes ficam pendentes até confirmação.

**Responsável Promotor**

Um utilizador “Responsável Promotor” pertence a uma entidade externa à *Evollu*. Estas entidades externas são contratadas pela *Evollu*, com o intuito de disponibilizarem promotores para a realização de rastreios auditivos. Assim, este tipo de utilizador, em conjunto com a equipa de suporte da *Evollu*, poderá gerir quais os promotores que irão efetuar cada rastreio auditivo. Toda a informação relativa às *leads* geradas nos seus rastreios deve ser apresentada de forma anonimizada, uma vez que representam informação de utentes reais, possuindo assim um carácter confidencial.

**Promotor**

Um utilizador “Promotor” tem um acesso restrito na plataforma. Este poderá apenas consultar quais os rastreios auditivos aos quais se encontra atribuído, assim como consultar as *leads* das suas sessões de rastreios de forma anonimizada como mencionado acima.

### ***Audiology Provider***

Um utilizador “*Audiology Provider*” pertence a uma entidade que adquire as *leads* geradas em quiosques físicos e em rastreios realizados nas farmácias. Posteriormente, estes utilizadores irão recorrer à plataforma para registar a evolução do estado de todas estas.

### **Audiologista**

Um utilizador “Audiologista” tem acesso a todas as suas consultas agendadas e a todos os dados dos seus utentes, isto é, poderá consultar todas as consultas agendadas e o historial dos mesmos, com o objetivo de possibilitar a estes um acompanhamento personalizado.

#### **4.2.2. Lista de Requisitos**

A Tabela 1 representa apenas parte das tarefas desenvolvidas onde uma listagem mais detalhada com a previsão e custo de cada uma das tarefas pode ser consultada no Anexo I. Os requisitos são listados agrupados pelo tipo de serviço onde se encontram, de realçar que a ordem é completamente aleatória não expressando qualquer tipo de importância.

**Tabela 1 - Listagem de requisitos principais da plataforma *Evollu***

Descrição da Funcionalidade		Nível de Importância	Tipo de Utilizador
Requisitos Gerais			
RF0001	Um utilizador pode autenticar-se na plataforma.	Alta	Utilizador Comum
RF0002	Um utilizador pode recuperar a sua palavra-passe.	Média	Utilizador Comum
RF0003	Um utilizador consegue visualizar a sua página de perfil com todos os seus dados de utilizador, permissões que tem em cada	Baixa	Utilizador Comum

	entidade assim como configurações adicionais.		
RF0004	Um utilizador pode filtrar tabelas e gráficos da plataforma tendo em conta entidades, agrupamento de entidades, intervalos de tempo e pesquisa por texto.	Média	Utilizador Comum
RF0005	Um utilizador ao utilizar filtros, os seus dados devem ser persistidos localmente para que sejam restaurados em futuras utilizações.	Baixa	Utilizador Comum
RF0006	Todos os ecrãs da plataforma devem ser o mais responsivos possíveis, para garantir uma utilização natural em ecrãs de dimensões menores.	Média	Utilizador Comum
RF0007	Toda a plataforma deve estar disponível nos idiomas Português, Inglês e Espanhol.	Alta	Utilizador Comum
Serviço de Agendamento			
RF0101	Um utilizador do tipo <i>Evollu</i> pode criar, editar e remover disponibilidades de farmácia para uma dada entidade.	Alta	Administrador
RF0102	Um utilizador do tipo <i>Evollu</i> ou de uma entidade que exerce uma atividade deve conseguir associar, editar e remover um profissional de saúde a uma disponibilidade de farmácia.	Alta	Administrador
RF0103	Um utilizador do tipo Farmácia pode criar, editar e remover marcações associadas a uma disponibilidade de profissional de saúde.	Alta	Responsável Farmácia; Farmácia
RF0104	Um utilizador pode criar, editar e remover disponibilidades de farmácia e de profissionais de saúde recorrendo a frequências de recorrências.	Média	Utilizador Comum
RF0105	Um utilizador <i>Evollu</i> pode consultar uma classificação das farmácias existentes, para	Alta	Administrador

	vários critérios acerca de todos os rastreios auditivos realizados em farmácias.		
RF0106	Um utilizador <i>Evollu</i> pode consultar uma classificação dos promotores existentes, para vários critérios acerca de todos os rastreios auditivos realizados em farmácias.	Alta	Administrador
RF0107	Um utilizador <i>Evollu</i> ou Promotor deve conseguir enviar o código QR associado a uma sessão de rastreio auditivo.	Média	Administrador; Responsável Promotor; Promotor
RF0108	Um utilizador Promotor deve poder exportar todos os seus rastreios auditivos para o seu calendário pessoal.	Média	Administrador; Promotor
RF0109	Um utilizador Responsável Promotor e Promotor deve conseguir visualizar uma listagem anonimizada de todas as <i>leads</i> provenientes de cada rastreio auditivo.	Baixa	Responsável Promotor; Promotor
RF0110	Como utilizador da plataforma, quero conseguir fazer toda a gestão de disponibilidades e marcações através de uma agenda com as vistas do tipo dia, semana e mês.	Alta	Utilizador Comum
RF0111	Um utilizador da Aplicação Móvel deve conseguir aceder a uma listagem de rastreios auditivos mais próximos de si.	Média	<i>Lead</i>
RF0112	Um utilizador da Aplicação Móvel deve conseguir aceder a uma listagem de <i>slots</i> disponíveis para um dado rastreio.	Média	<i>Lead</i>
RF0113	Um utilizador da Aplicação Móvel deve conseguir criar, listar, alterar e remover marcações associadas a um rastreio auditivo.	Média	<i>Lead</i>
Serviço de autenticação			

RF0201	Um utilizador pode consultar uma listagem de entidades a que tem acesso.	Alta	Utilizador Comum
RF0202	Um utilizador ao selecionar uma entidade pode consultar as farmácias, quiosques e utilizadores da mesma.	Alta	Utilizador Comum
RF0203	Um utilizador pode criar, associar, alterar e remover uma farmácia de uma entidade.	Média	Utilizador Comum
RF0204	Um utilizador com permissões pode criar, alterar e remover salas associadas a uma farmácia.	Baixa	Utilizador Comum
RF0205	Como utilizador com acesso a determinada farmácia, quero poder visualizar todas os serviços realizados pela mesma.	Baixa	Utilizador Comum
RF0206	Um utilizador pode visualizar e alterar os dados relativos aos quiosques da sua entidade.	Baixa	Utilizador Comum
RF0207	Um utilizador, pode criar, associar, alterar e remover utilizadores de uma entidade.	Alta	Utilizador Comum
RF0208	Um utilizador ao consultar os detalhes de outro utilizador pode visualizar os detalhes do mesmo, assim como todas as permissões e atividades que este exerce em cada entidade.	Alta	Utilizador Comum
RF0209	Um utilizador ao consultar as funções de outro utilizador, é-lhe dada a possibilidade de visualizar e alterar estas, agrupando as informações tanto por entidade como por tipo de função.	Alta	Utilizador Comum
RF0210	Um utilizador ao alterar as funções de outro utilizador por tipo de função pode atribuir entidades utilizando um agrupamento de entidades.	Baixa	Utilizador Comum
RF0211	Um utilizador <i>Evolu</i> deve conseguir criar, listar, editar e remover agrupamentos que representem um aglomerado de entidades.	Média	Administrador

RF0212	Um utilizador <i>Evollu</i> deve conseguir associar determinado agrupamento a outros utilizadores da plataforma.	Baixa	Administrador
RF0213	Um utilizador <i>Evollu</i> deve conseguir criar, visualizar, editar e remover <i>roles</i> de todos os micro serviços existentes.	Média	Administrador
RF0214	Um utilizador <i>Evollu</i> deve conseguir visualizar todas as <i>subroles</i> associadas a uma <i>role</i> , assim como associar novas <i>roles</i> à mesma.	Média	Administrador
Serviço de <i>Leads</i>			
RF0301	Um utilizador <i>Lead</i> ao registar-se num balcão de uma farmácia deve poder receber um email, uma <i>sms</i> ou um papel impresso de forma a que possa fornecer o seu consentimento, de forma a ser contactado por <i>call-centers</i> .	Média	<i>Lead</i>
RF0302	Um utilizador <i>Lead</i> ao efetuar o teste auditivo num quiosque deve receber um email com o resultado do seu teste auditivo.	Média	<i>Lead</i>
RF0303	Um utilizador Farmácia, ao criar uma <i>lead</i> de balcão deve conseguir visualizar uma indicação, se a <i>lead</i> já forneceu consentimento para que seja contactado.	Baixa	Responsável Farmácia; Farmácia
RF0304	Um utilizador deve poder criar, listar, editar e remover etiquetas associadas a uma cor.	Baixa	Utilizador Comum
RF0305	Um utilizador deve poder indicar se uma etiqueta criada pelo mesmo pode ser visualizada por todos os membros da sua entidade ou apenas pelo próprio.	Baixa	Utilizador Comum
RF0306	Um utilizador deve conseguir associar <i>leads</i> a uma ou várias etiquetas.	Baixa	Utilizador Comum

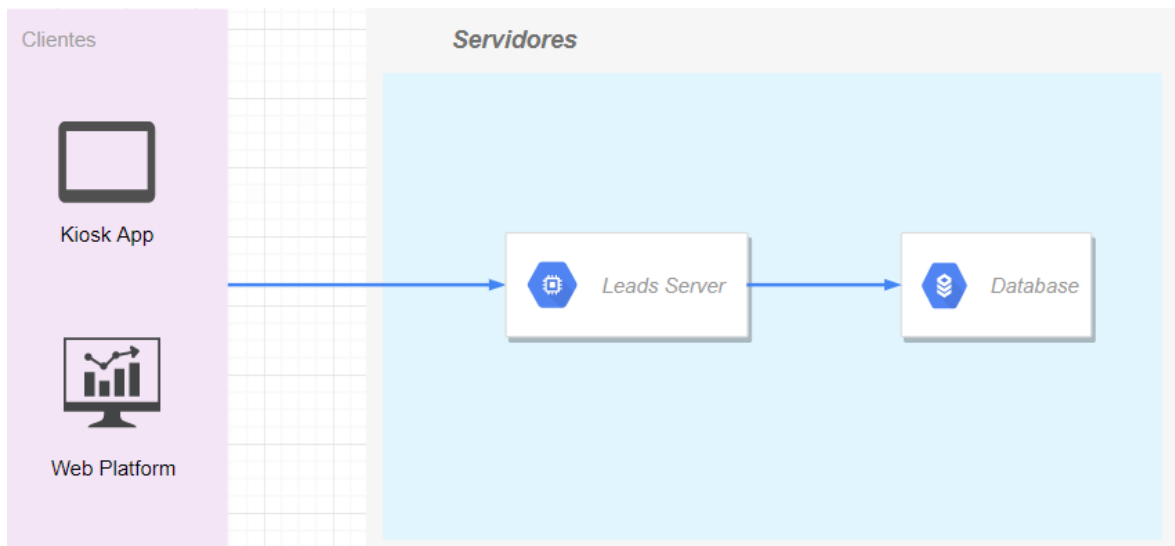
RF0307	Um utilizador deve poder filtrar a listagem de <i>leads</i> existentes por uma ou várias etiquetas.	Baixa	Utilizador Comum
RF0308	Um utilizador deve poder visualizar todas as entidades das quais deseja receber notificação via email, das <i>leads</i> criadas nas mesmas.	Média	Utilizador Comum
RF0309	Um utilizador deve conseguir visualizar a listagem de quiosques da qual a sua entidade é proprietária, assim como os que lhe estão afetos.	Baixa	Utilizador Comum
RF0310	Um utilizador deve conseguir aceder a uma listagem de regras de notificação, que consistem nos critérios utilizados no envio de emails de novas <i>leads</i> .	Baixa	Utilizador Comum
Serviço de Registos Clínicos			
RF0401	Um utilizador Audiologista deve conseguir visualizar uma listagem de todos os pacientes a que tem acesso.	Média	Audiologista
RF0402	Um utilizador Audiologista deve conseguir visualizar o histórico de marcações de um determinado paciente.	Média	Audiologista
RF0403	Um utilizador Audiologista deve conseguir criar, alterar, remover e visualizar uma listagem de aparelhos auditivos pertencentes a um dado paciente.	Alta	Audiologista
RF0404	Um utilizador Audiologista deve conseguir criar, listar, editar e remover anamneses a um dado paciente.	Média	Audiologista
RF0405	Um utilizador Audiologista deve conseguir criar, listar, editar e remover registos clínicos de um determinado paciente.	Alta	Audiologista
RF0406	Um utilizador Audiologista deve poder visualizar todo o histórico de testes auditivos	Baixa	Audiologista

	realizados por determinado paciente provenientes da aplicação <i>Evollu</i> .		
--	---	--	--

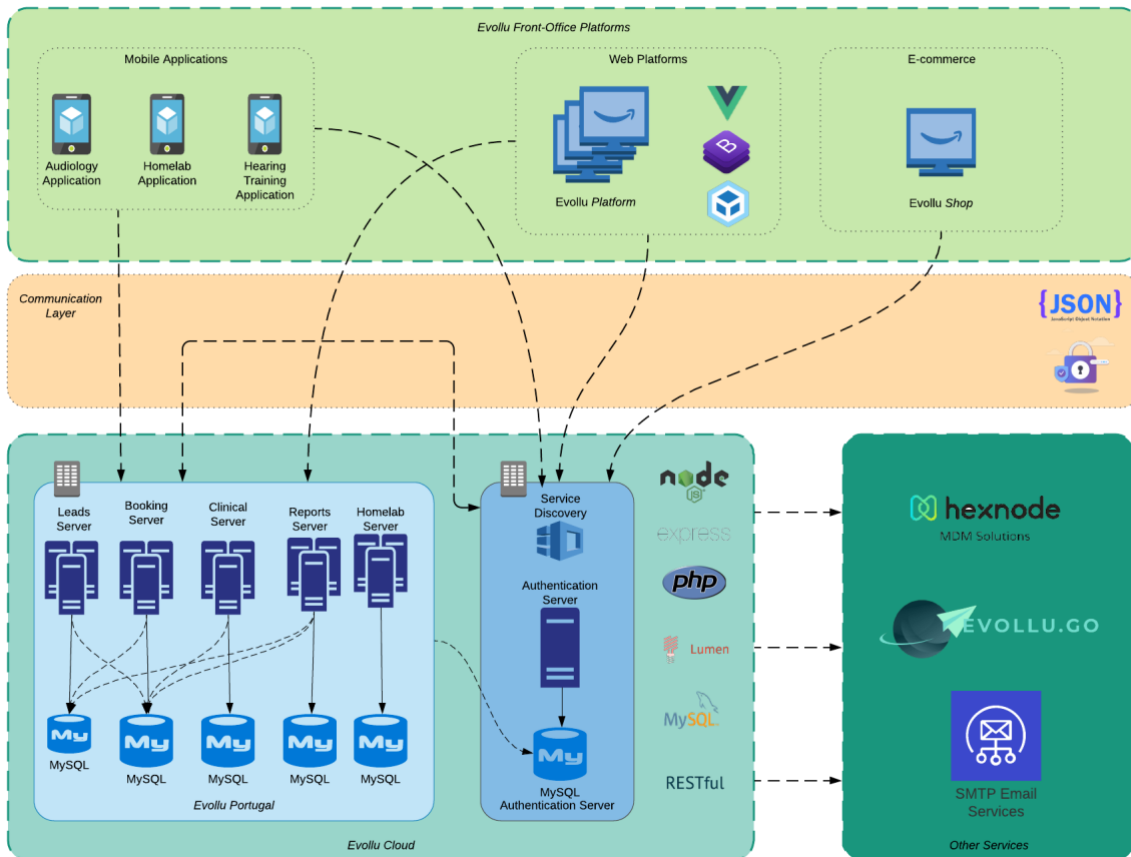
### 4.3. Arquitetura da Plataforma *Evollu*

À data inicial do estágio, a *Evollu* era constituída por uma arquitetura monolítica, onde a mesma era responsável por autenticar e autorizar utilizadores a aceder ou a criar recursos. Este tipo de arquitetura não era tão escalável nem adequada à visão que a *Evollu* teria da sua plataforma futura, visto que um dos objetivos da mesma passava por criar uma plataforma, que integrasse vários serviços associados à saúde com foco na área da audiolgia.

A Figura 12 e Figura 13 representam, respetivamente, as diferenças arquiteturais de todo o ecossistema *Evollu* desde a data inicial do estágio (01 de agosto de 2018) até à presente data.



**Figura 12 - Arquitetura inicial plataforma *Evollu***



**Figura 13 - Arquitetura final ecossistema *Evollu***

Na Figura 12 é possível observar uma estrutura monolítica, em que apenas existem duas aplicações que interagem com o servidor de *Leads*, sendo elas a aplicação quiosque, a aplicação que funciona como angariador de *leads* através do teste de audição concebido na *Evollu*; e a plataforma Web, onde é possível consultar as *leads* que foram criadas em cada um dos quiosques, assim como as respostas fornecidas durante o questionário e detalhes específicos sobre o teste auditivo.

Já a Figura 13, representa a arquitetura atual (31 de julho de 2019), onde é possível observar que, com a quantidade de serviços existentes, a arquitetura seria bastante complexa, se seguisse uma abordagem monolítica.

É possível observar ainda que, a camada de apresentação é constituída por: diversas aplicações móveis, a plataforma e uma loja online. Existem várias versões da plataforma disponíveis, mais concretamente uma para cada país, onde a mesma é comercializada, em que as diferenças consistem em diversas configurações de linguagens padrão, entre outras características.

A camada do transporte/comunicação é realizada de forma segura, devido à utilização do protocolo HTTPS (*Hypertext Transfer Protocol Secure*) e todas estas utilizam JSON (*Javascript Object Notation*) como formato de dados.

Por fim, a camada do servidor é constituída por:

- Inúmeros *clusters* de micro serviços específicos de audiologia, sendo eles o serviço de *leads*, agendamento, registos clínicos e relatórios. É de realçar que todos estes serviços e respetivas bases de dados são replicados nos vários países onde a *Evollu* opera. Assim, tanto o Reino Unido, como a Espanha possuem um conjunto de todos estes servidores e respetiva plataforma Web.
- Um único servidor de autenticação que atua igualmente como *service discovery*, uma vez que todos os micro serviços desenvolvidos da *Evollu* são-lhe conhecidos. Assim, para um utilizador que aceda à plataforma no Reino Unido, o servidor de autenticação é responsável por indicar qual o conjunto de serviços que este deve utilizar.
- O micro serviço *Homelab* diz respeito a uma nova atividade que a *Evollu* se encontra a integrar na sua plataforma, responsável por agilizar o processo de realização de análises clínicas ao domicílio, em Portugal.
- O *Hexnode* é um serviço que permite controlar remotamente todos os quiosques pertencentes à *Evollu*, possibilitando a criação de *policies* e restrições na utilização dos mesmos, para que seja assegurado, como, por exemplo, que a aplicação *Evollu* esteja sempre em funcionamento nos *tablets*, e que a versão desta se encontre atualizada sem intervenção humana, a todo o momento.
- Um conjunto de bases de dados, sendo que cada micro serviço possui apenas uma. Apesar disso, é possível a um micro serviço efetuar a leitura a bases de dados de outros serviços, para que possa facilitar a interdependência entre os mesmos.
- Outros serviços como: (1) uma API (*Application Programming Interface*) interna desenvolvida, denominada por *Evollu GO*, que consiste num URL (*Uniform Resource Locator*) *shortener*, isto é, permite a transformação de um URL extenso num outro de apenas 6 caracteres; (2) uma loja *e-commerce* que possui autenticação *Evollu*, e onde são vendidos artigos relacionados com a manutenção de aparelhos auditivos, como pilhas, borrachas, entre outros.

A Figura 14 demonstra o esquema dos micro serviços necessários para gerir todo o processo envolvido desde a criação de uma *lead* através dos quiosques, até ao agendamento das respetivas consultas de acompanhamento, sendo que os micro serviços intervenientes em todo o processo são:

- Serviço de autenticação: Responsável por autenticar os utilizadores encarregues de desempenhar as funções necessárias na plataforma, assim como expor todos os outros micro serviços.
- Serviço de *Leads*: Responsável por agregar todas as *leads* provenientes dos vários métodos de aquisições das mesmas, assim como o processo necessário desde que uma *lead* é submetida até à marcação de uma consulta.
- Serviço *Booking*: Responsável por agendar a marcação de consultas de pacientes para um audiolologista, e também pela marcação de rastreios auditivos em farmácias.
- Serviço Clínico: Responsável por armazenar as informações provenientes em consultas de audiologia.
- Serviço Relatórios: Responsável por gerar inúmeros relatórios de faturação às várias entidades intervenientes na realização de rastreios auditivos. De realçar, que este micro serviço não irá ser abordado durante todo o documento, uma vez que a sua implementação foi realizada por outros elementos da empresa.

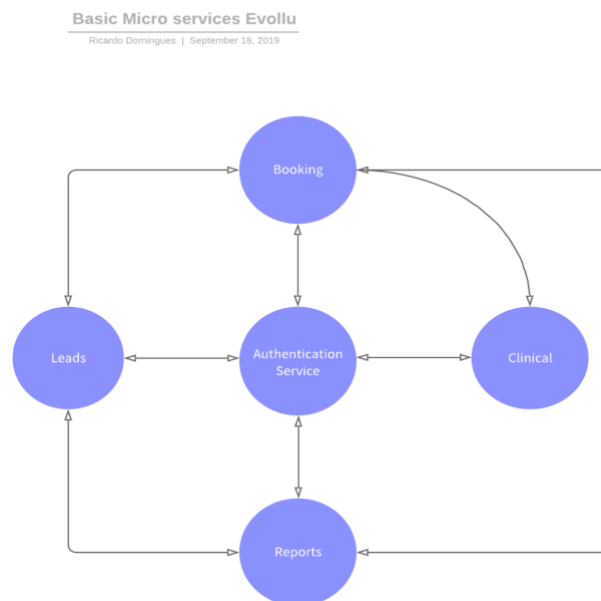


Figura 14 - Organização dos micro serviços no processo de Audiologia

Desta forma, o esquema em micro serviços descrito anteriormente torna-se bastante vantajoso, uma vez que todo o processo, desde a aquisição de possíveis utentes com dificuldades auditivas até ao tratamento dos mesmos, é composto por vários módulos independentes, onde cada um tem as suas responsabilidades delineadas.

Para além disso, cada módulo pode ser comercializado individualmente, visto que por vezes as empresas utilizadoras da plataforma já possuem os seus sistemas internos, tendo interesse apenas em módulos específicos.

Ainda assim, a arquitetura implementada traz inúmeras complicações no desenvolvimento de software. Alguns dos desafios encontrados no desenvolvimento de uma arquitetura em micro serviços foram: (1) delimitação de fronteiras; (2) gestão de permissões; e (3) comunicação entre micro serviços.

Delimitar as fronteiras poderá ser uma tarefa bastante árdua, visto que torna-se complicado decidir quais as corretas responsabilidades de cada serviço existente [27]. Por vezes é necessária uma reflexão sobre o serviço, onde determinada funcionalidade, que é inerente a múltiplos micro serviços, deve ser introduzida.

Durante a fase de implementação, este tipo de problema ocorreu inúmeras vezes e, causou um atraso significativo no desenvolvimento da plataforma. Um exemplo concreto foi, que inicialmente, toda a gestão de pacientes que envolve vários dados distintos como moradas dos mesmos, pessoas de contacto em caso de emergência e cartões de saúde, foi desenvolvida no micro serviço responsável pelo agendamento de consultas. Mais tarde, com a introdução de novos serviços e funcionalidades, foi necessário mover toda esta lógica de negócio para o micro serviço de autenticação., Uma vez que este foi desenvolvido em *NodeJS* foi necessário reescrever todo o código de PHP para *JavaScript*.

No que diz respeito à gestão de permissões, numa fase inicial, todas as permissões de utilizadores encontravam-se no serviço de autenticação, o que acabou por se tornar dificultoso à medida que foram adicionados mais micro serviços em todo o ecossistema. Rapidamente, se percebeu que a gestão de permissões deveria ser ao nível do próprio serviço, pois o serviço de autenticação não tem como conhecer todas as diferentes permissões existentes nos inúmeros serviços existentes.

Assim, todas estas mudanças em conjunto com as alterações referidas anteriormente relativas à gestão de pacientes coincidiram e foram resolvidas em simultâneo, o que levou a

que estas alterações perdurassem aproximadamente dois meses, dado que impuseram mudanças drásticas ao nível da arquitetura da plataforma.

Por fim, a comunicação entre serviços é algo que neste tipo de padrão arquitetural se torna um requisito fundamental, visto que o processo de transição de uma *lead* para um agendamento e respetivas consultas representam vários módulos distintos da plataforma.

Nos primeiros meses de desenvolvimento, os vários micro serviços existentes conseguiam escrever diretamente na base de dados de outros serviços, isto acabou por facilitar todo o processo de desenvolvimento. Rapidamente se observou que a abordagem seguida não ia de encontro com as boas práticas de desenvolvimento em micro serviços, dado que poderia tomar efeitos indesejados [27]. Um dos exemplos foi a criação de um utilizador a partir de um serviço sem ser o de autenticação, em que todas as etapas associadas à criação do mesmo não eram despoletadas. Algumas das etapas passavam por o envio de um *welcome* email, a atribuição de permissões e outras notificações envolvidas no processo, o que resultava em inconsistências nos micro serviços.

Deste modo, decidiu-se que a melhor abordagem seria a criação de novos *endpoints* nas API dos serviços para a comunicação entre estes. Apesar desta solução solucionar o problema descrito, existem outras soluções como a utilização de comunicações baseadas em eventos assíncronos, visto que a utilização de pedidos HTTP (*Hypertext Transfer Protocol*) é considerada *fault tolerant*, dado que se um dos serviços responder com erro, todas as comunicações seguintes falham.

## **4.4. Principais Tecnologias Utilizadas**

Neste capítulo são descritas as principais linguagens e tecnologias utilizadas no desenvolvimento da plataforma subdivididas entre cliente e servidor.

### **4.4.1. Servidor**

A Figura 15 ilustra as diferentes tecnologias utilizadas, até ao momento, na implementação dos vários micro serviços existentes na plataforma *Evollu*.



Figura 15 - Linguagens e tecnologias utilizadas nos micro serviços

O servidor de autenticação utiliza um conjunto de tecnologias pertencentes ao ecossistema de *NodeJS*. Deste modo, este micro serviço foi implementado na linguagem *Javascript*, em que foi empregue uma *framework* denominada de *express*. Para lidar com acessos à base de dados foi utilizado um ORM (*Object Relational Mapping*) denominada de *sequelize*, que retrata o ORM mais maturo no respetivo ecossistema.

O *NodeJS* é um interpretador *open-source* de código *JavaScript* de natureza assíncrona e orientado a eventos, isto é, ao utilizar um ciclo de eventos, o *node* interpreta, numa única *thread*, todos os pedidos de forma assíncrona ao invés do processo sequencial tradicional, que por sua vez impede que hajam possíveis bloqueios na execução de código [28]. Assim, ao realizar uma transação à base de dados ou ao ler um ficheiro, este não espera que a ação conclua e processa a próxima instrução na fila de execução. Quando a mesma é concluída é lançado um evento, e uma função denominada de *callback* é executada como resultado da ação inicial, o que o torna ideal para desenvolver aplicações escaláveis [28].

O *ExpressJS* é a *framework NodeJS* mais popular. Esta é flexível e minimalista, mas mesmo assim providencia um conjunto robusto de funcionalidades para o desenvolvimento de aplicações Web e API [29].

Nos restantes micro serviços foi utilizado uma *stack PHP* com a utilização de uma *framework* minimalista do *Laravel*, denominada de *Lumen*.

O PHP (*PHP: Hypertext Preprocessor*) é uma linguagem *open-source*, que pode ser instalada de forma gratuita em qualquer sistema operativo. Esta é amplamente utilizada e direcionada para o desenvolvimento Web, especificamente para aplicações do lado do servidor. Apesar de muito utilizada, é uma linguagem *weakly typed* que apresenta alguns problemas de consistência [30].

O *Lumen* é uma *framework* PHP, baseada em *Laravel*, que foi desenvolvida com o propósito de construir apenas serviços e API com tempos de resposta extremamente rápidos, sendo uma *framework* PHP muito rápida. O seu criador é Taylor Otwell, a mesma pessoa responsável pela *framework* *Laravel* [31].

O *Lumen* é recomendado para projetos que possuem uma arquitetura baseada em micro serviços, isto é, ao utilizar uma arquitetura em micro serviços obtém-se uma aplicação bastante modular, uma vez que é possível ter cada serviço a correr individualmente, com diferentes tipos de linguagens de programação e/ou diferentes técnicas de armazenamento de dados, o que resulta em aplicações mais flexíveis e escaláveis [32].

#### 4.4.2. Cliente

Como é possível observar pela Figura 16, toda a plataforma da *Evollu* consiste na utilização de *JavaScript*, em que é utilizada a biblioteca de *Vue.js*, com o uso de *Bootstrap 4* para a construção de UI e *Webpack* para compilação e *bundle* de ficheiros estáticos como CSS (*Cascading Style Sheets*) e imagens.



Figura 16 - Linguagens e tecnologias utilizadas no *front-end*

*Vue* é uma *framework* progressiva de *JavaScript* utilizada para construir UI. A biblioteca *core* foca-se na camada da apresentação, sendo uma *framework* bastante fácil de aprender. Apesar disto, esta também é capaz de construir aplicações robustas como é o caso de SPA (*Single Page Application*) [33].

O *Bootstrap* é um *toolkit open-source* que permite a construção de UI responsivas e orientadas a dispositivos móveis, sendo a biblioteca de CSS mais popular. Para além disso, esta biblioteca fornece ao utilizador inúmeros componentes reutilizáveis que permitem implementar rapidamente a interface a partir de protótipos [34].

O *Webpack* é uma ferramenta que permite o *bundle* de módulos estáticos para aplicações *JavaScript* modernas [35]. Ao processar a aplicação, este gera um gráfico que mapeia cada módulo e as respetivas dependências, como mostra a Figura 17.

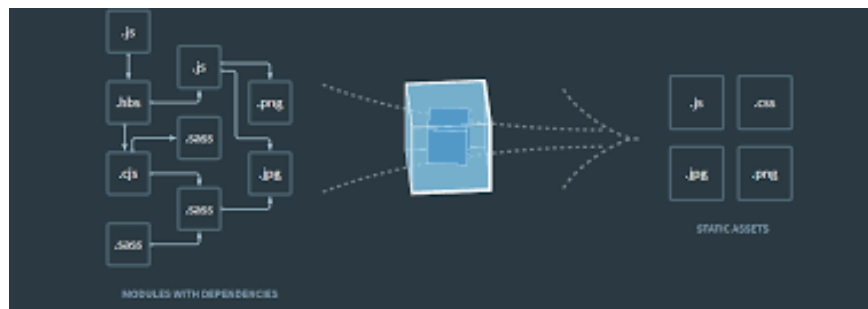


Figura 17 - *Bundle* de dependências utilizando *Webpack* [35]

## 4.5. Atividades desenvolvidas

Esta secção descreve parte das atividades desenvolvidas ao longo do estágio curricular. Começa por descrever o micro serviço de autenticação, responsável pela gestão de toda a informação relativa aos utilizadores e respetivas entidades (Secção 4.5.1), de seguida são abordadas algumas das funcionalidades implementadas no micro serviço já existente (Secção 4.5.2), o micro serviço de agendamento, que possibilita a marcação de consultas e rastreios auditivos (Secção 4.5.3), e por fim, o micro serviço de registos clínicos de audiologia que permite a gestão de toda a informação clínica associada a um utente (Secção 4.5.4).

### **4.5.1. Serviço de Autenticação**

Há alguns anos, quando os computadores representavam sistemas *standalone*, a informação associada aos utilizadores, tanto informações pessoais como credenciais de acesso encontravam-se armazenadas no mesmo local.

Com o passar dos anos, e com o desenvolvimento de novos padrões e tecnologias é possível, hoje em dia, a utilização da mesma informação de login em múltiplas plataformas e aplicações móveis. Isto foi conseguido devido à adoção de padrões em toda a indústria de desenvolvimento de software [36].

Assim, existem inúmeros protocolos que especificam como construir todo o processo de autenticação e autorização em plataformas, referindo-se estes a como gerir a identidade dos utilizadores, como mover dados pessoais de utilizadores pelos vários serviços e decidir quem pode aceder aos mesmos.

Assim alguns dos protocolos de autorização utilizados hoje em dia são o *OAuth1*, *OAuth2*, *Open ID Connect*, *JSON Web Tokens*, *SAML (Security Assertion Markup Language)*. Na secção seguinte irá ser abordado o protocolo *OAuth2.0*, visto ter sido o protocolo utilizado no desenvolvimento do servidor de autenticação da *Evollu*.

#### **4.5.1.1. Processo de Escolha de Tecnologias**

O servidor de autenticação representa uma das componentes mais importantes em todo o ecossistema, uma vez que se encontra totalmente centralizado no mesmo. Inicialmente, o orientador de estágio possibilitou a possibilidade de escolha das tecnologias a utilizar, sendo que devia ser uma entre *Java* e *JavaScript*. Apesar de todos os restantes micro serviços desenvolvidos na *Evollu* se encontrarem em *PHP*, o servidor de autenticação, sendo um serviço central em toda a plataforma, necessita de algumas características que o permita tornar-se numa aplicação escalável.

Assim, de modo a conhecer entre as linguagens exigidas, a que mais se adequa, foram realizados pequenos testes entre as mesmas, de modo a perceber algumas características como a performance, a legibilidade do código, que permite avaliar o quão fácil será com a evolução do projeto de serem introduzidas novas funcionalidades e, por fim, o respetivo *deployment*, ou seja, todas as etapas necessárias para que o servidor fique disponível e acessível a partir da Web.

Os testes consistiram na criação de duas aplicações REST (*Representational State Transfer*), uma em *Java* e outra em *JavaScript* com alguns modelos simples. Os resultados indicaram que em termos de performance, a linguagem de *JavaScript* alcançou melhores ganhos, sendo que para uma simples aplicação demo de CRUD (*Create Read Update Delete*) foi no total 47% mais eficaz numa bateria de 10 testes para cada uma das seguintes ações:

- Listagem de utilizadores existentes com os respetivos *posts* e comentários;
- Inserção de um utilizador;
- Atualização de um utilizador;
- Remoção de um utilizador;
- Inserção de um comentário de um dado utilizador e *post*;
- Atualização de um comentário de um utilizador e *post*;
- Remoção de um comentário de um utilizador e *post*.

Em relação à legibilidade de código, o *Java* é mais conciso e organizado, uma vez que foi utilizada a *framework Spring*, onde existem padrões e estruturas de ficheiros a seguir. Ao contrário, no *JavaScript* existem inúmeras abordagens quanto à própria estrutura do projeto e não enfoca nenhuma prática específica.

Por fim, quanto ao *deployment*, a aplicação de *JavaScript* revelou-se mais simples, talvez devido à experiência adquirida no decorrer de todo o percurso académico, onde apenas com uma simples configuração tanto em *Nginx* e *Apache* (servidores Web), é possível disponibilizar uma aplicação. No *Java* são necessárias mais configurações.

Deste modo, foi tomada a decisão, em conjunto com o supervisor de estágio, de prosseguir com a linguagem *JavaScript*, devido aos resultados favorecerem mais esta linguagem e, também devido aos conhecimentos previamente adquiridos na mesma, durante o percurso académico. Foi selecionada a *framework express* e, inicialmente não foi utilizado nenhum ORM, uma vez que a complexidade das funcionalidades não o justificou. Apenas mais tarde, com o crescimento do projeto e com a introdução de novas funcionalidades foi introduzido um ORM denominado de *sequelize*.

#### **4.5.1.2. OAUTH 2**

Muitos processos de autenticação envolvem o uso de credenciais de acesso fornecidas pelo utilizador, nos casos mais simples, com recurso a um email e a uma password. Normalmente, as plataformas necessitam que o utilizador se autentique através da inserção das respetivas

credenciais num formulário HTML (*HyperText Markup Language*), que após serem submetidas é realizado um pedido a um servidor que irá verificar as mesmas. De seguida o servidor cria uma *cookie* com os dados da respetiva sessão, sendo que enquanto a sessão estiver assinalada como ativa, é possível a um utilizador aceder a recursos protegidos. Devido à sua simplicidade, este processo ainda é muito utilizado por muitas aplicações Web, mas apesar disso existem algumas desvantagens associadas à mesma, sendo elas: (1) processos de autenticação baseados em *cookies* são *stateful*, isto é, o servidor necessita de armazenar informação de forma a conseguir identificar as várias sessões existentes; (2) *cookies* não funcionam convenientemente em aplicações móveis; e (3) *cookies* não suportam a integração de aplicações ou clientes *third-party* [37].

A *framework OAuth* resolve os problemas listados acima. Esta *framework* é bastante utilizada hoje em dia devido ao facto de ser uma *framework* de autorização que permite às aplicações acederem a informação de todos os tipos de serviço. Assim, o *OAuth2* permite que clientes aleatórios, como uma aplicação móvel, acedam a informação de um utilizador através de um servidor de autorização de maneira segura, confiável e eficiente.

Existem alguns conceitos associados à *framework OAuth2* sendo eles: (1) *Resource Owner* que representa uma entidade capaz de fornecer acesso a um recurso protegido (quando o *Resource Owner* é um utilizador, este é referido como utilizador final); (2) *Resource Server* (*API Server*) que representa o servidor que possui a informação protegida, é capaz de receber pedidos e enviar as respetivas respostas através do uso de *access tokens*; (3) *Client* que representa uma aplicação que realiza os pedidos protegidos em nome do *Resource Owner* com a sua autorização (a aplicação pode ser executada num servidor, computador ou noutro dispositivo); e (4) *Authorization Server* que como o próprio nome indica é o servidor que gera os *access tokens* depois de validar a autenticação do *Resource Owner* [38].

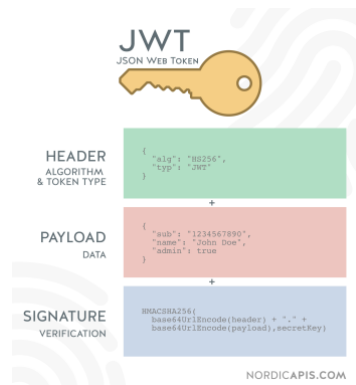
O *OAuth2* define cinco fluxos distintos que permitem aceder a recursos protegidos sendo eles o *Authorization Code*, *Implicit Grant*, *Resource Owner Credentials Grant*, *Client Credentials Grant* e, por fim, *Refresh Token Grant*. Apenas irão ser abordados dois dos fluxos apresentados, visto serem os únicos que são utilizados em todo o ecossistema da *Evollu*.

O *Resource Owner Credentials Grant* é um fluxo que é utilizado essencialmente entre *first party clients*, isto é, entre clientes que são conhecidos entre si e pode ser utilizado tanto entre aplicações Web como entre aplicações móveis [38]. O processo inicia-se pelo *client* (pode

ser tanto a plataforma da *Evollu* como as próprias aplicações) que solicita as credenciais de um utilizador em conjunto com o *client id* e o *client secret*. O servidor de autorização ao validar todos os campos devolve ao mesmo um *Access Token*, o respetivo *Refresh Token*, assim como a duração na qual o *Access Token* irá ser válido, que no caso da *Evollu* corresponde ao período de 1 mês.

O *Refresh Token Grant* é utilizado apenas para renovar o *Access Token* depois do mesmo ter expirado, sendo que a própria ação deverá ser invisível para o utilizador final quando um dos servidores refutar o *Access Token*. O cliente deve enviar o *Refresh Token* para o servidor de autorização para que este envie um novo par *Access Token*, *Refresh Token* [38].

Os *tokens* gerados seguem um padrão extremamente utilizado hoje em dia denominado de JWT (*Json Web Tokens*), que de uma forma simples consiste numa chave constituída por três partes distintas sendo elas o cabeçalho, os dados propriamente ditos que se pretendam que sejam transmitidos e, por último, a respetiva assinatura como mostra a Figura 18 [39].



**Figura 18 - Constituição de uma chave JWT**

A primeira parte da chave é composta pelo tipo de chave que representa, assim como qual o algoritmo utilizado para realizar a assinatura. A segunda parte corresponde a todo o *payload* de dados a transmitir, basicamente correspondem a *claims*, isto é, representam declarações sobre a própria chave, como sobre o utilizador que representam.

Apesar de não existir qualquer obrigatoriedade quanto às *claims* a utilizar, as boas práticas referem que devem ser incluídas as seguintes sendo elas: (1) *issuer* que especifica a entidade que gerou o *token*; (2) *exp* representa a data até quando o *token* é válido; (3) *subject* caracteriza o próprio utilizador a que a chave pertence; e (4) *nbj* diz respeito à data em que o *token* foi gerado. Por fim, a assinatura compõe a última parte do *token*, em que a chave

utilizada para obter a mesma deve ser o mais protegida possível para que não hajam utilizadores a impersonificar outros [39].

#### 4.5.1.3. Diagrama de Classes - Autenticação

As principais classes pelas quais o servidor de autenticação é responsável são os utilizadores, entidades, instalações, moradas e agrupamentos representados na Figura 19. De realçar que cada classe apresentada representa uma tabela na base de dados.

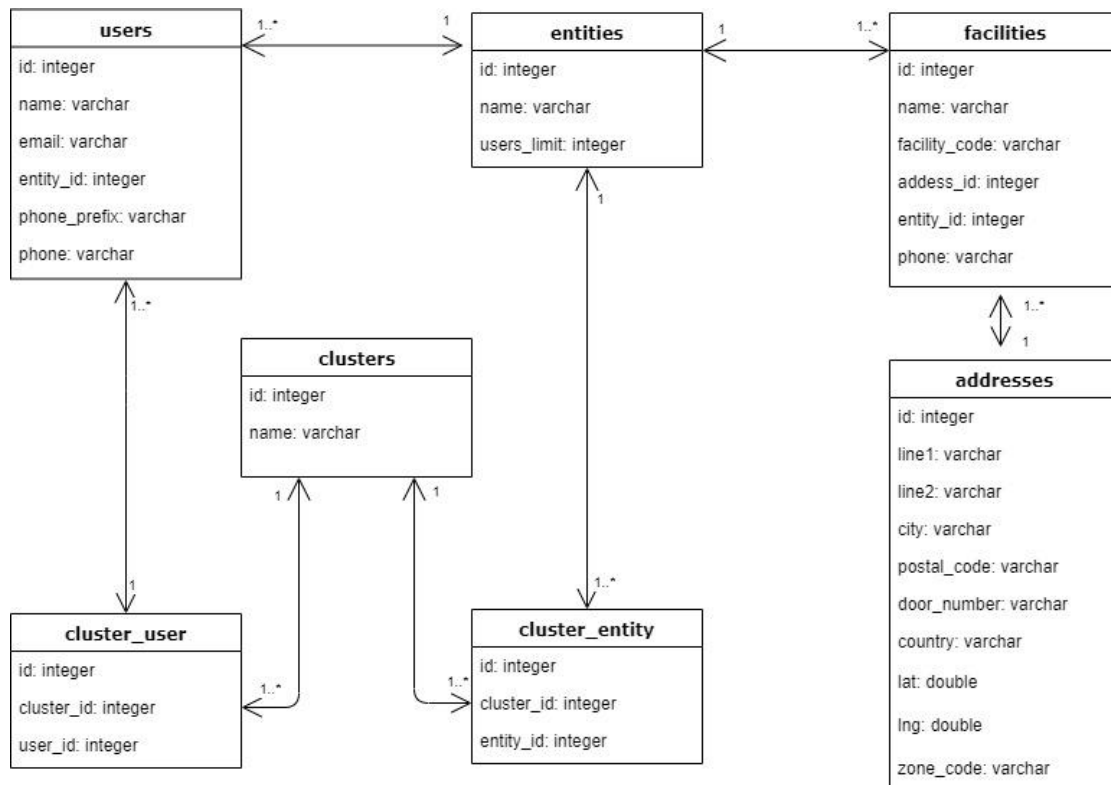


Figura 19 - Diagrama das classes principais servidor autenticação

- Um utilizador é uma pessoa que consegue autenticar-se na plataforma *Evollu*. Existe outra tabela (não representada na imagem) que guarda informações adicionais para além das apresentadas.
- Na plataforma *Evollu* as entidades (*Entities*) representam empresas ou *audiology* providers que usufruem da mesma, assim como empresas que detêm uma ou mais farmácias.
- Instalações que representam unicamente as farmácias aderentes a todo o projeto de audiologia da *Evollu* e pertencem a uma entidade e podem ser agendadas serviços

nas mesmas. Nas instalações são guardados o nome, o número de telefone, uma morada.

- Agrupamentos correspondem, como o próprio nome indica, a um aglomerado de entidades, no momento apenas podem ser geridos por membros da *Evollu*, mas é possível partilhar os mesmos com outros utilizadores. Os agrupamentos têm um papel fulcral, pois é possível filtrar a informação através dos mesmos em todas as páginas da plataforma onde são listados dados, assim como dados estatísticas e atribuição de permissões.
- *Addresses* contém todos os dados necessários para representar uma morada, de uma instalação ou de um paciente. As moradas servem exclusivamente para a aplicação móvel *Evollu* mostrar as farmácias onde existem rastreios mais perto do utilizador e para os promotores se conseguirem deslocar às mesmas.

De realçar que as tabelas descritas representam apenas uma parte do servidor de autenticação. Foram criadas outras, como os pacientes, os seus cartões, as suas pessoas de contacto (caso ocorra algo em relação aos mesmos), responsáveis de entidade e outras.

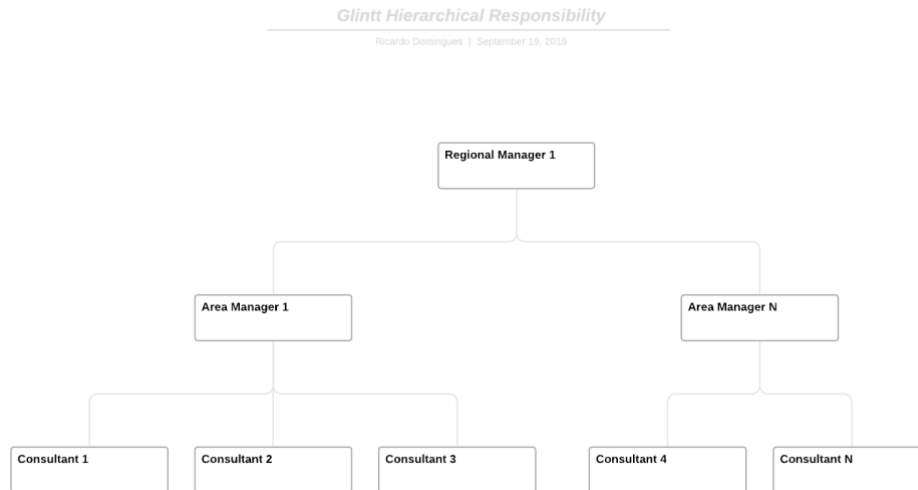
#### **4.5.1.1. Processo de Criação de Entidades**

A plataforma da *Evollu* é uma plataforma *multi-tenancy*, uma vez que possui uma arquitetura de software em que uma única instância da aplicação serve múltiplos clientes distintos, estes clientes representam entidades na plataforma. Deste modo, o processo de criação de entidades é um dos processos fulcrais que ocorre na plataforma, onde uma empresa denominada de *Glintt* intervém nos mesmos.

A *Glintt* é uma empresa que desenvolve software para hospitais, farmácias e clínicas em toda a península ibérica [40]. Em parceria com a *Evollu*, a *Glintt* desempenha o papel de incentivar as suas farmácias a utilizar a plataforma *Evollu*, de modo a serem realizados rastreios auditivos nas mesmas. As farmácias em questão obtêm uma comissão por cada *lead* qualificada gerada nas mesmas. Assim, foi necessário a criação de funcionalidades específicas para a *Glintt*, com o objetivo de os utilizadores da mesma poderem inserir as farmácias que pretendem aderir ao programa.

De forma sucinta, a *Glintt* pode ser representada por uma árvore hierárquica, como mostra a Figura 20.

Como é possível observar pela Figura 20, existem três níveis hierárquicos sendo eles os *regional managers*, *area managers* e *consultants*, em que o papel desempenhado por todos estes consiste na gestão das farmácias nas regiões que lhes competem.



**Figura 20 - Hierarquia de responsabilidades *Glantt***

De modo a agilizar no processo de criação de entidades provenientes da *Glantt*, foi facultado um ficheiro que contém todas as farmácias detidas por eles, e o mesmo foi importado para uma tabela na base de dados do servidor de autenticação. Todas as entidades criadas pela *Glantt* consistem em farmácias individuais, onde, mais tarde, irão ser efetuados rastreios organizados pela própria *Evollu*.

O formulário de criação de um pedido de entidade é iniciado com a introdução de um código único associado à farmácia. Ao preencher este campo, os dados da farmácia são carregados automaticamente, sendo necessário introduzir apenas os dados da pessoa responsável da farmácia (por norma corresponde ao dono da farmácia), o número de telefone da mesma e, o *regional manager*, *area manager* e *consultant* responsáveis por esta, como mostra a Figura 21.

Depois de toda a informação se encontrar validada e o pedido de entidade efetuado, é despoletado um processo complexo entre múltiplos serviços como mostra a Figura 22 em que os passos a realizar consistem no servidor de autenticação validar todos os dados inseridos, criar a entidade, a instalação e o utilizador responsável da entidade. Para além disso, é realizado um pedido HTTP para os serviços evidenciados no diagrama com o objetivo de serem criadas permissões para os vários responsáveis da entidade como o próprio dono, o *regional manager*, o *area manager* e o consultor da mesma.

**NOME DA FARMÁCIA:**  
Farmácia Central

---

**TELEFONE DA FARMÁCIA:**  
911111111

---

**PESSOA RESPONSÁVEL PELA FARMÁCIA:**  
Nome Responsável

---

**E-MAIL DA FARMÁCIA:**  
email@responsavel.com

**Morada:**

**PAIS:**  
Portugal

---

**LINHA 1 (RUA / PRAÇA / AV):**  
Edif. Bela Vista

---

**LINHA 2 (N.º / ANDAR / PORTA):**  
RC, Fr. C-D

---

**CIDADE:**  
Lugar do Cruzeiro

---

**CÓDIGO POSTAL (0000-000):**  
4540-302 ESCARIZ

---

**DISTRITO:**  
Aveiro

---

**CONCELHO:**  
Arouca

---

**FREGUESIA:**  
Escariz

---

Localização Automática

**REGIONAL MANAGER:**  
Regional Manager 1

---

**AREA MANAGER:**  
Area Manager 2

---

**CONSULTOR:**  
Consultor 2

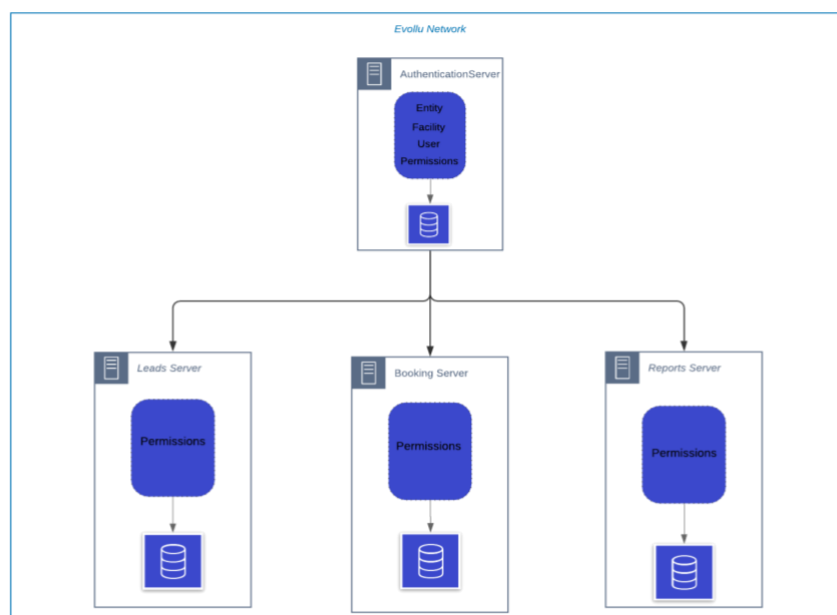
---

Interesse no Kiosk

Observações

**Figura 21 - Formulário de criação de entidade**

No caso de um dos responsáveis for alterado, todo o *workflow* referenciado na Figura 22 tem de ser realizado novamente de modo a remover as permissões dos responsáveis anteriores e, a atribuir permissões aos novos.



**Figura 22 - Workflow na criação de uma entidade**

#### 4.5.1.2. Processo de Autenticação

O processo de autenticação corresponde ao processo no qual o utilizador acede à plataforma e preenche o formulário de *login* até ao momento em que a plataforma carrega todas as informações necessárias da conta do utilizador, para que este possa usufruir de todas as funcionalidades que a plataforma lhe pode proporcionar.

Na plataforma da *Evollu*, todos os *menus* são construídos dinamicamente consoante as permissões e/ou a entidade do utilizador. Isto permite uma enorme flexibilidade, pois é possível especificar os *menus* que devem ser apresentados aos utilizadores a partir da base de dados. O diagrama apresentado na Figura 23 representa todo o processo de autenticação, assim como as etapas que são realizadas em *background* e transparentes ao utilizador.

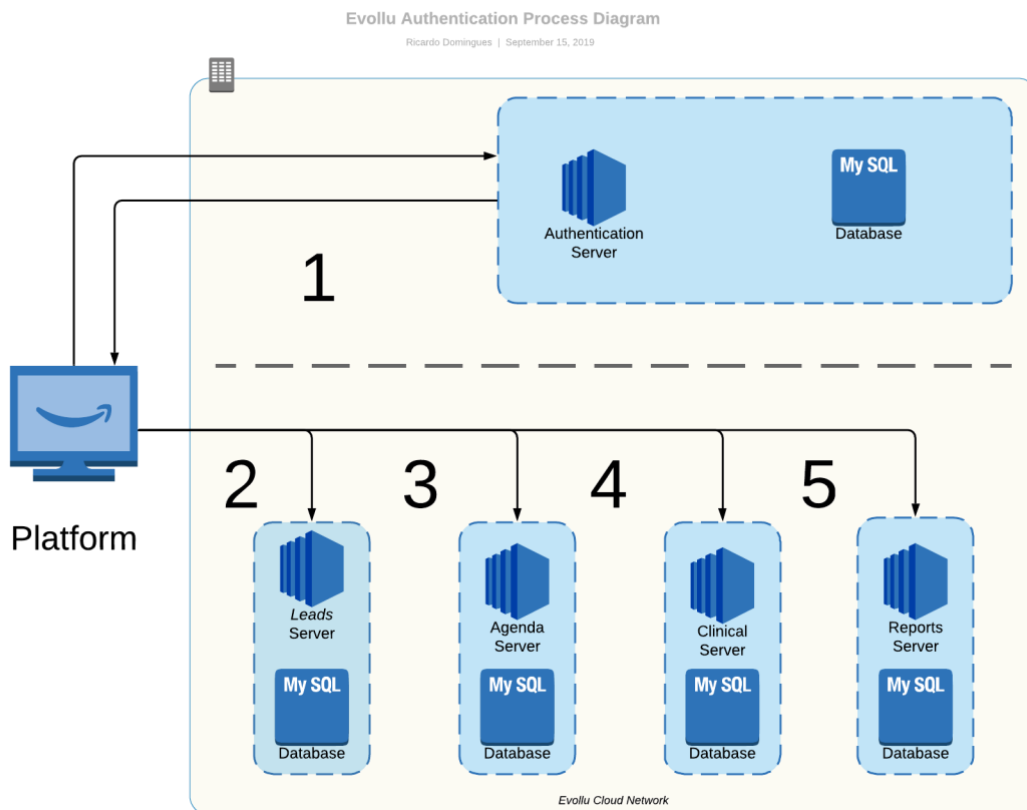


Figura 23 - Processo autenticação plataforma *Evollu*

Como se pode observar pelo diagrama, o utilizador preenche o formulário de *login* e depois, o servidor de autenticação valida as credenciais e envia todos os dados associados ao mesmo, desde informações acerca da sua conta, a sua entidade e todos os serviços aos quais este tem acesso.

De seguida, para cada um dos serviços provenientes da resposta apresentada na Figura 24, é realizado um pedido HTTP adicional para cada um dos serviços provenientes da resposta anterior. A resposta de cada um dos serviços contém informações, tais como, as permissões que o utilizador possui nesse serviço, assim como as entidades a que o mesmo tem acesso e quais os componentes que devem ser desenhados em cada um dos *menus* com base nas suas permissões.

```
{
  "id": 6,
  "name": "Ricardo Domingues",
  "entity": {},
  "services": [
    {
      "id": 1,
      "name": "Authentication Server",
      "slug": "authentication",
      "url": "https://auth.evollu.com",
      "components": [
        {
          "id": 1,
          "name": "Users",
          "icon": "fa fa-user",
          "meta": null
        },
        {},
        {}
      ],
      "user_permissions": [],
      "user_roles": [],
      "entities": []
    },
    {
      "id": 2,
      "name": "Leads Server",
      "slug": "leads-pt",
      "url": "https://leads.evollu.com"
    },
    {
      "id": 2,
      "name": "Agenda Server",
      "slug": "agenda-pt",
      "url": "https://booking.evollu.com"
    },
    {},
    {}
  ]
}
```

Figura 24 - Resposta do servidor de autenticação a um *login* válido

Cada componente especificado, na tabela de *componentes*, representa uma página distinta na plataforma *Evollu*. Todo o processo descrito em cima, resulta na estrutura de dados apresentada na Figura 25.

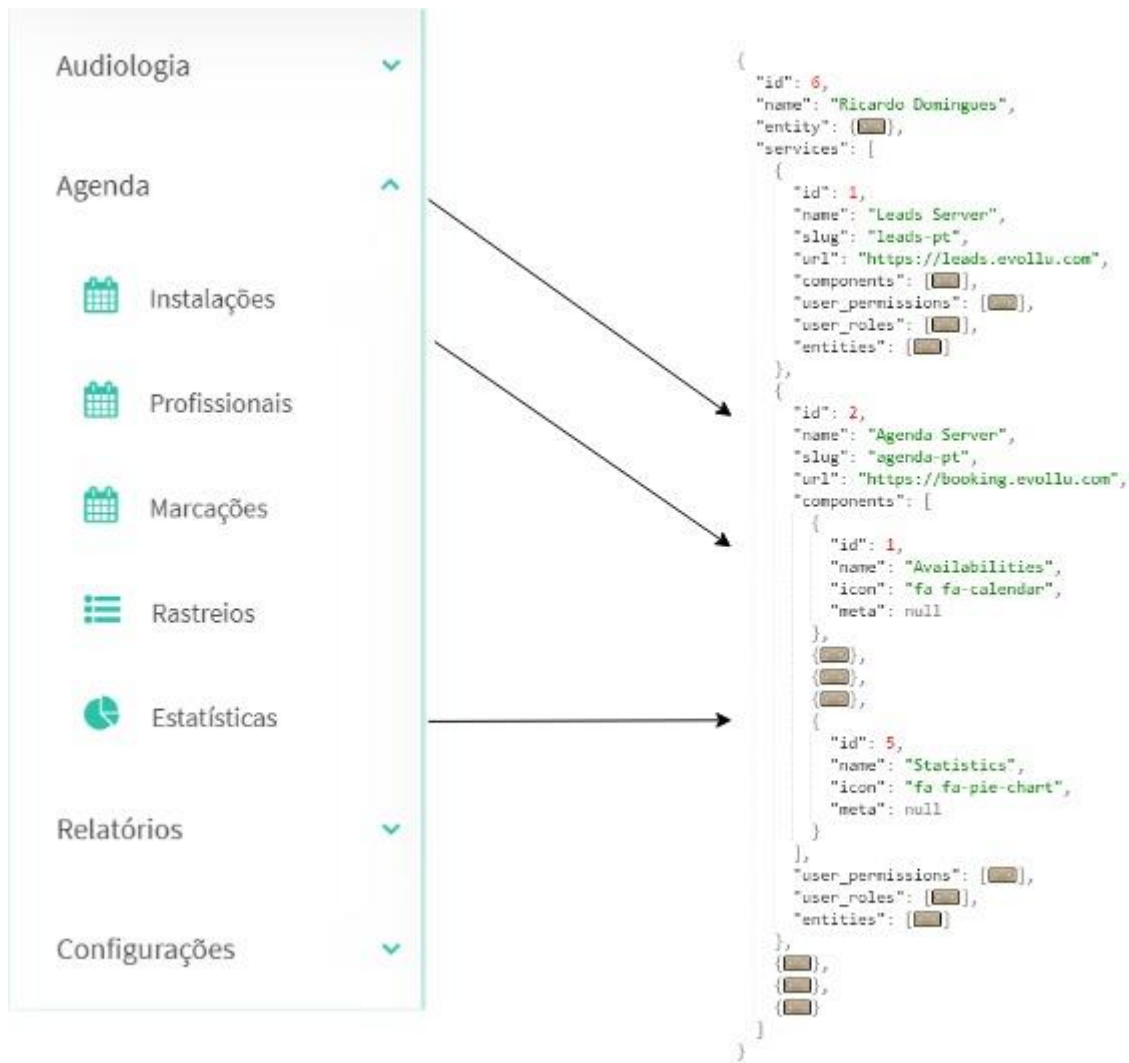


Figura 25 - Resultado final do menu a partir de JSON

Quando a plataforma recebe as respostas dos micro serviços, toda esta informação é armazenada no *Vuex*. O *Vuex* representa uma biblioteca pertencente ao ecossistema de *Vue*, que permite o controlo de estados da aplicação num só ponto, permitindo a todos os componentes acederem e modificar toda esta informação centralizada [33].

Para além disto, a aplicação neste momento possui toda a informação acerca dos componentes a que o utilizador autenticado tem permissões e pode iniciar o processo de apresentar os *menus*. Cada serviço apresentado na Figura 25 representa um *menu*, sendo que todos os componentes que se encontrem incluídos no mesmo irão representar um *submenu*. Para cada *submenu* é utilizado o nome do serviço em conjunto com o nome do componente para procurar o respetivo ficheiro que representa a vista que deve ser carregada ao clicar nesse *submenu*.

Apesar da solução acima ser bastante efetiva, surge o problema de não existir flexibilidade para agrupar componentes provenientes de vários serviços no mesmo *menu*. Por exemplo, no caso de todas as estatísticas existentes nos inúmeros serviços, seria bastante útil ter um *menu* denominado de Estatísticas, onde era possível agregar todos estas. Para isto, foi adicionado um campo à tabela de componentes denominado de *meta*, que de uma forma simples, especifica em que menu é que aquela página deve ser colocada. Caso já exista um menu com a designação inserida, a página é agregada a esse *menu*, caso contrário, é criado um novo menu.

Na Figura 26, é possível visualizar a existência de um novo menu designado como *Estatísticas*, e que o submenu designado como estatísticas, que anteriormente, se encontrava no menu de *Agenda*, encontra-se agora no menu *Estatísticas* com o nome de *Rastreios*, sendo este o nome que a página dispõe no JSON da resposta.



Figura 26 - Utilização de meta components na sidebar

#### **4.5.2. Serviço de *Leads***

O módulo de *leads* é, no momento, um dos serviços mais fulcrais providenciado pela *Evollu*, dado que este é o módulo no qual todo o processo de angariação de *leads* é realizado, para além disso, é o módulo mais comercializado na *Evollu*.

O objetivo deste módulo é o de processar e armazenar todos os dados provenientes dos quiosques. Inicialmente estes encontravam-se apenas fisicamente nas farmácias, mas mais tarde, com a implementação do módulo do Serviço de Agendamento, foi possível a atribuição de quiosques a promotores que realizavam rastreios nas farmácias das regiões, onde se encontravam designados (quiosques móveis).

Assim, as empresas fornecedoras de serviços na área da audiologia podem consultar as suas *leads*, geradas através dos quiosques, e proceder ao acompanhamento destas nas suas plataformas internas.

De realçar que na data de início do estágio, parte do módulo já se encontrava implementado, sendo que foi necessário efetuar alterações às funcionalidades já existentes, de modo a possibilitar que este fosse integrado na nova arquitetura que foi construída.

Foram acrescentadas novas funcionalidades devido a este ser o módulo mais utilizado por clientes, logo é natural que sejam feitos novos pedidos de funcionalidades de maneira a agilizar e melhorar todo o processo de transformação de uma *lead*.

Para além de toda esta informação, ainda compete a este módulo gerir toda a informação acerca dos quiosques, como a que promotores se encontram designados e a atividade dos mesmos, onde é possível consultar para cada dia, quantos questionários foram realizados, quantos pedidos de contatos foram feitos, entre outros.

Por fim, foram realizadas métricas para avaliar toda a informação guardada, assim como, a evolução entre os estados de uma *lead*, visto que esta possui inúmeros estados desde a criação do próprio registo até se considerar uma *lead* qualificada, sendo que os mais relevantes são: (1) se é um contacto válido; (2) se efetuou uma consulta com um audiolologista; (3) se na consulta foi confirmado que tinha dificuldades auditivas; (4) se resulta numa experiência, ou seja, se irá experimentar um aparelho auditivo e avaliar a adaptação ao mesmo; e por último, (5) se comprou o aparelho.

#### 4.5.2.1. Sistema de Notificações

Devido ao elevado número de *leads* recebidas na plataforma através de todos os canais de angariação das mesmas, foi necessário a implementação de um sistema de notificações de modo a filtrar todo este volume de informação, para otimizar o tempo despendido com as potenciais *leads*.

Inicialmente, já se encontrava desenvolvida uma solução, onde estavam a ser despoletadas notificações sempre que uma nova *lead* era submetida para a plataforma e que cumprisse com requisitos pré-definidos e estáticos.

Na fase inicial, a solução acima servia, mas rapidamente originou adversidades devido ao acréscimo de entidades e, respetiva intenção de diferentes critérios para estas. Assim, era necessária a existência de um painel de configurações onde cada entidade pudesse configurar os seus critérios para as *leads* provenientes dos seus quiosques.

Em suma, foi concebida uma página onde cada entidade pudesse definir as suas próprias regras, tanto para o valor dos questionários efetuados como para o resultado do teste auditivo. Com o envio de uma nova *lead*, primeiramente todas as regras definidas pela própria entidade são cruzadas com os valores da *lead*. Caso a entidade não possua nenhuma regra definida, é verificado se a entidade é pertencente a uma outra entidade, caso seja, é feito o mesmo para as regras da entidade “pai”. Por fim, caso as duas não possuam regras específicas são aplicadas regras *default*.

#### 4.5.2.2. Critérios de Notificações

A Figura 27 e a Figura 28 correspondem, à listagem de todas as regras de notificação para uma entidade, e ao formulário de criação de novas regras de notificação, respetivamente.



QUESTIONÁRIO	TESTE AUDITIVO	PEDIDO DE CONTACTO
Maior ou igual 0	Maior ou igual 0	Não
Maior ou igual 40	Maior ou igual 40	Não

2 registos

Figura 27 - Listagem de critérios de notificação de uma entidade.

TIPO	OPERADOR	VALOR (ENTRE 0 E 100)
Questionário	Maior ou igual	0
Teste Auditivo	Maior ou igual	0

Pedido de Contacto

**Figura 28 - Formulário de criação de novos critérios de notificação**

Por fim, as notificações via *email* eram enviadas a todos os membros pertencentes à respetiva entidade, algo que se revelou em certos casos indesejável, uma vez que cada entidade possui múltiplos quiosques em farmácias e, muitas vezes, são determinadas pessoas responsáveis por um subgrupo de quiosques que acabariam por receber notificações de *leads* de quiosques que não eram representados por eles. Tendo isto em conta, na página do perfil do utilizador foi dada a possibilidade de os mesmos subscreverem os quiosques para os quais desejariam ser notificados (ver Figura 29).

**Subscrição de Notificações** ✎

Irá receber emails de notificação de novos contactos para as entidades seleccionadas.

Acústica Médica

Evollu

Farmácia Benfica

Farmácia de demonstração

GLINTT

Widex

**Figura 29 - Subscrição de notificações a quiosques**

#### 4.5.2.3. Resultado Teste Auditivo

Uma funcionalidade pedida por clientes que usufruíram dos serviços da *Evollu* foi o do envio do resultado do teste auditivo realizado para os próprios utentes, para que pudesse não só criar uma maior aproximação entre os clientes com todas as pessoas que realizassem o teste, como também para que estes pudessem trazer o resultado com eles nas consultas que se seguissem com audiologistas certificados, de forma a agilizar todo o respetivo progresso.

Com isto, foi necessário a implementação de uma solução que permitisse às várias entidades existentes construir o seu resultado através de um *template* fornecido pelos próprios, como mostra a Figura 30.

Este *template* consiste, essencialmente, num ficheiro em formato *pdf* com os respetivos formulários e gráficos todos limpos, isto é, por preencher. Quando um teste auditivo é realizado, é enviado para a plataforma., todos os dados são analisados e é nesse momento que o *pdf* é construído recorrendo a várias bibliotecas, que permitem tanto a leitura como a escrita através de coordenadas (x, y) no próprio ficheiro, transformando-se assim no resultado ilustrado na Figura 30.

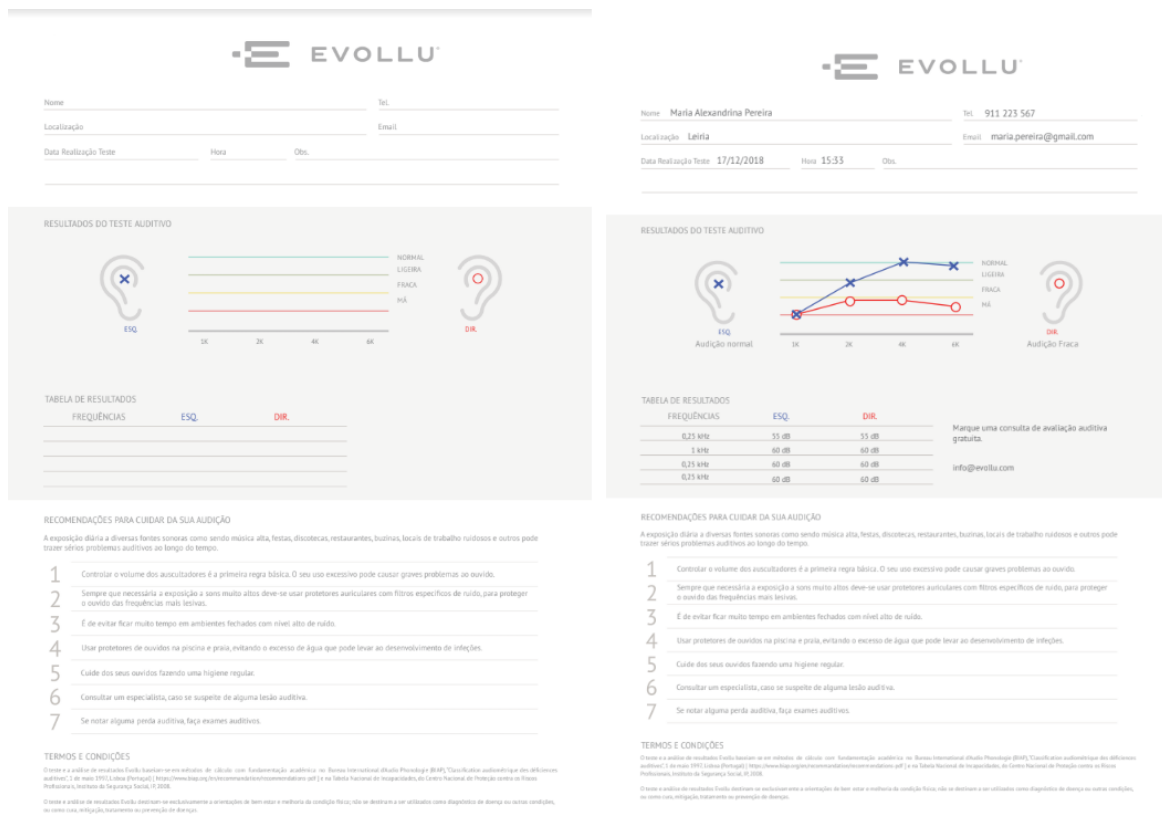


Figura 30 - Antes e depois do resultado do teste auditivo

É necessário ter em conta que cada entidade pode ter inúmeros *templates* em várias línguas. Consoante vários fatores, como a língua em que foi realizado, o teste ou a linguagem padrão da entidade, é decidido qual o *template* a utilizar.

### 4.5.3. Serviço de Agendamento

Numa fase inicial, foi solicitado um micro serviço que conseguisse interligar as *leads* geradas a partir dos inúmeros canais de angariação existentes, até às respetivas consultas de acompanhamento. Assim, este serviço iria ser benéfico não só para a *Evollu*, como também para outras empresas providenciarem consultas diretamente nas suas farmácias aderentes.

Durante a fase de elaboração, constatou-se que todo o processo de agendamento era comum a outros tipos de atividade. Desta forma, este serviço poderia ser generalizado com a possibilidade de comercialização a empresas associadas a outras especialidades na área da saúde.

Este serviço é responsável por inúmeras funções, que se encontram descritas nos seguintes pontos:

- Gestão de espaços físicos que, usualmente, correspondem às farmácias aderentes onde é possível agendar consultas e rastreios;
- Gestão de funcionários, isto é, possibilitar a atribuição de um profissional de saúde a uma farmácia, capaz de fornecer um tipo de serviço específico;
- Marcação de utentes consoante a disponibilidade de uma farmácia e de um profissional de saúde que irá realizar as suas consultas, para que, mais tarde, seja possível existir um histórico de consultas realizadas por farmácias, profissionais de saúde, assim como de pacientes.

#### **4.5.3.1. Processo de Agendamento**

O processo de agendamento inicia-se com a criação de instalações. Cada instalação poderá ter várias salas associadas à mesma. Através da agenda, é possível abrir disponibilidade de uma determinada sala, para um determinado dia, num intervalo número de horas.

É necessário ainda atribuir ou especificar profissionais de saúde ou auxiliares, que poderão vir a realizar os serviços na farmácia pretendida. É exigido que o responsável da farmácia especifique quais os profissionais que poderão realizar as consultas na sua farmácia. De seguida, o profissional de saúde poderá ver a listagem de todas as disponibilidades para as farmácias onde pode realizar consultas, e assim, marcar-se como disponível para as mesmas.

Já com um profissional de saúde disponível numa determinada instalação, os utentes e restantes funcionários da mesma poderão marcar consultas associadas a esta disponibilidade. De notar, que no caso de uma marcação efetuada através da aplicação *Evollu*, esta deve ficar num estado pendente até que um utilizador da plataforma valide e confirme a mesma.

Em caso de cancelamento, todos os intervenientes associados à marcação, como a própria farmácia, o profissional de saúde e respetivo paciente, devem ser notificados e a vaga deve ser libertada.

## Instalações

As instalações representam somente farmácias, onde é possível a realização de consultas e rastreios, de modo a angariar o maior número de utentes possível.

Foi criada uma página onde é possível a um utilizador visualizar todas as farmácias, nas quais o mesmo tem permissões. Nela é possível visualizar os detalhes da mesma, como o nome, o número de telefone, várias informações relativas à morada da mesma e, por fim todas as salas onde é possível realizar eventos (ver Figura 31).

The screenshot displays the 'Instalações' (Installations) page. On the left, there is a search bar labeled 'FILTRAR' with the placeholder 'Texto de pesquisa'. Below it is a table listing pharmacies:

NOME	TELEFONE	CIDADE	SALAS
Farmácia Leiria	244000000	Leiria	3
Farmácia Lisboa	244000000	Leiria	2
Farmácia Porto			0

Below the table, it indicates '3 registos'. On the right side, there is a detailed view for 'Farmácia Leiria' with the following information:

- Telefone: 244000000
- Cidade: Leiria
- Latitude: 39.7540405
- Longitude: -8.8759987
- Morada: Rua A nº 12, 2ºE 2400

Below the details, there is a section for 'Salas' (Rooms) with an 'ADICIONAR' button. The table shows three rooms:

#	NOME	AÇÕES
1	room1	[Edit]
2	room2	[Edit]
3	room3	[Edit]

Figura 31 - Listagem de instalações e respetivos detalhes

## Disponibilidades de Sala

De modo a criar uma disponibilidade de uma farmácia para consultas através da plataforma *Evollu*, é exigido que o utilizador possua permissões de escrita ou de administração para a mesma. Estas disponibilidades podem ser privadas, sendo que não são listadas publicamente e apenas utilizadores específicos podem visualizar as mesmas.

Na criação deste tipo de disponibilidades é necessário o preenchimento da data de início, data de término, tipo de serviço que irá ser efetuado, assim como, qual a sala onde irá ser realizado o evento, caso a farmácia possua várias salas. É possível ainda a atribuição de um profissional de saúde, caso já se saiba esta informação antecipadamente, como mostra a Figura 32.

**Criar Disponibilidade - Saraiva S.A. (Francode Nossa Senhora)**

**INÍCIO**  
2019-10-03 18:00:00

**FIM**  
2019-10-03 22:00:00

**TIPO**  
Audiologia

**LOCAL**  
Espaço Promoção

**RECORRENTE**  
 Evento recorrente

**PROFISSIONAIS**  
 Atribuir Staff

**PROFISSIONAIS**  
Pedro Fonseca

CANCELAR GUARDAR

Figura 32 - Formulário de criação de disponibilidade de farmácia

### Disponibilidades de Profissionais de Saúde

Ao existir uma disponibilidade de farmácia, um profissional de saúde pode marcar-se como disponível para realizar consultas na sua área de serviço. Para o mesmo se registar na respetiva disponibilidade, é necessário que esteja autorizado a realizar o respetivo serviço na farmácia em questão.

Toda a logística dos utilizadores que se encontrem autorizados a realizar um serviço específico pode ser realizada na página de entidade, por utilizadores com permissões para o efeito, como é possível observar pela Figura 33.

Para finalizar, é possível a profissionais distintos marcarem-se como disponíveis para a mesma disponibilidade de farmácia, desde que não exista nenhuma sobreposição de horários, como por um exemplo, o Profissional A realizar consultas da parte da manhã e o Profissional B da parte da tarde (ver Figura 34).

Detalhes do Utilizador

**Salvador**

E-mail: Yango27@gmail.com  
Entidade: Promoter's Provider

Permissões | **Atividades**

**Atividades**

ATIVIDADE	ENTIDADE
Análises Clínicas	Albuquerque - Franco, Albuquerque - Martins, Albuquerque - Moreira, Mais 404...
Audiologia	Entidade Demo Norte, Entidade Demo Centro, Entidade Demo Sul
Enfermagem	O usuário não fornece enfermagem em nenhuma entidade.
Nutrição	O usuário não fornece nutrição em nenhuma entidade.
Rastreio Auditivo	Albuquerque - Franco, Albuquerque - Martins, Albuquerque - Moreira, Mais 404...
Rastreio Local	O usuário não fornece rastreio local em nenhuma entidade.

**Figura 33 - Gestão de atividades associadas a um utilizador**

Atribuir Profissional - Saraiva S.A. (Francode Nossa Senhora)

INÍCIO: 2019-10-02 09:15:00 | FIM: 2019-10-02 15:00:00

PROFISSIONAIS: Pedro Fonseca

INSTALAÇÃO: Saraiva S.A. | SALA: Espaço Promoção

CANCELAR | GUARDAR

**Figura 34 - Formulário de Atribuição de Profissional de Saúde**

## Consultas

As marcações correspondem a utentes que pretendam realizar consultas de acompanhamento numa disponibilidade de um profissional de saúde. Estas podem ser inseridas na plataforma

através de utilizadores da farmácia, dos próprios profissionais de saúde ou até pelos próprios pacientes, através da aplicação *Evollu*.

Nas marcações através da aplicação *Evollu*, são fornecidos apenas intervalos de tempos fixos consoante o tipo de consulta que o mesmo tencione realizar, isto porque uma primeira consulta é, naturalmente, mais demorada do que uma consulta de rotina. Quanto às marcações na plataforma por utilizadores com permissões, não existe qualquer restrição horária, sendo que estes são responsáveis por gerir as suas próprias agendas, como mostra a Figura 35.



Criar Marcação - Quarta-feira, 2 de outubro de 2019 - 09:15 / 15:00

Nome Telefone

E-mail

Registo de contacto já existe? [Clique aqui](#) para procurar um registo.

INÍCIO INÍCIO

2019-10-02 10:00:00 2019-10-02 11:00:00

TIPO

Primeira consulta

Observações

CANCELAR GUARDAR

Figura 35 - Formulário de criação de consulta

#### 4.5.3.2. Agenda

Tendo em conta a secção “Processo de Agendamento”, foi proposto que a interface, na implementação de todo este processo de agendamento, fosse semelhante a uma agenda. Isto porque, trata de um conjunto de funcionalidades, que tem como base intervalos de horários dentro de intervalos de datas. Os próprios utilizadores, num uso pessoal, estão acostumados a lidar com este tipo de interfaces; e por isso é possível alcançar uma melhor *User Experience* através de ações, como o de seleção de um intervalo de tempo, ou até a deslocação de um evento para um horário distinto.

De maneira a usufruir e proporcionar ao utilizador final a melhor experiência de utilização possível, foram implementadas diversas vistas na agenda, sendo elas as seguintes:

- Vista por dia, onde é possível visualizar todos os eventos existentes para as farmácias selecionadas num determinado dia;
- Vista semanal, onde é possível visualizar todos os eventos existentes para as farmácias selecionadas numa determinada semana;
- Vista mensal, onde é possível visualizar todos os eventos existentes para as farmácias selecionadas num determinado mês;
- Vista em listagem, onde consta uma lista de todos os eventos num determinado mês.

Na Figura 36 e na Figura 37, é possível observar o calendário para três farmácias distintas na vista diária e mensal, onde é possível visualizar a grelha de horário com o intervalo de horas entre as 09h da manhã e as 23h da noite, horário este definido pelos órgãos de chefia da *Evollu*.

Em relação aos eventos, é mostrada ao utilizador uma breve descrição sobre o mesmo, contendo informação simplificada, como a data de início e fim, o nome da farmácia, tipo de serviço a que se refere e, por fim, a sala onde irá ser realizado. No entanto, com um clique sobre o evento, é possível consultar todos os detalhes acerca do mesmo e, é possibilitada a edição ou remoção do mesmo.

Um requisito solicitado na elaboração da agenda foi a visualização de todos os feriados obrigatórios, para que não fossem marcados eventos sobrepostos a estes. A solução para este critério foi desenvolvida com o recurso a uma biblioteca *JavaScript* de *full-calendar*. Esta biblioteca é responsável por construir o esqueleto da agenda nas inúmeras vistas, sendo que a maior parte do CSS necessitava de ser customizado.

Ao longo da implementação de todas as funcionalidades relativas à agenda, surgiram inúmeras dificuldades devido ao facto da biblioteca ser baseada em *JavaScript* e a mesma depender da manipulação de eventos dos próprios elementos da página HTML, ao contrário do *Vue*, que se caracteriza por ser *data-driven*.

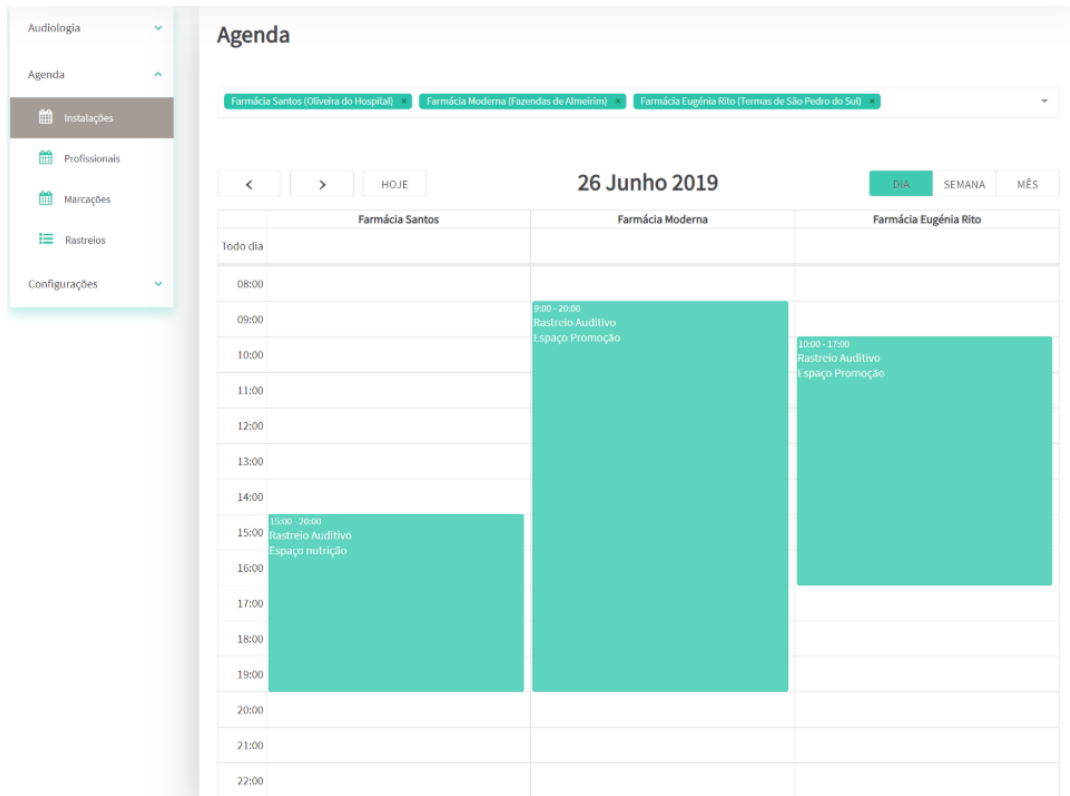


Figura 36 - Agenda vista diária

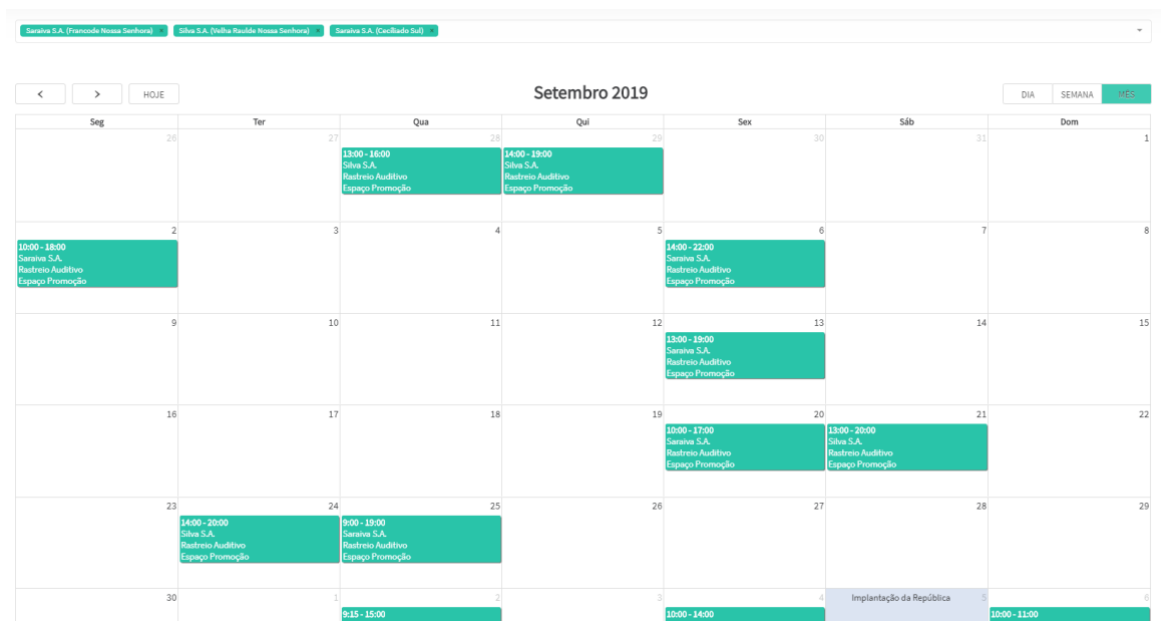


Figura 37 - Agenda vista mensal

### 4.5.3.3. Eventos Recorrentes

De forma a facilitar a gestão das várias instalações, nomeadamente, a abertura de *slots* de disponibilidades para uma dada farmácia, foi proposta a elaboração de eventos recorrentes,

ou seja, possibilitar ao utilizador a oportunidade de realizar toda a gestão de eventos que se repetem ao longo do tempo.

A implementação de um sistema que suporte eventos recorrentes, dependendo dos requisitos necessários, pode representar uma tarefa bastante complexa, tanto a nível arquitetural como a nível de representação dos dados. Dessa forma, foi necessário um planeamento prévio para garantir que a solução implementada cumprisse com todos os requisitos, que não impusesse nenhuns obstáculos no futuro e, que permitisse cumprir com todas as funcionalidades desejadas.

Após a especificação dos requisitos e funcionalidades necessárias, chegou-se à conclusão que o sistema teria de ser capaz de:

- Criação de eventos com tipo de recorrência diária, semanal e mensal, sendo que no caso da última deverá ser possível especificar a semana em que o evento se deve repetir, ou seja, entre a primeira semana e a quinta semana de cada mês, visto que existem meses onde existem mais de quatro semanas, ou mesmo o último dia da semana possível de cada mês.
- Possibilidade de especificação de intervalos sobre quais os eventos se devem repetir, ou seja, de quanto em quanto tempo é que a recorrência deve ocorrer, que varia consoante o tipo de recorrência escolhida.
- Possibilidade de especificação de e até que data o evento se deve repetir, tendo como alternativas a escolha de uma data posterior à data do evento até o limite de dois anos, ou até o número de vezes em que o evento se deve repetir.
- Sempre que um evento recorrente seja alterado ou eliminado deve ser apresentado ao utilizador a possibilidade de modificar ou remover apenas o evento selecionado, e os que se repetem no futuro, ou por último, todos os eventos associados à repetição.

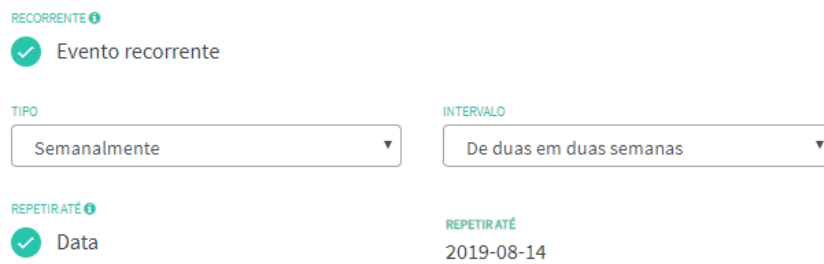
Depois de uma pesquisa, verificou-se que existem pelo menos duas abordagens distintas à elaboração de eventos recorrentes, podendo-se considerar uma como uma abordagem mais simples ao problema, e a outra como uma abordagem mais complexa, sendo que são caracterizadas por, respetivamente: (1) ao armazenamento de todas as possíveis instâncias distintas de um evento recorrente como linhas distintas numa tabela e (2) ao armazenamento de um padrão de recorrência associado a um evento recorrente, onde é necessário gerar todos os eventos passados e futuros programaticamente.

A primeira abordagem tem como vantagem o facto de ser uma abordagem bastante simples, o que acabará por levar a uma implementação facilitada e, como desvantagem, o facto de ser necessário guardar todas as instâncias de um evento na base de dados, o que para um evento que se repita semanalmente durante um ano em inúmeras farmácias pode, rapidamente, representar um vasto volume de dados na base de dados. Outra desvantagem prende-se ao facto de que a atualização de eventos recorrentes pode tornar-se bastante complexa, isto porque caso um dos eventos recorrentes seja modificado, leva a que todas as instâncias associadas à repetição sejam alteradas, o que por sua vez pode causar um elevado número de *queries* adicionais e, caso ocorra um erro durante todo o processo de atualização ou remoção, é necessário que a exceção seja tratada adequadamente para não levar a inconsistências na base de dados.

Quanto à segunda abordagem, as suas vantagens são que solucionam todas as desvantagens da primeira abordagem, mas carece de uma estruturação bastante complexa a nível do modelo de dados, com várias tabelas adicionais para controlar todos os casos possíveis.

Assim, tendo em conta as vantagens e desvantagens de cada um dos possíveis métodos de implementação do problema proposto, foi escolhida a abordagem mais simples, onde todas as instâncias de eventos são armazenadas na base de dados, isto leva a que haja um maior número de registos, mas tendo em conta que existe uma duração máxima de intervalo de tempo nos eventos recorrentes, acaba por não causar um grande impacto.

A interface implementada para gerir os eventos recorrentes encontra-se representada tanto na Figura 38 como na Figura 39, e todos os requisitos necessários foram tidos em conta no desenvolvimento da mesma.



RECORRENTE ⓘ  
✓ Evento recorrente

TIPO  
Semanalmente ▼

INTERVALO  
De duas em duas semanas ▼

REPETIRATÉ ⓘ  
✓ Data

REPETIRATÉ  
2019-08-14

**Figura 38 - Interface evento recorrente semanal**

RECORRENTE ⓘ  
 Evento recorrente

TIPO  
Mensalmente ▼

INTERVALO  
Todos os meses ▼

SEMANA  
1º Domingo ▼

REPETIR ATÉ ⓘ  
 Data

Número de ocorrências (1 - 100) ▼

Figura 39 - Interface evento recorrente mensal

#### 4.5.3.4. Rastreios Auditivos

Todas as funcionalidades em redor do micro serviço de agendamento procuraram ir de encontro ao modelo de negócio atual. Este modelo de negócio era caracterizado por uma abordagem direta, ou seja, a *Evollu* iria disponibilizar toda a plataforma onde, farmácias parceiras e aderentes pudessem disponibilizar o seu próprio espaço, nos quais audiologistas, principalmente internos, isto é, pertencentes à *Evollu*, mas também externos, conseguissem gerir e realizar consultas.

Face a este modelo, surgiu a oportunidade de a *Evollu* proporcionar a realização de rastreios auditivos em grande volume nas farmácias portuguesas. De forma consequente, este aumento iria causar um elevado número de *leads* geradas, sendo que a *Evollu* só teria de assegurar a existência de promotores para cada um dos rastreios auditivos realizados. Os promotores são indivíduos que se deslocam às farmácias para promover os rastreios e incentivar pessoas a realizarem o teste auditivo. Estes promotores eram adquiridos através de empresas promotoras externas. Deste modo, estes promotores iriam contribuir para a *Evollu*, realizando ações de rastreios diários, em inúmeras farmácias em Portugal.

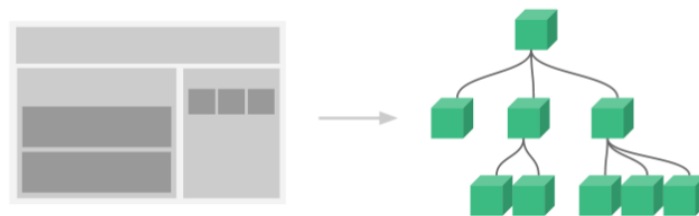
É importante realçar que a inclusão dos rastreios auditivos no micro serviço de agendamento fez com que este dispusesse de duas vertentes sendo eles, respetivamente, o propósito inicial de agendamento de consultas em farmácias parceiras à *Evollu*, que permite realizar o acompanhamento dos mesmos através do serviço de registos clínicos detalhado na secção 4.5.4, e o agendamento de rastreios auditivos em diferentes farmácias, onde todas as *leads* angariadas nas sessões seriam vendidas a empresas utilizadoras da plataforma.

Em termos de implementação, tornou-se bastante acessível adaptar todas as funcionalidades existentes à realização de rastreios, uma vez que se trata de um evento específico, efetuado numa farmácia e num horário específico, em que o profissional de saúde é o próprio

promotor. Deste modo, foi possível conciliar toda a lógica de negócio já existente aos rastreios, sendo que foi apenas necessário a adição de um novo tipo de atividade designado como rastreios auditivos à tabela de atividades e, adicionar essa mesma atividade a todas as farmácias onde a *Evollu* iria organizar os mesmos.

Apesar disso, a agenda possui algumas desvantagens, nomeadamente, o facto de não dispor de uma visualização global do estado de todos os rastreios. Devido à existência de um número elevado de rastreios é impossível visualizar todo este número de farmácias.

Tendo em consideração o pormenor descrito acima, foi desenvolvida uma nova página que iria colmatar as desvantagens de visualização por parte da agenda. Assim, levou-se a cabo a migração de parte das funcionalidades, como foi o caso da criação de disponibilidades de farmácia, e a própria gestão dos promotores que iriam a cada rastreio. A adaptação das funcionalidades tornou-se bastante trivial devido ao facto de todas as funcionalidades serem independentes e reutilizáveis devido à possibilidade de reutilização de componentes *Vue* ao longo do projeto. Um componente *Vue* descreve uma funcionalidade independente que pode ser reaproveitada em inúmeras páginas, levando à reutilização de código e permitindo possuir elementos com comportamentos específicos testados [33], como é possível observar pela Figura 40.



**Figura 40 - Estrutura de componentes numa página Web**
















Como se pode observar pela Figura 41, existem várias vantagens proporcionadas por esta nova página, tendo em conta este modelo de negócio baseado em rastreios auditivos, sendo que o mapa permite obter uma visão global dos rastreios pelas inúmeras regiões pertencentes a Portugal.

Existem vários filtros possíveis:

- Filtro por texto, onde é realizada uma pesquisa local, isto é, uma pesquisa com todos os dados carregados localmente no *browser*, sem ir ao servidor;

- Filtro por entidade, onde é possível efetuar a filtragem por várias entidades ou até por um agrupamento de entidades;
- Filtro por data, em que, por defeito, as datas selecionadas correspondem aos próximos dois meses;
- Organização da listagem do seguinte modo: instalação que agrupa os vários rastreios por farmácia; promotor que agrupa os rastreios pelo promotor que irá realizar os mesmos; e por datas em que os rastreios são agrupados por data de início.

The screenshot displays the 'Rastreios' (Audits) interface. At the top right, there is a green button labeled 'CRIAR RASTREIO'. Below it, there are search and filter options: a search box with 'Texto de pesquisa', a dropdown for 'ENTIDADES' set to 'Todas as Entidades', a date range '2019-07-15 a 2019-09-15', and a dropdown for 'OPÇÕES DE VISUALIZAÇÃO' set to 'Por instalação'. The main content area is titled 'Visualização Por Instalação' and features a table of audit records. To the right of the table is a map of Portugal with a 'Mapa' and 'Satélite' toggle, and an 'Esconder Mapa' button.

Farmácia Santana (Rua de Santana - Leça do Balio)				
2019-07-15 10:00 - 18:00	Farmácia Santana (Rua de Santana - Leça do Balio)	Angélica Amorim	Glintt Wixed 13	  
2019-08-19 09:30 - 18:00	Farmácia Santana (Rua de Santana - Leça do Balio)	Promotor indefinido	Sem Quilosque Atribuído	  
Farmácia Central (Rua República - Figueira da Foz)				
2019-07-15 10:00 - 18:00	Farmácia Central (Rua República - Figueira da Foz)	Cátia Costa	Glintt Wixed 6	  
Farmácia Areal (Avº Sul - Urbanização Sull Park - São João de Ver)				
2019-07-15 10:00 - 18:00	Farmácia Areal (Avº Sul - Urbanização Sull Park - São João de Ver)	Luís Teixeira	Glintt Wixed 14	  
Farmácia Ascensão Nunes (Rua Anta da Aqualva - Aqualva Cacém)				
2019-07-16 09:30 - 18:00	Farmácia Ascensão Nunes (Rua Anta da Aqualva - Aqualva Cacém)	Carolina Chanoca	Sem Quilosque Atribuído	  
2019-08-13	Farmácia Ascensão Nunes		Sem Quilosque	

**Figura 41 - Interface implementada para a página dos Rastreios Auditivos**

De modo a proporcionar ao utilizador uma melhor experiência de utilização, os filtros são persistidos localmente no *browser*, para que sejam guardados em futuras utilizações.

Uma vez que, grande parte da utilização da página pode ser efetuada em parte pelos promotores, e estes podem utilizar ecrãs de dimensões reduzidas, é importante referir que existiu o cuidado de adaptar a listagem a ecrãs de dimensões inferiores como é representado na Figura 42.



Figura 42 - Listagem de rastreios em ecrãs de dimensões inferiores

#### 4.5.3.5. Identificação de Sessões de Rastreios Auditivos

A adição dos rastreios auditivos provocou um aumento considerável no número de *leads* geradas, sendo que o modo de angariação das mesmas aplicou-se de dois modos, sendo respetivamente: (1) nas farmácias com quiosques físicos e (2) através dos promotores em sessões de rastreios auditivos que utilizavam um quiosque portátil.

Foi necessário distinguir quais as *leads* geradas em sessão por promotores, das que foram geradas através dos quiosques físicos. Das *leads* geradas em sessão, é necessário identificar qual o promotor que realizou a mesma, para que seja possível a realização de todos os pagamentos a estes e, também perceber quais os promotores que se destacam. Tendo isto em conta, foi solicitada a implementação de uma abordagem que permitisse solucionar a questão referida. Tendo por base a problemática enunciada, chegou-se à conclusão de que a utilização de *QR Code* seria a mais adequada.

O *QR Code* é uma tecnologia que permite o armazenamento de várias informações como hiperligações, coordenadas, entre outros [41], para que variados dispositivos, neste caso os quiosques, possam realizar a leitura do mesmo através do *scan*, recorrendo às câmaras fotográficas dos mesmos.

Assim, foi proposta uma nova funcionalidade através da página dos rastreios. Esta funcionalidade consiste na introdução de um email de um utilizador na aplicação, neste caso o de um promotor. Posto isto, seria enviado para o mesmo todas as informações necessárias

sobre a sessão, tais como os contactos da farmácia, como chegar à mesma, a data de início e fim do rastreio e, por fim, um código QR gerado (ver Figura 43), que permitiria ser lido na aplicação de teste auditivo. A informação contida no código QR correspondia a uma hiperligação para o serviço de *leads*, que possibilitava iniciar a sessão, para que todas as *leads* enviadas no intervalo da sessão ficassem registadas na mesma.

#### Rastreio Auditivo

Caro(a) promotor(a),

A equipa da Evollu relembra que tem um rastreio auditivo agendado para o dia 05/07/2019 das 09:30h até às 18:00h na farmácia Farmácia Medeiros.

Se por algum motivo identificar que irá chegar atrasado(a), contate de imediato a farmácia e em seguida a Evollu, para que estes informem os utentes do seu atraso.

Para consultar as leads realizadas na respetiva sessão [clique aqui](#).

Caso necessite de alguma indicação adicional pode contactar a farmácia através do número: 253311123.



Morada da farmácia:

- Praça 5 de Outubro 78 Vila Verde 4730-731 VILA VERDE [Google Maps](#).

Melhores cumprimentos  
Equipa Evollu



Figura 43 - Exemplo email com QR Code

#### 4.5.3.6. Exportação de eventos para calendários externos

Em consequência da inutilização da plataforma pelos promotores existentes, foi necessário tomar providências adicionais para além dos emails automáticos apresentados na secção 4.7, de forma a combater falhas esporádicas de rastreios por parte dos mesmos.

Assim, foi proposta a integração do calendário pessoal dos promotores na plataforma por serviços providenciados por terceiros, como é o caso do *Google Calendar*, *Apple Calendar*, o calendário do *Outlook*, entre outros, para que fosse possível aos promotores acenderem às suas agendas pessoais na aplicação de calendário predefinida pelos seus *smartphones*, providenciando uma conveniência adicional aos próprios, dado que todos estes calendários suportam a especificação de alertas customizados para cada um dos seus eventos.

## ICal

O *ICal*, também designado por *ICalendar*, é um ficheiro que possui um formato específico de dados para representação de eventos desenvolvidos pela *Apple*. Este tipo de ficheiro suporta eventos recorrentes, assim como, campos adicionais como o estado do evento, descrição, localização, entre outros [42].

Como podemos observar pelo diagrama da Figura 44, o controlador *ICalController* é responsável por gerir os recursos associados ao *ICal*, desde criar, listar e até ativar/desativar a exportação de eventos.

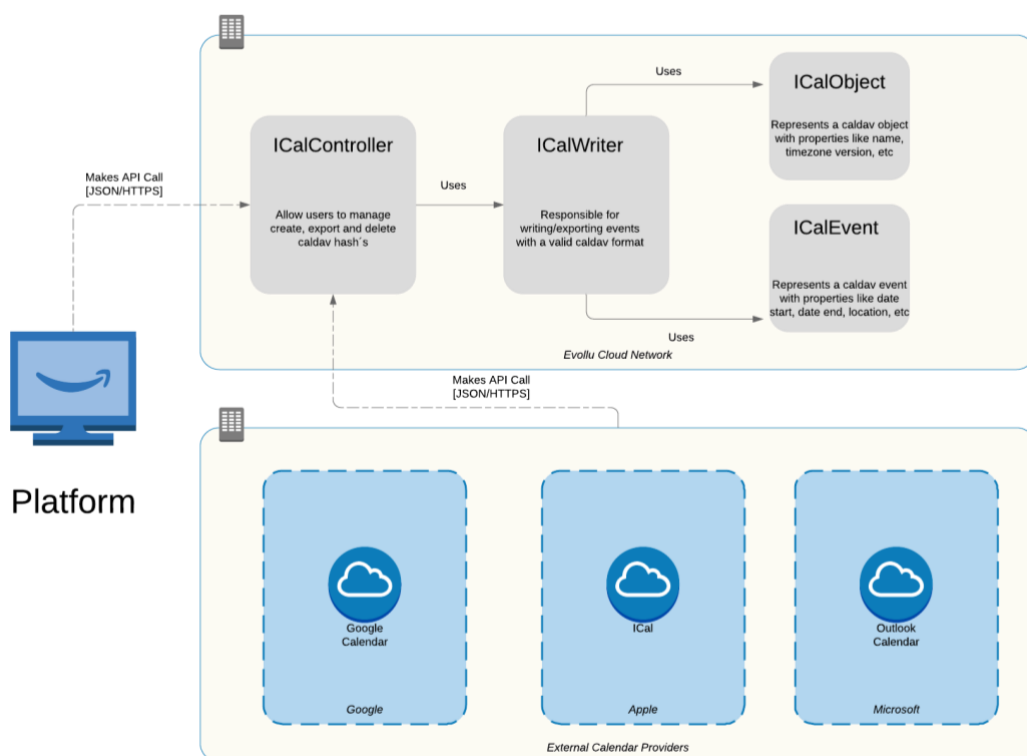
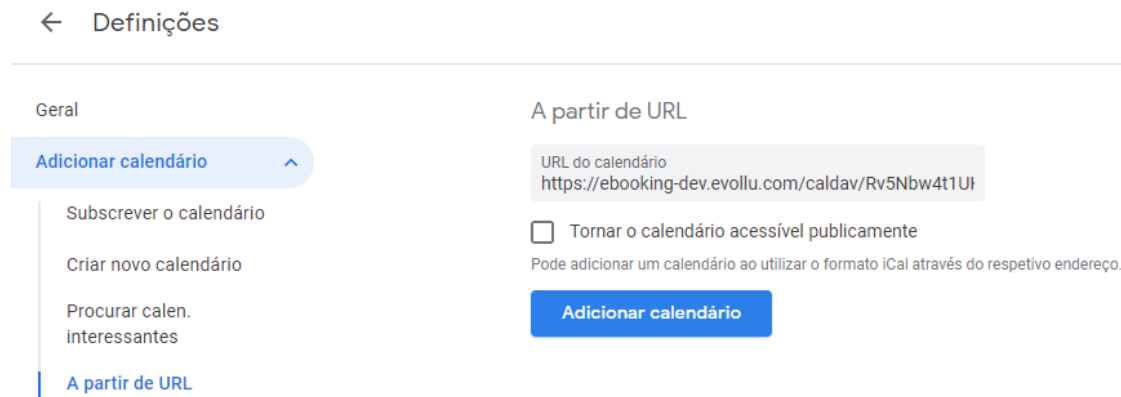


Figura 44 - Diagrama de fluxo da exportação de eventos

Ao gerar um *ICal* é gerada uma hiperligação única para o próprio utilizador onde são devolvidos todos os rastreios, para qual o utilizador tem permissões de leitura. A hiperligação devolve um ficheiro no formato *.ics* e com um *contente-e* designado de *text/calendar*. O utilizador pode assim, copiar o endereço de URL criado e utilizar o mesmo para subscrever a um calendário no seu calendário pessoal (ver Figura 45).



**Figura 45 - Subscrição calendário a partir de url no Google Calendar**

Em termos de *interface*, foi pedido que na página de perfil do próprio utilizador fosse possível visualizar se este já teria gerado uma exportação de todos os seus eventos, e caso não tivesse, fosse possível a sua criação (ver Figura 46). Também deveria ser possível ativar ou desativar um calendário (ver Figura 47).

## Configurações

### SINCRONIZAÇÃO CALENDÁRIO

Sem url de sincronização com calendário definido. [Clique aqui](#) para gerar um novo.

**Figura 46 - Criação de ICal no perfil de utilizador.**

## Configurações

### SINCRONIZAÇÃO CALENDÁRIO



<https://ebooking-dev.evollu.com/calDav/OZkd77fXRj58goSyZuYMyC9ewbxbtHwAQnsgxEcj3N9SQklXj4>

Quer um novo endereço? [Clique aqui](#) para gerar uma nova hiperligação.

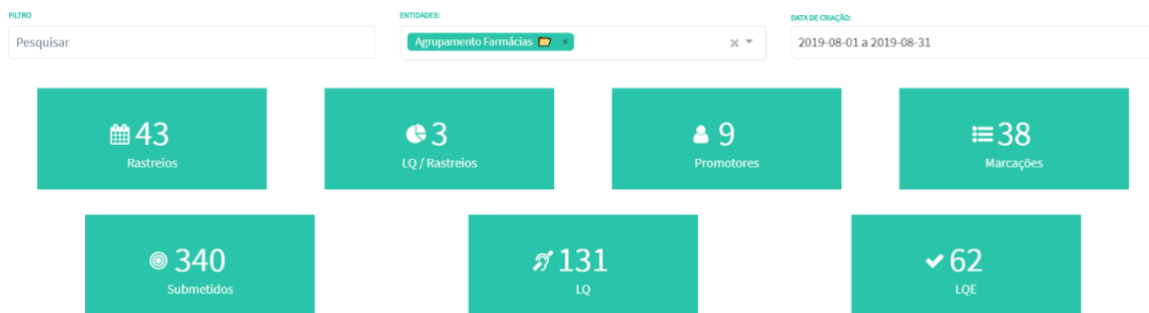
**Figura 47 - Interface de visualização de ICal gerado**

### 4.5.3.7. Métricas e Estatísticas de agendamentos

De modo a permitir avaliar o sucesso do projeto foi necessária a implementação de métricas e KPI (*Key Performance Indicator*) de visualização, para que fosse possível determinar vários aspetos sobre o projeto em si, como quais as farmácias de onde provêm mais *leads*, destas quantas passam a qualificadas, quais os promotores mais eficientes (uma vez que os promotores desempenham um papel essencial neste modelo de negócio, sendo que é necessário encorajar as pessoas a efetuarem o teste auditivo), entre outros.

Com o recurso a todos estes dados, torna-se possível potencializar o projeto, de modo a torná-lo mais eficiente, tentando descobrir padrões, como por exemplo quais as melhores regiões de Portugal para que se possa alocar mais rastreios e promotores, de forma a angariar o maior número de *leads*.

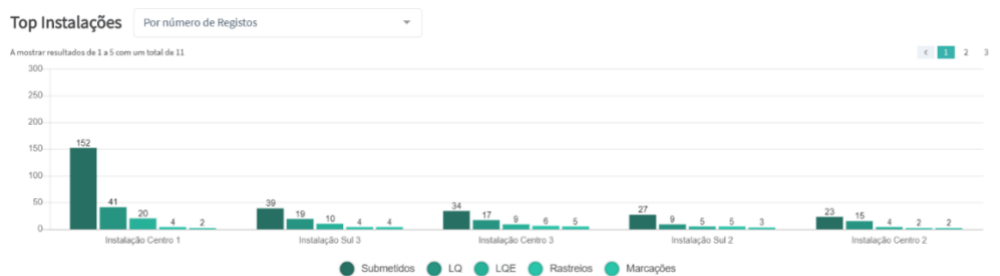
Assim, foi pedido um painel de estatísticas onde se pudesse visualizar de forma imediata dados gerais acerca do projeto (ver Figura 48), como: (1) o número de rastreios efetuados; (2) o número de promotores; (3) o número de marcações efetuadas, através da plataforma para todos os rastreios; (4) o número de testes auditivos realizados; (5) o número de testes auditivos realizados onde foi detetado sinais de perda auditiva; e (6) o número de *leads* qualificadas provenientes dos testes realizados.



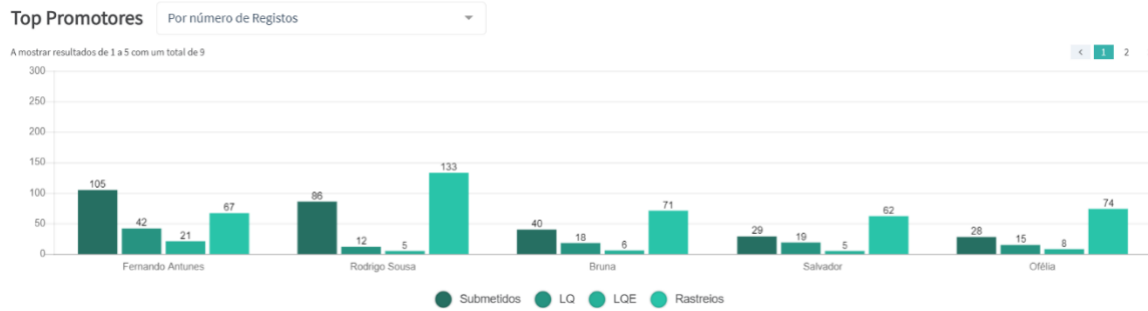
**Figura 48 - Métricas e filtros gerais dos rastreios**

Para além de todos estes dados, era necessário visualizar tanto o top de farmácias, como o top de promotores para inúmeras métricas definidas pelos responsáveis do projeto (ver Figura 49 e Figura 50).

De modo a possibilitar aos utilizadores uma melhor experiência de visualização foram realizados filtros, como: (1) por um intervalo temporal; (2) por um conjunto de entidades; (3) por agrupamentos específicos de entidades; e (4) por promotores específicos.



**Figura 49 - Top Instalações**



**Figura 50 - Top Promotores**

#### 4.5.4. Serviço de Registos Clínicos de Audiologia

O módulo de registos clínicos é um serviço que irá ser providenciado pela *Evollu* com o objetivo de realizar todo o acompanhamento de consultas específicas de audiologia provenientes do Serviço de Agendamento. O intuito deste módulo é processar e armazenar todos os dados dos utentes, como, por exemplo, os seus aparelhos auditivos, a anamnese que consiste no historial clínico do utente, como também os sintomas narrados por este, e o histórico de todos os testes auditivos realizados, quer através da aplicação móvel da *Evollu*, quer realizados em consultas acompanhadas por audiologistas. Todos estes dados correspondem à informação necessária para existir um prognóstico e um respetivo tratamento auditivo nos utentes. É importante referir que as informações anteriormente enunciadas foram determinadas por um audiologista certificado integrante na *Evollu*.

No futuro, com a introdução de outras áreas da atuação, como a nutrição, oftalmologia, entre outros, será necessário a criação de novos módulos representativos de registos clínicos para cada uma destas áreas.

Foi requisitada a implementação de todo o *front-end* necessário para ser possível o audiologista realizar as suas consultas, como também guardar toda a informação imprescindível para as mesmas, sendo que toda a API que a plataforma iria consumir já se encontrava desenvolvida por um outro membro da equipa.

Em termos de requisitos foi apenas solicitado que toda a gestão de conteúdos das consultas fosse realizada na mesma página, de modo a tornar a plataforma o mais simples possível. Tendo isto em conta, foi decidido que o serviço de registos clínicos iria consistir apenas numa página e, basear-se-ia num processo em dois passos, sendo eles respetivamente, na

seleção de um paciente através da listagem de utentes e, na gestão de todas as informações clínicas do mesmo, como mostra a Figura 51 e a Figura 52.

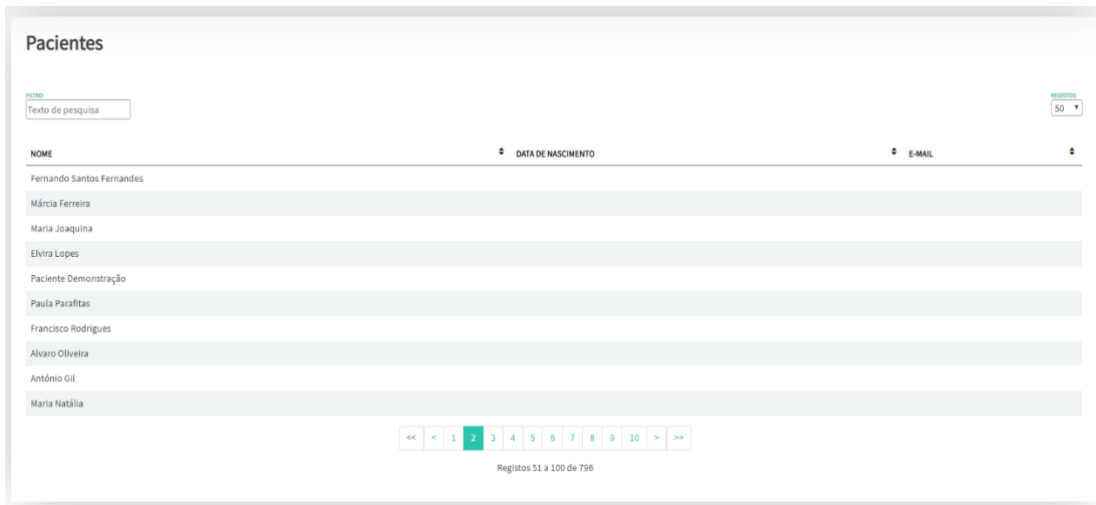


Figura 51 - Listagem de pacientes - audiologia

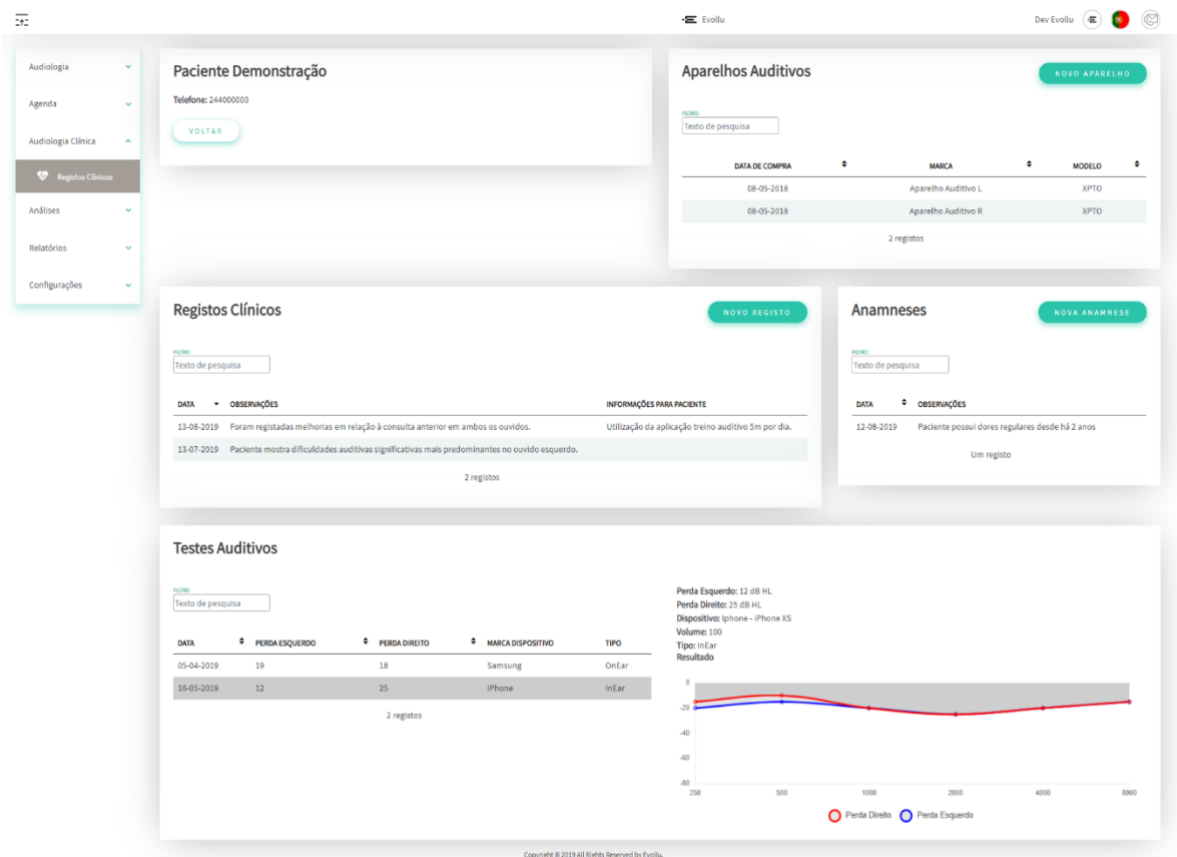


Figura 52 – Gestão de informação clínica de paciente

É necessário relembrar que todo este módulo ainda não se encontra a ser comercializado para clientes externos, sendo que apenas se encontra em utilização, por audiologistas pertencentes à *Evollu*, para a realização das suas consultas. A sua utilização acaba por

corresponder a uma *alpha release*, visto que só membros internos possuem acesso à mesma, e no futuro irão ser adicionadas novas funcionalidades, tais como, a possibilidade de realização de consultas através da internet com recurso a uma chamada de vídeo e, também uma melhoria em toda a UI.

#### **4.6. Autorização – Permissões e *Roles* de utilizadores**

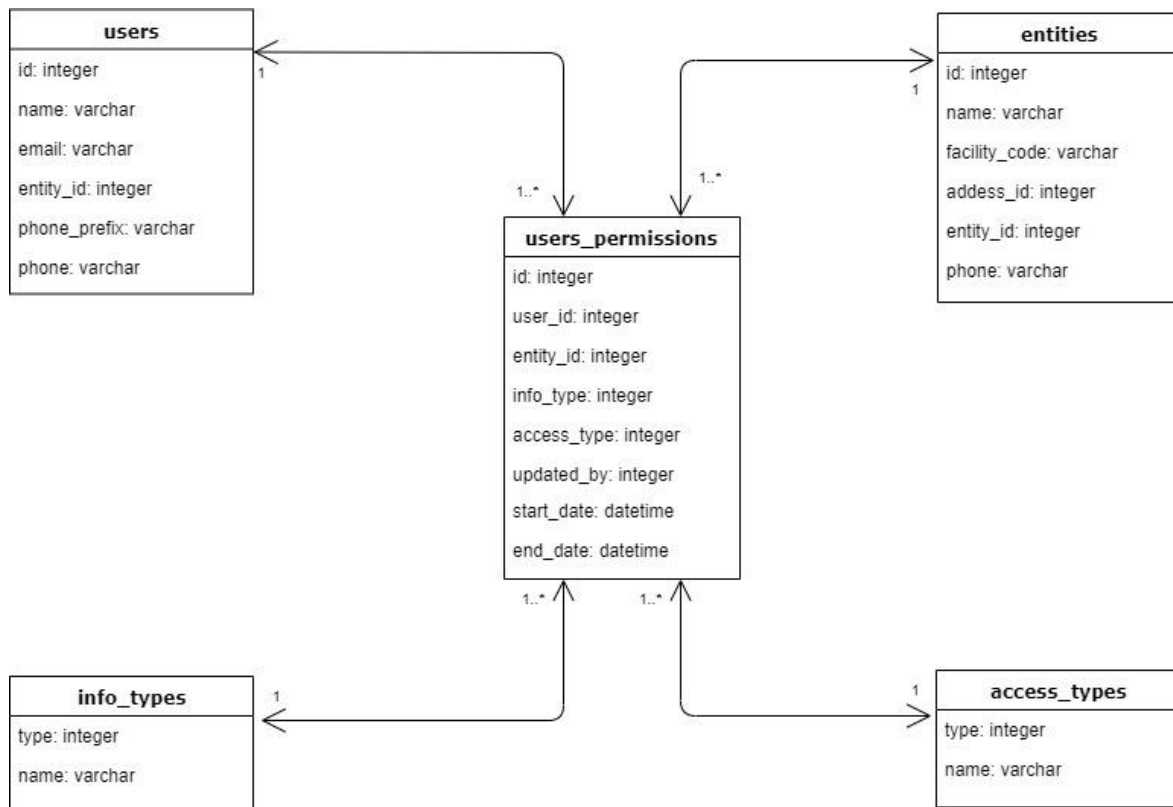
A autorização é uma prática bastante importante em qualquer empresa, pois esta garante a segurança de software. A autorização acontece depois da identidade do utilizador ser validada pelo sistema e, permite possibilitar o acesso total a recursos específicos como informações, ficheiros, bases de dados, entre outros.

Numa arquitetura baseada em micro serviços, cada um destes deve ser capaz de autorizar um utilizador a realizar as inúmeras ações disponibilizadas por estes [43]. Deste modo, todos os micro serviços utilizam mecanismos de autorização semelhantes, o que levou a que toda a lógica associada à mesma fosse extraída e adicionada ao projeto *Git* (o *Git* é um sistema de controlo de versões, que permite facilitar o desenvolvimento de *software* em equipa [44]) de onde todos os micro serviços na *Evollu* são originados.

Uma das desvantagens da implementação de uma arquitetura em micro serviços como mencionado na secção 2.4.2, é que foi necessário implementar toda a lógica de autorização como as migrações das tabelas, os controladores e tudo o resto, tanto em *JavaScript* como em PHP, visto que existem micro serviços implementados com linguagens distintas, o que por sua vez pode resultar em obstáculos no desenvolvimento de software.

Numa fase inicial, foi implementado um sistema de permissão associado a um utilizador que, de forma sucinta, descreve o acesso a um tipo de informação correspondente a um utilizador para uma dada entidade.

Para implementar o descrito acima, foram utilizadas um total de cinco classes em que cada classe representa uma tabela na bases de dados (ver diagrama de classes na Figura 53).



**Figura 53 - Diagrama de classes intervenientes no processo de autorização**

A tabela *access\_types* descreve o nível de acesso associado a uma permissão, sendo que os valores possíveis variam entre leitura, escrita e administrador, embora estes estejam dependentes de serviço para serviço e alguns apenas possam necessitar dos dois primeiros. Por sua vez, a tabela *info\_types* possui todos os possíveis tipos de informação, estando associados ao tipo de informação que cada serviço abrange.

As tabelas *users* e *entities*, como indicado na secção 4.5.1.3, dispõem, respetivamente, dos utilizadores registados na plataforma e, das várias entidades às quais os utilizadores pertencem.

Por fim, a tabela *users\_permissions* pretende agrupar todas as informações das tabelas caracterizadas anteriormente, onde cada registo representa uma permissão dada a um utilizador sobre uma entidade para um tipo específico de informação e, em termos de gestão de permissões foi decidido, em equipa, que cada utilizador poderia dar apenas as permissões que o mesmo possuía.

Esta abordagem procurou ir ao encontro dos requisitos originais e mostrou-se bastante flexível face ao número de permissões que iriam crescendo consoante o desenvolvimento de cada serviço na plataforma.

Com a adição de rastreios auditivos no modelo de negócio existente na *Evollu*, detetou-se que o sistema de permissões atual poderia ser melhorado, uma vez que, com a criação de um grande volume de entidades existiam bastantes utilizadores que tinham a necessidade de obter permissões para as mesmas.

Assim, com a criação de uma nova entidade, seria necessário atribuir em média entre 10 a 15 permissões distintas por todos os serviços existentes, para um variado número de utilizadores, o que a longo prazo poderá resultar num elevado número de linhas na base de dados, apesar da enorme flexibilidade fornecida por este tipo de solução.

Tendo em conta todos os fatores enunciados acima, ponderou-se a substituição de permissões para *roles*. Uma *role* em termos práticos é constituída por um aglomerado de permissões, onde, para o caso enunciado acima, só seriam necessárias entre 2 a 4 *roles*.

Com esta substituição, pretendeu-se não só solucionar o problema do elevado número de linhas, mas também, proporcionar um sistema de autorização que fosse conhecido por parte dos utilizadores finais, visto que a maioria das plataformas deste formato implementam este tipo de solução, como por exemplo *Google Cloud Platform*, *Amazon Web Services*, entre outros.

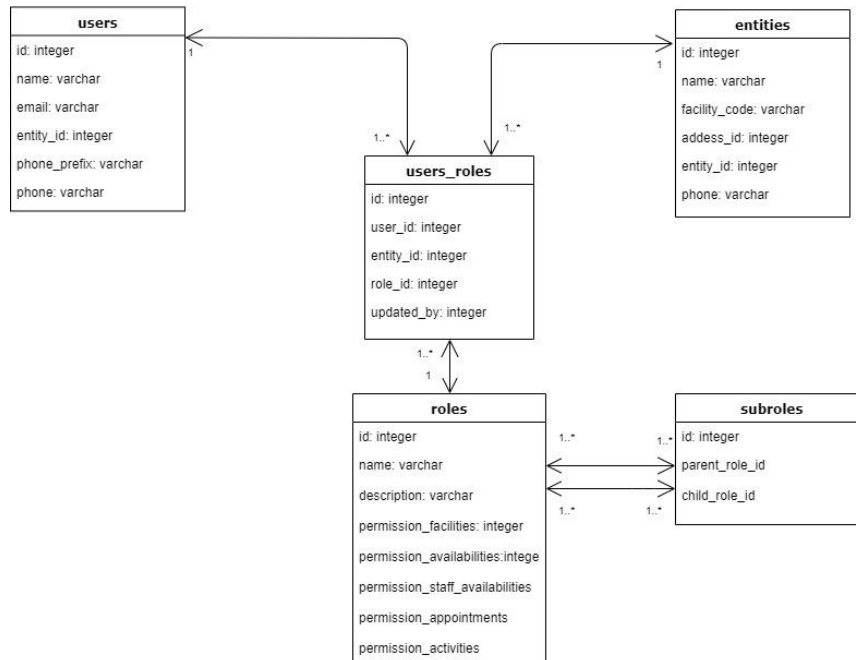
Na Tabela 2 é possível observar uma análise comparativa entre a utilização de permissões e *roles* para o mesmo problema. O problema consiste na atribuição de permissões ou *roles* para 300 entidades distintas e 100 utilizadores, o que é bastante semelhante aos casos que se verificam na plataforma da *Evollu*.

**Tabela 2 - Análise comparativa do nº linhas entre permissões e *roles***

Abordagem	Nº Entidades	Nº Utilizadores	Nº Permissões / Roles	Total Linhas
Permissões	300	100	10	300 000
Roles	300	100	3	90 000

Desta forma, é possível verificar uma enorme diferença entre o número de linhas das duas soluções. Em suma, a segunda abordagem resultou numa redução de 70% nas mesmas em relação à primeira, mantendo o mesmo nível de granularidade. Deste modo, foi realizada a substituição da tabela *users\_permissions* por *user\_roles* que possui referências diretas para um utilizador, uma entidade e uma *role*. Foi ainda introduzida uma outra tabela de *roles*,

responsável por armazenar as informações da mesma, como o nome da *role*, uma breve descrição do que a *role* permite realizar, e todas as permissões associadas à mesma, como se pode observar pelo diagrama representa na Figura 54.



**Figura 54 - Diagrama de classes - sistema de roles**

Todavia, a implementação de um sistema de autorização baseado em roles possui algumas desvantagens em relação à de permissões únicas, sendo elas:

- Perda da possibilidade de atribuição de permissões entre intervalos de tempo, que apesar da flexibilidade que poderia possibilitar era algo que até ao momento ainda não tinha sido muito utilizado.
- Para a atribuição de *roles*, era necessário que um utilizador tivesse todas as *roles* que pode atribuir, porque apenas é possível atribuir *roles* que se possam.
- Para o utilizador final, poderá não ser claro identificar as informações às quais uma *role* fornecia acesso, visto ser um conjunto de várias permissões.

Para colmatar algumas das desvantagens enunciadas, foi decidido a criação de uma tabela *pivot* denominada de *subroles*, esta responsável por mapear uma *parent\_role* a uma *child\_role*, como é possível observar pela Figura 54. Esta tabela especifica todas as *roles* filho que uma *role* pai pode atribuir, isto é, um utilizador com a *role* A pode conceder a outros utilizadores a *role* A, assim como todas as *child roles* que se encontrem na tabela de *subroles* associadas à mesma.

No entanto, foi verificado que em certos serviços, como o Serviço de Registos Clínicos de Audiologia, por se tratar de dados restritos, existe a necessidade de existirem permissões ao nível de um recurso, isto é, ao invés de proporcionar acesso a um audiologista a todos os registos clínicos de um utente ou entidade, poderão existir certos casos, onde este apenas tem acesso a um registo clínico específico. Neste caso, a solução de *roles* não funciona com estes tipos de casos. Assim, para todos os serviços que necessitem deste tipo de permissões específicas são mantidos ambos os esquemas de tabelas, sendo que o serviço é responsável por verificar nestas, se o utilizador tem autorização para realizar determinada ação.

#### **4.6.1. Super User**

Com a necessidade de avaliar todo o projeto e, como todos os utilizadores só teriam acesso às entidades nas quais teriam permissões, tornou-se necessário implementar uma solução onde para utilizadores internos, pertencentes à *Evollu*, não fosse necessário a atribuição de *roles* para as poder visualizar, visto que se faltassem permissões para uma entidade iria provocar inconsistências na visualização das mesmas. A solução encontrada acabou por ser bastante simples, dado que foi decidido que se um utilizador tivesse uma *role* para uma *entity\_id* com valor nulo, significava que esse utilizador possuía essa *role* para todas as entidades existentes, o que facilitou bastante o facto de garantir que a qualquer momento um utilizador tivesse permissões para todas as entidades.

#### **4.6.2. Gestão e Atribuição de Permissões**

De modo a proporcionar a melhor experiência de utilização possível na atribuição de *roles*, foi possibilitado ao utilizador duas vistas semelhantes que agrupam os dados de forma distinta, isto é, a possibilidade de visualizar as *roles* por entidade ou por *role* como se pode verificar na Figura 55 e na Figura 56. Foi também adicionado um campo de pesquisa que permite filtrar a informação disponível, uma vez que consistiu numa das problemáticas evidenciadas pelos utilizadores na UI anterior.

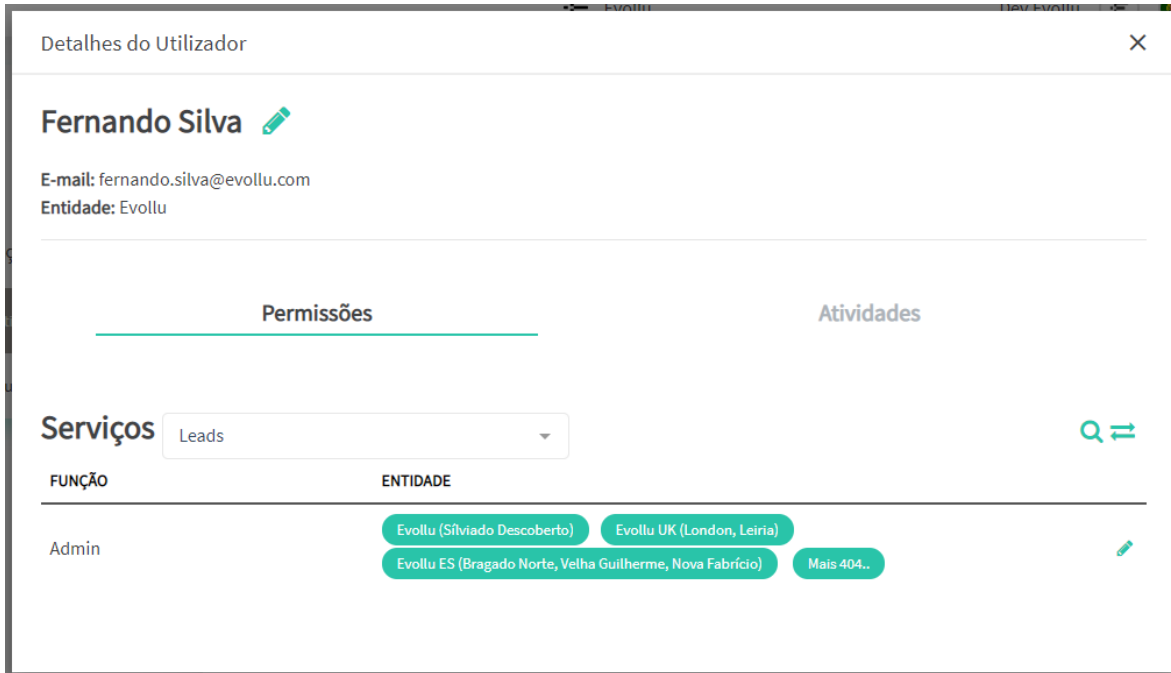


Figura 55 - Vista permissões agrupadas por permissão

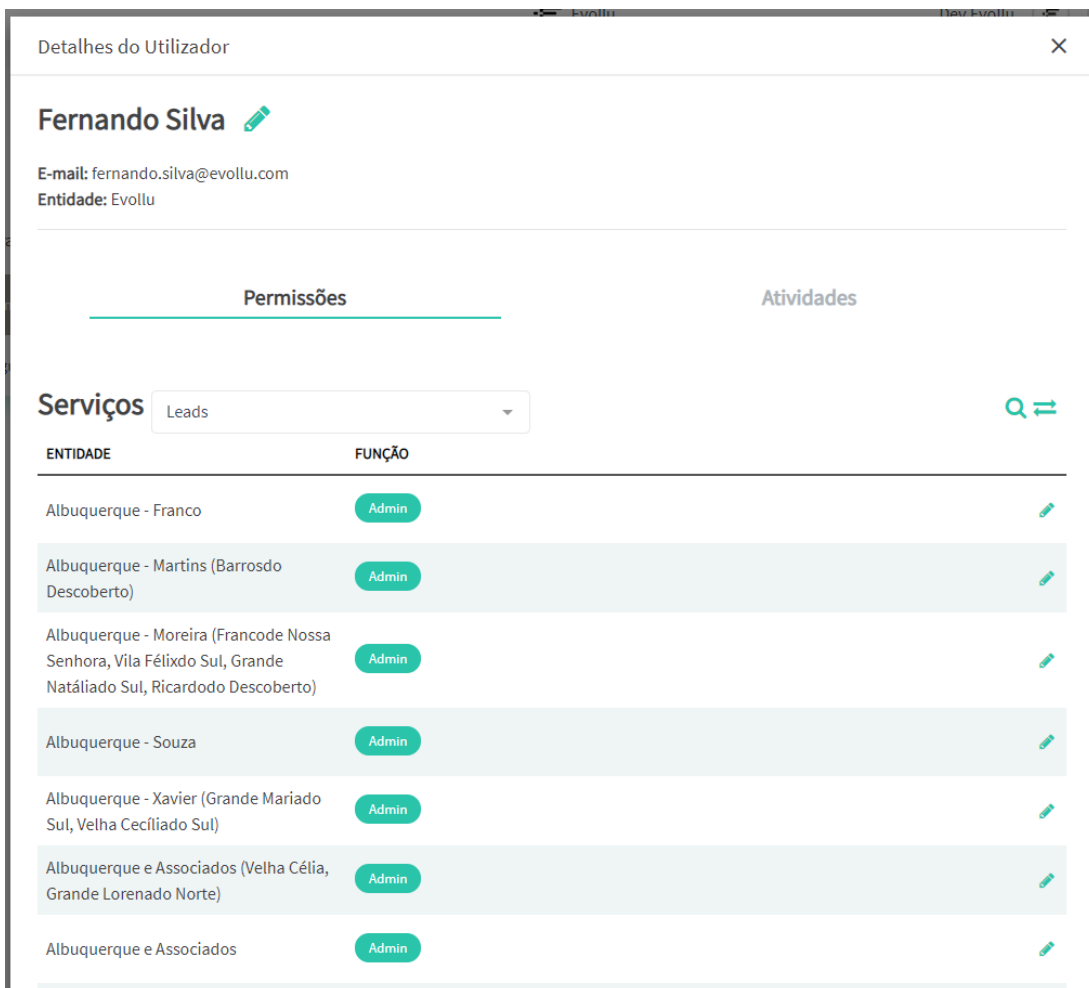


Figura 56 - Vista permissões agrupadas por entidade

Foram necessários cuidados na implementação da funcionalidade de atribuição de permissões, visto existirem certos casos específicos que requerem condições adicionais como é o caso das *super roles* mencionadas na secção 4.6.1, uma vez que se trata de um caso especial, não é possível retirar essa *role* a um utilizador. A partir da UI implementada apenas é permitido atribuir ou retirar as *roles* que um utilizador possui para uma dada entidade. Caso um utilizador seja um *super user*, este poderá atribuir essa mesma *role* a qualquer utilizador para qualquer entidade.

#### **4.7. Notificações Automatizadas**

A plataforma da *Evollu* pode considerar-se uma plataforma ampla devido a todas as funcionalidades que providencia, assim foi necessário a automatização de algumas tarefas para tornar a gestão de todos os processos mais acessíveis e simples, caso contrário seria necessário a perda de tempo significativo a realizar certas funções.

Deste modo, a utilização de notificações automáticas foi realizada através da utilização de *Cron Jobs*. *Cron Jobs* é uma ferramenta *Linux* que permite planear a realização de tarefas automáticas em intervalos de tempo regulares [45].

Alguns dos casos mais comuns onde este tipo de ação se pode revelar extremamente útil é em casos como os descritos anteriormente, onde é necessário um sistema de notificações, cópias de bases de dados, ou até tarefas que podem levar algum tempo a serem executadas.

Foi decidido que algumas das notificações teriam de ser enviadas através da utilização de envio de mensagens SMS (*Short Message Service*), devido ao facto da perda auditiva ser mais predominante em populações mais envelhecidas, como referido na secção 2.2, o que significa que parte destas pessoas não possuem endereços de e-mail.

De entre as opções disponíveis para o envio de SMS, foi escolhida a utilização de uma API, visto que a outra opção envolveria adquirir um servidor físico com a única finalidade de enviar SMS, o que para uma primeira fase não se revelou necessário.

Assim, foi realizado um estudo a várias API existentes, em que o objetivo seria o de procurar a solução mais barata, dado que este é um serviço pago e porque o envio destas iria ser utilizado em diversas funcionalidades, o que poderia refletir-se em preços elevados ao fim de algum tempo. A Tabela 3 representa várias API analisadas, onde, com a aprovação da

*Evollu*, foi escolhida a *Gateway API* para o envio das mensagens, uma vez que possui os preços mais em conta.

**Tabela 3 - Análise comparativa de várias API de envio de SMS**

API Provider	Preço	Observações
Gateway API	0.021€	Possuí documentação em diversas linguagens de como realizar o envio utilizando a respetiva API.
SMS API	0.026€	Permite receber SMS de volta.
Twilio API	0.045€	Possuí documentação em diversas linguagens de como realizar o envio utilizando a respetiva API.
ClockWork	0.06€	Preço elevado.

A Tabela 4 representa uma listagem de todas as notificações que foram implementadas com o recurso a *Cron Jobs*.

**Tabela 4 - Listagem de notificações automatizadas**

Notificação	Destinatários	Tipo de Notificação	Frequência
Envio de <i>QR Code</i>	Promotor de Rastreio Auditivo	Email	Dia anterior às 18h
Notificação de Rastreios	Responsável da Farmácia	Email	Dois dias anteriores às 17h
Notificação de Consultas	Utente	Email / SMS	Dia anterior às 18h
Estado Quiosques Físicos	<i>Evollu</i>	<i>Email</i>	Todos os dias às 9h
Licenças de Quiosques a Expirar	<i>Evollu</i>	<i>Email</i>	Uma semana antes de expirarem às 9h

Se seguida são descritos os diversos tipos de notificações implementados.

### **Envio *QR Code* Automático**

Devido ao facto de parte dos promotores serem pessoas subcontratadas por empresas externas que participam em eventos, congressos ou lançamentos de novas marcas e produtos, fez com que a maioria dos promotores sejam de faixas etárias baixas e, que deste modo, existisse uma grande rotatividade de promotores. As especificidades inerentes aos mesmos, levou a que estes não utilizassem a plataforma com regularidade, onde era possível visualizar os seus próximos rastreios auditivos.

Tendo isto em conta, todos os dias o responsável do projeto pertencente à *Evollu* teria de enviar um email aos promotores que tivessem um rastreio designado para o dia seguinte com a informação do *QR Code*, para que estes pudessem iniciar a sessão quando chegassem à farmácia para que todas as *leads* ficassem registadas.

### **Notificação de Rastreios**

Também se revelou a necessidade de informar as farmácias da ocorrência de futuros rastreios auditivos, para que pudessem ser tomadas as medidas necessárias para o evento e, também para lembrar as mesmas da necessidade de realizar possíveis marcações de utentes para os rastreios.

### **Notificação de Consultas**

Por sua vez, todos os utentes das marcações associadas tanto a rastreios como a consultas gerais, deveriam ser notificados para lembrar os mesmos dos seus compromissos, para que pudessem comparecer nos mesmos.

### **Estado dos Quiosques Físicos**

Dado que os quiosques físicos se encontram em farmácias onde é possível realizar o teste auditivo sem nenhum tipo de acompanhamento, caso acontecesse algo com os quiosques, seria necessário procurar solucionar o problema o mais rapidamente possível com o objetivo de poder ser utilizado pelas pessoas.

Assim, através da utilização da API do *Hexnode*, que permite verificar o estado dos quiosques e realizar ações remotamente sobre os mesmos, era necessário verificar todos os dias se eles se encontravam a comunicar, caso contrário seria enviado um e-mail para a

*Evollu* a notificar quais os quiosques que não se encontravam a comunicar, assim como a razão para o mesmo, que, usualmente, seria ou por falta de bateria ou por falta de WIFI.

### **Licenças de Quiosques a Expirar**

Na compra de um quiosque tanto físico como portátil, é criada uma licença com a duração padrão de 1 ano. Numa fase inicial, a quantidade de quiosques não justificava a automatização da tarefa, mas com o crescimento na venda de quiosques nos vários países, onde atualmente já foram comercializados aproximadamente 110, foi necessário enviar uma notificação para relembrar a membros da equipa de suporte da *Evollu* que as licenças estariam a terminar, para se dar início ao processo de renovação.

## 5. Processo de Desenvolvimento de Testes

O desenvolvimento de software caracteriza-se por um processo complexo. Desta forma, em qualquer etapa do seu desenvolvimento poderão ocorrer inúmeras falhas. É necessário identificá-las e mitigá-las o mais previamente possível, de modo a interromper a sua propagação e reduzir os custos de verificação e correção associados. Para além disso, os testes asseguram a qualidade de um produto, pois estes são capazes de validar todos os fluxos de execução possíveis e não o que é expectável de acontecer [46].

Até ao momento, a implementação de testes ao nível do código como testes unitários e testes de integração não foram de modo algum um requisito durante a implementação dos mesmos, uma vez que algumas vezes devido à pressão da *Evollu* angariar novos clientes ou mesmo por clientes terem solicitado funcionalidades de carácter urgente, foi necessário a implementação das mesmas rapidamente, sem que houvesse uma fase de elaboração de testes, ainda assim as principais funcionalidades e *endpoints* cruciais em cada um dos serviços possui testes.

No servidor de autenticação, pela implementação em *NodeJS*, foram utilizadas as bibliotecas *chai* e *chai-http*, que de um modo simples são bibliotecas de asserção BDD (*Behaviour Driven Development*)/TDD (*Test Driven Development*) para *JavaScript* e *browser* e, também permitem simular pedidos HTTP. Em conjunto com estas bibliotecas, foi ainda utilizado uma biblioteca denominada de *mocha*, que permite a realização do teste em si e permite a realização de testes assíncronos.

Quanto aos restantes servidores, nomeadamente desenvolvidos em PHP na *framework Laravel*, foi utilizado o *PHPUnit* em conjunto com as classes que auxiliam na realização dos testes já implementadas no projeto raiz da *framework Lumen*.

Ainda assim, o processo de desenvolvimento realizado na *Sensing Evolution*, no que diz respeito à utilização do *Git* conta com a utilização de *feature branches*, onde todas as funcionalidades são implementadas. Com este procedimento, é possível realizar a integração para o *branch master* apenas quando toda a funcionalidade se apresentar completa. O *branch master* é onde se encontram todas as funcionalidades que são realizadas numa *sprint*. Na data final do *sprint*, todas as funcionalidades que se encontrem prontas em *master* são conduzidas para um *branch staging*, onde permanecem durante uma ou duas semanas para

serem testadas, tanto pela equipa de desenvolvimento, como por outros elementos pertencentes à *Evolu* e só após estes concederem aprovação é que se torna possível aplicar estas nos ambientes de produção.

Desta forma, independentemente da inexistência de um elevado número de testes, é possível constatar que todas as funcionalidades são testadas generosamente antes de prosseguir para ambientes de produção.

## 6. Conclusão

Com o término do estágio na *Sensing Evolution*, é possível afirmar que o mesmo foi bastante desafiante tanto a nível profissional como a nível pessoal e, revelou-se uma experiência deveras enriquecedora, devido a todas as competências adquiridas e, pelo modo como me preparou para o ambiente empresarial.

As dificuldades sentidas prenderam-se ao facto da empresa ainda se encontrar numa fase de crescimento bastante inicial, o que conseqüentemente instruiu uma maior necessidade de assumir responsabilidades, e por outro lado, uma maior desorganização devido à inexistência de processos associados à mesma; às frequentes mudanças nos requisitos, devido à tentativa de previsão de funcionalidades e de ir ao encontro das necessidades dos clientes e; por último, devido à pressão de submeter uma nova versão. Por vezes foi necessário acelerar o processo de desenvolvimento, e desta forma não foi possível realizar um adequado número de testes unitários e de integração.

Toda a plataforma acabou por ser reimplementada tanto a nível arquitetural, como também a nível de interfaces, sendo que a maior parte das funcionalidades já existentes foram sofrendo alterações no decorrer do estágio. O maior desafio encontrado foi o desenvolvimento da plataforma numa arquitetura baseada em micro serviços, pois esta arquitetura representa uma área não abordada no decorrer de todo o percurso académico. Um outro desafio foi toda a gestão do micro serviço de agendamento dado à complexidade que o sistema exigiu, como também devido às constantes mudanças nos requisitos do mesmo.

Durante o decorrer do estágio foi realizado *continuous delivery*, em média foram efetuadas atualizações da plataforma de 3 em 3 semanas. Desta forma, foi exigida uma maior responsabilidade e cuidados na realização da mesma para garantir uma maior credibilidade e confiança em toda a plataforma. Isto levou a que conhecimentos adicionais fossem adquiridos no âmbito de *DevOps* uma vez que requereu a participação em todos os processos e tarefas de atualização dos servidores do ecossistema da *Evollu*.

Apesar dos escassos testes realizados ao código, foi possível testemunhar a importância dos mesmos, uma vez que devido à frequente mudança de requisitos e às alterações que estas mesmas induziram, pode-se afirmar que a existência de uma inúmera quantidade de testes

são uma mais valia para que todas as mudanças efetuadas no futuro possam ser feitas sem que haja danos funcionais.

No futuro, a *Evollu* tem como objetivo introduzir canais de angariação de *leads* na internet, através de um teste auditivo realizado no browser. Para além disto, é necessária a introdução de mais funcionalidades na plataforma existente, assim como alargar os serviços disponibilizados e realizar um *redesign* para melhorar certos aspectos de UX.

O balanço final do estágio é nitidamente positivo, todas as tarefas e desafios foram ultrapassados com sucesso. Ambas as partes encontram-se satisfeitas, uma vez que à data final do estágio foi recebido um convite para continuar a colaborar com a empresa e, com todo o agrado, o mesmo foi aceite.

## Bibliografia

- [1] Evollu, “Evollu,” 2018. [Online]. Available: <https://evollu.com/pt/index>. [Accessed: 12-Oct-2018].
- [2] Evollu, “Evollu,” 2018. [Online]. Available: <https://evollu.com/pt/whoweare>. [Accessed: 12-Oct-2018].
- [3] “Evollu.” [Online]. Available: <https://evollu.com/pt/ecosystem>. [Accessed: 13-Oct-2018].
- [4] “Audiologia - Ponte Saúde.” [Online]. Available: <http://www.pontesaude.pt/events/audiologia/>. [Accessed: 21-Oct-2018].
- [5] S. N. de Saúde, “Infográfico INSEF – Dificuldades auditivas - INSA,” 2018. .
- [6] I. N. D. R. Jorge, “Dificuldades auditivas,” 2015.
- [7] M. Som, “Deficiência auditiva afeta um milhão em Portugal,” 2014. [Online]. Available: <https://www.minisom.pt/artigos/deficiencia-auditiva-afeta-um-milhao-em-portugal>.
- [8] W. H. Organization, “A perda auditiva é um problema de saúde pública,” 2017. [Online]. Available: <https://www.dgs.pt/em-destaque/dia-mundial-da-audicao-3-de-marco-ppsx.aspx>.
- [9] “Web Application Development Process Flow - EXISTEK - Medium.” [Online]. Available: <https://medium.com/existek/web-application-development-process-flow-5645e2683b5d>. [Accessed: 08-Dec-2018].
- [10] Martin, “How Basic Web Development Languages HTML and CSS work,” *CLEVERISM*, 2015. .
- [11] “Seven Principles of Modern Web App Development - DZone Web Dev.” [Online]. Available: <https://dzone.com/articles/7-principles-of-modern-web-application-development>. [Accessed: 02-Nov-2018].
- [12] “Web Application Architecture | Types of Web Application Architecture.” [Online]. Available: <https://www.sataware.com/app-development/web-application-architecture/>. [Accessed: 16-Dec-2018].
- [13] “15 benefits of software architecture you should know | Apiumhub.” [Online]. Available: <https://apiumhub.com/tech-blog-barcelona/benefits-of-software-architecture/>. [Accessed: 27-Nov-2019].
- [14] “Monolithic Architecture pattern.” [Online]. Available:

- <https://microservices.io/patterns/monolithic.html>. [Accessed: 17-Mar-2019].
- [15] M. Villamizar *et al.*, “Evaluating the monolithic and the microservice architecture pattern to deploy web applications in the cloud,” in *2015 10th Computing Colombian Conference (10CCC)*, 2015, pp. 583–590.
- [16] “Microservice Architecture pattern.” [Online]. Available: <https://microservices.io/patterns/microservices.html>. [Accessed: 17-Mar-2019].
- [17] J. Soldani, D. A. Tamburri, and W.-J. Van Den Heuvel, “The pains and gains of microservices: A Systematic grey literature review,” *J. Syst. Softw.*, vol. 146, pp. 215–232, Dec. 2018.
- [18] G. McGrath and P. R. Brenner, “Serverless Computing: *Design, Implementation, and Performance*,” in *2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, 2017, pp. 405–410.
- [19] Amazon Web Services, “Computação sem servidor – Amazon Web Services,” 2019. [Online]. Available: <https://aws.amazon.com/pt/serverless/>. [Accessed: 05-Apr-2019].
- [20] T. Dybå and T. Dingsøy, “Empirical studies of agile software development: A systematic review,” *Inf. Softw. Technol.*, vol. 50, no. 9–10, pp. 833–859, Aug. 2008.
- [21] A. Manifesto, “Principles behind the Agile Manifesto,” 2001. [Online]. Available: <https://agilemanifesto.org/iso/en/principles.html>. [Accessed: 03-May-2019].
- [22] D. WEST, “Agile Scrum Roles | Atlassian,” 2019. [Online]. Available: <https://www.atlassian.com/agile/scrum/roles>. [Accessed: 06-May-2019].
- [23] D. WEST, “Sprint Planning | Atlassian,” 2019. [Online]. Available: <https://www.atlassian.com/agile/scrum/sprint-planning>. [Accessed: 06-May-2019].
- [24] D. WEST, “Sprints | Atlassian,” 2019. [Online]. Available: <https://www.atlassian.com/agile/scrum/sprints>. [Accessed: 06-May-2019].
- [25] D. RADIGAN, “Three steps to better sprint reviews | Atlassian,” 2019. [Online]. Available: <https://www.atlassian.com/agile/scrum/sprint-reviews>. [Accessed: 06-May-2019].
- [26] SCRUM PORTUGAL, “Scrum - framework de desenvolvimento de produtos, abordagem ágil,” 2017. [Online]. Available: <http://www.scrumportugal.pt/scrum/>. [Accessed: 11-Sep-2019].
- [27] D. A. IBM; Fachat, “Challenges and benefits of the microservice architectural style – IBM Developer,” 2017. [Online]. Available: <https://developer.ibm.com/articles/challenges-and-benefits-of-the-microservice->

- architectural-style-part-1/. [Accessed: 17-May-2019].
- [28] NodeJS, “About | Node.js,” 2019. [Online]. Available: <https://nodejs.org/en/about/>. [Accessed: 22-Sep-2019].
- [29] “Express - Node.js web application framework.” [Online]. Available: <https://expressjs.com/>. [Accessed: 22-Sep-2019].
- [30] Jorge Amado Soria Ramírez, “PHP: Anatomía del ma diseño.”
- [31] “Lumen - PHP Micro-Framework By Laravel.” [Online]. Available: <https://lumen.laravel.com/>. [Accessed: 22-Sep-2019].
- [32] “Lumen Programming Guide: Writing PHP Microservices, REST and Web Service APIs - Paul Redmond - Google Livros.” [Online]. Available: [https://books.google.pt/books?hl=pt-PT&lr=&id=6ksiDQAAQBAJ&oi=fnd&pg=PR6&dq=what+is+the+use+of+lumen+php&ots=fzNHMEpB4e&sig=ThCWBtcD-bQZ6oLoELyn3cwPA\\_8&redir\\_esc=y#v=onepage&q=what is the use of lumen php&f=false](https://books.google.pt/books?hl=pt-PT&lr=&id=6ksiDQAAQBAJ&oi=fnd&pg=PR6&dq=what+is+the+use+of+lumen+php&ots=fzNHMEpB4e&sig=ThCWBtcD-bQZ6oLoELyn3cwPA_8&redir_esc=y#v=onepage&q=what+is+the+use+of+lumen+php&f=false). [Accessed: 22-Sep-2019].
- [33] “Vue.js.” [Online]. Available: <https://vuejs.org/>. [Accessed: 10-Aug-2019].
- [34] “Bootstrap · The most popular HTML, CSS, and JS library in the world.” [Online]. Available: <https://getbootstrap.com/>. [Accessed: 22-Sep-2019].
- [35] “webpack.” [Online]. Available: <https://webpack.js.org/>. [Accessed: 22-Sep-2019].
- [36] “WebAuthn Introduction - Learn about Web Authentication W3C Spec.” [Online]. Available: <https://auth0.com/blog/introduction-to-web-authentication/>. [Accessed: 23-May-2019].
- [37] Yjvesa Balaj, “Token-Based vs Session-Based Authentication: A survey,” 2017.
- [38] “Uma introdução ao OAuth 2 | DigitalOcean.” [Online]. Available: <https://www.digitalocean.com/community/tutorials/uma-introducao-ao-oauth-2-pt>. [Accessed: 25-May-2019].
- [39] “JSON Web Token Introduction - jwt.io.” [Online]. Available: <https://jwt.io/introduction/>. [Accessed: 24-May-2019].
- [40] “Sobre a Glintt.” [Online]. Available: <https://www.glintt.com/pt/o-que-somos/sobreglintt/Paginas/default.aspx>. [Accessed: 10-Sep-2019].
- [41] “What Is A QR Code And Why Do You Need One? - Search Engine Land.” [Online]. Available: <https://searchengineland.com/what-is-a-qr-code-and-why-do-you-need-one-27588>. [Accessed: 12-Aug-2019].
- [42] “ICS File (What It Is & How to Open One).” [Online]. Available:

- <https://www.lifewire.com/ics-file-2622749>. [Accessed: 16-Aug-2019].
- [43] “Authentication and Authorization in Microservices - DZone Microservices.” [Online]. Available: <https://dzone.com/articles/authentication-and-authorization-in-microservices>. [Accessed: 22-Sep-2019].
- [44] Git-SCM, “Git - What is Git?” [Online]. Available: <https://git-scm.com/book/en/v2/Getting-Started-What-is-Git%3F>. [Accessed: 26-Sep-2019].
- [45] “A Beginners Guide To Cron Jobs - OSTechNix.” [Online]. Available: <https://www.ostechnix.com/a-beginners-guide-to-cron-jobs/>. [Accessed: 22-Sep-2019].
- [46] Luciano Baresi and Mauro Pezz`e, “An Introduction to Software Testing.”

## **Anexos**



## Anexo I – Tarefas Realizadas

Nome da tarefa	Data de conc...	Effort	Cost
✔ Correções agenda	31 jul	2	2
✔ Edição de disponibilidade	2 ago	1	1
✔ Alteração de rastreio	2 ago	1	1
✔ Listar disponibilidade sem staff	2 ago	7	7
✔ Agenda uk	26 jul	3	3
✔ Adicionar total de appointments no dashboard e envio resumo mensal desde inicio do projeto	24 jul	1	1
✔ Videos demo rastreios	19 jul	4	4
✔ Reporting and analytics for appointments API (basic KPIs)	19 jul	3	3
✔ Nome do iCal	12 jul	1	1
✔ Correções Agenda 3 🗨	5 jul	1	1
✔ Notificações farmácias	5 jul	2	3
✔ Reconfigurar roles de agenda	5 jul	1	1
✔ Dividir permissões facilities e novas colunas 6 🗨	14 jun	3	4
✔ Verificar facilities sem as persons corretas a desempenhar funções 5 🗨	26 jun	2	2
✔ Configuração CRON Jobs	24 jun	1	1
✔ Envio email automático para os promotores de uma sessão com a informação do QRCode	21 jun	1	1
✔ Testar datas e correções	14 jun	3	3
✔ Caldav 8 🗨	7 jun	4	8
✔ Handler de errors nos formulários 4 🗨	7 jun	3	3
✔ Testar refactor to start and end datetimes 8 🗨	7 jun	4	4
✔ Garantir que o duplo click no front-end não crie	11 jun	1	1
✔ Gerar sessão de rastreio automaticamente com recurso a encriptação 2 🗨	7 jun	2	2

Nome da tarefa	Data de conc...	Effort	Cost
✔ Testar Roles "Super"	31 maio	1	1
✔ Endpoint para rastreios	31 maio	1	1
✔ Email rastreio QRCode	31 maio	1	1
✔ Migrate start time and end time to datetime Front-end	21 maio	4	5
✔ Cancelar emails de consulta de marcações	24 maio	1	1
✔ Migrar persons para users em PROD	24 maio	1	1
✔ Atualização PDF's Clínica	17 maio	1	1
✔ Migrate start time and end time to datetime API	10 maio	6	8
✔ Permissões baseadas em Roles	10 maio	4	16
✔ Refactor Código	10 maio	3	3
✔ Roles (Vue)	10 maio	4	5
✔ Rastreios por distância (ebooking)	8 maio	2	3
✔ Converter Person para Users	3 maio	4	7
✔ Resolução e alteração erros clinical	2 maio	1	1
✔ Correções appointments	2 maio	2	2
✔ Roles em Booking	3 maio	2	2
✔ Rastreios apenas com um botão de edição 2 🗨	26 abr	2	2
✔ Permissões em /services	26 abr	2	2
✔ Marcações em Sessão de promoção 1 🗨	18 abr	1	1
✔ Modificações página screenings 6 🗨	16 abr	2	2
✔ Página de rastreios responsiva em mobile	11 abr	1	1
✔ Eliminação de rastrios	11 abr	2	2

Nome da tarefa	Data de conc...	Effort	Cost
✔ Adição de rastreio na listagem	5 abr	3	5
✔ Bug fixing teste 2	20 mar	4	4
✔ Link para leads na listagem	5 abr	3	3
✔ Remover BDs não utilizadas	29 mar	1	1
✔ Create Address based on entity request for facility	29 mar	1	1
✔ Fix clinical	29 mar	2	2
✔ Endpoint facilities mobile app	25 mar	1	1
✔ Marcações de períodos na agenda (Web)	22 mar	1	1
✔ Não dar para fazer select na agenda se não tiver permissões de escrita	15 mar	1	1
✔ Adicionar QR code às disponibilidades de staff 3 🗨	15 mar	3	3
✔ Criar tabela people com os campos adicionais e atualizar controladores	11 mar	3	3
✔ Atualizar ebooking-srv para não usar permissões do auth	1 mar	1	3
✔ Atualizar clinical-srv para usar permissões com entity_id	1 mar	1	1
✔ Middleware para guardar testes auditivos requerer apenas que seja uma pessoa em vez de paciente	1 mar	1	1
✔ Refactor endpoints	27 fev	2	2
✔ Alterações Agenda	26 fev	3	3
✔ Appointments mais responsivo e traduções	20 fev	1	1
✔ Permitir às farmácias fazer marcações de utentes nas promoções	20 fev	2	3
✔ Colocar feriados no calendário da agenda 1 🗨 1 🗨	28 jan	3	3
✔ Alterações	28 jan	2	2
✔ Dar permissões necessárias ao carlos para realizar as suas acções e migrar as disponibilidades antigas dele	29 jan	1	1
✔ Obrigar no front end a especificar o staff para a disponibilidade	24 jan	1	1

Nome da tarefa	Data de conc...	Effort	Cost
✔ Adicionar entity_id às permissions	23 jan	1	1
✔ CRUD facilities tendo em conta entities	23 jan	1	1
✔ Add hasAppointments field to services	23 jan	2	2
✔ Refactor agenda para utilizar a nova tabela dos serviços	23 jan	3	4
✔ Criação tabela facility service person 2 ↵	21 jan	4	4
✔ Adicionar entity_id às facilities	21 jan	2	2
✔ Ao atualizar disponibilidade se poder atualizar staff availability atualizar	18 jan	2	2
✔ Preselect service person if auth user in one of them	18 jan	1	1
✔ Ver erro ao associar cartão ao paciente	14 jan	1	1
✔ Limitar número de anos até onde se pode escolher disponibilidades	14 jan	1	1
✔ Quando não existe disponibilidades de staff mostrar períodos a cinzento	14 jan	2	3
✔ Zoom de todas as vistas semanais com agenda consoante breakpoints de ecrãs caso mais do que uma instalação selecionada	11 jan	3	4
✔ Vista semanal com múltiplas instalações fica com scrollable para o lado	10 jan	1	1
✔ Ao criar disponibilidade de sala poder especificar, também, o staff	11 jan	3	3
✔ Verificar sobreposições na plataforma de prod na agenda vista semanal	10 jan	1	1
✔ Criar role para pacientes 1 ↵	9 jan	2	2
✔ Permitir criar 2 aparelhos auditivos de uma vez	8 jan	2	1
✔ Permitir colapsar resultado da anamnese mais recente	8 jan	1	1
✔ Rever erro ao atualizar services	8 jan		
✔ Mostrar sala na agenda pessoal lista/mensal	8 jan	1	1
✔ Alterar horário da agenda das 8h às 24h	8 jan	1	1
✔ Corrigir mensagem de erro para informar falta de disponibilidade em vez de hora inválida	8 jan	2	1

Nome da tarefa	Data de conc...	Effort	Cost
✔ contact request add facility	21 dez, 2018	1	1
✔ Send email with contact request	18 dez, 2018	2	2
✔ Ao remover um patient verificar se as related persons podem ser apagadas (se não for contact person de + ninguém e não for paciente nem staff)	17 dez, 2018	1	1
✔ Alinhar labels da hora da realização do teste com a marca do dispositivo	17 dez, 2018		
✔ Sincronizar nome, email, número de telefone de utilizadores quando alterações são realizadas	6 dez, 2018	3	2
✔ Configure queues and supervisor in Widex server	3 dez, 2018	1	
✔ Endpoint to hearing test pdf result	29 nov, 2018	1	1
✔ Permitir repetições nas disponibilidades (Web) 4 🐛	26 nov, 2018	8	12
✔ Calendário UI 4 🐛	27 nov, 2018	6	10
✔ Permitir repetições nas disponibilidades (API & DB) 6 🐛	23 nov, 2018	8	11
✔ Agenda deve apresentar horas na vertical 5 🐛	19 nov, 2018		5
✔ Implement SMS notification for scheduled appointment 1 🗨	8 nov, 2018	3	
✔ Appointment booking notification (email and SMS) 4 🐛	9 nov, 2018	5	
✔ Delete appointment record	6 nov, 2018	1	
✔ Ver testes auditivos feitos na app	5 nov, 2018 – 9 no...	2	
✔ Listar, procurar e consultar detalhes de consulta (Appointment record)	8 nov, 2018	1	
✔ CRUD Appointment record (Web) 2 🐛	8 nov, 2018	2	
✔ Gestão ocupação/disponibilidade de audiologistas (room & staff_availability) (API)	2 nov, 2018	8	
✔ Gestão ocupação/disponibilidade de audiologistas (room & staff_availability) (Web)	2 nov, 2018	8	
✔ Apagar Pacientes	6 nov, 2018	2	
✔ Configure supervisor on server to send emails using database driver and document those in technical docs	2 nov, 2018		
✔ Send email using database driver			

Nome da tarefa	Data de conc...	Effort	Cost
✔ Tabela de entity_activity_provider e entity_activity_person	13 set	2	2
✔ Gestão tipo de actividades	13 set	6	6
✔ Alterações página de utilizadores	6 set	2	2
✔ Fundir menus (meta) a serviços	6 set	4	4
✔ Rever eliminar utilizador	6 set	1	1
✔ Page config per user API 3 🗨	30 ago	6	3
✔ Page config per user Vue 1 🗨	30 ago	4	1
✔ An user can be associated to services	2 ago	3	3
✔ Entity com campo país	2 ago	2	2
✔ Edição da língua no perfil do utilizador	2 ago	3	2
✔ Atribuição de roles a agrupamentos	26 jul	8	3
✔ Update Entity Profile	19 jul	5	5
✔ Criar pessoa no registar de forma a guardar o genero e a data de nascimento	12 jul	2	1
✔ Password reset invalida tokens antigos 2 🗨	12 jul	2	2
✔ Responsável entidade 6 🗨	4 jul	4	6
✔ Definir servidores para entidades 4 🗨	5 jul	3	3
✔ Adicionar Roles a novo utilizador 2 🗨	28 jun	4	5
✔ Atribuir subroles a utilizadores API 3 🗨	28 jun	4	4
✔ universal link apps	27 jun	2	2
✔ Actualizações de campos perfil	28 jun	1	1
✔ Gestão de subroles	21 jun	2	2
✔ Garantir que emails se encontrem na base de dados sempre em lowercase	7 jun	1	1

Nome da tarefa	Data de conc...	Effort	Cost
✔ Testar Roles "Super"	31 maio	1	1
✔ Quando um utilizador tenta entrar e não tem serviços enviar um email para suporte	7 jun	1	1
✔ Menu Configurações a funcionar com entity_id null e novo formato VUEX 4 🐛	31 maio	6	6
✔ Ao criar novo utilizador herdar linguagem default da entidade 2 🐛	7 jun	1	1
✔ Permissões e Roles de autenticado	31 maio	1	1
✔ Auth emails suporte assunto em portugues, espanhol e ingles	30 maio	1	1
✔ Super user 4 🐛	24 maio	8	11
✔ Pedidos de entidade com utilizador já existente 2 🐛	18 maio	2	2
✔ Rever query de roles 1 🐛	24 maio	1	1
✔ Command para Roles	17 maio	8	4
✔ Permissões baseadas em Roles	10 maio	4	16
✔ Resolução de problemas relacionados com apps	10 maio	2	2
✔ Roles (Vue)	10 maio	4	5
✔ Atribuir/remover services em função de Roles 4 🐛	3 maio	2	3
✔ Pesquisar instalações ordenadas por distância	2 maio	3	3
✔ Build staging	29 abr	1	2
✔ Remover Roles (Front-end/API)	30 abr	3	3
✔ Msg quando o numero de utilizadores chega a 5 2 🐛	26 abr	1	1
✔ Regex password	29 abr	1	1
✔ Alterações Auth 3 🐛	26 abr	1	1
✔ Nodejs folder/code organization	26 abr	3	3
✔ Permissões em /services	26 abr	2	2

Nome da tarefa	Data de conc...	Effort	Cost
✔ Implementar Roles (API) 6 🗨	26 abr	2	2
✔ Testar Login 3 🗨		2	2
✔ Unit Tests Auth 5 🗨	19 abr	3	4
✔ Profile de utilizador	5 abr	3	2
✔ Edição de utilizadores 4 🗨	12 abr	12	12
✔ Código da instalação 1 🗨	4 abr	1	1
✔ Revisão e alterações resposta person por entidade	10 abr	2	2
✔ Revisão e alterações moradas para aplicação Evollu	10 abr	3	3
✔ Persons de um utilizador (API)	5 abr	1	1
✔ Endpoints 3 🗨	8 abr	3	3
✔ Templates email Homelab	29 mar	1	1
✔ Remover BDs não utilizadas	29 mar	1	1
✔ Logs de entrada/saída	29 mar	3	3
✔ Alterações 6 🗨		3	5
✔ Alterações de email fica sujeito a confirmação de conta	29 mar	3	3
✔ Criar serviços para todos os users consoante permissões atuais	29 mar	3	3
✔ Zonas administrativas PT	29 mar	1	2
✔ Pagina Gestão Staff (API & Front-end)	21 mar	4	5
✔ Bug fixing teste 1 12 🗨	15 mar	4	5
✔ Alterações 3 🗨	13 mar	1	1
✔ Atualizar front-end para instalações pertencerem menu Admin	13 mar	2	2
✔ Passar Persons para Auth	11 mar	5	12

Nome da tarefa	Data de conc...	Effort	Cost
✔ Entidade (API) devolve cidade	8 mar	1	1
✔ Passar Clusters para Auth	6 mar	3	4
✔ Facilities em Auth	7 mar	2	5
✔ Documentar Endpoints Auth	1 mar	3	3
✔ Utilizar API para criar a pessoa no servidor de homelab	21 fev	1	1
✔ Adicionar entity_id a utilizadores vindo de App's	21 fev	1	1
✔ Colocar New Auth branch in Dev	21 fev	1	1
✔ Endpoints CRUD entidades	6 fev	2	3
✔ Envio de emails com lingua dinamica	30 jan	2	2
✔ Change permissions to glintt	28 jan	1	1
✔ Atualizar library bcrypt to bcryptjs	25 jan	1	1
✔ Mudar mensagens de erro ao fazer reset token para não mostrar que o token é invalido	25 jan	1	1
✔ Atualizar criar pessoa ao fazer login	25 jan	1	1
✔ Rever limite de utilizadores para enviar email quando é criado limite - 1	18 jan	1	1
✔ Update de permissão tendo em conta permissões atuais	17 jan	2	2
✔ Log de autenticações na plataforma	10 jan	1	1
✔ user com created_by, updated_by e deleted_by	10 jan	1	1
✔ Change new account email template	10 jan	1	1
✔ Fix redirect uri	10 jan		
✔ Adicionar validade aos códigos de password reset	8 jan	2	2
✔ prefix on edit profile	17 dez, 2018	3	3
✔ pretty message when account is already confirmed	12 dez, 2018	1	1

Nome da tarefa	Data de conc...	Effort	Cost
✔ Localidade de entidade	26 jul	3	3
✔ Audits com GPS	26 jul	1	1
✔ Address_lat, address_lng and address_country on entity_request	23 jul	1	1
✔ Lista de testes nos rastreios	26 jul	2	2
✔ Reporting and analytics for appointments Vue (basic KPIs) - Part 2	23 jul	8	8
✔ Reporting and analytics for appointments Vue (basic KPIs) - Part 1	19 jul	6	6
✔ Extração das roles dadas pelo entity Request para 1 classe à parte e endpoints	19 jul	3	3
✔ Atualizar Responsáveis de entidade altera user roles nos vários micro serviços	18 jul	7	7
✔ Correções para Leads-uk	16 jul	2	2
✔ Atualizar Dev 4 🗑	15 jul	1	1
✔ Build Production UK	15 jul	1	1
✔ Correções	12 jul	1	1
✔ Tag Leads (API)	12 jul	3	3
✔ Tag Leads (VUE)	12 jul	5	7
✔ Marcadores / tags para leads (API)	12 jul	3	4
✔ Atualizar Demo 6 🗑	5 jul	8	13
✔ Componentes Genéricos	5 jul	2	2
✔ Marcadores / tags para leads (Vue)	5 jul	3	4
✔ Testar Roles 4 🗑	5 jul	1	1
✔ Alterações roles 6 🗑	4 jul	5	5
✔ Atualizar DEV com novas funcionalidades (roles/divisão de permissões)	5 jul	1	1
✔ Transferência de leads 4 🗑	3 jul	3	6

Nome da tarefa	Data de conc...	Effort	Cost
✔ Colocar dados em PROD 6 🗄️	28 jun	2	2
✔ Atribuir subroles a utilizadores VUE 3 🗄️	28 jun	10	18
✔ Correção pedido de entidade sem consultor	14 jun	2	2
✔ Criar atividade "Rastreio local"	14 jun	1	1
✔ Aplicar alterações das datas no servidor de leads + componente que mostra os dados da sessão atual	12 jun	1	1
✔ Mail Acústica	14 jun	1	1
✔ Ver todas as roles por entidades num modal	12 jun	1	1
✔ Publicar produção 7 🗄️	6 jun	2	2
✔ Devolver entidades com cidade no servidor e mostrar nos selects do front-end 4 🗄️	6 jun	2	2
✔ Correções página de user roles 6 🗄️	4 jun	2	2
✔ Testar Roles "Super"	31 maio	1	1
✔ Página detalhe de rastreios 3 🗄️	7 jun	2	2
✔ Menu Audiologia e Agenda a funcionar com entity_id null e novo formato VUEX	7 jun	4	3
✔ Atualizar VUEX para novo formato de dados devolvidos por /services 3 🗄️	31 maio	3	3
✔ Correção página de settings menu audiology e página de entidades 3 🗄️	3 jun	1	1
✔ Colocar Build para staging e produção	31 maio	1	1
✔ Rever permissões a partir de roles	31 maio	2	2
✔ Colocar servidores e front-end produção 6 🗄️	30 maio	2	2
✔ Refactor update roles de utilizador (Api) 6 🗄️	29 maio	4	4
✔ Refactor Roles interface 8 🗄️	29 maio	5	5
✔ Logs de acesso a rastreios	21 maio	2	2
✔ Remover "Tablet Luis"	21 maio	1	1

Nome da tarefa	Data de conc...	Effort	Cost
✓ Ensaios e documentação para PROD	18 maio	4	4
✓ Alterações e correção para PROD	17 maio	5	5
✓ Alterações pedidos de entidade 4 🗨	13 maio	2	2
✓ Agenda/Marcações/ o filtro por semana quando estão varias entidades selecionadas não funciona e está desformatado 3 🗨	17 maio	2	2
✓ Permissões baseadas em Roles	10 maio	4	16
✓ Adicionar cidades aos pedidos por entidades	10 maio		
✓ leads anonimizadas com total de testes	3 maio	1	1
✓ Correção /services leads join multiple connections error	29 abr	1	1
✓ Msg quando o numero de utilizadores chega a 5 2 🗨	26 abr	1	1
✓ Listar user roles de utilizador e atualizar as mesmas (API)	26 abr	2	2
✓ Adaptar roles para permitir a atualização de roles de vários serviços a partir do mesmo componente	26 abr	2	2
✓ Leads coluna marketing	26 abr	1	1
✓ Relatórios incluir title and surname	16 abr	1	1
✓ Assinatura de mail com relatório	12 abr	1	2
✓ Atualizar hierarquia utilizadores da Glintt	12 abr		
✓ Verificação Prod -> newAuth 6 🗨	5 abr	5	25
✓ Email de Relatório de teste 2 🗨	5 abr	1	1
✓ Alterar nomes de menus	29 mar	1	1
✓ Remover BDs não utilizadas	29 mar	1	1
✓ Restringir Glintt de Leads/Refs a leitura	29 mar	1	1
✓ acesso plataforma por promotor	19 mar	2	2
✓ Envio de email para promotores	19 mar	1	2

Nome da tarefa	Data de conc...	Effort	Cost
✔ Endpoint para envio de leads e informação necessária para o promotor	15 mar	2	2
✔ Email para lead com resultado teste auditivo em EN	12 mar	2	2
✔ Alterações PDF Acustica	8 mar	1	1
✔ Atualizar front-end para usar clusters a partir do auth	6 mar	1	1
✔ Delay nos pedidos ao servidor dos filtros das tabelas (não fazer 1 pedido por letra)	8 mar	1	1
✔ Atualizar front-end para não usar permissões do auth	1 mar	4	7
✔ Atualizar evollead-srv para não usar permissões do auth	1 mar	1	2
✔ Corpo de email do envio do teste auditivo	25 fev	3	3
✔ Colocar Leads Dev with New Auth branch	22 fev	2	3
✔ Colocar Front-end with New Auth branch	21 fev	1	1
✔ Remover Front end de envio de sms para leads ao balcao	20 fev	1	1
✔ pronto para começar o projeto nacional, arranque oficial dia 18/02	15 fev		
✔ Refazer página Utilizadores (com permissões auth)	8 fev	2	2
✔ página "Gestão da Entidade"	7 fev	2	2
✔ Endpoints "subscriptions"	6 fev	2	2
✔ Criar instalação e sala	24 jan	1	1
✔ Criar Entity Requests persons and permissions	24 jan	2	3
✔ Adicionar NIF à entidade	18 jan	1	1
✔ Passar new user para modal e alterar selects to single multiselect	17 jan	2	3
✔ Criar utilizador glintt servidor	16 jan	2	2
✔ Criar utilizador Glintt front end	16 jan	2	3
✔ Atualizar front end para preencher automaticamente regional managers, area managers e consultores depedendo do utilizador	15 jan	2	2

Nome da tarefa	Data de conc...	Effort	Cost
✓ Devolver todos os responsáveis acima e todos os users nos quais um utilizador é pai na criação de pedidos de entidade	15 jan	2	2
✓ Enviar report para email da APP do kiosk da WIDEX - com gráfico e mensagem conforme resultado	7 jan	2	2
✓ Referência na entidade para um Cliente 3 🗨	7 jan	1	1
✓ Introduzir logo da entidade no template do teste auditivo dinamicamente 1 🗨			
✓ Adicionar coluna no componente de Contactos (References) e Leads com a indicação se tem Perda Auditiva (PA) ou não tem perda auditiva (NPA)	26 dez, 2018	2	2
✓ Ao criar utilizador enviar email para ser ele a definir a password?	21 dez, 2018	2	2
✓ Alterar o relatório enviado as leads com as sugestões da Widex	28 dez, 2018		
✓ Enviar e-mail a partir da plataforma			
✓ Esconder dropdown ao seleccionar lingua		1	1
✓ Fazer uma trait com métodos de pesquisa de entidade, estado e datas para reutilizar nos vários controladores	20 dez, 2018	1	1
✓ Adicionar Filtros de entidade/cluster, estado e datas à tabela de contactos	20 dez, 2018	3	3
✓ Dividir painel de References em componentes	19 dez, 2018	2	2
✓ Atualizar PDF template de resultado teste auditivo	18 dez, 2018	1	1
✓ Permitir alterar o nome do kiosk e enviar email a informar que quiosque foi alterado	18 dez, 2018	2	2
✓ Página Quiosques mobile dropdown z-index incorrecto		1	1
✓ Construir pdf de consentimento dinamicamente tendo em conta a entidade	14 dez, 2018	2	2
✓ Utilizar servidor de shortner para enviar url's de confirmação de conta/consentimento		3	4
✓ Criar servidor capaz de dar shortner a url's	12 dez, 2018	4	2
✓ Envio do resultado do teste para um email a indicar a partir da plataforma	11 dez, 2018	4	4
✓ Modificar icon de limpar filtros	11 dez, 2018	1	1
✓ Colocar alterações em DEV	10 dez, 2018	1	1
✓ Dividir painel de Leads em componentes	10 dez, 2018	2	3

Nome da tarefa	Data de conc...	Effort	Cost
✓ Criar painel de leads em componentes			
✓ Verificação diária apenas manda info sobre quiosques ativos	10 dez, 2018	2	3
✓ Corrigir acentos ao exportar resultado do teste auditivo	4 dez, 2018	1	1
✓ No filtro texto pesquisa ter a opção de apagar o texto com um X 1	30 nov, 2018	2	2
✓ Default value for leads that don't have consent	29 nov, 2018	1	1
✓ Remover mail automático para o suporte@evollu.com 2	20 nov, 2018	1	
✓ Criação do PDF com o resultado do teste auditivo 5	28 nov, 2018	8	9
✓ Envio do resultado do teste para o paciente quando é disponibilizado email	28 nov, 2018	2	1
✓ Adicionar label Hz nas frequências no resumo do quiosque e adicionar mensagens de erro ao introduzir regras notificacao 3		1	
✓ Colocar em staging regras de notificação e testar	21 nov, 2018	1	1
✓ Não permitir alteração da entidade de uma lead (VUE + API)	16 nov, 2018	1	
✓ Criar interface VUE para agrupamento de entidades	16 nov, 2018	4	
✓ Adicionar botão para consentimento na página das referências (permitir imprimir também) 2	15 nov, 2018	4	
✓ Ecrã de resumo da entidade	14 nov, 2018		
✓ Add phone number with prefix to references in order to save it in db	12 nov, 2018		
✓ API e VUE para editar notificações 2	14 nov, 2018	4	
✓ Verificar critérios de envio de email	13 nov, 2018	3	
✓ Schedule emails about kiosk's activity to send daily 4	7 nov, 2018	5	
✓ Manter fixos os filtros quando mudamos de página (ex: referencias para os filtros) 1	7 nov, 2018	1	
✓ Log out automático quando o utilizador não tem roles			
✓ Gestão de utilizadores (API) 6			
✓ Implementar "sorting" das "references" da API (/leads)	16 ago, 2018		