



ESCOLA SUPERIOR
DE TECNOLOGIA
E GESTÃO

Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Eng.^a Informática – Computação Móvel

EVOLLU AUDIOLOGY FOR ALL, A4A

ESTUDANTE MICAEL LIMA FETEIRA OLIVEIRA

Leiria, setembro de 2023



ESCOLA SUPERIOR
DE TECNOLOGIA
E GESTÃO

Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Eng.^a Informática – Computação Móvel

EVOLLU AUDIOLOGY FOR ALL, A4A

ESTUDANTE MICAEL LIMA FETEIRA OLIVEIRA
Número: 2210613

Relatório de estágio realizado sob orientação da Professora Doutora Anabela Moreira Bernardino (anabela.bernardino@ipleiria.pt).

Leiria, setembro de 2023

AGRADECIMENTOS

Ao concluir este projeto, gostaria de agradecer e enunciar todas as pessoas que de certa forma, direta ou indiretamente, estiveram presentes no desenvolvimento do mesmo, pois todas desempenharam um papel crucial para a realização e a execução do projeto.

Primeiramente, agradecer à *Evollu* por disponibilizar esta oportunidade que me providenciou experiência profissional, assim como um desenvolvimento das minhas capacidades sociais, pelo trabalho em equipa e pela aprendizagem de uma nova *framework*. Certamente, levo boas recordações e amigos próximos desta experiência.

Ao Pedro Miguel Prates Alves, o meu orientador e supervisor na *Evollu*, que foi o meu pilar em todas as questões e dúvidas que me surgiam. Esteve sempre disponível para ajudar e fornecer o melhor de si para os outros. Todo o *feedback* fornecido, quer seja a nível tecnológico, social ou profissional foram de extrema importância durante todo o percurso, pois sem a sua ajuda o projeto não teria tomado um rumo de desenvolvimento tão fluido e dinâmico.

À orientadora Professora Doutora Anabela Moreira Bernardino, pela sua grande dedicação ao fornecer todo o *feedback* necessário perante o desenvolvimento deste relatório. Providenciou uma ajuda fulcral ao estar sempre disponível para qualquer dúvida que surgisse, assim como fornecer todo o apoio e motivação.

Ao Instituto Politécnico de Leiria, por todos estes anos de formação letiva e educacional, assim como a todos os professores que me acompanharam, sempre disponíveis para ajudar no que fosse preciso.

A todas as pessoas que testaram e forneceram *feedback* sobre o comportamento e estado da aplicação. Este foi extremamente importante para a compreensão de possíveis erros que existiam, assim como possíveis melhorias a efetuar.

À minha família, que esteve sempre disponível para me ajudar. Todo o apoio, motivação e valores transmitidos foram cruciais para o desenvolvimento de todo o projeto. À minha namorada, Ema Catarina Fernandes Neves, por toda a sua disponibilidade e força, sendo um pilar incondicional em toda a minha vida.

Um muito obrigado a todos os que de certa forma, me ajudaram na realização deste estágio.

RESUMO

O presente relatório pretende descrever o trabalho que foi desenvolvido ao longo do estágio curricular, no âmbito do [Mestrado em Engenharia Informática - Computação Móvel \(MEI-CM\)](#), da [Escola Superior Tecnologia e Gestão \(ESTG\)](#) do [Instituto Politécnico de Leiria \(IPL\)](#).

O estágio foi realizado na entidade *Sensing Evolution* com uma duração de 9 meses e a tempo inteiro. A *Sensing Evolution* ou *Evollu* é uma empresa de desenvolvimento de *software*, nomeadamente na área da saúde, através do desenvolvimento de plataformas e aplicações que providenciem ao utilizador a possibilidade de realizarem rastreios e marcar consultas com profissionais na área. Do mesmo modo, são desenvolvidas diversas soluções empresariais a empresas que atuem na área da saúde, assim como parcerias com entidades escolares.

O principal objetivo residiu no desenvolvimento de uma aplicação que providenciasse 4 testes ao utilizador, para que este conseguisse realizar rastreios à sua saúde auditiva, conseguindo assim realizar uma monitorização do seu perfil auditivo, quando devidamente autenticado. Este projeto faz parte do [Portugal 2020 \(PT2020\)](#), em colaboração com a [Escola Superior de Tecnologia da Saúde do Instituto Politécnico de Coimbra \(ESTeSC-IPC\)](#), e pretende ajudar na deteção precoce de perturbações na audição e democratizar o acesso aos serviços de saúde audiológica. Apesar do âmbito do projeto e dos seus objetivos, foram identificadas oportunidades de melhoria e reestruturações necessárias para continuar a cumprir estes objetivos e, ao mesmo tempo, permitir uma mais fácil e rápida evolução da aplicação num futuro próximo.

Durante o desenvolvimento do projeto, surgiram novas ideias e funcionalidades que seriam de extrema importância para o correto funcionamento e uso da aplicação, assim como reestruturações necessárias para obter uma aplicação estável e que correspondesse aos objetivos pretendidos.

A interação com uma nova *framework*, *SwiftUI*, permitiu desenvolver novos conhecimentos e competências no desenvolvimento *mobile*, assim como a aquisição de experiência profissional pelo contacto com os diversos membros de equipa, plataformas e métodos utilizados pela empresa, tendo sido o primeiro contacto em ambiente profissional. Foi providenciada uma aplicação estável com o serviço

de autenticação, comunicações com *Application Programming Interface (API)* e realização de testes em conformidade com os requisitos pretendidos.

Palavras-chave: Perfil auditivo, Rastreo Auditivo, Desenvolvimento de *Software* para saúde audiológica, Aplicação móvel *iPhone Operating System (iOS)*, Serviços de saúde audiológica.

ABSTRACT

The present report intends to describe the work that was developed during the curricular internship in the scope of the Master in Computer Engineering - Mobile Computing, of [ESTG](#) of the [IPL](#).

The internship was carried out at the entity Sensing Evolution with a duration of 9 months and as a full-time regime. Sensing Evolution or Evollu is a software development company, namely in the health branch, through the development of platforms and applications that provide the user with the possibility of carrying out screenings and making appointments with professionals in the area. Likewise, several business solutions are developed for companies that operate in the health branch, as well as partnerships with school entities.

The main objective resided in the development of an application that provided 4 tests to the user, so that he could carry out tracking of his hearing health, thus enabling the monitorization of the user's auditory profile. In order to provide greater security and functionality to the application, an authentication service was also implemented. This project is part of [PT2020](#) in collaboration with the [ESTeSC-IPC](#) and aims to help people with hearing disorders and allow for easier access to audiological health services. During the development of the project, new ideas and functionalities emerged that would be of extreme importance for the correct functionality and use of the application, as well as necessary restructurings to obtain an application that corresponded to the intended objectives.

The interaction with a new SwiftUI framework allowed the development of new knowledge and skills in mobile development, as well as the acquisition of professional experience through contact with the various team members, platforms and methods used by the company, having been the first contact in a professional environment. This internship provided a stable application with authentication service, [API](#) communications, testing services in accordance with the intended requirements.

Keywords: *Auditory profile, Hearing Screening, Software development about Audiological health, [iOS](#) Mobile Application, Audiological Health Services.*

ÍNDICE

Agradecimentos	i
Resumo	iii
Abstract	v
Índice	vii
Lista de Figuras	xi
Lista de Tabelas	xv
Lista de Abreviaturas	xvii
1 Introdução	1
1.1 Interesse pela candidatura	1
1.2 Objetivos	1
1.2.1 Projeto <i>Audiology for All (A4A)</i>	2
1.2.2 Benefícios de soluções antecipadas	2
1.3 <i>Evollu</i>	3
1.3.1 História da <i>Evollu</i>	3
1.3.2 Equipa e condições de trabalho	3
1.3.3 Modelo de negócio	4
1.4 Cronograma do estágio	5
1.5 Estrutura do relatório	8
2 Enquadramento	9
2.1 Definição do problema	9
2.1.1 Saúde em Portugal	9
2.1.2 Audiologia	10
2.1.3 Perda Auditiva	10
2.2 Estudo de aplicações	12
2.2.1 <i>Mimi Hearing Test</i>	12
2.2.2 <i>Biosom</i> - Teste Auditivo	14
2.2.3 <i>Evollu App</i>	15
3 Metodologia de Desenvolvimento	19
3.1 Metodologias <i>Agile</i>	19

3.2	<i>Scrum</i>	20
3.3	<i>Plataforma Jira</i>	22
3.3.1	Registo de tempo	23
4	Desenvolvimento	25
4.1	Arquitetura	25
4.1.1	Linguagem de Programação	25
4.1.2	<i>Data Flow</i>	30
4.1.3	Tarefas específicas	31
4.1.4	Arquitetura geral	32
4.2	Testes	36
4.2.1	Processamento auditivo central (PAC)	36
4.2.2	Esforço Auditivo	37
4.2.3	Zumbidos ou <i>Tinnitus</i>	37
4.2.4	<i>Tonal</i>	38
4.3	<i>UserStories</i>	38
4.4	Tecnologias utilizadas	42
4.4.1	<i>Xcode</i>	42
4.4.2	<i>TestFlight</i>	43
4.4.3	<i>GitLab</i>	44
4.4.4	<i>GitKraken</i>	45
4.4.5	<i>Bash Command-line interface (CLI) MacBook</i>	46
4.4.6	<i>Figma</i>	46
4.4.7	<i>Jira</i>	47
4.4.8	<i>Postman</i>	48
4.4.9	<i>WireGuard</i>	48
4.4.10	<i>MailTrap</i>	49
4.4.11	<i>Overleaf</i>	50
4.5	<i>Funcionalidades comuns</i>	50
4.6	<i>Interação com Servidor</i>	52
4.7	<i>Autenticação</i>	53
4.7.1	<i>Login principal</i>	54
4.7.2	<i>Login secundário</i>	54
4.7.3	Registar um utilizador	55
4.7.4	<i>Logout</i>	55
4.8	<i>Métodos que requerem autenticação</i>	55
4.8.1	Alterar a palavra-passe	55
4.8.2	Repor a palavra-passe	56

4.8.3	Editar o perfil	56
4.8.4	Eliminar de conta	56
4.8.5	Enviar os resultados de testes/subtestes	56
4.8.6	Obter resultados dos testes	57
4.9	Aspetos técnicos importantes	57
4.10	<i>Dificuldades Sentidas</i>	72
5	Produto final	81
5.1	Alterações e testes	81
5.1.1	<i>Design</i>	81
5.1.2	Funcionalidades	87
5.1.3	Validade dos subtestes do PAC	87
5.2	Testes de usabilidade	88
5.2.1	Secções	88
5.2.2	Respostas	95
5.3	Produto Final	106
5.3.1	Abertura da aplicação	106
5.3.2	Autenticação	106
5.3.3	Página inicial	113
5.3.4	Definições	119
5.3.5	Perfil	123
5.3.6	Teste PAC	127
5.3.7	Teste Esforço Auditivo	138
5.3.8	Teste Zumbidos/ <i>Tinnitus</i>	147
5.3.9	Teste Tonal	153
6	Conclusões	163
6.1	Considerações finais	163
6.2	Melhorias	164
6.3	Trabalho futuro	165
	Bibliografia	167
	Declaração	173

LISTA DE FIGURAS

Figura 1	Cronograma do desenvolvimento do projeto	7
Figura 2	Aplicação <i>Mimi Hearing iOS</i>	13
Figura 3	Continuação da aplicação <i>Mimi Hearing iOS</i>	14
Figura 4	Aplicação <i>Biosom - Teste Auditivo iOS</i>	15
Figura 5	Aplicação <i>Evollu</i>	17
Figura 6	Colunas do quadro <i>Kanban</i>	22
Figura 7	Registo de tempo no <i>Jira</i>	23
Figura 8	Como funciona o <i>SwiftUI</i> . Imagem proveniente de <i>Atipik</i> [18]	26
Figura 9	Exemplos de dinamização de código.	28
Figura 10	Árvore de decisão do <i>SwiftUI</i> . Imagem proveniente de <i>Matteo Manferdini</i> [25]	31
Figura 11	Arquitetura geral da aplicação	33
Figura 12	Comunicação com o servidor	34
Figura 13	Programa <i>Xcode</i>	43
Figura 14	Plataforma <i>AppStoreConnect</i>	44
Figura 15	Plataforma <i>GitLab</i>	45
Figura 16	Programa <i>GitKraken</i>	45
Figura 17	Terminal do <i>MacBook</i>	46
Figura 18	Programa <i>Figma</i>	47
Figura 19	Programa <i>Postman</i>	48
Figura 20	Programa <i>Wireguard</i>	49
Figura 21	Plataforma <i>MailTrap</i>	50
Figura 22	Pausar o teste/subteste	51
Figura 23	Sair do teste	52
Figura 24	O <i>GeometryReader</i>	58
Figura 25	Verificar o tamanho de ecrã	58
Figura 26	Continuação de verificar o tamanho de ecrã	59
Figura 27	<i>Loading Screen</i>	60
Figura 28	<i>Toast</i>	60
Figura 29	Utilização de <i>CoreData</i>	62
Figura 30	Fluxograma para guardar os dados da autenticação localmente	63
Figura 31	Fluxograma de verificação do <i>Token</i>	63

Figura 32	Exemplo de notificações implementadas	64
Figura 33	Fluxograma para despoletar uma notificação de erro no envio de um teste	65
Figura 34	Fluxograma do <i>schedule</i> de notificações	65
Figura 35	Fluxograma do <i>scheduled notification dates</i>	66
Figura 36	Notificações com ações integradas	67
Figura 37	Uso de <i>thread</i> na lógica do teste Tonal	68
Figura 38	<i>LazyLoading</i> para obter os resultados dos testes	70
Figura 39	Fluxograma do <i>lazy loading</i>	71
Figura 40	<i>Package</i> do teste Tonal	72
Figura 41	Exemplo de estruturas aplicadas	73
Figura 42	Exemplo de <i>ForEach</i> de <i>id: \.id</i>	74
Figura 43	Exemplo de <i>Singleton</i> para o teste Esforço Auditivo	75
Figura 44	Um dos métodos implementados para a atualização dos botões	76
Figura 45	Exemplos de conteúdo estático no teste Zumbidos	78
Figura 46	Exemplos de conteúdo dinâmico no teste Zumbidos	79
Figura 47	Versão desatualizada do ícone da aplicação	82
Figura 48	Versão desatualizada da <i>SplashScreen</i>	82
Figura 49	Versão desatualizada da página de <i>Login</i>	83
Figura 50	Versão desatualizada da página inicial da escolha de testes	83
Figura 51	Versão desatualizada da página inicial do teste PAC	84
Figura 52	Versão desatualizada dos resultados dos subtestes do PAC	85
Figura 53	Versão desatualizada da página de resultados do subteste Esforço Auditivo	86
Figura 54	Versão desatualizada da página de toque de sons do subteste Esforço Auditivo	87
Figura 55	Questionário da secção 1	89
Figura 56	Questionário da secção 2	90
Figura 57	Questionário da secção 3	92
Figura 58	Questionário da secção 4	94
Figura 59	Questionário da secção 5	95
Figura 60	Respostas da secção 2	96
Figura 61	Continuação de respostas da secção 2	96
Figura 62	Respostas da secção 3	97
Figura 63	Respostas da secção 3	98
Figura 64	Continuação de respostas da secção 3	99
Figura 65	Continuação de respostas da secção 3	100
Figura 66	Continuação de respostas da secção 3	101

Figura 67	Respostas da secção 4	102
Figura 68	Continuação de respostas da secção 4	103
Figura 69	Continuação de respostas da secção 4	104
Figura 70	Continuação de respostas da secção 4	105
Figura 71	Respostas da secção 5	106
Figura 72	<i>Loading screen</i>	107
Figura 73	Página inicial de <i>login</i>	108
Figura 74	Erros na página inicial de <i>login</i>	109
Figura 75	Página de recuperação da palavra-passe	110
Figura 76	Processo de registar uma nova conta.	112
Figura 77	Página de seleção de testes	113
Figura 78	<i>Side Bar</i>	114
Figura 79	Ecrã de resultados.	115
Figura 80	Ecrã de processamento de áudio.	116
Figura 81	Continuação do ecrã de processamento de áudio.	117
Figura 82	Ecrã de testes por enviar.	118
Figura 83	Página de definições	119
Figura 84	Página de termos e condições	120
Figura 85	Página de política de privacidade	121
Figura 86	Enviar <i>email</i> para o suporte <i>Evolllu</i>	121
Figura 87	Partilhar a aplicação <i>Hear App</i>	122
Figura 88	Notificação para confirmar <i>logout</i>	123
Figura 89	Página de perfil	124
Figura 90	Página de editar os dados do utilizador	125
Figura 91	Alterar a palavra-passe	126
Figura 92	Modificar as notificações	126
Figura 93	Eliminar a conta	127
Figura 94	<i>SplashScreen</i> PAC	128
Figura 95	Página inicial do PAC	129
Figura 96	Descrição do subtteste Memória Sequencial de sons não-verbais	130
Figura 97	Ecrãs de instrução	130
Figura 98	Opções dos subttestes do PAC	131
Figura 99	Auscultadores dos subttestes do PAC	132
Figura 100	Toque prévio dos sons	132
Figura 101	Ecrã antes de iniciar a sequência	133
Figura 102	Toque dos sons da sequência	134
Figura 103	Selecionar os sons da sequência	135
Figura 104	Resultados do subtteste	136

Figura 105	Avançar para o próximo subteste	137
Figura 106	Desbloqueio da recomendação	137
Figura 107	Recomendação do PAC	138
Figura 108	<i>Splash Screen</i> do teste Esforço Auditivo	139
Figura 109	Descrição do teste Esforço Auditivo	139
Figura 110	Questionário do teste Esforço Auditivo	140
Figura 111	Ajuste de volume do teste Esforço Auditivo	141
Figura 112	Descrições dos subtestes do teste Esforço Auditivo	142
Figura 113	Instruções dos subtestes do teste Esforço Auditivo	142
Figura 114	Toque dos sons nos subtestes do teste Esforço Auditivo	143
Figura 115	Continuação do toque dos sons nos subtestes do teste Esforço Auditivo	144
Figura 116	Escolher os sons nos subtestes do teste Esforço Auditivo	145
Figura 117	Resultado dos subtestes do teste Esforço Auditivo	146
Figura 118	Recomendação do teste Esforço Auditivo	147
Figura 119	<i>SplashScreen</i> do teste Zumbidos	147
Figura 120	Descrição do teste Zumbidos	148
Figura 121	Questionário do teste Zumbidos	149
Figura 122	Ajuste Grosseiro do teste Zumbidos	150
Figura 123	Ajuste Fino do teste Zumbidos	151
Figura 124	Resultados do teste Zumbidos	152
Figura 125	Recomendação do teste Zumbidos	152
Figura 126	<i>SplashScreen</i> do teste Tonal	153
Figura 127	Descrição do teste Tonal	154
Figura 128	Questionário do teste Tonal	155
Figura 129	Escolher tipo de auscultador no teste Tonal	156
Figura 130	Instruções do teste Tonal	157
Figura 131	Colocar volume no máximo	158
Figura 132	Decorrer do teste Tonal	159
Figura 133	Ecrã de resultados do teste Tonal	159
Figura 134	Resultados para cada ouvido	160
Figura 135	Gráfico sobre perda auditiva	161
Figura 136	Recomendação do teste Tonal	161

LISTA DE TABELAS

Tabela 1	Autenticação perante um utilizador não autenticado	38
Tabela 2	Autenticação perante um utilizador autenticado	39
Tabela 3	Opções disponibilizadas ao utilizador	39
Tabela 4	Funcionalidades do teste PAC	40
Tabela 5	Funcionalidades do teste Esforço Auditivo	41
Tabela 6	Funcionalidades do teste Zumbidos	41
Tabela 7	Funcionalidades do teste Tonal	42

LISTA DE ABREVIATURAS

A4A	<i>Audiology for All.</i>
ADN	Ácido desoxirribonucleico.
AIFF	<i>Audio Interchange File Format.</i>
API	<i>Application Programming Interface.</i>
APNS	<i>Apple Push Notification Service.</i>
App	<i>Application.</i>
CLI	<i>Command-line interface.</i>
CTO	<i>Chief technology (or technical) officer.</i>
dB	<i>Decibel.</i>
DEEC	Departamento de Engenharia Eletrotécnica e de Computadores.
DEV	<i>Development.</i>
ESTeSC-IPC	Escola Superior de Tecnologia da Saúde do Instituto Politécnico de Coimbra.
ESTG	Escola Superior Tecnologia e Gestão.
FHD	<i>Full High Definition.</i>
GUI	<i>Graphical User Interface.</i>
HD	<i>High Definition.</i>
HTTP	<i>Hypertext Transfer Protocol.</i>
HTTPS	<i>Hypertext Transfer Protocol Secure.</i>

Lista de Abreviaturas

ID	<i>Identification.</i>
iOS	<i>iPhone Operating System.</i>
IPL	Instituto Politécnico de Leiria.
JSON	<i>JavaScript Object Notation.</i>
MEI-CM	Mestrado em Engenharia Informática - Computação Móvel.
MP3	<i>MPEG-1 Audio Layer-3.</i>
OMS	Organização mundial de saúde.
ORL	Otorrinolaringologista.
PAC	Processamento auditivo central.
PROD	<i>Production.</i>
PT2020	Portugal 2020.
QA	<i>Quality assurance.</i>
RPAC	Rastreo processamento auditivo central.
SUS	<i>System Usability Scale.</i>
TAES	Tópicos Avançados de Engenharia de Software.
UC	Universidade de Coimbra.
VPN	<i>Virtual Private Network.</i>
WAV	<i>Waveform Audio.</i>

INTRODUÇÃO

Este capítulo fundamenta o interesse pela candidatura à proposta de estágio da entidade acolhedora *Evolllu* (Secção 1.1). Deste modo, é descrita a motivação e os objetivos do estágio (Secção 1.2), assim como a apresentação da entidade acolhedora (Secção 1.3). O cronograma referente ao desenvolvimento do projeto (Secção 1.4) também se encontra presente, além da apresentação da estrutura do relatório (Secção 1.5).

1.1 INTERESSE PELA CANDIDATURA

A proposta da empresa *Evolllu* suscitou interesse pela oportunidade de trabalhar a tempo integral no desenvolvimento de aplicações para dispositivos móveis, concretamente para a plataforma *iPhone Operating System* (iOS). O entusiasmo por este ramo surgiu especialmente pelo comparecer na unidade curricular de desenvolvimento de aplicações para dispositivos móveis no [Mestrado em Engenharia Informática - Computação Móvel \(MEI-CM\)](#). O primeiro contacto com a *framework* *UIKit*, proveniente da *Apple*, transmitiu o ideal do que seria a futura proposta a candidatar. Após algumas reuniões e conversações com a empresa, foi estabelecido um acordo que me permitia realizar um estágio no período de 1 de setembro de 2022 até 31 de Maio de 2023. Deste modo, surge assim o projeto desenvolvido ao longo do estágio, *Audiology for All (A4A)*, sendo que este projeto remete-se para os dispositivos móveis com sistema operativo *Android* e *iOS* e o estágio residiu-se apenas no desenvolvimento nativo para *iOS* com o uso da *framework* *SwiftUI*.

1.2 OBJETIVOS

Nesta secção são contextualizados os principais objetivos relativos ao desenvolvimento do projeto (Secção 1.2.1). Nomeadamente, a importância da implementação de uma solução para um problema de extrema importância, mas de certa forma negligenciado (Secção 1.2.2).

1.2.1 *Projeto A4A*

No dia 7 de julho de 2021 foi anunciado a parceria entre a [Universidade de Coimbra \(UC\)](#) e a empresa *Evolllu* para o dinamismo do rastreio e treino auditivo. A docente e investigadora do projeto, Margarida Serrano, da [Escola Superior de Tecnologia da Saúde do Instituto Politécnico de Coimbra \(ESTeSC-IPC\)](#), referiu a importância do desenvolvimento desta aplicação para o seu uso no quotidiano das pessoas. Este projeto permite a criação de aplicações para que seja possível trabalhar com populações dos 3 aos 95 anos e estas poderão ser utilizadas por profissionais de saúde ou por qualquer cidadão que queira realizar uma monitorização do seu perfil auditivo. Estas aplicações vão ajudar pessoas com perturbações na audição e ajudar a prevenir o desenvolvimento de problemas de aprendizagem, isolamento, frustração, depressão ou aumento da probabilidade de declínio cognitivo entre os 40 e os 50 anos, nas pessoas que não sejam acompanhadas no tempo devido [1]. Assim, surgiu o projeto de desenvolvimento da aplicação [A4A](#), de nome *Hear App*.

1.2.2 *Benefícios de soluções antecipadas*

Atualmente é aconselhado a ida a um centro auditivo, pelo menos uma vez por ano, para a realização de um teste auditivo, pois na maioria dos casos detetados precocemente é possível restituir a audição à pessoa [2]. A *Evolllu* desenvolve aplicações gratuitas para qualquer utilizador conseguir utilizar e realizar testes auditivos, cientificamente comprovados, de modo a conseguir perceber qual o estado da sua saúde auditiva.

A empresa *Evolllu* foca-se sobretudo no desenvolvimento e disponibilização de diversos materiais que providenciam a possibilidade de realização de testes auditivos. Deste modo, é possível identificar a presença de um problema auditivo e de igual forma marcar uma consulta com um especialista, ou simplesmente verificar que se encontra tudo bem e não existe a necessidade de recorrer a um especialista. Os testes são apenas informativos e uma ajuda opcional, permitindo disponibilizar uma solução rápida e eficaz para que qualquer pessoa interessada ou com dúvidas da sua atual situação auditiva, consiga rapidamente realizar um teste auditivo e obter *feedback* sobre a sua condição auditiva.

O principal objetivo consiste em informar o utilizador sobre a possibilidade de apresentar um problema auditivo e providenciar algumas opções para o utilizador, como por exemplo reabilitação auditiva. Por vezes, em casos casos mais graves

onde a ação clínica não é eficaz, os aparelhos auditivos são a solução recomendada, podendo melhorar significativamente a audição perdida.

1.3 EVOLLU

Esta secção fundamenta o enquadramento relativo à entidade *Evollu*. Primeiramente, uma breve introdução à história que levou ao surgimento da *Evollu* (Secção 1.3.1), seguido da equipa e condições de trabalho que se podem encontrar na entidade (Secção 1.3.2). Por fim, um breve enquadramento da atual proposta de solução da empresa no mercado (Secção 1.3.3 e Secção 1.3.3.1).

1.3.1 História da *Evollu*

A empresa *Evollu* surgiu em 2015, derivado da ideologia do fundador e co-fundador que pretendiam transformar o acesso à *healthcare* relacionado à audiolgia. O tempo médio da perda de audição até à pessoa procurar ajuda referente a essa mesma perda são 7 anos [3]. Tal problemática é potenciada e agudizada devido à desaprendizagem do cérebro, pois quanto mais tempo uma pessoa sofre de uma perda auditiva maior será o grau de dificuldade no processo de reabilitação.

Nos últimos anos, a *Evollu* tem vindo a investir no apoio à prestação de cuidados de saúde de forma sustentada e inovadora. Os resultados obtidos nos projetos de investigação e desenvolvimento em que a *Evollu* participou, permitiram a identificação de técnicas e estratégias inovadoras na promoção dos serviços de saúde, nomeadamente associados ao desenvolvimento de aplicações e ferramentas de rastreio disponibilizadas às populações.

1.3.2 Equipa e condições de trabalho

A *Evollu* tem crescido ao longo do tempo devido à integração de novos projetos e à formação de novas parcerias com outras empresas.

Apresenta ainda uma grande afinidade com a formulação de estágios, sejam estes a nível nacional, como é o caso do estágio em que se este relatório se baseia, como estágios a nível internacional, devido à presença de dois estagiários de nacionalidade Dinamarquesa. Estes estagiaram na *Evollu* pelo período de 3 meses em funções de

Web Development. O ambiente de trabalho sempre foi bastante positivo e amistoso entre todos os elementos da equipa, no qual todos se encontravam disponíveis e motivados para ajudar em qualquer dúvida que surgisse.

Aquando do início do estágio foi disponibilizado o material necessário para o desenvolvimento do projeto [A4A](#), sendo um *MacBook* (*MacBook Pro i7* de *16gb RAM*) para o desenvolvimento e um *iPhone* (modelo *iPhone 12*) para o *testing* da aplicação desenvolvida. O decorrer do projeto contou com a ajuda de diversos membros da equipa para a disponibilização e desenvolvimento de etapas importantes, como a implementação do *backend* para a comunicação com a aplicação em desenvolvimento e as etapas de *Quality assurance (QA)*.

1.3.3 Modelo de negócio

A *Evollu* apresenta-se ao mercado de distintas formas, tentando alicerçar a sua proposta de valor no seguinte:

- A presença de uma plataforma de *software* que possibilita a realização de um pré-diagnóstico, que consegue entregar algo simples e rápido aos utilizadores.
- A forte aposta por parte da empresa consiste nesta se querer transformar no “*Uber*” da facilitação de serviços de saúde às farmácias portuguesas.
- O maior objetivo consiste em apresentar um *suit* de rastreio, no qual a entidade *Evollu* consiga conectar tudo e não apenas os dados referentes à audiológia, garantindo aos profissionais de saúde um conjunto de dados que permitem avaliar holisticamente a saúde do utilizador.

1.3.3.1 Solução 1-2-3

A *Evollu* tem como lema e objetivo de desenvolvimento de todas as suas soluções tecnológicas que permitam configurarem-se como soluções 1-2-3, sendo dividido da seguinte forma: *Diagnostic*, *Result* e por fim *Care*. Como os próprios nomes indicam, o início da etapa consiste na realização de testes de diagnóstico, de modo a compreender qual o estado atual do paciente. Após a realização do(s) teste(s) os dados são guardados e analisados, o que leva à terceira e última etapa, *Care*. Nesta serão aplicadas soluções para o tratamento e/ou reabilitação do paciente.

Para o futuro, existe uma forte aposta no desenvolvimento e disponibilização de uma plataforma que interligue diversos pontos do mercado. Ora, sucintamente o ponto principal consiste na construção de uma plataforma integrada que:

- Possibilita a identificação de *needs* antes que outros parceiros na área da saúde intervenham;
- Baseado em tecnologias únicas e patenteadas que juntas conseguem apresentar soluções imediatas. De igual forma, é possível monitorizar os diversos dados que são recolhidos ao longo do tempo;
- Facilita o acesso à *healthcare*, tornando-a mais acessível e aumenta simultaneamente o lucro aos parceiros associados à empresa.;
- O principal objetivo consiste sempre no trabalho de prevenção, de modo a possibilitar e/ou construir uma melhor qualidade de vida.

1.4 CRONOGRAMA DO ESTÁGIO

O estágio decorreu desde setembro de 2022 a maio de 2023, perfazendo um período para o desenvolvimento do projeto de 9 meses.

De seguida é apresentado, de forma sucinta, o trabalho realizado em cada mês do estágio:

- Setembro → No primeiro mês existiu um período de integração com a equipa e as tecnologias utilizadas na empresa, assim como um estudo sobre a linguagem *Swift* e *framework SwiftUI*. Foi dado o início ao desenvolvimento da aplicação, através da criação de um projeto e definição das primeiras etapas a realizar, nomeadamente o desenvolvimento do teste [Processamento auditivo central \(PAC\)](#). Introdução ao *Apple Store Connect* para a publicação de uma versão da aplicação de forma a ser possível testar a mesma noutros dispositivos.
- Outubro → Foi disponibilizada para a [ESTeSC-IPC](#) a primeira versão estável do teste [PAC](#). Iniciado o desenvolvimento do teste Esforço Auditivo. Foi também iniciada a escrita do relatório de estágio.
- Novembro → Primeira reunião com a [ESTeSC-IPC](#), tendo sido definidos alguns novos requisitos e modificações para o teste [PAC](#) e Esforço Auditivo. Foram efetuadas as alterações necessárias e foi disponibilizada a nova versão estável do teste [PAC](#) e a primeira versão estável do teste Esforço Auditivo. Iniciado o desenvolvimento do teste Zumbidos.

- Dezembro → Primeira versão estável do teste Zumbidos. Dado início à implementação da comunicação com o servidor.
- Janeiro → Efetuadas novas alterações ao teste Esforço Auditivo, provenientes do *feedback* de uso da última versão disponibilizada. Implementação dos métodos de autenticação na aplicação.
- Fevereiro → Segunda reunião com a [ESTeSC-IPC](#), onde foram projetadas novas melhorias e alterações ao teste [PAC](#) e Esforço Auditivo. Início da reestruturação de substituição dos dados estáticos na aplicação por pedidos de dados ao servidor.
- Março → Término da reestruturação de obtenção dos dados dinâmicos do servidor. Primeira versão do teste Zumbidos disponibilizada à [ESTeSC-IPC](#). Novas alterações a todos os testes. Implementação da comunicação com a [Application Programming Interface \(API\)](#) do [Departamento de Engenharia Eletrotécnica e de Computadores \(DEEC\)](#) para processamento de áudio. Reestruturação para envio dos dados dos testes para o servidor, assim como a sua obtenção.
- Abril → Início do desenvolvimento do teste Tonal. Novas alterações e reformulações necessárias a todos os testes.
- Maio → Primeira versão estável e disponibilização do teste Tonal. Aprimoramento de detalhes e resolução de problemas diagnosticados através do uso da aplicação por parte da [ESTeSC-IPC](#) (de referir que esta etapa foi sendo constantemente aplicada ao longo do desenvolvimento). Disponibilização da última versão estável da aplicação. A escrita deste relatório foi um processo contínuo ao longo do estágio.

Na Figura 1 é apresentado o cronograma referente ao decorrer das etapas de desenvolvimento do projeto.



Figura 1: Cronograma do desenvolvimento do projeto

1.5 ESTRUTURA DO RELATÓRIO

O relatório encontra-se dividido nos seguintes 6 capítulos:

- Introdução (Capítulo 1): capítulo onde está presente uma introdução ao estágio realizado, os objetivos do projeto, o cronograma e a estrutura do relatório.
- Enquadramento (Capítulo 2) : capítulo onde está presente a definição do problema e o estudo de aplicações móveis de paradigma equivalente.
- Metodologia de Desenvolvimento (Capítulo 3): capítulo onde são descritas as metodologias utilizadas durante o projeto.
- Desenvolvimento (Capítulo 4): capítulo que descreve a arquitetura, tecnologias utilizadas, requisitos funcionais da aplicação (*User Stories*), testes desenvolvidos, aspetos técnicos importantes, dificuldades sentidas, entre outros.
- Produto Final (Capítulo 5): capítulo que refere os resultados dos testes externos realizados à aplicação e as alterações efetuadas à aplicação. Descreve ainda, de forma pormenorizada, as funcionalidades da aplicação e o seu estado final.
- Conclusões (Capítulo 6): capítulo que apresenta a conclusão e considerações sobre o desenvolvimento do projeto, assim como o trabalho futuro.

ENQUADRAMENTO

Neste capítulo será retratada a definição do problema, nomeadamente o tema da saúde em Portugal, do que trata a audiologia e a importância da atenção à perda auditiva (Secção 2.1). Será, de igual modo, retratado o estudo das aplicações de paradigma equivalente à aplicação desenvolvida (Secção 2.2).

2.1 DEFINIÇÃO DO PROBLEMA

Esta secção permite descrever o motivo pelo qual surgiu a ideia do desenvolvimento da aplicação A4A. Encontra-se dividido em 3 partes: saúde em Portugal (Secção 2.1.1), audiologia (Secção 2.1.2) e perda auditiva (Secção 2.1.3).

2.1.1 *Saúde em Portugal*

A saúde é um tema cada vez mais presente e fundamental a ter em conta ao longo dos últimos tempos. O surgimento da pandemia de Covid-19 veio abalar sensivelmente todos os meios e demonstrar mais uma vez o papel principal da saúde no nosso bem estar. Esta é considerada o equilíbrio presente em cada um de nós e influencia diretamente a nossa qualidade de vida, devido a estar relacionada com o bem-estar físico, mental e social, seja na vida pessoal ou profissional. De certo modo, não existe um termo moralmente correto ou errado sobre o que é a saúde, pois esta é influenciada por diversos fatores que variam entre sociais, pessoais, profissionais entre outros [4].

A [Organização mundial de saúde \(OMS\)](#) lançou em 2010 o relatório mundial da saúde, no qual é referido: “A boa saúde é essencial para o bem-estar humano e para o desenvolvimento económico e social sustentado”. Esse mesmo relatório apresentava como objetivo engrandecer e abrir portas para a descoberta universal de saúde, isto é, conseguir globalizar o acesso a mais e melhores meios que providenciassem uma melhor saúde [5]. Deste modo, seria possível melhorar a qualidade de vida das pessoas através da realização dos pontos presentes no relatório referenciado.

2.1.2 *Audiologia*

A audiolgia trata-se de um ponto bastante importante no bem-estar, sendo fundamental para uma boa qualidade de vida, pois apresenta como objetivo avaliar, preservar e reabilitar as duas principais funções do ouvido, sendo elas a audição e o equilíbrio. Scaranello [6] refere que a deficiência auditiva exerce um impacto importante a nível social, económico e psicossocial, seja no indivíduo portador de deficiência auditiva, como na sua família e comunidade em que se insere [7].

É em 1952 que surge o primeiro laboratório de audiolgia em Portugal, sendo este o Hospital dos Capuchos na cidade de Lisboa. Ao longo dos anos, foram criados cursos relacionados com as várias especialidades, de modo a permitir o surgimento de novas carreiras profissionais que pudessem formar técnicos de diagnóstico e terapia na área da audiolgia. Foi ainda criado um mestrado em audiolgia em 1999, pela faculdade de ciências médicas da Universidade Nova de Lisboa, em parceria com a universidade de Kansas (EUA) [8].

2.1.3 *Perda Auditiva*

É estimada a existência de cerca de 1 milhão de pessoas em Portugal que sofram uma perda auditiva, o que representa 10% da população, sendo este um valor bastante considerável [3]. No entanto, este valor sobe para os 500 milhões quando são analisados todos os casos de perda auditiva em todo o mundo. A causa deste extraordinário aumento refere-se a estarmos cada vez mais expostos a ruídos no nosso dia a dia, como são os casos de concertos e ao trânsito das cidades, além da maior afluência no consumo de tabaco, álcool, entre outros [9]. Existe ainda o atual acesso facilitado a novos e melhores equipamentos de reprodução de som, como *headsets*, *headphones*, intra auriculares, auriculares, colunas, *soundbars*, entre diversos outros, que não têm facilitado a estagnação deste problema, podendo ser uma das principais causas para o aumento [10].

Segundo a *Minisom*, as principais causas da perda de audição são as seguintes [11]:

- Exposição a ruídos altos;
- Fatores hereditários e algumas doenças, como diabetes e hipertensão;
- Medicamentos ototóxicos;

- Neuroma acústico;
- Cera nos ouvidos;
- Infecções virais;
- Traumatismo na cabeça;
- Exposição prolongada a ruído;
- Consumo de tabaco, drogas, álcool e excesso de peso.

A perda auditiva não se identifica apenas em pessoas com uma idade mais avançada, sendo que este problema pode acontecer praticamente em qualquer altura da vida e com igual incidência em ambos os sexos. Geralmente, a maior parte das pessoas sofre perdas de audição de forma gradual com o aumento da idade e é deste modo que se explica a elevada incidência da doença e/ou problema na população mais sénior. Infelizmente, existe um desdém e incultura que providencia que uma pessoa demore cerca de 7 anos a procurar ajuda para o seu problema auditivo [3]. Este longo tempo de atuação é extremamente desfavorável, pois é de grande importância tratar o problema o mais rápido possível, devido a existir uma maior possibilidade de reabilitação auditiva nos primeiros tempos do surgimento e/ou consciência do problema.

É esperado que 60% da população nacional com mais de 65 anos venha a padecer deste problema. Esta perda auditiva é gradual e quando se começa a “ouvir mal” (expressão popular) pode não ser logo evidente. Tal proporciona o demorado tempo para procurar ajuda e conseguinte a origem de problemas mais graves e alguns mesmo irreversíveis [11].

Sintomas de uma pessoa que sofra de perda auditiva [12]:

- Pede constantemente para repetirem o que foi dito;
- Tendência para falar mais alto;
- Ouvir conteúdo multimédia com um som mais alto;
- Tem dificuldade em acompanhar conversas de grupo;
- Sente algum desconforto e intolerância aos sons intensos.

Já as consequências da perda auditiva, segundo a *Minisom* são as seguintes [11]:

- Dificuldade de concentração;
- Baixa autoestima;
- Medo de relacionamentos sociais;

- Depressão e Stress;
- Dor de cabeça;
- Tensão muscular;
- Pressão arterial mais elevada.

2.2 ESTUDO DE APLICAÇÕES

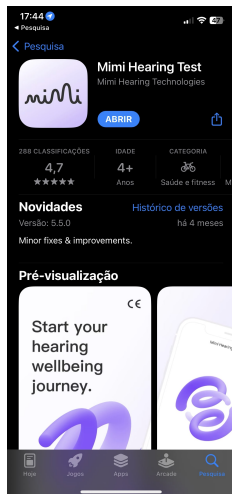
Esta secção pretende descrever o estudo realizado sobre outras aplicações móveis existentes na loja *App Store*. Este estudo permite encontrar um equilíbrio entre as diversas aplicações que já existem no mercado e que são comparáveis à aplicação a ser desenvolvida, de modo a verificar novos ideais. São descritas 3 aplicações (Secção 2.2.1, Secção 2.2.2 e Secção 2.2.3), sendo que a escolha das 2 primeiras deve-se à sua popularidade na *App Store* (principais opções devolvidas aquando da realização da pesquisa) e a última trata-se de uma aplicação desenvolvida pela entidade acolhedora. Embora já se encontrasse estipulado um *design* para algumas funcionalidades na versão *Android* da aplicação, foi recorrido ao estudo de aplicações que apresentam funcionalidades equiparáveis. Este estudo apresenta apenas um valor de curiosidade, pois não afeta diretamente o desenvolvimento do projeto, devido a ser necessário corresponder aos requisitos estabelecidos pela empresa. Apresenta, no entanto, um ponto de consideração para a possível formulação de melhorias ao desenvolvimento e uso da aplicação. Foi utilizada a pesquisa de “Teste auditivo” na loja *App Store* do **iOS**. De referir que as informações a seguir representadas referem-se ao momento da escrita do relatório e podem, a qualquer momento, ficar desatualizadas.

2.2.1 *Mimi Hearing Test*

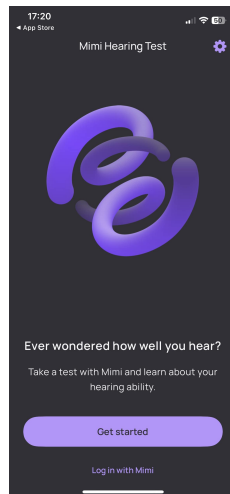
A aplicação *Mimi Hearing Test* (ver Figuras 2 e 3) é o primeiro resultado perante a pesquisa anteriormente referida. Apresenta uma pontuação de 4.7 estrelas com 288 classificações. Os comentários providenciados tendem a classificar a aplicação como: “Funcional”, “Simples” e “Fiável”. Do mesmo modo, existem comentários que referem alguns *bugs* presentes na aplicação, mas que aparentemente já se poderão encontrar resolvidos, devido à aplicação apresentar novas versões com a descrição “*Fixes & Improvements*” e os comentários serem anteriores a essas versões.

De facto, a aplicação apresenta um fluxo bastante simples e intuitivo, sendo necessário providenciar o acesso ao microfone, de modo a garantir que o utilizador se encontra num espaço tranquilo/calmo (“*quiet*“ é o termo presente na aplicação). Existe ainda a necessidade de indicar o ano de nascimento, a utilização de auscultadores e manter um nível de volume de 50% para a realização de um dos dois testes disponíveis.

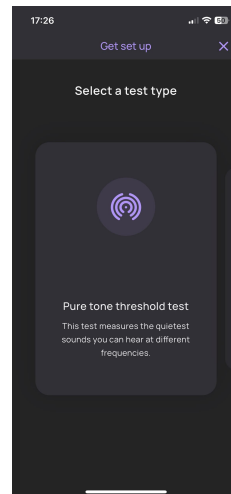
De seguida, são apresentadas algumas fotos relativas à aplicação *Mimi Hearing Test*:



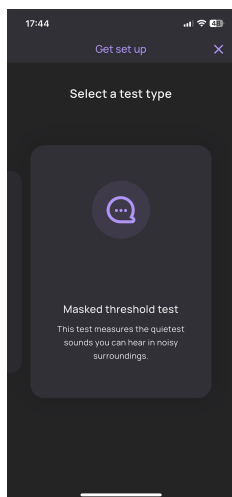
(a) *Mimi Hearing* na App Store



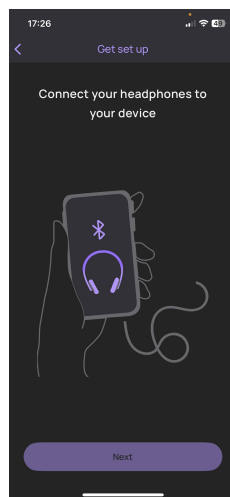
(b) Página inicial



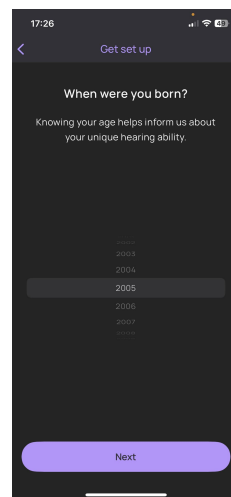
(c) Teste 1



(d) Teste 2



(e) Uso de auscultadores



(f) Data de nascimento

Figura 2: Aplicação *Mimi Hearing* iOS

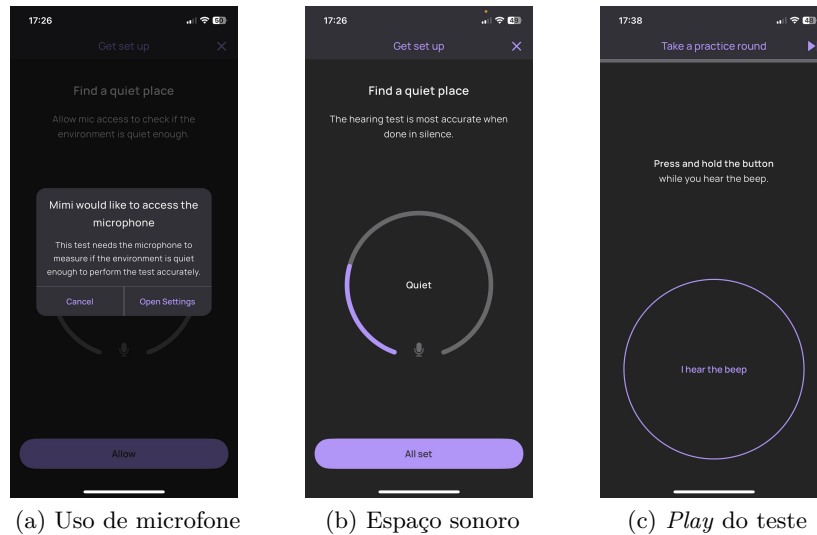


Figura 3: Continuação da aplicação *Mimi Hearing iOS*

2.2.2 *Biosom - Teste Auditivo*

A aplicação *Biosom - Teste Auditivo* (ver Figura 4) encontra-se após a *Mimi Hearing* na *App Store*. Infelizmente, para a utilização da mesma é necessário proceder ao registo de conta e devido a ser suscetível o recebimento de *spam* com a criação da mesma, não foi criado um registo. Assim, foram analisadas as fotografias presentes na página da aplicação na *App Store* e é possível concluir que esta aplicação aparenta ser mais complexa e menos dinâmica do que a anterior, *Mimi Hearing*. Apresenta um visual mais complexo e menos intuitivo para a utilização por parte do utilizador. Do mesmo modo, parece ser apenas disponibilizado um teste, comparativamente aos dois testes disponibilizados pela *Mimi Hearing*. É de igual forma obrigatório o uso de auscultadores, assim como a realização do teste num lugar tranquilo/calmo. Não existe a obrigatoriedade de manter o nível de som nos 50%, mas sim verificar o quão baixo é possível ouvir os sons. Não existem classificações ou comentários atribuídos à aplicação.

De seguida, são apresentadas algumas fotos da aplicação, disponibilizadas na sua página na *App Store*, e outras provenientes da instalação da mesma:

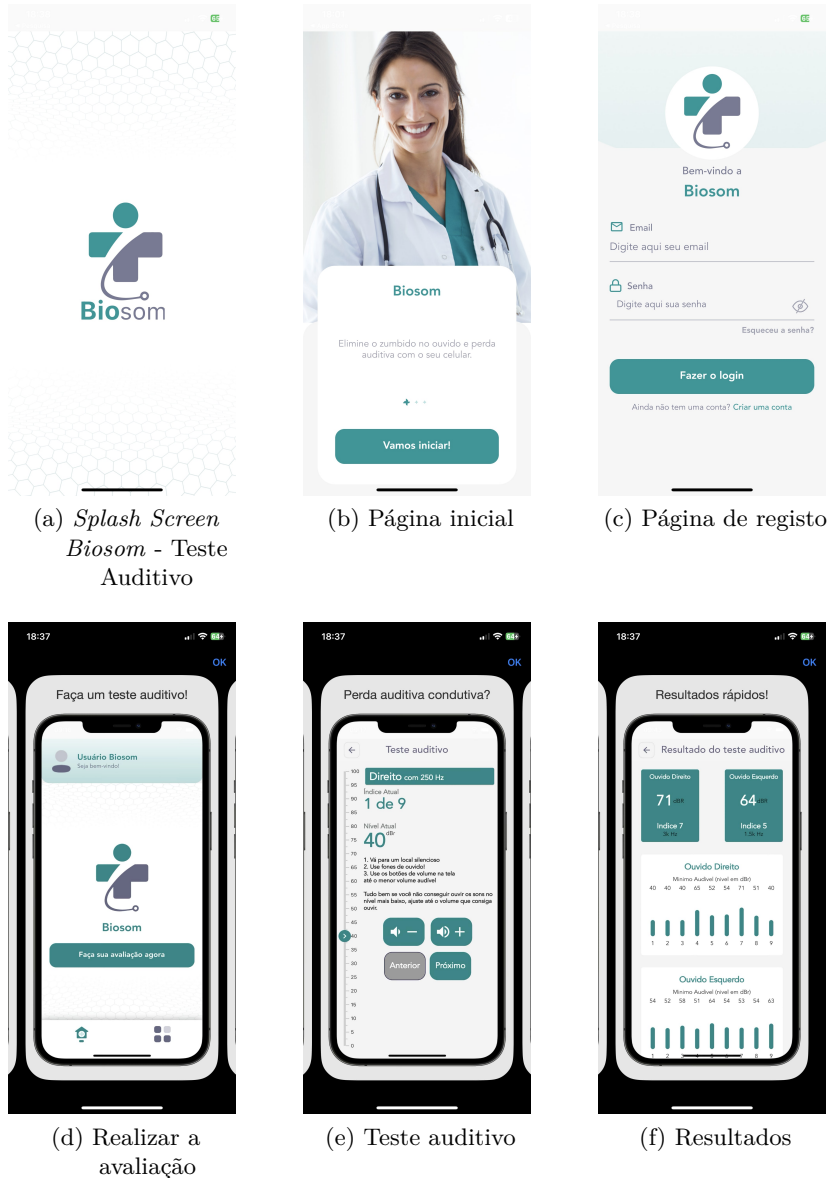


Figura 4: Aplicação *Biosom* - Teste Auditivo *iOS*

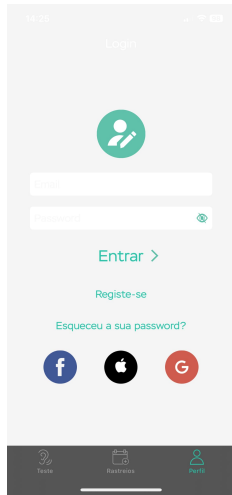
2.2.3 *Evollu App*

A aplicação *Evollu* (ver Figura 5) foi desenvolvida pela *Evollu* e trata-se da disponibilização de um teste Tonal para permitir que os utilizadores diagnostiquem problemas auditivos. São requeridos alguns requisitos como o nível de volume a 100%, o uso de auscultadores e o *login* de conta para obter acesso aos resultados. Apresenta 2 classificações de 5 estrelas sem nenhum comentário associado e um

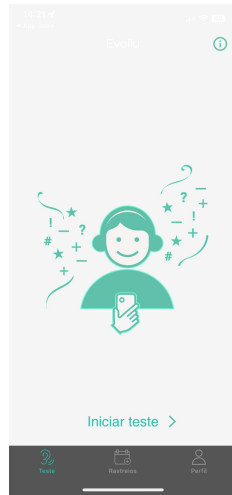
longo histórico de lançamento de versões, que acabou por ser terminado há dois anos atrás.

No geral, a aplicação encontra-se funcional e apresenta um *design* diferente dos atuais padrões da *Evolvu*, proporcionando uma experiência de utilização pouco acomodativa comparativamente aos outros projetos da *Evolvu*. A escolha de cores também não é a mais correta, pois são utilizados brancos por cima de tons de bege, o que torna muito difícil a leitura de diversas palavras.

De seguida, são apresentadas algumas fotos da aplicação *Evolvu*:



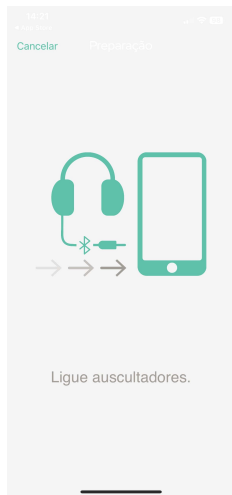
(a) Autenticação



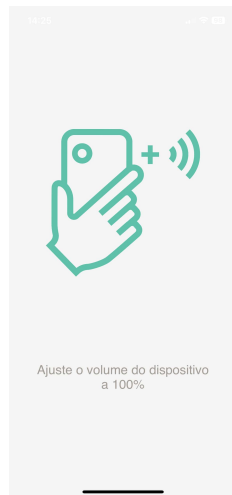
(b) Página inicial



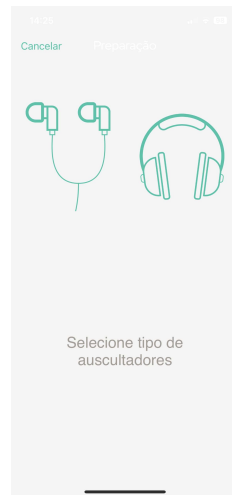
(c) Informação da aplicação



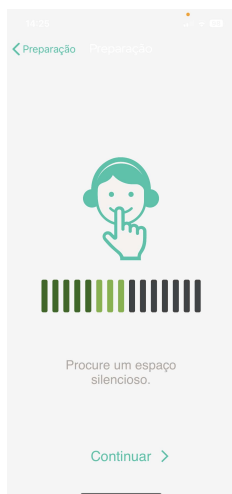
(d) Ligar os auscultadores



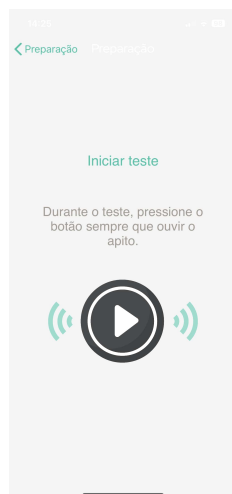
(e) Colocar o volume no máximo



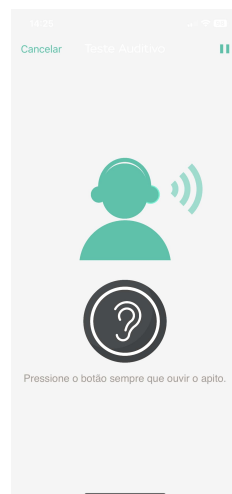
(f) Escolher os auscultadores



(g) Espaço sonoro



(h) Informações antes de iniciar o teste



(i) Teste em execução

Figura 5: Aplicação *Evolu*

METODOLOGIA DE DESENVOLVIMENTO

Neste capítulo é descrito o termo de metodologia *Agile* (Secção 3.1), a metodologia utilizada pela *Evolvu*, assim como a descrição do tipo de *sprints* realizadas na empresa (Secção 3.2). De seguida, é retratada a plataforma utilizada pela empresa, o Jira (Secção 3.3).

3.1 METODOLOGIAS AGILE

A *Evolvu* integra uma metodologia de desenvolvimento *Agile*, nomeadamente o *Scrum*. As metodologias *Agile* surgiram no início do anos 2000, mais concretamente em fevereiro de 2001, sendo apresentada por 17 especialistas em desenvolvimento de *Software* [13]. De acordo com o relatório *Success Rates Rise* do *Pulse of Profession* de 2017, estas metodologias rapidamente modificaram o mercado e o **Ácido desoxirribonucleico (ADN)** das empresas, sendo que nesse ano cerca de 71% das empresas estudadas utilizavam uma metodologia *Agile* para o desenvolvimento dos seus projetos [14].

Diferenciam-se de outros modelos de desenvolvimento mais rígidos, como o caso do *Waterfall*, que apresenta um método de planear o desenvolvimento do projeto do início ao fim, sem que seja dado valor e/ou lugar a possíveis imprevistos que possam surgir. Em contrapartida, a implementação de metodologias *Agile* possibilita que sejam resolvidos contratempos e/ou funcionalidades que não estivessem inicialmente previstas no início do projeto em tempo útil. Sucintamente, estas metodologias permitem adotar a melhor estratégia para cada momento, sem que seja comprometido o desenvolvimento do projeto, assim como aprimorar a colaboração entre os clientes e a equipa, melhorando desta forma a satisfação do cliente [13].

Principais pontos para a adoção das metodologias *Agile* [15] :

- *Software* com boa *performance*;
- Integrações entre pessoas;
- Resposta rápida para imprevistos e mudanças;

- Maior colaboração com o cliente → Melhor atendimento dos requisitos do cliente → Aumento da produtividade;
- Processo incremental - Adaptabilidade - Simplicidade.

3.2 SCRUM

Como referido na secção anterior (Secção 3.1), a metodologia *Agile* utilizada pela *Evollu* é o *Scrum*. Esta permite aos programadores de *software* optarem por uma abordagem que facilite a resolução dos problemas e paradigmas complexos, tal como a possibilidade da entrega do produto segundo as necessidades do cliente. Esta última deve-se ao aumento da produtividade e criatividade conseguido pela abordagem da metodologia [16]. O *Scrum* é definido através de *Sprints* (nome provém da analogia de *Sprint* → corrida de velocidade) sendo um esforço para concretizar todas as *UserStories* (tarefas) o mais rapidamente possível e entregar funcionalidades a funcionar ao cliente [15].

As *sprints* na *Evollu* apresentam a duração de 14 dias (2 semanas), iniciam/terminam normalmente à segunda-feira e são acompanhadas por diversas reuniões:

- *Daily Meeting*;
- *Sprint Review*;
- *Sprint Retrospective*;
- *Sprint Planning*.

As *Daily Meeting*, como o próprio nome indica, são realizadas todos os dias. Trata-se de uma reunião para a agregação de valores e informação por parte de toda a equipa, sendo que cada membro indica o que realizou no dia de trabalho anterior. Essas informações referem-se a todas as reuniões em que terá participado, os objetivos e/ou funcionalidades desenvolvidas, interações com outros elementos de equipa (caso se tenha recorrido a outros membros da equipa para o desenvolvimento de alguma funcionalidade, para auxílio em tarefas, entre outros), o estado atual das suas tarefas, a possível presença de *Blockers* (situações que impedem de avançar no desenvolvimento das tarefas) e o plano de trabalho para o dia atual.

A *Sprint Review* é uma reunião realizada no dia anterior ao encerramento da *Sprint*. Esta tem como objetivo realizar uma avaliação geral sobre as *stories/tasks/bugs*, entre outros, que foram realizados ou que terão que ficar para a próxima *Sprint*. De igual forma, são pontuadas todas as tarefas que atualmente não possuam nenhum

Score atribuído, para caso essa tarefa seja dada como *Accepted*, ser possível encerrar a *Sprint* (*Sprints* não podem ser encerrados com tarefas concluídas que não tenham pontuação associada).

A *Sprint Retrospective* é igualmente realizada no dia anterior ao encerramento da *sprint*. Esta permite realizar um *overview* aos pontos positivos e negativos sobre como ocorreu a *sprint* atual. Suscita que sejam diagnosticados problemas e são desta forma exploradas soluções que possam resolver esses problemas, assim como a verificação da aplicação das soluções definidas na *sprint* anterior.

A *Sprint Planning* é realizada antes do início da *Sprint*. Equipara-se à *Sprint Review*, mas neste caso são analisadas e discutidas todas as funcionalidades, *bugs*, entre outros, que serão tidos em conta ao longo da nova *Sprint*. Apenas as *Tasks* carecem de serem pontuadas (*Story Points*) e esta pontuação é atribuída tendo em conta um conjunto de métricas: tempo de desenvolvimento, tempo de pesquisa, tempo de teste, valor que a funcionalidade irá acrescentar ao projeto, entre outros. Todos os membros de equipa participam na pontuação, podendo atribuir um valor tendo em conta a sequência de *Fibonacci* (1,2,3,5,8,+8). É realizada uma moda a todas as pontuações e é atribuída essa moda à pontuação da tarefa (em determinados casos pode ser necessário discutir melhor a tarefa, por exemplo, em casos dos valores de pontuações serem muito díspares em cada membro, ou empate entre valores). Nos casos de pontuação +8 deve-se dividir o contexto da tarefa em mais tarefas, por exemplo criar um *Epic*. Deste modo, encontra-se assim definido um *Sprint Backlog*.

Existem 3 *Roles* no *Scrum* sendo:

- *Product Owner* → é o responsável por maximizar o valor do produto, o valor da equipa de desenvolvimento, fazer o levantamento de requisitos, encontrar-se com o cliente, intermediário entre a equipa de desenvolvimento e o cliente, identificar as funcionalidades a implementar e a prioridade das mesmas. Este último trata-se do *Product Backlog*.
- *Scrum Master* → é a pessoa que garante que a metodologia *Scrum* é aplicada e compreendida corretamente, é o líder que ajuda a equipa a moldar-se e adaptar-se à metodologia.
- *Equipa Desenvolvimento* → são os *developers* que desenvolvem as funcionalidades.

3.3 PLATAFORMA JIRA

A *Evollu* utiliza a plataforma *Jira* para o desenvolvimento de *software*, de modo a conseguir gerir todos os projetos que se encontram em desenvolvimento. Esta plataforma é tida como uma das mais populares e utilizadas pelas empresas com equipas que trabalham em metodologias *Agile* [17].

A plataforma disponibiliza quadros *Kanban* (onde são inseridas todas as *Tasks*, *Stories*, *Bugs*, entre outros) para que o *Product Owner* consiga definir o *Product Backlog* e o *Sprint Backlog*. Estes quadros *Kanban* encontravam-se inicialmente divididos em 7 colunas, sendo estas:

1. *Not Started* → Todas as *Tasks*, *Stories*, *Bugs*, entre outros que fazem parte da *Sprint* atual, mas ainda nenhum programador começou a trabalhar nelas;
2. *Blocked* → Tudo o que se encontra nesta coluna significa que atualmente não é possível continuar o desenvolvimento devido a algum contratempo;
3. *In Progress* → Tudo o que está atualmente a ser desenvolvido por pelo menos um programador;
4. *Peer Review* → Outro programador está a validar o conteúdo desenvolvido pelo anterior.
5. *QA* → As funcionalidades foram implementadas e precisam de ser validadas por um programador que não tenha participado diretamente no desenvolvimento das mesmas;
6. *Acceptance* → As funcionalidades foram aprovados no teste de *QA* anterior. Carecem atualmente de serem validadas pelo *Product Owner*;
7. *Done* → As funcionalidades foram implementadas e aceites, sendo assim dadas por finalizadas.

Posteriormente, foram acrescentadas as colunas de (ver Figura 6):

1. *Ready to Demo* → Quando um produto se encontra em fase estável para ser disponibilizado em ambiente de desenvolvimento;
2. *Ready to Production (PROD)* → Quando um produto se encontra em fase estável para ser disponibilizado em ambiente de produção.



Figura 6: Colunas do quadro *Kanban*

Desta forma, além da definição de prioridade de cada funcionalidade, este planeamento permite dar a conhecer aos desenvolvedores qual o atual estado de todas as funcionalidades e equilibrar o esforço e a quantidade de trabalho de toda a equipa de desenvolvimento.

3.3.1 Registo de tempo

O registo de tempo no desenvolvimento de cada tarefa é muito importante, pois permite que a equipa responsável pela análise dos dados no *Jira* consiga obter mais detalhes sobre o desenvolvimento (ver Figura 7). Desta forma, será mais fácil e preciso, estipular qual será o tempo necessário para desenvolver uma funcionalidade, por exemplo, o que permite que seja, de igual forma, mais fácil estipular quantas *tasks* serão colocadas na *sprint* aquando do início da mesma.

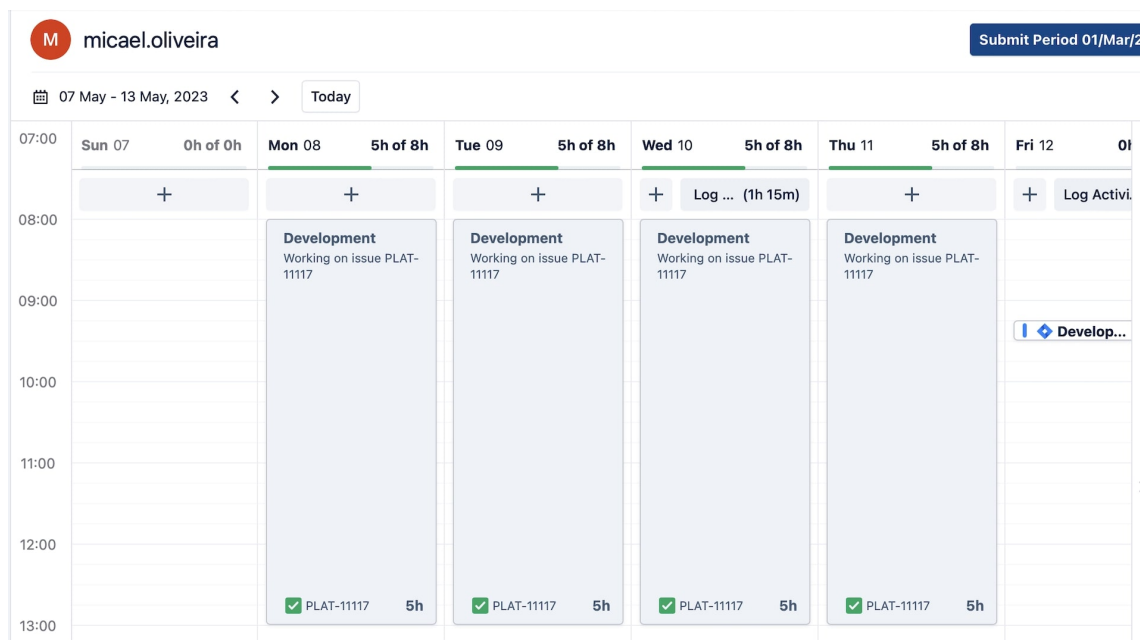


Figura 7: Registo de tempo no *Jira*

DESENVOLVIMENTO

Neste capítulo é descrita a arquitetura da aplicação (Secção 4.1), os testes desenvolvidos (Secção 4.2), assim como as *UserStories* (Secção 4.3), as tecnologias utilizadas (Secção 4.4) e as funcionalidades comuns entre os testes (Secção 4.5). De seguida é apresentada a interação com o servidor (Secção 4.6), a autenticação (Secção 4.7), métodos que requerem autenticação (Secção 4.8), aspetos técnicos importantes no desenvolvimento (Secção 4.9) e as dificuldades sentidas no desenvolvimento (Secção 4.10).

4.1 ARQUITETURA

Nesta secção é descrita a linguagem de programação utilizada no desenvolvimento do projeto, nomeadamente a *framework SwiftUI*. O porquê do uso do *SwiftUI*, assim como as suas vantagens e desvantagens perante o *UIKit* (Secção 4.1.1), termos do fluxo de dados (Secção 4.1.2) e de tarefas do *SwiftUI* (Secção 4.1.3) também são referidos. Por fim, é retratada a arquitetura geral da aplicação, o servidor e o uso de *packages* (Secção 4.1.4).

4.1.1 Linguagem de Programação

Esta secção descreve a linguagem de programação que foi utilizada no desenvolver da aplicação A4A. Encontra-se dividida nas secções de uso do *SwiftUI* (Secção 4.1.1.1), vantagens do uso da linguagem (Secção 4.1.1.2) e desvantagens do uso da mesma (Secção 4.1.1.3).

4.1.1.1 Uso do *SwiftUI*

Como referido na Secção 1.1 (Interesse pela candidatura), o desenvolvimento de projetos através da *framework* de programação *UIKit* permitiu desenvolver os conhecimentos mínimos relacionados com a linguagem *Swift*. No entanto, esta

mesma *framework* não foi aplicada no estágio, devido a algumas considerações que são apresentadas de seguida:

- Futuro do desenvolvimento de aplicações para o software *iOS*:

Com a introdução do *SwiftUI* por parte da *Apple* em 2019, foi concebido um novo paradigma e possibilidade para a gestão e desenvolvimento de aplicações móveis para o ecossistema da *Apple*. Com a presença do *UIKit* no mercado por 10 anos, surge o *SwiftUI*, uma *Declarative Framework*, que sucintamente, significa que o programador está a declarar algo que quer que aconteça, ao contrário do *UIKit* que se trata de uma *Imperative Framework* - *stating how to do something*.

Exemplos concretos:

SwiftUI → Pedir a alguém que desenhe um carro. Não interessa como é o processo de *design*, apenas interessa o resultado obtido.

UIKit → Pedir para que seja desenhado um carro. São providenciadas direções de como deve ser realizado o processo, de modo a que o resultado final seja o esperado.

O *SwiftUI* foi desenhado para *reactive programming*, o que se está a tornar cada vez mais popular e foi inspirado por outras *frameworks* de *Declarative UI*, como o *React*. Já o *UIKit* não foi inicialmente desenhado para essa abordagem. De um modo geral, a aplicação reage a um evento (interação por parte do utilizador, *timer*, retorno de *webservice*, entre outros) para alterar o seu estado de acordo com o mesmo. Sempre que o estado é alterado, o *SwiftUI* cria uma nova *view tree* e compara-a com a versão anterior, realizando apenas o *rendering* das *views* modificadas [18] (ver Figura 8). Este processo não é computacionalmente exigente, devido ao poder do *Swift struct* [19].

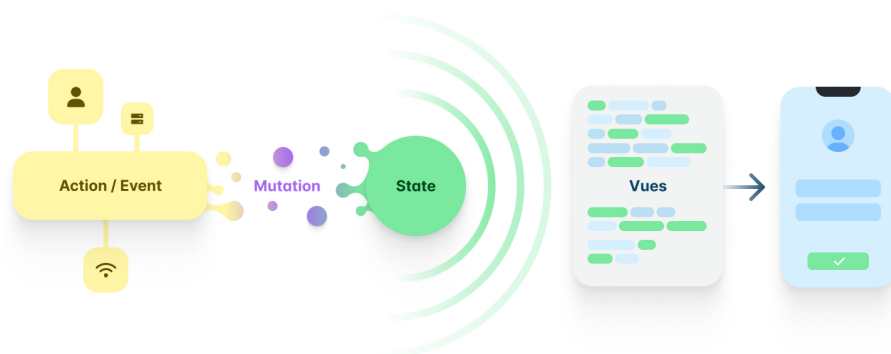


Figura 8: Como funciona o *SwiftUI*. Imagem proveniente de *Atipik* [18]

Mesmo que o *SwiftUI* seja mais otimizado para os novos dispositivos e providencie elementos de programação mais fortes, o que permite que os programadores desenvolvam melhores aplicações, é improvável a *Apple* decidir iniciar a ação de *Deprecate* do *UIKit*. O *Objective C* e *AppKit* nunca obtiveram uma ação de *Deprecate* por parte da *Apple*, nem o é esperado devido ao compilador ser escrito em *C++* e utilizar *Clang* para realizar a compilação (o mesmo *Clang compiler* utilizado no *Objective C*) [20] [21].

No entanto, existem algumas opiniões que referem o *Deprecate* do *UIKit* [22] [23].

- Requisito da empresa:

A *Evolllu* já apresenta no mercado algumas aplicações para dispositivos móveis, sejam estas de sistema operativo *Android* ou *iOS*, assim é uma decisão tecnológica relativamente fácil de ser tomada, tendo em conta o *know-how* da equipa e o conhecimento existente na empresa. Tal permite que sejam estabelecidos diversos padrões de desenvolvimento e a equipa de desenvolvimento da empresa encontrar-se mais a par de todos os projetos referentes a determinada *Framework*. Com a possibilidade de existirem manutenções e/ou reaproveitamento de componentes derivados das aplicações desenvolvidas é extremamente importante manter uma coerência nas plataformas utilizadas. De igual forma, aquando do início do projeto *A4A*, a *Evolllu* utilizava uma metodologia de desenvolvimento nativa para o desenvolvimento de aplicações *mobile*, o que implica que não seriam usadas linguagens de programação híbridas, como *React Native*, *Angular*, *Flutter*, entre outras. De referir que a versão *Android* da aplicação *A4A* encontra-se a ser desenvolvida em *Kotlin*.

- Dinamização de código:

O desenvolvimento em *SwiftUI* é bastante dinâmico através da possibilidade de criação de componentes e da posterior reutilização dos mesmos. À medida que os testes iam sendo desenvolvidos, era aproveitado algum tempo para analisar qual seria a melhor proposta a avançar. Diversas etapas e possíveis soluções eram tidas em conta, de modo a que no final fosse possível analisar qual seria a opção mais viável para o atual problema, assim como para quando surgisse um problema de origem/solução equivalente. Deste modo, além de conseguir satisfazer e resolver o problema atual, era possível combater os possíveis problemas equivalentes que pudessem surgir. Isto permite otimizar bastante a escrita de código, pois não existe a necessidade de estar sempre a escrever o mesmo código para problemas iguais ou equivalentes.

Num exemplo concreto: O teste [PAC](#) apresenta um componente que se repete diversas vezes, o componente retângulo com os cantos arredondados. Foi criada uma *view* que contém esse mesmo retângulo e sempre que for necessário a sua utilização, essa *view* será chamada e será adicionado à mesma o *overlay* com os dados pretendidos (ver Figura 9).

Outro caso concreto: O botão *back* que se encontra presente em cerca de 90% das *views* do teste [PAC](#). Foi criada uma vista que é responsável por voltar atrás na navegação, ou seja, sempre que a mesma é pressionada irá voltar para a vista anterior. Para a sua utilização é apenas necessário integrar a *view* dentro de outra *view* (ver Figura 9).

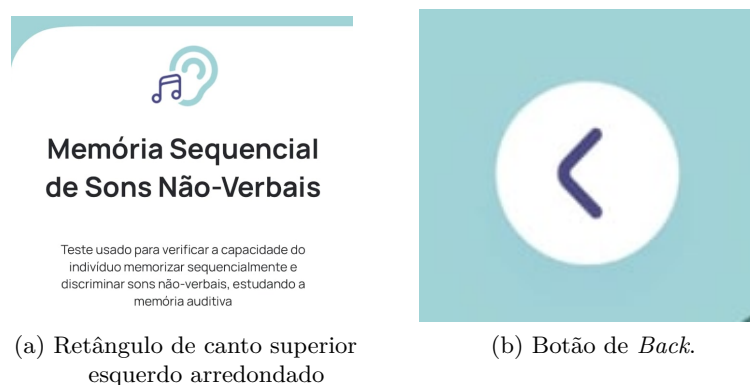


Figura 9: Exemplos de dinamização de código.

4.1.1.2 Vantagens do *SwiftUI*

O *SwiftUI* apresenta algumas vantagens comparativamente ao *UIKit*, de seguida são enunciadas algumas dessas vantagens [24]:

- *Navigation API*:

É possível definir o estado da inicialização, gerir transições entre classes, responder a *links* diretos e muito mais, graças ao controlo programático sobre o comportamento da navegação da aplicação.

- *Advanced Layout Control*:

Permite construir e reutilizar *layouts* para o desenvolvimento das vistas da aplicação. O *SwiftUI* apresenta uma nova *GRID API* que permite oferecer um alinhamento das vistas, seja simultaneamente na horizontal ou vertical, além da implementação das *VStacks*, *HStacks* e *ZStacks* (como o próprio nome indica, as *VStack* irão criar um componente/grupo vertical, *Vertical Stack*,

as *HStack* vão criar um componente/grupo horizontal e as *ZStacks* vão criar um componente/grupo que possibilita colocar elementos uns em cima dos outros). O novo *low-level custom Layout API* permite ter um controlo total sobre a construção exata do *layout* que o programador deseja para a sua *view*. Existe ainda a *ViewThatFits* que permite especificar múltiplas variações de uma determinada *view*, sendo depois o *SwiftUI* a escolher automaticamente a variante que melhor se adapta ao espaço disponível.

- *Better UIKit Interoperability:*

Devido à presença do *UIKit* no mercado há mais de 10 anos, é esperado que o mesmo seja mais completo e adaptável para a resolução de determinadas tarefas. O pouco tempo de mercado do *SwiftUI* poderá não apresentar o mesmo número de documentação e/ou resolução de problemas intuitivamente como o *UIKit*, no entanto, este irá ganhar cada vez mais capacidades ao longo do tempo. O *SwiftUI* foi desenhado para que seja possível trabalhar com outras *interface frameworks*. É possível desenvolver *custom UICollection View cells* ao usar a sintaxe declarativa do *SwiftUI* e ao mesmo tempo integrar essas mesmas *custom cells* com o *UIKit*. Esta interação providencia as funcionalidades esperadas, seja as *swipe actions* ou as *cell backgrounds*, por exemplo.

4.1.1.3 Desvantagens do SwiftUI

O *SwiftUI* apresenta, de igual modo, algumas desvantagens que se encontram seguidamente enunciadas:

- *Framework* recente: A *framework* é bastante recente comparativamente ao *UIKit*, o que significa que muitos recursos ainda não se encontram disponíveis no *SwiftUI*.

Num exemplo concreto: Aquando do desenvolvimento de um *slider*, foi necessário conjugar as duas *frameworks*, pois o componente *slider* que existe no *SwiftUI* não apresenta todas as opções de customização como no *UIKit*.

- Menos documentação disponibilizada: Como referido no ponto anterior, trata-se de uma *framework* recente e o número de documentação disponível ainda é limitado comparativamente ao *UIKit*, por exemplo.
- Infraestrutura baseada em *UIKit* e *AppKit*: As *APIs* do *SwiftUI* são componentes implementados pelo *UIKit* e *AppKit*, o que significa que a estrutura encontra-se limitada em comparação com as imensas possibilidades das estruturas sobre as quais foi construída [18].

4.1.2 Data Flow

No *SwiftUI* existem diversos *property wraps* para permitir o fluxo de dados perante as *views* (ver Figura 10). De seguida, são apresentados os que foram utilizados no projeto [25].

- *State*: Guardar uma variável mutável (passível de sofrer alterações no seu valor) numa *view*.

Exemplo concreto: ter uma variável inicialmente definida com o valor *false*, mas o clique num botão altera o seu valor para *true*.

- *Binding*: Alterar o valor de uma variável que se encontre noutra *view* de hierarquia maior (como se tratasse de uma *view* pai).

Exemplo concreto: Dentro de uma *view*, chamar outra *view* e passar o valor de uma variável para essa segunda *view*. Alterar o valor na segunda *view*, alterando de igual forma na primeira *view*.

- *Environment*: Permite passar dados para outra *view* na hierarquia, independentemente do quão longe se encontram uma da outra e ignorando as *views* intermediárias.

Exemplo concreto: Aceder ao *PresentationMode* em qualquer *view*, de modo a voltar à *view* anterior.

- *Published*: Quando sofrem alterações no valor avisam dinamicamente as outras *views* que necessitam delas sob a ocorrência de uma alteração e realizam um *refresh* nas mesmas. São aplicadas a objetos que utilizem o protocolo *ObservableObject*.

Exemplo concreto: Alterar o valor de uma variável *count* numa *view*, dinamicamente atualiza na outra *view* que utiliza essa variável.

- *StateObject*: Permite a uma *view* criar objetos locais. Normalmente utilizadas para receber eventos como *asynchronous callbacks*.

Exemplo concreto: Aceder a instâncias que já foram criadas anteriormente noutra *view*, ou criar novas instâncias.

- *ObservedObject*: Permite que uma *view* receba um objeto por injeção de dependências de um dos seus antecessores.

Exemplo concreto: Uma *view* acede a um objeto e a *view* posterior que também for aceder terá acesso ao objeto já com os valores alterados na *view* anterior.

- *EnvironmentObject*: Permite que qualquer *view* aceda a objetos globalmente partilhados.

Exemplo concreto: Duas ou mais *views* necessitam de aceder a um objeto ao mesmo tempo e saber quando este foi alterado, sendo que esse valor tem que ser partilhado entre ambas.

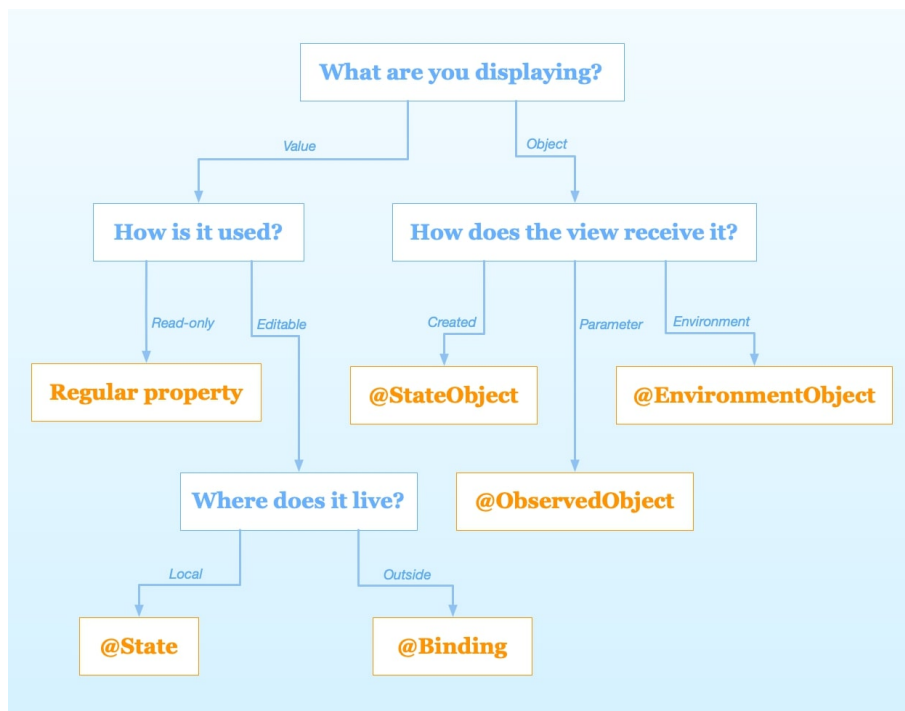


Figura 10: Árvore de decisão do *SwiftUI*. Imagem proveniente de *Matteo Manferdini* [25]

4.1.3 Tarefas específicas

De seguida são referidos alguns *property wraps* do *SwiftUI* utilizados durante o desenvolvimento para a realização de tarefas específicas [25].

- *AppStorage*: Permite aceder ao sistema de *User Default* em qualquer *view*.
Exemplo concreto: Aceder aos dados guardados através do *UserDefaults*. Mais informações na Secção 4.9 ([Aspetos técnicos importantes](#)).
- *UIApplicationDelegateAdaptor*: Permite indicar uma *UIKit/AppKit app delegate* a uma *SwiftUI app*. Uma *app delegate* obtém notificações de diversos eventos globais que ainda não se encontram disponíveis no *SwiftUI*.

Exemplo concreto: Necessário definir uma *app delegate* para o correto funcionamento do despoletar de notificações na aplicação. Mais informações na Secção 4.9.

- *FetchRequest*: Permite à *view* obter informações guardadas no *Core Data* perante um *fetch request*. Utiliza objetos globalmente mapeados no ambiente do *SwiftUI*.

Exemplo concreto: Obter os testes que se encontram guardados localmente no dispositivo. Mais informações na Secção 4.9 ([Aspetos técnicos importantes](#)).

- *GestureState*: Conecta uma propriedade armazenada ao estado de um gesto. Caso o gesto se torne inativo, a propriedade retorna ao valor inicial.

Exemplo concreto: Obter informações relativamente à localização do *input* do utilizador para o teste Zumbidos. Mais informações na Secção 4.10 ([Dificuldades Sentidas](#)).

- *Namespace*: Propriedade utilizada para sincronizar animações de uma *view* para outra.

Exemplo concreto: Na integração da *ScrollView* dentro de outra *view*. Mais informações na Secção 4.9 ([Aspetos técnicos importantes](#)).

4.1.4 Arquitetura geral

A aplicação conta com uma comunicação a um servidor, de modo a que seja possível realizar todos os pedidos necessários para o correto funcionamento da aplicação. De igual forma, foram utilizadas outras *frameworks* para conseguir cumprir os requisitos requeridos. Na Figura 11 é apresentada a arquitetura geral da aplicação, sendo posteriormente explicada em maior detalhe nas Secções 4.1.4.1, 4.1.4.2, 4.1.4.3, 4.6 ([Interação com Servidor](#)), 4.7 ([Autenticação](#)) e 4.9 ([Aspetos técnicos importantes](#)).

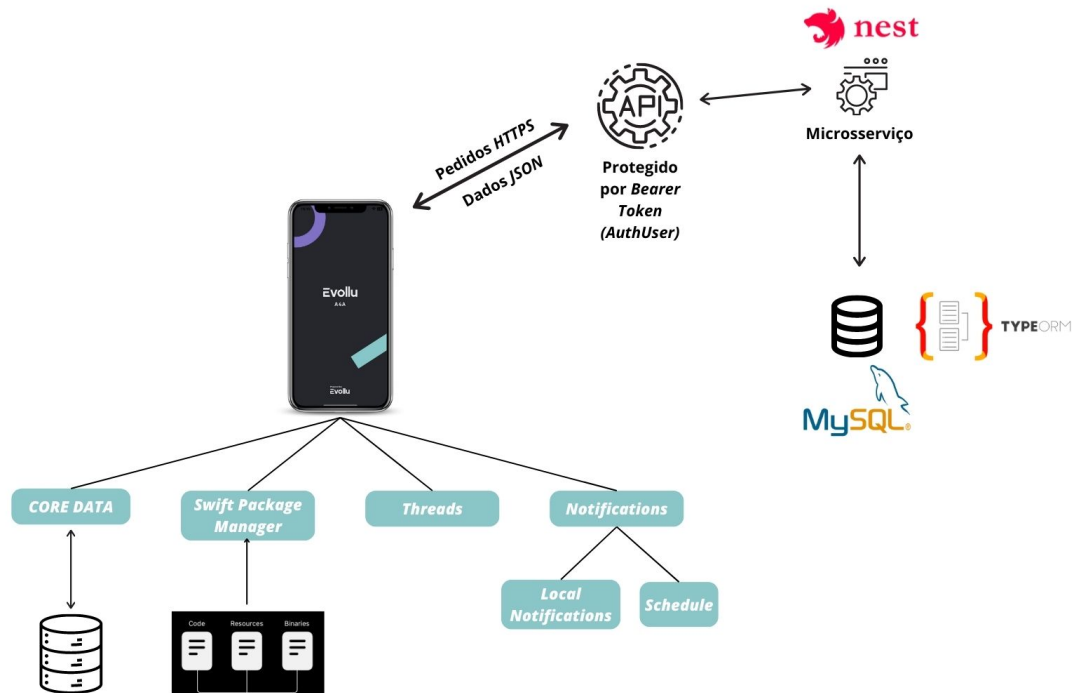


Figura 11: Arquitetura geral da aplicação

4.1.4.1 Base de dados

Foi implementada uma base de dados pelos membros da equipa responsáveis pelo desenvolvimento do servidor (Secção 4.1.4.2). Esta permite guardar os dados de utilização da aplicação, como por exemplo, os testes que cada utilizador realizou e os seus resultados. Desta forma, é possível ter um ponto centralizado com os dados do utilizador e aceder aos mesmos através de pedidos ao servidor. Devido à imposição de confidencialidade por parte da empresa, não é possível referir mais detalhe sobre o tópico.

4.1.4.2 Servidor

A aplicação foi sendo desenvolvida sem recurso a qualquer interação com um servidor até meio de dezembro de 2022. Inicialmente todos os dados presentes nos testes, nomeadamente dados relativos aos textos descritivos dos testes, recomendações providenciadas pela realização dos testes, dados dos questionários, entre outros, existiam apenas localmente na aplicação. No entanto, toda esta lógica deve ser proveniente de um servidor, pois torna-se mais prático e fácil a sua posterior alteração e/ou utilização. Os dados dos questionários e testes poderão ter que ser

consultados numa plataforma *Web*, uma vez que, sem a presença dos dados num servidor, teriam de ser enviados todos os dados dos testes como perguntas, respostas seleccionadas, respostas não seleccionadas, entre outros, o que seria impraticável e desnecessário.

A aplicação envia pedidos RESTful *Hypertext Transfer Protocol Secure* (HTTPS) com dados em formato *JavaScript Object Notation* (JSON) para uma API protegida por *Bearer Token*, baseado na *auth* do utilizador. Os pedidos vão para um micro-serviço em *NEST.js*, sendo depois comunicado internamente com a *MySQL Database* de ORM (ferramenta para mapear entidades com as tabelas da base de dados e automatizar a conversão de *object-to-table and table-to-object* [26]) *TypeORM* (ferramenta de ORM escrita em *TypeScript*, o que facilita o seu uso com a *framework NEST* [27]).

De referir, que na Figura 12 encontra-se representada a API, sendo que esta foi desenvolvida em articulação com os elementos da equipa de desenvolvimento responsáveis pelo desenvolvimento e disponibilização dos métodos necessários à aplicação, desde a autenticação até ao envio dos resultados dos testes.

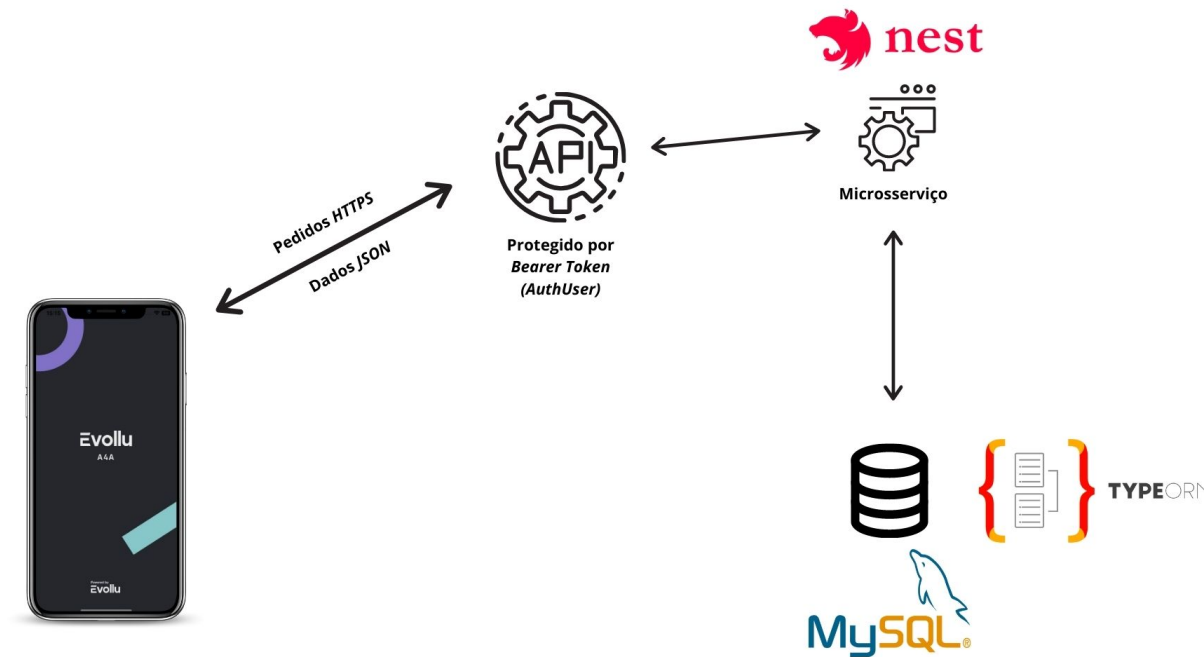


Figura 12: Comunicação com o servidor

4.1.4.3 *Uso de packages*

Ao longo do desenvolvimento do projeto foram utilizados diversos *packages*, de forma a conseguir cumprir o desenvolvimento de diversas funcionalidades. De seguida, são enumerados alguns *packages* que foram utilizados, seguidos de uma breve descrição dos mesmos.

- *ActivityIndicatorView*:

Este *package* foi utilizado para conseguir transmitir ao utilizador um *feedback* perante a realização de pedidos ao servidor. Sempre que são realizados pedidos ao servidor (necessitam de obter uma resposta do mesmo) é apresentada uma *view* de *loading*, permitindo informar o utilizador que a aplicação encontra-se à espera da resposta do servidor. Mais informações podem ser obtidas na Secção 4.9 ([Aspetos técnicos importantes](#)).

- *AudioKit*:

A utilização deste *package* foi extremamente importante para o *play* e alteração dos parâmetros dos sons/áudios, como por exemplo, no teste de zumbidos. Este permite sintetizar, processar e analisar áudio, desenvolvido para as plataformas de *iOS*, *macOS* e *tvOS* [28]. Mais informações podem ser obtidas na Secção 4.10 ([Dificuldades Sentidas](#)).

- *AudioKitEX*:

O *AudioKitEX* faz parte do *AudioKit*, sendo um *C-backed AudioKit DSP* [28].

- *SoundPipeAudioKit*:

O *SoundPipeAudioKit* também faz parte do *AudioKit* e rege-se como a adição de opções de osciladores, efeitos, filtros, entre outros, que podem ser posteriormente aplicados aos sons [28]. Este é extremamente importante para as opções de alteração de intensidade e frequência no teste de zumbidos.

- *Google SignIn*:

Este *package* é necessário para ser possível realizar o *login* através do *Google*. Esta funcionalidade ainda não se encontra totalmente implementada. Mais informações podem ser obtidas na Secção 4.7.2 ([Login secundário](#)) e Secção 6.2 ([Melhorias](#)).

4.2 TESTES

Esta secção fundamenta as funcionalidades principais dos testes e subtestes implementados no desenvolvimento do projeto. Estas informações encontram-se presentes na Secção 4.2.1, 4.2.2, 4.2.3 e 4.2.4.

4.2.1 *PAC*

O projeto começou pelo desenvolvimento do teste *PAC*. Este consiste na capacidade do utilizador conseguir distinguir quais foram os sons que ouviu. Este teste apresenta 4 subtestes que, por sua vez, se podem dividir em 2 modelos, teste de memória ou teste de padrão:

- Teste de memória sequencial de sons não verbais:

Teste usado para verificar a capacidade do indivíduo memorizar sequencialmente e discriminar sons não verbais, estudando a memória auditiva. Este teste consiste no toque de 3 seqüências de 3 ou 4 sons não verbais (o que significa que poderão ser quaisquer sons excetuando palavras, siglas, entre outros) e o utilizador terá que conseguir identificar quais foram os sons que ouviu, assim como a ordem em que os ouviu. Apenas é possível acertar a seqüência, caso acerte todos os sons na ordem correta em que foram tocados.

- Teste de memória sequencial de sons verbais:

Este teste consiste na mesma lógica do anterior (teste de memória sequencial de sons não verbais), mas neste caso são tocados sons verbais.

- Teste de padrão de duração:

Teste usado para verificar a capacidade do indivíduo ordenar a informação temporal, memorizar sequencialmente e reconhecer padrões de duração e discriminar durações de sons tonais. Neste teste são tocadas doze seqüências de 3 ou 4 sons e o utilizador terá de, da mesma forma que os testes referidos anteriormente, identificar qual o som que foi tocado, assim como a ordem pela qual foi tocado.

- Teste de padrão de frequência:

Este teste segue o mesmo parâmetro que o anterior, mudando apenas os sons tocados para padrões de frequência.

4.2.2 *Esforço Auditivo*

Este teste apresenta inicialmente um questionário, onde é possível obter algumas informações sobre o atual estado de saúde auditiva do utilizador. Após responder ao questionário, o utilizador irá realizar um teste de esforço auditivo.

O subteste de esforço auditivo consiste em tocar 3 sequências com uma determinada lista de palavras, em que o número de palavras pertence ao intervalo de {5,6,9,10,12} palavras. O utilizador irá ouvir todas essas palavras e no fim terá de escolher as últimas 3 palavras que ouviu, sempre num conjunto de 9 palavras disponíveis. Ou seja, vai aparecer uma tabela com nove palavras e terá que selecionar as 3 últimas palavras que ouviu. A ordem pela qual seleciona as palavras não é importante, pois o que é realmente avaliado é se o utilizador conseguiu selecionar a antepenúltima palavra que foi tocada.

Nota importante: o utilizador terá apenas 18 segundos para responder quais as palavras que ouviu. Caso não selecione as 3 palavras a tempo, o teste irá prosseguir automaticamente para a próxima sequência.

Após a conclusão do subteste de esforço auditivo, o utilizador terá de realizar o subteste de inteligibilidade. Este subteste é composto por 16 sequências e em cada uma irá tocar 1 som. Posteriormente o utilizador vai ter que selecionar o som que ouviu entre 4 palavras foneticamente parecidas (terá 6 segundos para responder).

4.2.3 *Zumbidos ou Tinnitus*

O zumbido ou acufeno é por definição qualquer som percebido na ausência de uma estimulação acústica [29]. Este teste é realizado pelos utilizadores que possam sentir um acufeno/zumbido permanente e pretendam obter informações/orientações sobre o mesmo. Assim como o teste Esforço Auditivo, este também apresenta um questionário, mas desta vez mais elaborado e detalhado, de forma a conseguir obter o máximo de informação relativamente à saúde auditiva do utilizador. Neste teste, o utilizador conseguirá descrever com o máximo de detalhe como é o zumbido que sente. O teste apresenta 4 ruídos disponíveis para que o utilizador possa escolher e customizar, de modo a reproduzir um ruído que seja equivalente ao zumbido que costuma sentir.

4.2.4 Tonal

Este teste permite verificar quais as frequências (associadas a diversos níveis de *Decibel* (dB)) que o utilizador consegue ouvir. Este termo tem o nome de teste Tonal. Consiste inicialmente num pequeno questionário, seguido das instruções e do teste propriamente dito. Serão tocadas diversas frequências, com variados níveis de dB e o utilizador terá de clicar num botão sempre que conseguir ouvir essa frequência.

4.3 USERSTORIES

De seguida, são apresentadas as tabelas (ver Tabelas 1, 2, 3, 4, 5, 6, 7) referentes às *UserStories* dos testes anteriormente referidos (Secção 4.2 Testes).

Tabela 1: Autenticação perante um utilizador não autenticado

Autenticação, Utilizador não autenticado		
Ação	Objetivo	Importância
Efetuar o <i>Login</i>	Utilizar a aplicação e aceder aos dados pessoais.	Alta
Alterar palavra-passe	Redefinir palavra-passe pessoal	Alta
Visualizar perfil	Visualizar os dados pessoais associados à conta	Médio
Editar perfil	Alterar os dados pessoais associados à conta	Médio
Eliminar conta	Apagar conta pessoal	Alta
Enviar resultados	Guardar dados dos testes/subtestes realizados no servidor, automaticamente ou manualmente	Alta
Obter resultados	Obter dados dos testes/subtestes anteriormente realizados	Alta

Tabela 2: Autenticação perante um utilizador autenticado

Autenticação, Utilizador autenticado		
Ação	Objetivo	Importância
Efetuar o <i>Logout</i>	Sair da conta pessoal.	Alta
Recuperar o acesso à conta	Voltar a ter acesso à conta pessoal	Alta
Realizar o registo na plataforma	Registar nova conta pessoal	Alta
Verificar os dados inseridos	Verificar que dados se encontram errados/carecem de mudança	Alta

Tabela 3: Opções disponibilizadas ao utilizador

Opções disponibilizadas		
Ação	Objetivo	Importância
Interagir com a API do DEEC	Processamento de áudio através de <i>upload</i> ou gravação de áudio.	Alta
Termos e condições	Visualizar os termos e condições do uso da aplicação	Médio
Política de privacidade	Visualizar a política de privacidade do uso da aplicação	Alto
Contactar suporte	Contactar o suporte da <i>Evolllu</i>	Médio
Partilhar aplicação	Partilhar aplicação com outros contactos	Médio
Notificações	Alterar as permissões relativas às notificações	Baixo

Tabela 4: Funcionalidades do teste PAC

Teste PAC		
Ação	Objetivo	Importância
Escolher o teste PAC	Escolher um dos subtestes disponíveis	Alta
Descrição subteste	Visualizar a descrição do subteste	Alta
Instruções subteste	Visualizar as instruções do subteste	Alta
Opções subteste	Escolher opções de realização do subteste	Alta
Pré-ouvir sons	Ouvir sons antes do subteste começar	Alta
Começar subteste	Ouvir os sons da sequência atual	Alta
Pausar subteste	Pausar o toque da sequência atual	Alta
Sair do subteste	Voltar à página inicial	Alta
Selecionar sons	Escolher sons ouvidos durante o toque da sequência	Alta
Resultados subteste	Visualizar resultados do subteste	Alta
Recomendação	Visualizar recomendação desbloqueada	Alta
Próximo subteste	Prosseguir para o próximo subteste ou página inicial	Alta

Tabela 5: Funcionalidades do teste Esforço Auditivo

Teste Esforço Auditivo		
Ação	Objetivo	Importância
Escolher o teste Esforço Auditivo	Iniciar o teste	Alta
Descrição do teste	Visualizar a descrição do teste	Alta
Questionário do teste	Responder às perguntas do questionário	Alta
Informação volume	Ajustar o volume do dispositivo	Alta
Opções subtteste	Escolher opções de realização do subtteste	Alta
Instruções subtteste	Visualizar as instruções do subtteste	Alta
Começar subtteste	Ouvir os sons da sequência atual	Alta
Pausar subtteste	Pausar o toque da sequência atual	Alta
Sair do subtteste	Voltar à página inicial	Alta
Selecionar sons	Escolher sons ouvidos durante o toque da sequência	Alta
Resultados subtteste	Visualizar resultados do subtteste	Alta
Recomendação	Visualizar recomendação desbloqueada	Alta
Prosseguir página inicial	Prosseguir para a página inicial	Alta

Tabela 6: Funcionalidades do teste Zumbidos

Teste Zumbidos		
Ação	Objetivo	Importância
Escolher o teste Zumbidos	Iniciar o teste	Alta
Descrição do teste	Visualizar a descrição do teste	Alta
Questionário do teste	Responder às perguntas do questionário	Alta
Pausar teste	Pausar o toque da sequência atual	Alta
Sair do teste	Voltar à página inicial	Alta
Ajuste grosseiro	Ajustar o ruído grosseiramente	Alta
Ajuste fino	Ajustar o ruído detalhadamente	Alta
Resultados teste	Visualizar resultados do teste	Alta
Recomendação	Visualizar recomendação desbloqueada	Alta
Prosseguir página inicial	Prosseguir para a página inicial	Alta

Tabela 7: Funcionalidades do teste Tonal

Teste Tonal		
Ação	Objetivo	Importância
Escolher o teste Tonal	Iniciar o teste	Alta
Descrição do teste	Visualizar a descrição do teste	Alta
Questionário do teste	Responder às perguntas do questionário	Alta
Instruções teste	Visualizar as instruções do teste	Alta
Começar teste	Ouvir os sons da sequência atual e clicar no botão indicado	Alta
Pausar teste	Pausar o toque da sequência atual	Alta
Sair do teste	Voltar à página inicial	Alta
Página resultados	Escolher o resultado para visualizar	Alta
Resultados teste	Visualizar resultados do teste	Alta
Prosseguir página inicial	Prosseguir para a página inicial	Alta

4.4 TECNOLOGIAS UTILIZADAS

Nesta secção são descritas as tecnologias que foram utilizadas durante o desenvolvimento do projeto. O estágio permitiu que fossem conhecidas e utilizadas diversas tecnologias, sendo que na maior parte, foi a primeira interação com a tecnologia.

4.4.1 *Xcode*

Como referido anteriormente, na Secção 1.1 ([Interesse pela candidatura](#)), o estágio consistiu essencialmente no desenvolvimento de aplicações para o sistema operativo [iOS](#) através do uso do *Xcode* (ver Figura 13). Esta é uma ferramenta para a criação e desenvolvimento de aplicações para a plataforma da *Apple*, não existindo atualmente outra opção para o desenvolvimento nativo em *SwiftUI/UIKit*. Esta ferramenta permite assim gerir o fluxo/*workflow* de todo o desenvolvimento para o ecossistema *Apple*, desde a criação, *testing*, *optimizing* e disponibilização na *App Store*. Esta ferramenta já tinha sido utilizada na unidade curricular do [MEI-CM](#), Desenvolvimento de Aplicações para Dispositivos Móveis [30].

Figura 13: Programa *Xcode*

4.4.2 *TestFlight*

O desenvolvimento de aplicações acarreta a necessidade de disponibilizar versões (normalmente funcionais) para que outros elementos da equipa possam testar e validar o *flow* e a funcionalidade das aplicações (testes de QA). De igual forma, os clientes das aplicações também necessitam de acesso para realizarem os seus testes e validações antes da aplicação ser publicada na biblioteca de aplicações (neste caso na *App Store*). Deste modo, foi utilizada a ferramenta da *Apple* que permite facilitar esta distribuição de versões, o *TestFlight* [31].

Esta ferramenta presente na *App Store Connect* (ver Figura 14) possibilita que sejam convidados *internal testers* (fazem parte da equipa) ou *external testers* (pessoas externas à equipa) para terem acesso às versões das aplicações. Esta partilha é realizada através de:

- *Public Link* → normalmente para os *External Testers*, pois o acesso ao *link* permite instalar a aplicação no dispositivo.
- Convite por *email* → normalmente para os membros da equipa.

É também possível verificar o número de instalações da versão da aplicação, as sessões, as falhas diagnosticadas e os comentários submetidos pelos utilizadores.

Esta ferramenta nunca tinha sido utilizada antes e a utilização da mesma fez com que se tivesse uma opinião positiva pelas suas funcionalidades, usabilidade e simplicidade.

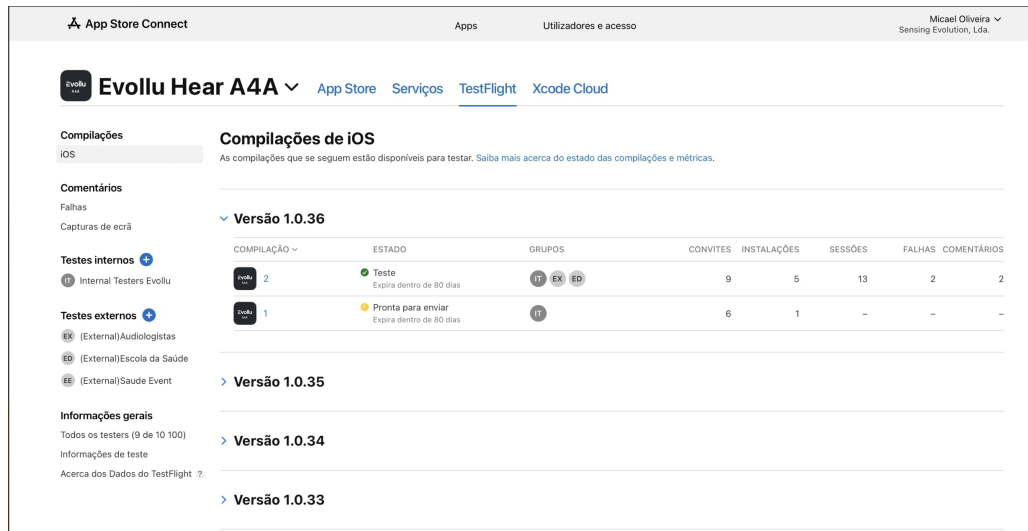


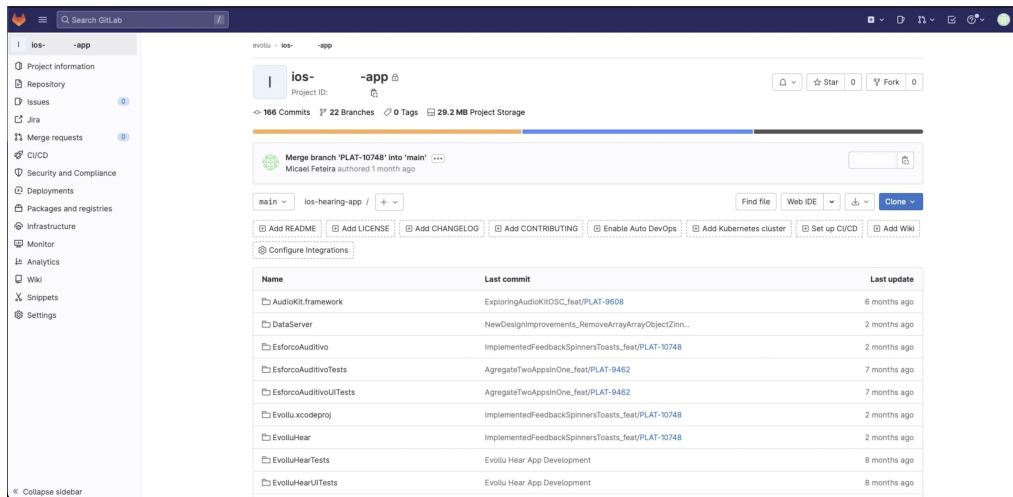
Figura 14: Plataforma *AppStoreConnect*

4.4.3 *GitLab*

Todo o código que é desenvolvido ao longo do projeto necessita de ser guardado num local seguro e de fácil acesso/distribuição/disponibilização para os diversos elementos da equipa. Deste modo, surge o conceito de repositórios, através do uso de plataformas interligadas com o *Git*.

O *GitLab* (ver Figura 15) é uma das diversas plataformas utilizadas para o planeamento da produção e a agregação dos diferentes membros de equipa num único local. Todos os membros da equipa estão sincronizados sobre todo o processo graças às ferramentas de planeamento e gestão, o que ajuda as equipas a desenhar, desenvolver e a gerir de forma segura, simples e rápida o código e os dados dos seus projetos através do sistema de distribuição de versões. De uma forma sucinta, os programadores enviam o código desenvolvido para um repositório que providencia uma fonte escalável, permitindo que as equipas possam ser produtivas sem terem que interromper os seus fluxos de trabalho [32].

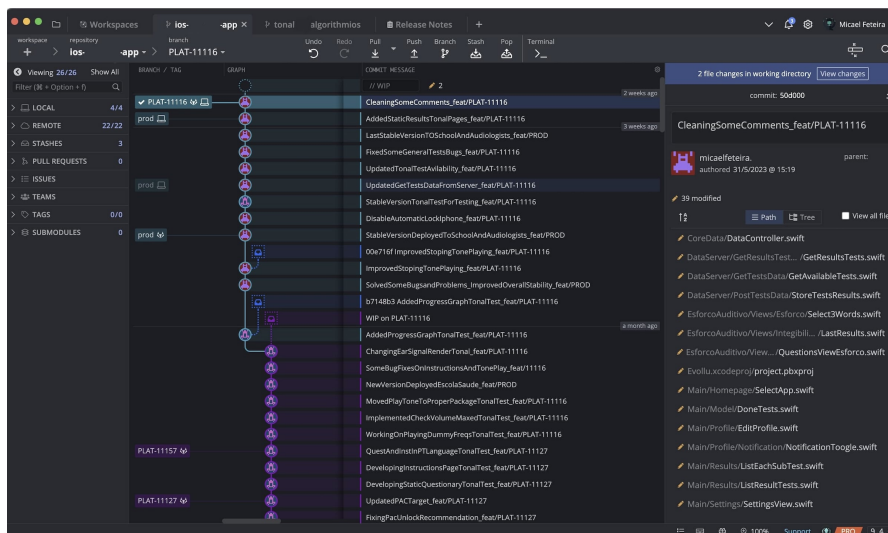
Esta plataforma nunca tinha sido utilizada, pois ao longo do percurso académico foi utilizada a plataforma *GitHub*. De um modo geral, a plataforma *GitLab* distinguiu-se como sendo mais completa do que a plataforma *GitHub*.

Figura 15: Plataforma *GitLab*

4.4.4 *GitKraken*

Existem diversas ferramentas/clientes para interação com as plataformas *Git* e para o desenvolvimento do projeto foi utilizada a ferramenta *GitKraken* (ver Figura 16).

Permite uma integração com o *GitLab* através de uma intuitiva *Git Graphical User Interface (GUI)* e de uma poderosa *Git Command-line interface (CLI)*, o que facilita todas as interações com o mesmo. É possível criar novos repositórios, realizar o *pull*, *commit*, *push*, *stash*, entre outros, de modificações, criação de novas *branches* entre tudo o que esteja relacionado com o *Git CLI* [33].

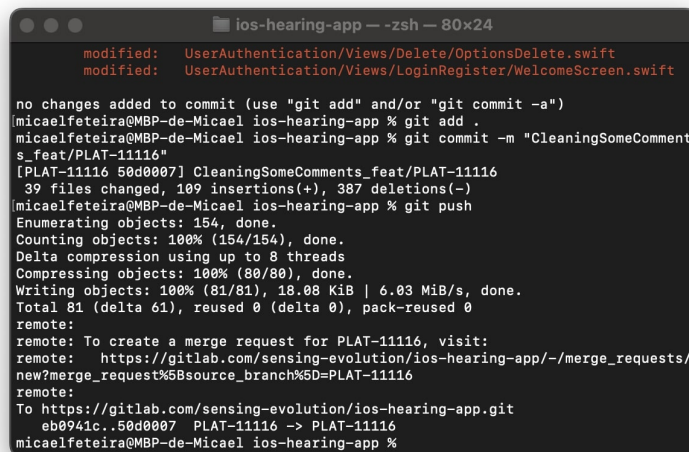
Figura 16: Programa *GitKraken*

A plataforma é bastante reconhecida por ser utilizada por diversas empresas de renome, tais como *Netflix*, *Ubisoft*, *Apple*, *BMW*, *Microsoft*, *Nasa* entre diversos outros nomes. Esta informação está disponibilizada no próprio *website* da plataforma [34].

Esta plataforma nunca tinha sido utilizada, pois ao longo do percurso acadêmico foi maioritariamente utilizado a *Bash CLI* de um *MacBook* para o efeito.

4.4.5 *Bash CLI MacBook*

Foi utilizada a *Bash* do *MacBook* (ver Figura 17) para a realização de comandos direcionados ao sistema de versionamento, de modo a completar com o uso da plataforma *GitKraken*. O uso desta rege-se pela alta fidelidade da mesma, assim como pelo hábito do uso recorrente ao longo de todo o percurso acadêmico na realização de projetos e outras atividades académicas.



```

ios-hearing-app --zsh -- 80x24
modified:  UserAuthentication/Views/Delete/OptionsDelete.swift
modified:  UserAuthentication/Views/LoginRegister/WelcomeScreen.swift

no changes added to commit (use "git add" and/or "git commit -a")
[miciaelfeteira@MBP-de-Micael ios-hearing-app % git add .
miciaelfeteira@MBP-de-Micael ios-hearing-app % git commit -m "CleaningSomeComments feat/PLAT-11116"
[PLAT-11116 50d0007] CleaningSomeComments_feat/PLAT-11116
 39 files changed, 109 insertions(+), 387 deletions(-)
miciaelfeteira@MBP-de-Micael ios-hearing-app % git push
Enumerating objects: 154, done.
Counting objects: 100% (154/154), done.
Delta compression using up to 8 threads
Compressing objects: 100% (80/80), done.
Writing objects: 100% (81/81), 18.08 KiB | 6.03 MiB/s, done.
Total 81 (delta 61), reused 0 (delta 0), pack-reused 0
remote:
remote: To create a merge request for PLAT-11116, visit:
remote:   https://gitlab.com/sensing-evolution/ios-hearing-app/-/merge_requests/new?merge_request%5Bsource_branch%5D=PLAT-11116
remote:
To https://gitlab.com/sensing-evolution/ios-hearing-app.git
 eb0941c..50d0007  PLAT-11116 -> PLAT-11116
miciaelfeteira@MBP-de-Micael ios-hearing-app %

```

Figura 17: Terminal do *MacBook*

4.4.6 *Figma*

O desenvolvimento de qualquer tarefa que envolva *Frontend* (algo que será visível para o cliente) requer que sejam desenvolvidos processos e protótipos de *design*. A ferramenta utilizada por todo o ambiente da empresa para esta interação é o *Figma* (ver Figura 18).

A plataforma permite que os membros da equipa se interliguem a todos os processos de *design* desenvolvidos dentro da plataforma. Apresenta diversas funcionalidades essenciais ao desenvolvimento de qualquer etapa de *design*, seja de aplicações móveis, *Web*, *marketing*, *papers* entre outros projetos [35].

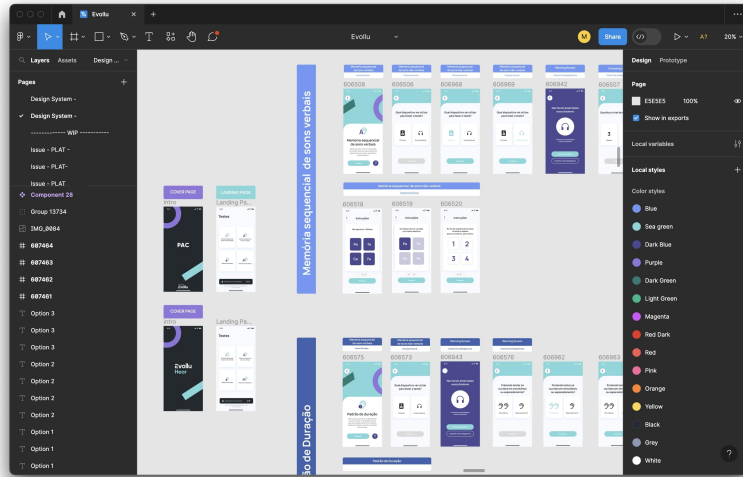


Figura 18: Programa *Figma*

Esta plataforma nunca tinha sido utilizada, no entanto é bastante simples de utilizar e compreender mesmo para quem nunca tenha interagido com a mesma. Apresenta um *tier* de uso gratuito e outros de custo associado, no entanto, comparativamente a outras ferramentas utilizadas ao longo do percurso académico como o *Balsamiq Wireframes* [36] ou o *Miro* [37], esta aparenta ser superior devido ao número de funcionalidades que apresenta [38].

4.4.7 *Jira*

O planeamento e a organização dos processos da equipa, através da metodologia *Agile* são conciliados com a plataforma *Jira*.

Trata-se de uma plataforma da *Atlassian* que permite configurar, organizar e gerir com maior facilidade o fluxo dos projetos. A monitorização de tarefas e o acompanhamento dos projetos e as suas atividades permitem à equipa organizar-se e ser mais autónoma na realização das tarefas [17].

Esta plataforma já tinha sido utilizada numa unidade curricular de licenciatura de Engenharia Informática, [Tópicos Avançados de Engenharia de Software \(TAES\)](#).

4.4.8 Postman

Foi utilizada a *Framework Postman* (ver Figura 19) para conseguir executar pedidos *Web* e obter *feedback* sobre a execução dos mesmos. Esta plataforma é muito importante, pois possibilita diversas informações necessárias à correta disponibilização de serviços de comunicação entre o *FrontEnd* e o *Backend*. É possível verificar se os pedidos devolvem as informações pretendidas, fazer testes com a passagem de diferentes variáveis nos *headers* ou *params* dos pedidos, visualizar os campos devolvidos aquando do sucesso no pedido, entre outros [39]. Trata-se de uma plataforma essencial para a interação com o servidor/API para os diversos pedidos *Hypertext Transfer Protocol (HTTP) Request Methods*, como o *Get*, *Post*, *Put*, *Delete*, *Patch*, entre outros.

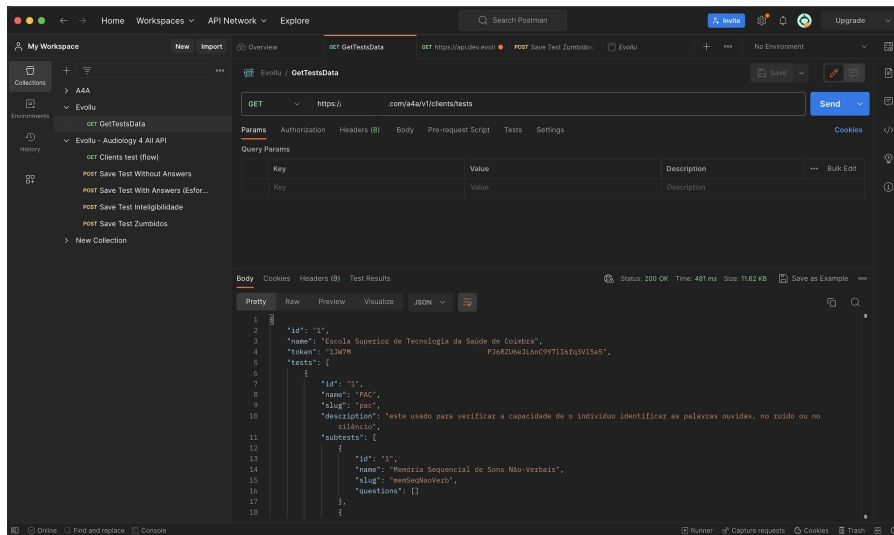


Figura 19: Programa *Postman*

Esta ferramenta já tinha sido utilizada durante o percurso académico.

4.4.9 WireGuard

O *WireGuard* (ver Figura 20) é um *Virtual Private Network (VPN)* rápido e moderno que utiliza um mecanismo de criptografia. Utiliza um *state-of-the-art cryptography* como o *Noise protocol framework*, *Curve25519*, *ChaCha20*, *Poly1305*, *BLAKE2*, *SipHash24*, *HKDF* e *secure trusted constructions* de modo a ser rápido, simples e mais prático. [40]. A sua utilização permitia aceder a serviços que apenas estariam acessíveis dentro da rede da empresa.

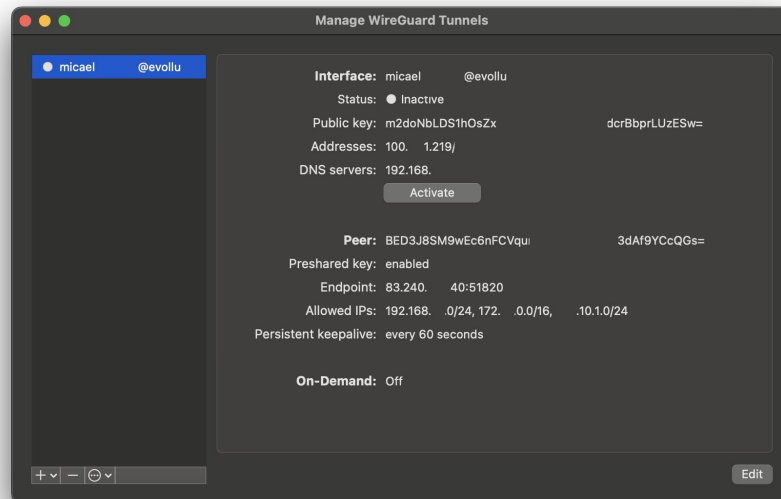


Figura 20: Programa *Wireguard*

Esta ferramenta nunca tinha sido utilizada antes, tendo sido utilizados outros serviços de [VPN](#), como o *OpenVPN - VPNBook*.

4.4.10 *MailTrap*

Algumas funcionalidades requerem o envio de *emails*, como por exemplo a funcionalidade de registrar utilizador ou de recuperar a palavra passe. No caso de registrar o utilizador, este receberá um *email* no *email* que utilizou para realizar o registo, de forma a confirmar e validar esse mesmo registo. Já a recuperação de palavra-passe rege-se pelo envio de um *email* para o *email* registado pelo utilizador (apenas possível caso o *email* esteja registado na plataforma) para que este possa reintroduzir uma palavra-passe para a sua conta. Estas funcionalidades já se encontram corretamente implementadas no (*Environment*)/ambiente de produção da empresa, o que significa que esta interação de diversas etapas funciona automaticamente sem qualquer problema. No entanto, o desenvolvimento inicia-se pelo ambiente de desenvolvimento (*Development (DEV)*) e esse ambiente necessita de uma plataforma como o *Mailtrap* (ver Figura 21) para simular o envio dos *emails* para os *emails* preenchidos [41].

Esta plataforma já tinha sido utilizada durante o percurso académico.

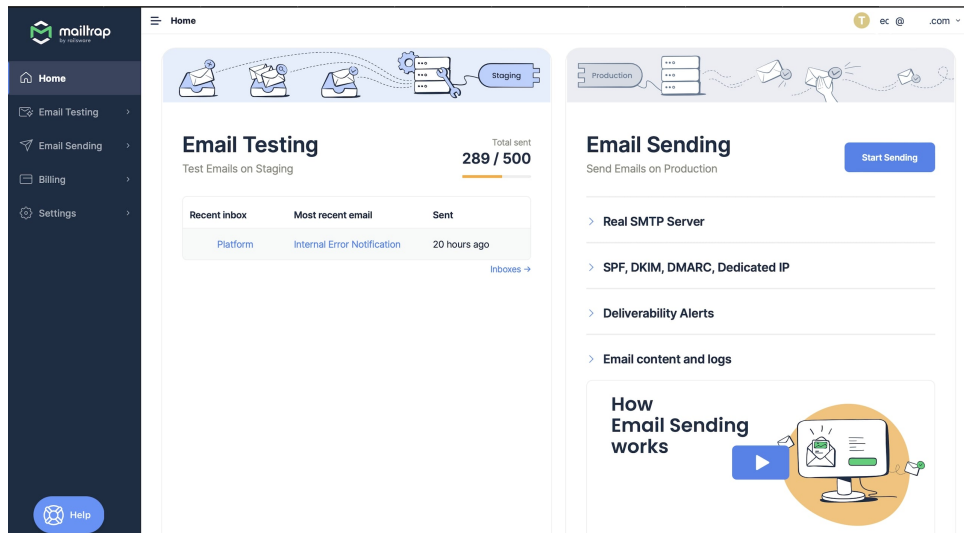


Figura 21: Plataforma *MailTrap*

4.4.11 *Overleaf*

O *Overleaf* é uma ferramenta *online* para a escrita e publicação de texto em *LaTeX* (sistema que inclui recursos destinados à produção de documentos técnicos e científicos [42]) e *Rich Text* (formato de documento específico para o processamento de texto desenvolvido pela *Microsoft* [43]) [44]. Facilita o processo de escrita e edição de documentos científicos, pois permite que o utilizador se foque na criação de conteúdos com qualidade, sem ter de se preocupar tanto com formatações de todo o género.

Esta plataforma foi utilizada pela primeira vez para a escrita deste relatório.

4.5 FUNCIONALIDADES COMUNS

Existem funcionalidades que são comuns ao funcionamento dos vários testes, de modo a tornar a experiência de utilização o mais dinâmica, didática e prática ao utilizador possível. Algumas destas funcionalidades são:

- Pausar o teste:

Devido a alguns testes serem bastante longos e à possível ocorrência de uma ação como a desconexão dos *headsets*, entre outros, é necessário implementar uma opção para a pausa do teste. O teste irá ficar em pausa no momento em que o utilizador clique no botão “X” (ver Figura 22) presente no canto superior direito, ou assim que o utilizador desligue os *headsets* (caso o utilizador

esteja a realizar o teste com recurso aos *headsets*). De referir que a ação de retirar/desconectar os *headsets Bluetooth*, caso estes tenham sensor de proximidade integrado, irá de igual modo provocar a pausa do teste e não apenas quando se desconecta o cabo dos *headsets* do telemóvel. Para continuar com o teste basta clicar na opção “Cancelar”.



Figura 22: Pausar o teste/subteste

Nota importante: Nos subtestes em que existe tempo para responder, não será possível pausar aquando do momento de resposta, de modo a não influenciar a escolha do utilizador. Este caso específico refere-se ao teste Esforço Auditivo, pois tanto o subteste de Esforço Auditivo como o da Inteligibilidade apresentam um tempo limite para responder. No entanto, continua a ser possível pausar enquanto estiverem a ser tocados os sons.

- Sair do teste/subteste:

O utilizador poderá querer sair/desistir de continuar a fazer o teste/subteste. Para tal, basta clicar no mesmo botão “X” presente no canto superior direito (ver Figura 22) e clicar na opção “Confirmar”. O utilizador será reencaminhado para a página inicial do teste (caso seja o teste Esforço Auditivo, Zumbidos ou Tonal), ou para a página de ouvir os sons (no caso de ser o teste PAC).

- Voltar à página de seleção testes:

O utilizador poderá querer sair do teste em que se encontra atualmente, de forma a conseguir realizar outro. Por exemplo, após realizar dois subtestes do

teste PAC pode querer realizar o teste zumbidos. Para tal, necessita apenas de se encontrar na página inicial de cada teste e clicar no botão “X” (ver Figura 23), que se encontra no canto superior direito e selecionar a opção “Confirmar”.



Figura 23: Sair do teste

4.6 INTERAÇÃO COM SERVIDOR

Num caso concreto, os dados dos questionários das aplicações foram todos definidos estaticamente e posteriormente alterados para receber esses mesmos dados do servidor. Tal implicou que toda a informação estática relativa a esses questionários fosse eliminada na aplicação e atualmente quando existe necessidade de alterar a *String* de uma frase por exemplo, basta fazer essa alteração no lado do servidor e será verificada no lado da aplicação quando for feito novamente o pedido.

Outro caso concreto, consiste na possibilidade de eliminar a conta do utilizador. Esta *feature* já necessita de comunicação com o servidor para eliminar o utilizador da base de dados.

Quando o utilizador seleciona a opção de apagar a sua conta é-lhe pedido para que escolha uma opção perante a razão pela qual deseja apagar a sua conta. Antes de considerar a possibilidade de obter estas opções pelo servidor, existia uma classe com um vetor que se encontrava preenchido com diversas opções de resposta, sendo

depois percorrido esse vetor para a apresentação das possibilidades de reposta. Através da interação com o servidor, é possível, de modo dinâmico, conseguir fazer o *link* das opções disponíveis no servidor com as opções *disponíveis* na *Application (App)*. De referir, que a *App* é sempre responsiva ao que o servidor retornar como opções. Na primeira análise ou perspectiva, este método ou opção parece mais complicado, no entanto, isto previne um conflito entre a resposta proveniente da *App* e das respostas presentes no servidor. Caso existisse uma determinada opção no servidor que fosse diferente da *App*, ou vice-versa, criava um conflito de resposta dada pelo utilizador e da resposta permitida/disponível no servidor.

O desenvolvimento do servidor para complementar o desenvolvimento da *App* foi iniciado a meados de dezembro de 2022 pela equipa.

4.7 AUTENTICAÇÃO

Após o início do desenvolvimento do *backend*/(servidor) foi dado início à implementação e desenvolvimento da autenticação na aplicação.

A *Evollu* já desenvolveu diversas aplicações e algumas encontram-se disponíveis para utilizar através do *download* das mesmas na *Apple Store* e/ou *Play Store*. Deste modo, a autenticação foi desenvolvida tendo em conta o domínio *Evollu Auth-Clinical* (domínio partilhado pelas aplicações de saúde da *Evollu*). Não foi necessário criar novos *endpoints*, mas sim adaptar alguns pedidos, como por exemplo, o pedido de registo de utilizador para aceitar determinado campo de dados, de modo a utilizar os serviços de autenticação já utilizados por outras aplicações.

O uso destes *endpoints* apresenta algumas vantagens:

- Desenvolvimento mais rápido:

Se já existem os *endpoints* necessários para realizar a autenticação é mais fácil e rápido de implementar a autenticação, devido a não ser necessário esperar pelo desenvolvimento dos *endpoints* ou de realizar testes muito intensivos e extensos aos *endpoints*.

- Já existem em produção:

Como referido, estes *endpoints* já se encontram a ser usados noutras aplicações o que significa que já se encontram funcionais e fiáveis (devido a já terem sido submetidos a diversos testes). Foram maioritariamente utilizados os *endpoints*

de ambiente de desenvolvimento (*dev*), mas também foi recorrido a *endpoints* de ambiente de produção para a realização de alguns testes.

- Possibilidade de *Login* entre *apps*:

Se for criada/registada uma conta numa aplicação que use o domínio *Evollu Auth-Clinical* para o registo dessa conta, será possível realizar o *login* dessa conta criada noutra aplicação que utilize esse mesmo domínio.

Nas Secções 4.7.1, 4.7.2, 4.7.2.1, 4.7.3 e 4.7.4 são apresentados os métodos de autenticação necessários para a primeira interação com a mesma.

4.7.1 *Login principal*

É necessário realizar um *login* para conseguir utilizar todos os recursos da aplicação. Este *login* é efetuado com recurso aos *endpoints* do domínio *Evollu Auth-Clinical*.

4.7.2 *Login secundário*

Inicialmente estava previsto o desenvolvimento de outros métodos de *login*, maioritariamente o *login* pelo *Facebook*, *Google* e *Apple*. O desenvolvimento dos dois últimos ainda foi iniciado, mas foram posteriormente desconsiderados devido a existirem tarefas com maior nível de prioridade. Já o *login* através do *Facebook* foi cancelado, devido à necessidade de corresponder a diversos protocolos exigidos pela plataforma *Facebook* e no momento não existia tempo e mão de obra para conseguir providenciar todas as informações requeridas à plataforma.

4.7.2.1 *Apple Login*

Sucintamente, para ser possível realizar o *login/signIn* pela *Apple* é necessário ativar o *SignIn with Apple* no *Target* do projeto e recorrer ao *SignInwithApple Button* para realizar o pedido/*request* que irá devolver um *Entity Token*. Esse *Entity Token* terá de ser enviado para o servidor para este conseguir retornar o *Auth Token* do utilizador. Estes passos referem-se apenas à preparação do projeto em *SwiftUI*, sendo também necessário configurar o servidor para lidar com esta autenticação. Nomeadamente a criação de uma chave única, que terá de ser gerada novamente em períodos de 6 em 6 meses, de acordo com as políticas da *Apple* [45].

Através dos *Guidelines* da *Apple* [46] é possível verificar que, se uma aplicação incluir opções de *login* através de outras plataformas como o *Facebook*, *LinkedIn*, *Amazon*, *Google*, entre outros, é necessário habilitar a opção de autenticação através do *Apple Sign-In*.

4.7.3 Registrar um utilizador

Possibilidade de criar/registar um novo utilizador na *App*. Inicialmente após o registo do novo utilizador era realizado o *login* automático com a conta criada. No entanto, essa funcionalidade foi desabilitada, devido a ser necessário confirmar primeiro a conta com recurso ao *email* que é enviado para o utilizador aquando do registo da conta. Esta funcionalidade apenas funciona em ambiente de produção.

4.7.4 Logout

Após o *login* de uma conta, o utilizador poderá realizar o *Logout* da mesma. Permite que seja possível trocar de conta para realizar o *login* com outra, ou realizar o registo de uma nova conta por exemplo.

4.8 MÉTODOS QUE REQUEREM AUTENTICAÇÃO

Nesta secção, são descritos todos os métodos que necessitam de autenticação por parte do utilizador, ou seja, caso o utilizador não esteja autenticado na aplicação ou o *Token* de autenticação (mais informações na Secção 4.9) tenha expirado, não irá ser possível realizar estes pedidos.

4.8.1 Alterar a palavra-passe

O utilizador poderá alterar a sua palavra-passe, sendo para isso necessário indicar qual a sua palavra-passe atual. Caso a palavra-passe atual indicada pelo mesmo coincida com a palavra-passe real do utilizador, este poderá alterar a palavra-passe desde que cumpra com os requisitos atuais (mais informações na Secção 5.3.2.3).

4.8.2 *Repor a palavra-passe*

Caso o utilizador se esqueça da sua palavra-passe atual, poderá realizar uma reposição da palavra-passe. É necessário selecionar a opção “Esqueceu a senha”, presente no ecrã inicial da aplicação, onde é realizado o *login*. Basta inserir o *email* da sua conta e receberá um *email* para esse mesmo *email* (apenas se esse *email* estiver associado a uma conta *Evolu* do domínio apropriado) junto com um *link* para proceder à reposição da sua palavra-passe. Esta funcionalidade apenas funciona no ambiente de produção.

4.8.3 *Editar o perfil*

Após a realização do *login*, o utilizador poderá alterar algumas informações associadas ao seu perfil, de modo a estarem em conformidade com os dados atuais do utilizador.

4.8.4 *Eliminar de conta*

O utilizador tem a liberdade de eliminar a sua conta assim que o desejar. Apenas terá de confirmar que realmente deseja proceder à eliminação da sua conta e escolher uma das opções que expliquem o porquê da sua escolha. Estas opções serão futuramente analisadas pela equipa, de modo a perceber se existem alguns pontos de melhoria que previnam que os utilizadores percam o interesse pelo uso da aplicação ou outras aplicações envolvidas no mesmo *domínio*.

4.8.5 *Enviar os resultados de testes/subtestes*

Quando o utilizador acaba de realizar um teste/subteste, vai ser despoletado automaticamente um pedido para enviar os dados desse teste/subteste para o servidor. Caso esteja tudo em conformidade (utilizador autenticado, *internet* disponível, entre outros) os dados serão corretamente enviados.

4.8.6 *Obter resultados dos testes*

Perante a situação anterior (envio dos resultados) o utilizador poderá posteriormente visualizar quais os testes/subtestes que já realizou ao aceder à opção “Resultados”, presente na *sidebar* da página de seleção de testes.

4.9 ASPETOS TÉCNICOS IMPORTANTES

Foram tomadas algumas decisões importantes ao longo do projeto, sendo de seguida apresentadas as mais importantes:

- Dinamizar ecrãs dinamicamente:

O desenvolvimento de aplicações móveis implica que se tenha atenção a um ponto muito importante. Com a enormidade de modelos disponíveis, as aplicações têm de estar adaptadas, de modo ao seu *design* ser responsivo para cada tamanho e/ou resolução de ecrã. Inicialmente, foi apenas testada a aplicação no telemóvel emprestado para o desenvolvimento do projeto, sendo que o mesmo apresenta um ecrã *Amoled Full High Definition (FHD)* de 5.9 polegadas. O design das páginas estava inicialmente a ser testado e configurado para esse mesmo dispositivo, através da integração de métodos como o *Padding*, *Offset*, entre outros, com valores que encaixavam perfeitamente nesse mesmo dispositivo. No entanto, após o teste da aplicação noutro dispositivo, um *iPhone 11* (que apresenta um ecrã de maiores dimensões com 6.1 polegadas e uma resolução inferior de apenas *High Definition (HD)*) foi visível que seria necessário utilizar outro método. Após a primeira interação com o *iPhone 11*, foi recorrido a vários emuladores no *XCode*, o que permitiu visualizar a presença de muitas *views* desformatadas. O mesmo impossibilitava, por vezes, o fluxo da interação com a aplicação, pois existia o desaparecimento de botões em ecrãs mais pequenos.

Deste modo, foi utilizado o *Geometry Reader* para conseguir dinamizar, de forma flexiva, o tamanho do *layout* [47]. Um exemplo concreto, consiste na definição da *VStack* principal da *view* com um *frame* de (*width: geometry.size.width, height: geometry.size.height*). Isto significa que a *VStack* irá ocupar todo o tamanho do ecrã em altura e largura, sendo depois aplicadas restrições aos componentes, como um *Text* que apenas poderá ocupar 80% da

largura do ecrã e 20% da altura do ecrã: (*width: geometry.size.width * 0.8*, *height: geometry.size.height * 0.2*) (ver Figura 24).

```
Text(recommendationString)
  .font(.manrope(.bold, size: Screen4Class.shared.isScreen4 ? 18 : Screen4Class.shared.isScreen47 ? 20 :
    25))
  .foregroundColor(Color.white)
  .multilineTextAlignment(.center)
  .frame(width: geometry.size.width * 0.8, height: geometry.size.height * 0.5)
```

(a) Utilização do *GeometryReader*

Figura 24: O *GeometryReader*

Foi ainda implementada uma classe que permite identificar quando se trata de um telemóvel de ecrã de 4 polegadas ou 4.7 polegadas (ver Figuras 25 e 26). Estas duas medidas são as que necessitam de um cuidado especial ao lidar com a dinamização do *design* das *views*, sendo para isso detetado assim que a aplicação é inicializada, se se trata de alguma destas medidas. Na Figura 24 é possível verificar que o texto terá diferentes tamanhos de fonte, consoante o tamanho do ecrã.

```
extension UIDevice{
  var iPhone: Bool {
    return UIDevice().userInterfaceIdiom == .phone
  }
  enum ScreenType: String {
    case iPhone4
    case iPhone5
    case iPhone6
    case iPhone6Plus
    case Unknown
  }
  var screenType: ScreenType? {
    guard iPhone else { return nil }
    switch UIScreen.main.nativeBounds.height {
    case 960:
      return .iPhone4
    case 1136:
      return .iPhone5
    case 1334:
      return .iPhone6
    case 2208:
      return .iPhone6Plus
    default:
      return nil
    }
  }
}
```

(a) Tamanhos de ecrã

Figura 25: Verificar o tamanho de ecrã

```

class Screen4Class {

    static let shared = Screen4Class()

    var isScreen4 : Bool = false
    var isScreen47 : Bool = false
    //Does not need to be a state Variable

    func checkScreen4(){
        if (UIDevice().screenType == .iPhone5) {
            isScreen4 = true
        }
    }
    func checkScreen47(){
        if (UIDevice().screenType == .iPhone6) {
            isScreen47 = true
        }
    }
}

```

(a) Funções para verificar os tamanhos de ecrã

```

init() {
    Screen4Class.shared.checkScreen4()
    Screen4Class.shared.checkScreen47()
}

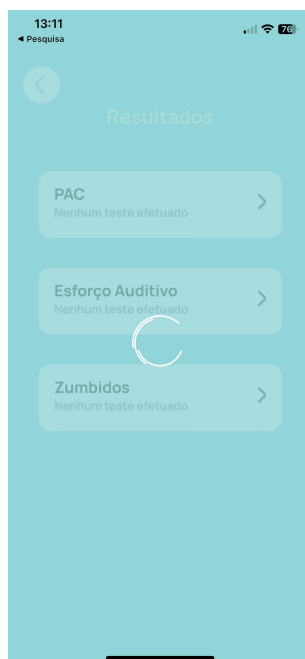
```

(b) Chamar funções na inicialização da `App`

Figura 26: Continuação de verificar o tamanho de ecrã

- *Feedback* ao utilizador: Perante a execução de pedidos ao servidor, foi necessário implementar *feedback* visual para conseguir demonstrar ao utilizador que a aplicação se encontra a realizar uma ação. Foi criada uma *ActivityIndicator-View* (Figura 27) que é mostrada enquanto é realizado um pedido ao servidor que irá fornecer dados importantes à aplicação.

Num caso concreto, quando é realizado um pedido ao servidor para obter os testes que foram realizados pelo utilizador, esta *view* é colocada por cima da *view* principal, o que ofusca a mesma e apresenta uma animação de *loading*:

(a) *View Loading Screen*

```
ZStack{
  ActivityIndicatorView(isVisible: $showLoadingIndicator, type: .arcs(count: 3, lineWidth: 2))
    .frame(width: 70.0, height: 70.0)
    .foregroundColor(Color(hex:0x7DD7DA))
}
.frame(width: geometry.size.width, height: geometry.size.height)
```

(b) Utilização do *ActivityIndicatorView*Figura 27: *Loading Screen*

Foram também adicionadas *toasts* (ver Figura 28) para informar o utilizador de determinadas ações como: *login* bem-sucedido, sessão expirada, erro ao enviar testes, entre outras.

Sessão Expirada

(a) *Toast* de sessão expirada

```
.toast(message: showError ? "Não foi possível enviar os dados, contacte o suporte." : showSuccess ? "Dados enviados com sucesso" : showNetNotOk ? "Verifique a sua conexão de internet." : "",
  isShowing: $showToast,
  duration: Toast.short)
```

(b) Utilização de *Toast*Figura 28: *Toast*

- Guardar dados de testes localmente:

A implementação do envio de dados após o utilizador terminar o teste levou à necessidade de acautelar problemas na comunicação com o servidor (por falta de ligação à *internet*, tempos de *downtime*, alterações em métodos resultantes de atualizações, entre outras). No sentido de prevenir a perda de resultados de testes, foram implementados mecanismos para a aplicação guardar de forma segura os dados e de permitir o *retry* do envio dos mesmos por parte do utilizador. Surgiu assim a possibilidade de guardar esses dados no armazenamento local do dispositivo, sendo para isso utilizado o *CoreData* providenciado pela *Apple*. Resumindo, trata-se de abstrair os detalhes do mapeamento dos objetos para uma *store*, o que facilita o método de guardar dados sem administrar uma base de dados [48]. Por *default*, o *Core Data* não encripta os dados persistidos, no entanto, como não se tratam de dados de cariz confidencial (como por exemplo *passwords*), não é considerado um problema (a única maneira de associar um teste a um utilizador é através do *id* do utilizador).

Na Figura 29 é possível verificar a estrutura dos dados que serão guardados localmente:

- *Identification (ID)*: Um **ID** gerado automaticamente para cada objeto guardado;
- *userId*: Guardar o **ID** do utilizador;
- *subtestId*: Guardar o **ID** do subteste;
- *stimulusType*: Guardar a opção de estímulo do teste. Valor opcional;
- *outputType*: Guardar a opção de *output* do teste;
- *difficultyLevel*: Guardar a opção de dificuldade do teste. Valor opcional;
- *sequencesCorrect*: Guardar o número de respostas corretas. Valor opcional;
- *answersGiven*: Guardar as respostas do questionário. Valor opcional;
- *frequenciesIntensities*: Guardar os dados do ruído personalizado no teste Zumbidos. Valor opcional;
- *timestamp*: Guardar a data de criação do objeto;
- *dateLastRetry*: Guardar a data da última tentativa de envio do objeto para o servidor;

- *dateStoredData*: Guardar a data do envio bem-sucedido do objeto para o servidor.

Attribute	Type
answersGiven	Transformable
dateLastRetry	Date
dateStoredData	Date
difficultyLevel	Integer 32
frequenciesIntens...	Transformable
id	UUID
outputType	String
sequencesCorrect	Integer 32
stimulusType	String
subtestId	Integer 32
timestamp	String
userId	Integer 32

(a) Estrutura do objeto a ser guardado

```
DataController().addTestData(subtestId: Int32(7), outputType: "headphones", answersGiven:
dataSequenceZinnitus.choosedValuesObject, frequenciesIntensitiesGiven:
dataSequenceZinnitus.frequencyintensity, context: managedObjContext)
```

(b) Exemplo de guardar um objeto no *CoreData*Figura 29: Utilização de *CoreData*

- Guardar dados da autenticação localmente:

Após uma realização de *login* bem sucedida, será guardado localmente no dispositivo, o *token* desse utilizador. Sempre que existe um método que necessita da autenticação do utilizador é efetuada uma análise à validade do *token* (encontra-se atualmente definido para 24 horas), ou seja, se existe e se foi criado há menos de 24 horas. Se o *token* não se encontrar válido, o utilizador será reencaminhado para a página de *login* e notificado com uma *toast* que foi efetuado o *logout*.

De igual forma, assim que o *token* é guardado, o *email* e *password* do utilizador também o serão. Esta funcionalidade permite que durante as 24 horas, desde que o *token* foi guardado, seja possível iniciar a aplicação sem ser necessário voltar a realizar o *login* (ver Figuras 30 e 31).

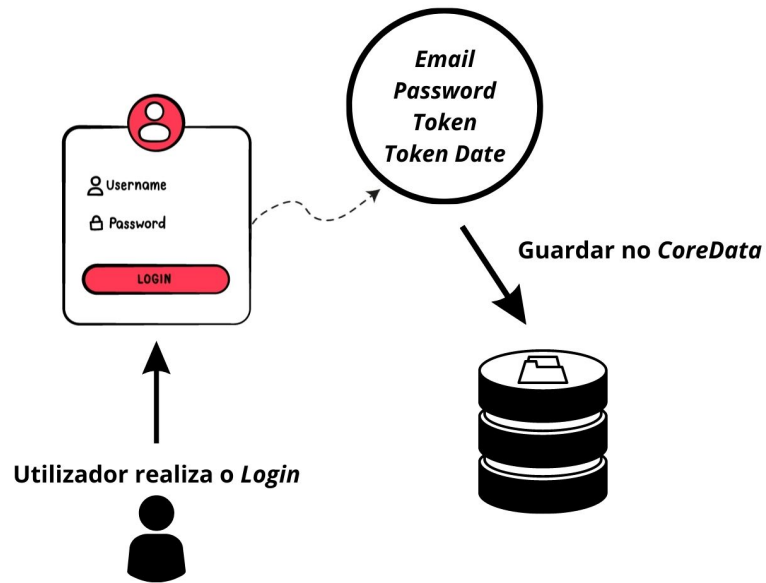
(a) Guardar o *Token*

Figura 30: Fluxograma para guardar os dados da autenticação localmente

(a) Verificar a validade do *Token*Figura 31: Fluxograma de verificação do *Token*

Após a conclusão do estágio, foi dado um maior destaque a este tema e foi possível verificar que o *UserDefaults* não deve ser usado para guardar

dados sensíveis. O *UserDefaults* não utiliza encriptação, sendo recomendável o uso do *Keychains* para o *store* de dados sensíveis [49]. De igual forma, para realizar alguns pedidos ao servidor, é necessário o *token* de servidor, sendo este guardado estaticamente na aplicação (para maior segurança deveria ser guardado com auxílio ao *Keychain*).

- Notificações:

Foi implementado o uso de notificações locais, através do *Local Notifications* (ver Figura 32). A escolha do uso das *Local Notifications* residiu na necessidade de despoletar notificações simples ao utilizador no próprio *smartphone*, sem existir alguma conexão ao servidor. Caso essas notificações fossem despoletadas num servidor, teriam de ser adotadas as *Push Notifications* (são enviadas pelo *Apple Push Notification Service (APNS)* sendo necessário implementar um serviço que comunique com o serviço) [50].



(a) Permissão para receber notificações

(b) Notificação de erro ao enviar o teste

(c) Notificação de sucesso ao enviar o teste

Figura 32: Exemplo de notificações implementadas

Sempre que é guardado um objeto no *CoreData* é despoletada uma notificação ao utilizador (ver Figura 33), afirmando que existem dados disponíveis para serem enviados ao servidor. Além destas notificações é ainda despoletado um método que irá ativar um *schedule* (ver Figura 34) de 2 dias e de 7 dias. Significa isto, que se decorrido 2 dias ainda existirem dados por enviar, o utilizador irá receber uma notificação (mesmo que a aplicação se encontre encerrada) a avisar que existem dados por enviar. O mesmo acontece se ao fim de 7 dias ainda existirem dados a ser enviados. Após os 7 dias não será

despoletada mais nenhuma notificação. De salientar, que apenas é despoletado o *schedule* destas notificações caso ainda não tenha sido despoletado essa ação. Caso já exista um *schedule* pendente, o *store* de um objeto no *CoreData* não irá ativar novamente o *schedule* e o período de *standby* de despoletar as notificações mantém-se inalterado (ver Figura 35).

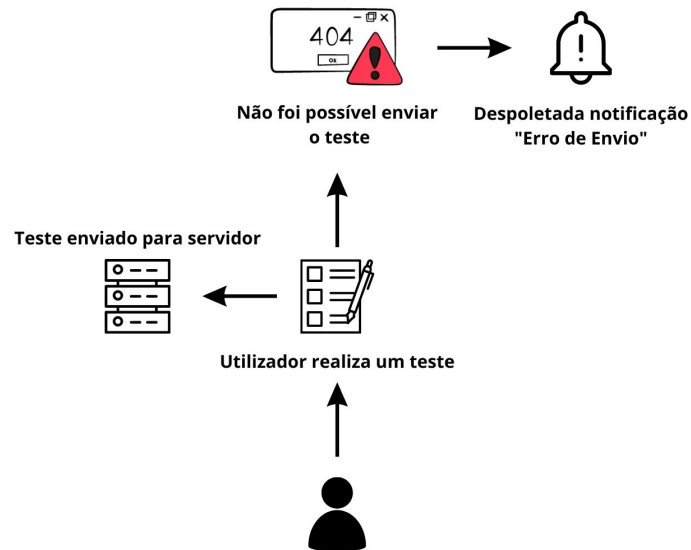


Figura 33: Fluxograma para despoletar uma notificação de erro no envio de um teste

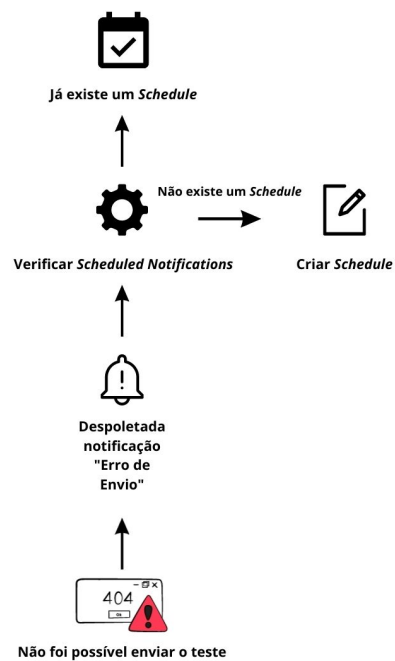


Figura 34: Fluxograma do *schedule* de notificações

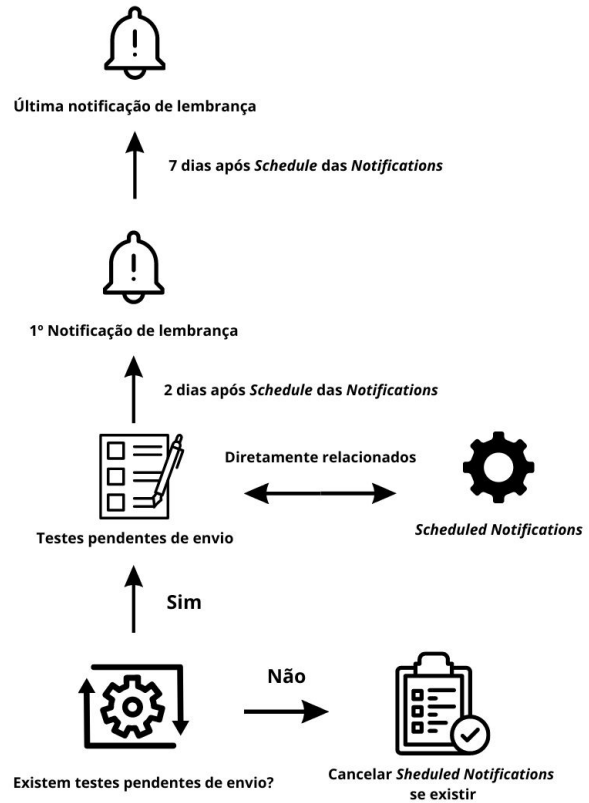
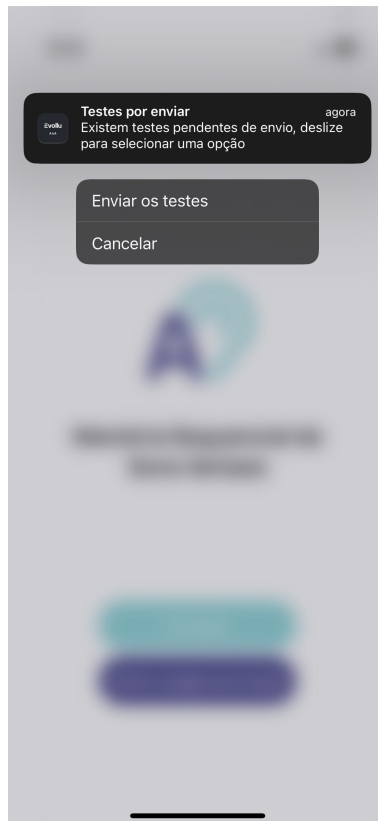


Figura 35: Fluxograma do *scheduled notification dates*

- Envio dos dados nas notificações:

As notificações que se encontram *scheduled* apresentam a possibilidade ao utilizador de este enviar os dados para o servidor sem necessitar de abrir a aplicação. Apresentam o nome de *Action Notifications* e se o utilizador clicar no botão “Enviar os testes“, irá ser ativado o método para o envio dos dados para o servidor (ver Figura 36). Se o envio for bem sucedido, será despoletada uma notificação a referir o sucesso da ação, caso não seja possível enviar, será despoletada uma notificação que informa essa impossibilidade.



(a) Notificação com ação

```
// This method will be called when the user clicks on an action in a local notification
func userNotificationCenter(_ center: UNUserNotificationCenter, didReceive response: UNNotificationResponse,
withCompletionHandler completionHandler: @escaping () -> Void) {
    if response.notification.request.content.categoryIdentifier == "myCategory" {
        switch response.actionIdentifier {
            case "confirm":
                // Handle the action
                if let selectApp = AppManager.shared.selectApp {
                    // Update your variable with the number of tests here
                    selectApp.handleLocalData(selectApp.localData) { nr in
                }
            }
            case "cancel":
                print("User Canceled Notification Action")

            default:
                break
        }
    }
    completionHandler()
}
```

(b) Método para lidar com a ação da notificação

Figura 36: Notificações com ações integradas

- Envio automático dos testes guardados localmente:

As notificações previamente descritas dão a possibilidade ao utilizador de enviar os dados que se encontram atualmente pendentes para envio, sem que este necessite de se encontrar na aplicação. No entanto, sempre que é realizada a abertura da aplicação, será realizado um pedido automaticamente

para tentar enviar todos os testes que se encontram pendentes de envio. Para tal, é realizado um pedido *FetchRequest* para obter todos os objetos que se encontram guardados no *CoreData* e um *ForEach* para percorrer cada um deles e proceder ao seu envio singular. De referir, que apenas será realizada esta operação automaticamente, sempre que for inicializada a aplicação. Se existir um insucesso no envio dos dados, o utilizador terá de aceder às outras possibilidades de envio de dados.

- Uso de *Threads*:

No desenvolvimento do teste Tonal era necessário ter em conta o *play* dos sons e o *feedback* do clique do utilizador no botão apropriado. Para conseguir lidar com estas duas situações, foi necessário recorrer ao uso de uma *thread* unicamente para lidar com toda a lógica de toque dos sons (ver Figura 37). Assim, toda a lógica encontra-se separada da *main thread* e previne a ocorrência de problemas de concorrência, pois quando era apenas utilizado a *main thread* para lidar com ambos os casos, poderia encadear um problema de concorrência que interrompia a lógica do toque dos sons quando o utilizador clicava no botão.

```
do {
    try engine.start()
    osc.start()

    audioThreadQueue.asyncAfter(deadline: .now() + duration) { [] in
        osc.stop()
        //Call completionHandler because audio stopped playing
        //Then it will continue with the logic inside PlayStopToneOscillator
        completionHandler?()
    }
} catch {
    print(error)
}
```

(a) Na geração de *Tones*

```
timer = Timer.scheduledTimer(withTimeInterval: 4.0, repeats: true) { [self] _ in

    if !isPlaying && self.index < frequencies.count {

        audioThreadQueue.async(flags: .barrier) {
            self.isPlaying = true
            self.canDecrement = true

            self.generateToneOscillator.initialisePlayTone(frequencyToPlay: self.index, duration: 1.0){
                self.audioThreadQueue.async {
                    self.isPlaying = false
                }
            }
        }
    }
}
```

(b) No toque de *Tones*

Figura 37: Uso de *thread* na lógica do teste Tonal

- *LazyLoading*:

A obtenção de todos os testes realizados pelo utilizador pode tornar-se uma lista muito extensa, tendo sido implementado no servidor uma disponibilização desses mesmos testes através de paginação. Isto significa que terão de ser realizados diversos pedidos para conseguir obter todos os testes realizados. Foi implementado o *lazy loading* para conseguir realizar um novo pedido sobre a próxima página, sempre que o utilizador chegue perto do final da lista de *scroll* (ver Figura 38). De modo a conseguir perceber quando o utilizador se encontrava perto do fim da lista, pois serão devolvidos no máximo 10 testes por pedido, foi utilizada uma *ObservableScrollView* para saber o *offset* atual. Foram efetuados diversos testes em diversos dispositivos até que foram estabelecidos valores para cada segmento de tamanho de ecrã. Num exemplo concreto, se o fim do *scroll* é 800 píxeis, será realizado um pedido ao servidor quando o utilizador se encontrar nos 700 píxeis, depois nos 1400, nos 2100, entre outros, pois foi implementada uma verificação de múltiplos do número inicial.

Um exemplo concreto em que existem 11 testes realizados, em que é inicialmente realizado o pedido inicial e são devolvidos 10 testes, quando o utilizador chegar perto do fim da lista vai ser realizado outro pedido, mas desta vez com o *index* do número de página incrementado e só irá devolver um teste. Já não será realizado mais nenhum pedido, pois o utilizador não conseguirá atingir o *offset* múltiplo do valor inicialmente definido.

```

.onAppear{
    //Verify what is the screen size, in order to see what value is needed to call lazyLoading
    scrollScreenSize = Screen4Class.shared.isScreen4 ? 1300 : Screen4Class.shared.isScreen47 ? 1600 : 1800
}

```

(a) Valores de *offset* iniciais por tamanho ecrã

```

.onChange(of: scrollOffset) { newScrollOffset in
    // Calculate the nearest multiple that's less than or equal to the new scroll offset
    let nearestMultiple = floor(newScrollOffset / scrollScreenSize) * scrollScreenSize

    // Check if the nearest multiple has changed since the last time this closure was
    // called
    if nearestMultiple > lastMultiple {

        numberPage += 1

        showLoadingIndicator = true

        //Verify if user is logged in (token valid):
        validateAndRetrieveUser() { result in
            switch result {
            case .success(_):
                getTestResults(token: UserDefaults.standard.string(forKey:
                    "authToken")!, testId: idTest, page: numberPage) { result in
                    switch result {
                    case .success(let resultTests):
                        resultEachTest = resultEachTest + resultTests
                        showLoadingIndicator = false
                    case .failure(let error):
                        print("Error getting Tests Results")
                        print(error)
                        showLoadingIndicator = false
                    }
                }
            case .failure(let error):
                print(error)
                makeNavigationWelcomeScreen = true
                showLoadingIndicator = false
            }
        }

        // Update the last multiple that triggered the function call
        lastMultiple = nearestMultiple
    }
}
}

```

(b) Realização do pedido para obtenção de resultados

Figura 38: *LazyLoading* para obter os resultados dos testes

Na Figura 39 é apresentado um fluxograma relativo ao mecanismo de *lazy loading* implementado.

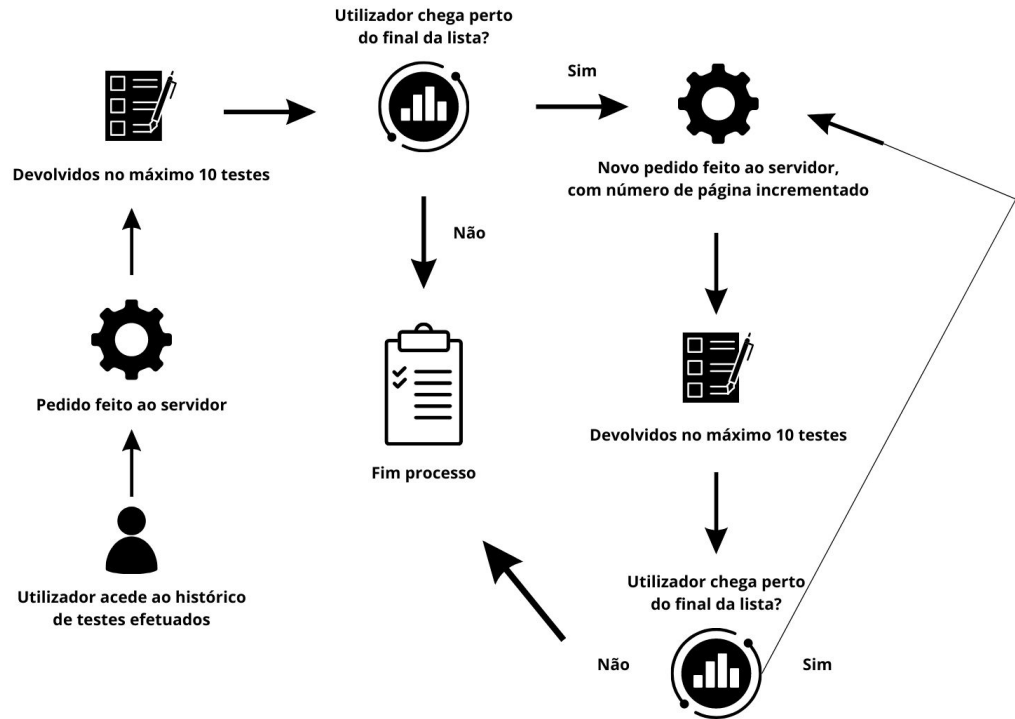
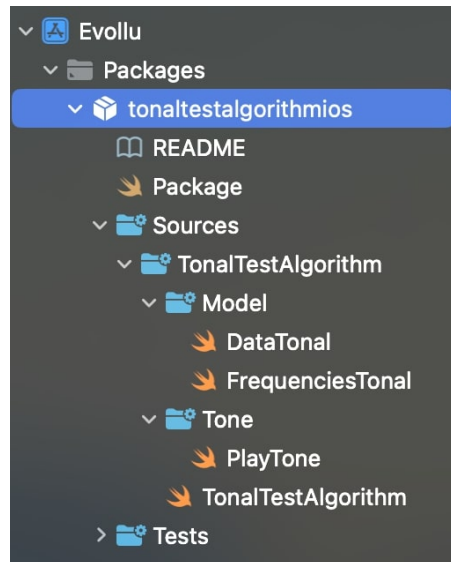


Figura 39: Fluxograma do *lazy loading*

- Desenvolvimento de um *package*:

Ao contrário dos testes [PAC](#), Esforço Auditivo e Zumbidos, o teste Tonal foi desenvolvido tendo em conta a sua posterior utilização noutra aplicação. Para tal, recorreu-se ao *Swift Package Manager*, no qual foi desenvolvido um *package* que permite ser integrado facilmente noutro projeto (ver [Figura 40](#)). Deste modo, toda a lógica do teste encontra-se disponível para utilização noutro projeto [iOS](#). Esta decisão deveria ter sido tomada para o desenvolvimento de todos os testes, mas devido a prazos no desenvolvimento e à possibilidade do uso do teste Tonal noutras plataformas e/ou aplicações, foi privilegiado o desenvolvimento deste.

(a) Estrutura do *package* criado

```

// swift-tools-version: 5.7
// The swift-tools-version declares the minimum version of Swift required to build this package.

import PackageDescription

let package = Package(
    name: "TonalTestAlgorithm",
    platforms: [
        .iOS(.v15) //minimum iOS version here
    ],
    products: [
        // Products define the executables and libraries a package produces, and make them visible to other packages.
        .library(
            name: "TonalTestAlgorithm",
            targets: ["TonalTestAlgorithm"]),
    ],
    dependencies: [
        // Dependencies declare other packages that this package depends on.
        // .package(url: /* package url */, from: "1.0.0"),
        .package(url: "https://github.com/AudioKit/AudioKit", from: "5.2.0"),
        .package(url: "https://github.com/AudioKit/SoundpipeAudioKit", from: "5.2.0"),
    ],
    targets: [
        // Targets are the basic building blocks of a package. A target can define a module or a test suite.
        // Targets can depend on other targets in this package, and on products in packages this package depends on.
        .target(
            name: "TonalTestAlgorithm",
            dependencies: ["SoundpipeAudioKit", "AudioKit"]),
        .testTarget(
            name: "TonalTestAlgorithmTests",
            dependencies: ["TonalTestAlgorithm"]),
    ]
)

```

(b) Ficheiro *Package.swift*Figura 40: *Package* do teste Tonal

4.10 DIFICULDADES SENTIDAS

Ao longo do projeto foram identificados alguns contratempos e problemas de maior intensidade, que necessitaram de um maior estudo e tempo de desenvolvimento para

serem superados. Em seguida são indicados os principais problemas identificados e posteriormente resolvidos:

- Variáveis entre *Views*:

A primeira grande questão residia na forma com que seriam interligados valores de uma determinada *view* para outra *view*. Ou seja, num caso concreto, se o utilizador escolher determinado teste e determinada categoria daquele teste (por exemplo escolher como *output* os *headsets*) essa configuração deve ser partilhada pelos ecrãs seguintes. Desta forma, seria possível utilizar esses mesmos dados sempre que fosse preciso. Após alguns *brainstorms* com a equipa, chegou-se à conclusão que a melhor solução passaria pelo uso de classes para a definição de estruturas de dados. Estas estruturas conteriam todos os atributos necessários para determinado objeto.

Na Figura 41 é possível verificar a estrutura implementada para um Teste, o qual deverá conter os seguintes atributos:

```
struct Test: Codable, Identifiable{

    var id: Int
    var title: String
    var description: String
    var image: String
    var selectedDevice: Device? = nil
    var selectedDifficulty: Difficulty? = nil
    var selectedEar: Ear? = nil
    var audiologies: Audiology? = nil
    var words: Words? = nil
}
```

(a) Estrutura do Teste

```
let difficulties = [
    Difficulty(id: 1, value: 3, title: ["Instrumentos", "Sílabas", "Sons"], image: "3"),
    Difficulty(id: 2, value: 4, title: ["Instrumentos", "Sílabas", "Sons"], image: "4")
]

struct Difficulty: Codable, Identifiable{

    var id: Int
    var value: Int
    var title: [String]
    var image: String
}
```

(b) Estrutura das Dificuldades

Figura 41: Exemplo de estruturas aplicadas

- **ID**: Cada teste apresenta um **ID** único associado ao mesmo (ver Figura 42). Importante para realizar um *foreach* e percorrer todos os testes num vetor por exemplo, pois num *foreach* é necessário atribuir um **ID** (que

permite identificar cada elemento no vetor como um valor único) e assim atribuímos este ID do teste [51].

```
ForEach(sessionTest.finishedSubtests, id: \.id) { finishedSubtest in
```

Figura 42: Exemplo de *ForEach* de *id: \.id*

- *title*: Cada teste apresenta um título.
 - *description*: Cada teste apresenta uma descrição.
 - *image*: Cada teste apresenta uma imagem.
 - *selectedDevice*: Cada teste poderá apresentar um dispositivo (trata-se de uma estrutura do tipo dispositivo). Como não é um atributo obrigatório, encontra-se definido como “;” e inicialmente declarado como *nil*.
 - *selectedDifficulty*: O mesmo que o atributo anterior, mas neste caso refere-se a uma dificuldade.
 - *selectedEar*: O mesmo que o atributo anterior, mas neste caso refere-se a atributos de ouvido.
 - *audiologies*: O mesmo que o atributo anterior, mas neste caso refere-se a atributos de audiologia.
 - *words*: O mesmo que o atributo anterior, mas neste caso refere-se a palavras.
- Guardar variáveis para aceder noutras *Views*:

Ao longo do desenvolvimento do projeto existiram variáveis que precisavam de ser partilhadas globalmente em toda a aplicação, mas que não pertenciam necessariamente a nenhuma estrutura. Deste modo, foram criados *singletons*, que são apenas criados uma vez e utilizados onde fosse necessário [52]. No exemplo a seguir representado (ver Figura 43), é possível verificar a existência de variáveis que permitem o fluxo de realização do teste Esforço Auditivo, como por exemplo, a *numberQuestion* que guarda o número da questão atual do questionário. De referir, que estas variáveis são do tipo *Published*.

```

class AnswersValueEsforco: ObservableObject {

    static let valuesEsforcoQuestionaire = AnswersValueEsforco()

    @Published var progressBarIndex: Int = 1
    @Published var numberQuestion: Int = 0

    @Published var indexSliderData: Int = 0
    @Published var sequenceResult: [AnswerTestDone] = []
    @Published var indexButtonData: [Int] = []
    @Published var sliderData: [Double] = []

    @Published var dataToStoreEsforco: [AnswerTestDone] = []

    //I will need to make a get to confirm if uer already made this test once
    @Published var alreadyMadeTest: Bool = false

    //These are required for when user finish and exit test, i need to reset values.
    @Published var sequenceResultInitial: [AnswerTestDone] = []
    @Published var indexButtonDataInitial: [Int] = []
    @Published var sliderDataInitial: [Double] = []

}

```

Figura 43: Exemplo de *Singleton* para o teste Esforço Auditivo

- *Listener* para a ligação dos *Headsets*:

Um dos requisitos do projeto regia-se pela possibilidade de realizar alguns testes com o uso de *Headsets*. Para tal, é necessário saber quando é que existem *Headsets* ligados ao dispositivo, de forma a habilitar essa opção ao utilizador. Deste modo, foi necessário criar um *Listener* que se encontra sempre a identificar a presença da ligação de *headsets*, sejam estes do tipo *wireless* ou por cabo.

- Pausar o teste com alteração do estado dos *Headsets*:

Este requisito depende do último referido (*Listener* para a ligação dos *Headsets*), pois depende do *listener*. Caso o utilizador esteja a realizar o teste com o uso de *Headsets* e estes sejam desconectados, o teste terá de ficar automaticamente em pausa. Foi necessário reformular todo o código referente ao *play* dos sons, de modo a ser possível pausar o teste automaticamente com a desconexão dos *Headsets* ou com o clique no botão de “Sair” do teste.

- *Feedback* dos botões dos questionários:

Inicialmente, na implementação do questionário do teste Esforço Auditivo, o *feedback* dos botões do questionário foi realizado de forma *hardcoded*. Devido ao tempo limitado para o desenvolvimento, de modo a entregar uma versão do teste estável, não existiu uma correta pesquisa e análise de soluções. Foi

adotada uma solução que não era viável devido a não ser escalável (não é facilmente alterada caso existam novos requisitos, como por exemplo a necessidade de adicionar um novo botão seria um trabalho muito difícil devido à implementação estática dos mesmos), muito extensa, complexa computacionalmente e de difícil leitura (*no readable*). No entanto, devido à complexidade do teste *Zinnitus* e à insatisfação com a anterior implementação no teste Esforço Auditivo, foi mesmo necessário despende mais tempo para conseguir encontrar uma solução viável. Foi possível desenvolver uma solução coerente, eficaz, mais otimizada e escalável, permitindo corresponder às expectativas atuais e a possíveis alterações que sejam necessárias (ver Figura 44).

```
func updateSelectedButtons(value: Int)
{
  //In order to change the color to White:
  //Need to check if the arrays does have that buttonindex, and if so remove
  if self.buttonList.contains(where: { $0.buttonIndex == value } )
  {
    self.buttonList.remove(at: self.buttonList.firstIndex(where: {$0.buttonIndex == value}))!

    self.borderAndtextList.remove(at: self.borderAndtextList.firstIndex(where: {$0.buttonIndex == value}))!
  }

  //In order to change the color to bluish
  //Need to remove all the elements from the array, in order to change the color in just the button that i
  clicked
  else
  {
    self.buttonList.removeAll()
    self.borderAndtextList.removeAll()

    self.buttonList.append(ColorButton(color: colors[self.colorIndex].value, buttonIndex: value))

    //ColorIndex + 1 since i want the second color, that is the white color
    self.borderAndtextList.append(ColorButton(color: colors[self.colorIndex + 1 ].value, buttonIndex: value))

    self.colorIndex += 1
    optionSelected = value

    //Reset colorIndex
    if self.colorIndex >= 1
    {
      self.colorIndex = 0
    }
  }
}
```

Figura 44: Um dos métodos implementados para a atualização dos botões

- Alterar a frequência e intensidade em *Real Time*:

Um dos grandes requisitos do teste *Zinnitus* é a possibilidade do utilizador conseguir definir em tempo real qual a frequência e a intensidade do ruído que mais se assemelha ao ruído que costuma ouvir. Foi necessário recorrer a diversas soluções e testes para conseguir encontrar uma solução que fosse de acordo com o requisito pretendido. Um dos grandes contratempos foi a necessidade de criar uma paleta onde fosse possível modificar o valor de x e y , de modo a conseguir alterar a frequência e intensidade ao mesmo tempo. A paleta teria de ser adaptável aos diferentes formatos e tamanhos de ecrã de cada dispositivo. Através dos valores de x e y obtidos, era então calculado

a intensidade e frequência, tendo em conta o uso do *Mixer* e *BitCrusher*, respetivamente. De referir, que os valores matemáticos apresentados, são necessários para conseguir obter resultados num intervalo de valor pretendido.

- Reestruturar questionários:

Inicialmente, todos os dados dos questionários eram definidos estaticamente na aplicação, sendo posteriormente necessário substituir pelos dados presentes no servidor. Foi assim necessário reestruturar toda a estrutura de dados existente para os questionários, de modo a conseguir dinamicamente receber e utilizar os dados provenientes do servidor. Na Figura 45 é possível verificar que numa versão anterior existiam muitos dados definidos estaticamente, como as questões e respostas do questionário, o filtro perante o tipo de questão, a necessidade de inicializar as respostas disponíveis para cada pergunta estaticamente, entre outros aspetos. Na Figura 46 verifica-se uma completa reestruturação dos dados, sendo que agora as questões, respostas, filtragem pelo tipo de questão, inicialização, entre outros, é tudo realizado de forma dinâmica consoante os dados obtidos pelo servidor.

```

var questionsTinnitus = [
    QuestionTinnitus(id: 1, description: "Qual a localização do seu zumbido?", answers: AnswersTinnitus.Test6),
    QuestionTinnitus(id: 2, description: "Há quanto tempo sente o seu zumbido?", answers: AnswersTinnitus.Test6),
    QuestionTinnitus(id: 3, description: "Acha que o zumbido interfere com a sua qualidade de vida?",
        answers: AnswersTinnitus.Test6),
    QuestionTinnitus(id: 4, description: "Quando tem maior percepção do seu zumbido? ", answers: AnswersTinnitus.Test6),
    QuestionTinnitus(id: 5, description: "Como descreve o seu zumbido quanto ao tipo de som?", answers:
        AnswersTinnitus.Test6),
    QuestionTinnitus(id: 6, description: "Como descreve o seu zumbido quanto à tonalidade?", answers:
        AnswersTinnitus.Test6),
    QuestionTinnitus(id: 7, description: "Qual a intensidade do seu zumbido?", answers: AnswersTinnitus.Test6),
    QuestionTinnitus(id: 8, description: "Qual dos sons abaixo mais se assemelha com o seu zumbido?", answers:
        AnswersTinnitus.Test6),
    QuestionTinnitus(id: 9, description: "Qual dos sons abaixo mais se assemelha com o seu zumbido?", answers:
        AnswersTinnitus.Test6),
    QuestionTinnitus(id: 10, description: "Como classifica a intolerância ao seu pior zumbido?", answers:
        AnswersTinnitus.Test6),
    QuestionTinnitus(id: 11, description: "Quais as estratégias desenvolvidas para lidar com o seu zumbido?", answers:
        AnswersTinnitus.Test6),

    QuestionTinnitus(id: 12, description: "Tem alguma patologia diagnosticada?", answers: AnswersTinnitus.Test6),
    QuestionTinnitus(id: 13, description: "Tem algum comportamento de risco?", answers: AnswersTinnitus.Test6),
    QuestionTinnitus(id: 14, description: "Ficaria mais tranquilo se falasse com um profissional de saúde especializado
        em zumbidos?", answers: AnswersTinnitus.Test6),
    QuestionTinnitus(id: 15, description: "Vamos agora caracterizar o seu zumbido. Indique por favor o ouvido onde o
        zumbido é mais intenso.", answers: AnswersTinnitus.Test6),
    QuestionTinnitus(id: 16, description: "Ajuste Grosso", answers: AnswersTinnitus.Test6),
    QuestionTinnitus(id: 17, description: "Ajuste Fino", answers: AnswersTinnitus.Test6),
]

```

(a) Questões definidas estaticamente

```

enum AnswersTinnitus: String, CaseIterable, Codable {

    case Test6

    var answer: [[String]]{
        switch self {
            case .Test6: return [
                ["Ouvido esquerdo", "Ouvido direito", "Em ambos os ouvidos", "Central"],
                ["Menos de 1 mês", "Menos de 6 meses", "1 ano ou menos", "Mais de 1 ano"],
                ["Sim", "Não", "Não sei"],
                ["Sempre", "Durante o dia", "Durante a noite", "No silêncio"],
                ["Contínuo", "Pulsátil"],
                ["Grave", "Agudo"],
                ["ND"],
                ["Ruído 1", "Ruído 2", "Ruído 3", "Ruído 4", "Outro"],
                ["Ruído 1", "Ruído 2", "Ruído 3", "Ruído 4", "Outro"],
                ["ND"],
                ["Ouvir música / TV", "Falar ao telefone", "Praticar esporte", "Gerador de ruído", "Outra", "Nenhuma estratégia"],
                ["Diabetes", "Hipertensão", "Cancro", "Depressão", "Stress", "Não"],
                ["Tabaco", "Álcool", "Caféina", "Medicação", "Exposição ao ruído", "Música em alta intensidade", "Não", "Outro"],
                ["Sim", "Não"],
                ["Esquerdo", "Direito"]
            ]
        }
    }
}

```

(b) Respostas definidas estaticamente

```

class ValuesButtonsTinnitus: ObservableObject {

    static let valuesButtonsT = ValuesButtonsTinnitus()

    var indexChosedButton: [[Int]] = [
        [-1], [-1], [-1], [-1, -1, -1, -1], [-1, -1], [-1, -1], [0], [-1], [-1], [0],
        [-1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1], [-1, -1, -1, -1, -1, -1, -1, -1], [-1, [-1], [0], [0]]
    ]

    var choosedValues: [[String]] = [
        ["ND"], ["ND"], ["ND"], ["ND", "ND", "ND", "ND"], ["ND", "ND"], ["ND", "ND"],
        ["slider1"], ["ND"], ["ND"], ["slider2"], ["ND", "ND", "ND", "ND", "ND", "ND"], ["ND", "ND", "ND", "ND", "ND",
        "ND"], ["ND", "ND", "ND", "ND", "ND", "ND", "ND"], ["ND"], ["ND"], ["rough"], ["thinSlider"]]
    ]

    var numberQuestion: Int = 0

    var valuesSlider: [[Double]] = [
        [-1.0, -1.0], [-1.0], [-1.0, -1.0]]
    ]

    var roughFit: [Double] = [-1.0, -1.0]

    var thumbLocation: CGPoint = CGPoint(x: 0.0, y: 0.0)
}

```

(c) Inicialização estática de respostas disponíveis para cada pergunta

Figura 45: Exemplos de conteúdo estático no teste Zumbidos

```

struct QuestionsViewTinnitus: View {

    var normalQuestions = [0, 1, 2, 13, 14]
    var isSlider = [6, 9]
    var fitSound = [15, 16]
    var multipleAnswersQuestions = [3, 10, 11, 12]
    var answers2Questions = [4, 5]
    var choosePlaySounds = [7,8]
}

```

(a) Filtro do tipo de questão definido estaticamente

```

class QuestionaryTinnitus: ObservableObject {
    static let questionnaireTinnitus = QuestionaryTinnitus()

    //Questions Questionary Tinnitus:
    @Published var questions: [String] = []

    //Normal Questions
    @Published var normalQuestion: [Int] = []

    //To store all sliders
    @Published var isSlider: [Int] = []

    //Horizontal slider:
    @Published var isSliderHorizontal: [Int] = []

    //2 Answers - Slider Vertical
    @Published var isSliderVertical: [Int] = []

    //Multiple answers
    @Published var multipleAnswers: [Int] = []

    //2 Answer - 2 Fors:
    @Published var answers2: [Int] = []

    //Sound Select:
    @Published var soundSelect: [Int] = []

    //Fit Sound:
    @Published var fitSound: [Int] = []

    //Vector to Store Description Slider:
    @Published var descriptionSlider: [[String]] = []

    //To store all Answers
    @Published var answersQuesTinnitus: [[AnswerOptionGet]] = []
}

```

(b) Dados dinâmicos

```

class ValuesButtonsTinnitus: ObservableObject {
    static let valuesButtonsT = ValuesButtonsTinnitus()

    var numberQuestion: Int = 0

    var roughFit: [Double] = [-1.0, -1.0]

    var thinFit: [Double] = [-1.0, -1.0]

    var thumbLocation: CGPoint = CGPoint(x: 0.0, y: 0.0)

    var postDone: Bool = false

    //Dynamic Data:

    //The others are not needed, since they not come in the request
    @Published var indexChoosedButton: [[Int]] = []

    @Published var choosedValuesObject: [AnswerTestDone] = []

    @Published var frequencyintensity: [FrequencyIntensityAnswer] = []

    @Published var valuesSlider: [[Double]] = []
}

```

(c) Inicialização dinâmica de variáveis

Figura 46: Exemplos de conteúdo dinâmico no teste Zumbidos

PRODUTO FINAL

Neste capítulo são descritas algumas das alterações efetuadas ao longo do desenvolvimento da aplicação (Secção 5.1), análise das repostas provenientes do questionário de usabilidade (Secção 5.2) e o estado final da aplicação (Secção 5.3), retratando todo o fluxo da aplicação.

5.1 ALTERAÇÕES E TESTES

Nesta secção são descritas algumas alterações que foram efetuadas à aplicação, assim como no decorrer geral do desenvolvimento da mesma. Encontra-se dividido em alterações relativas ao *design* (Secção 5.1.1) e às funcionalidades (Secção 5.1.2). De referir, que as fotos representativas do estado atual da aplicação encontram-se a partir da secção 5.3 (Produto Final). Ainda se encontra presente o questionário que foi respondido por alguns utilizadores que utilizaram a aplicação (Secção 5.2).

5.1.1 *Design*

Nesta secção, são demonstradas algumas das alterações que foram efetuadas em termos de *design* da aplicação ao longo do desenvolvimento do projeto.

5.1.1.1 *Ícone da aplicação*

O ícone da aplicação sofreu alterações (ver Figura 47), sendo que inicialmente era apenas apresentado o nome de **Rastreio processamento auditivo central (RPAC)**, sendo o nome inicial do teste **PAC**.

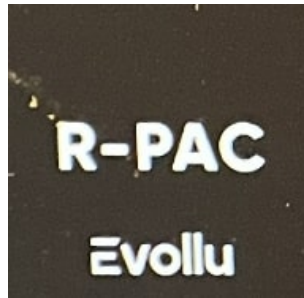


Figura 47: Versão desatualizada do ícone da aplicação

5.1.1.2 *SplashScreen*

A *SplashScreen* inicial apresentava apenas um logótipo (ver Figura 48), que era naquele momento utilizado para retratar a aplicação, tendo o nome de *Evollu Hear*. É possível visualizar a versão final na figura 72 (*Loading screen*).

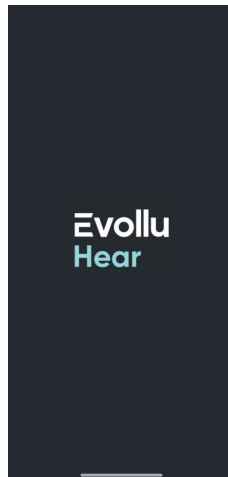


Figura 48: Versão desatualizada da *SplashScreen*

5.1.1.3 *Página para login*

Graças à impossibilidade de acabar o desenvolvimento do *login* através de outras plataformas, foram removidos os botões que tinham por objetivo dar acesso ao *login* para outras plataformas (ver Figura 49). É possível visualizar a versão final na figura 73 (*Página inicial de login*).



Figura 49: Versão desatualizada da página de *Login*

5.1.1.4 *Página inicial da escolha de testes*

Numa primeira fase, a página inicial (ver Figura 50) pretendia apenas possibilitar que os utilizadores conseguissem escolher um dos testes disponíveis. Tratava-se de uma página muito simples com a presença de 3 retângulos que identificavam um teste cada. Posteriormente, este ecrã foi substituído pelo atual (Secção 5.3.3), sendo este muito mais detalhado e de agradável utilização. É possível visualizar a versão final na figura 77 (*Página de seleção de testes*).



Figura 50: Versão desatualizada da página inicial da escolha de testes

5.1.1.5 *Página inicial do PAC*

Inicialmente, a página inicial do teste **PAC** (ver Figura 51) era consideravelmente mais simples, sendo que não contava com a informação relativa ao desbloqueio de recomendação e o *feedback* perante a realização de um teste. Existiu também uma reformulação relativamente ao tamanho e sombra dos *cards* de cada subteste, assim como o tamanho de letra. Como era previsto a realização de uma aplicação para cada teste, também não existia o botão “X” para sair do teste. É possível visualizar a versão final na figura 95 (*Página inicial do PAC*).



Figura 51: Versão desatualizada da página inicial do teste **PAC**

5.1.1.6 *Verificação de resultados do subteste do PAC*

Foi desenvolvido um ecrã (ver Figura 52) que pretendia mostrar a sequência que tinha sido tocada (sequência correta) e a sequência escolhida pelo utilizador. Após uma reunião com a **ESTeSC-IPC**, foi estipulado que este ecrã não era necessário e acabou por ser removido. É possível visualizar a versão final na figura 104 (*Resultados do subteste*).



Figura 52: Versão desatualizada dos resultados dos subtestes do PAC

5.1.1.7 Verificação de resultados do subteste Esforço Auditivo

No fim do subteste Esforço Auditivo, eram mostradas ao utilizador as sequências que tinha acertado em conjugação com as palavras corretas/erradas (ver Figura 53). Após uma reunião com a [ESTeSC-IPC](#), foi estipulado que este ecrã não era necessário e acabou por ser substituído por um ecrã que mostra um gráfico atualizado consoante o número de sequências certas. É possível visualizar a versão final na figura 117 ([Resultado dos subtestes do teste Esforço Auditivo](#)).



Figura 53: Versão desatualizada da página de resultados do subtteste Esforço Auditivo

5.1.1.8 *Feedback play de sons*

No caso dos subttestes Esforço Auditivo e Inteligibilidade, existiam uns “tracinhos” que procuravam dar um *feedback* ao utilizador relativamente ao progresso do toque de sons. Após uma reunião com a [ESTeSC-IPC](#), foi estipulado que este ecrã (ver Figura 54) ajudava o utilizador, pois, por exemplo, seria possível verificar quando seriam tocadas as 3 últimas palavras do subtteste Esforço Auditivo. Deste modo, este *feedback* acabou por ser alterado para mostrar apenas o número da sequência atual no subtteste Esforço Auditivo e substituído pela adição do número atual da sequência e atualização do gráfico perante esse mesmo número, no subtteste de inteligibilidade. É possível visualizar a versão final na figura 115 ([Continuação do toque dos sons nos subttestes do teste Esforço Auditivo](#)).

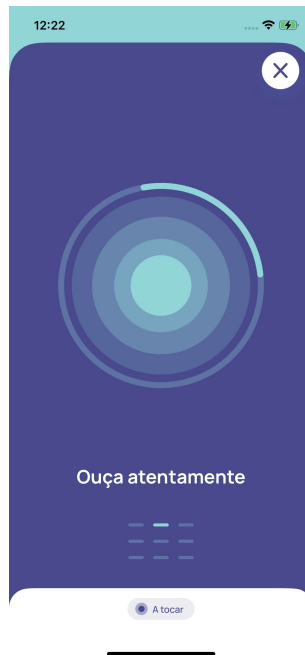


Figura 54: Versão desatualizada da página de toque de sons do subteste Esforço Auditivo

5.1.2 Funcionalidades

Nesta Secção 5.1.2, são demonstradas algumas das alterações que foram efetuadas em termos de funcionalidades da aplicação ao longo do desenvolvimento do projeto.

5.1.3 Validade dos subtestes do PAC

Os subtestes PAC passaram a ter uma validade de 30 dias, o que significa que ao fim de 30 dias esse teste já não será válido para o desbloquear de uma recomendação. Inicialmente, os subtestes realizados não apresentavam validade, no entanto, atualmente o utilizador terá de efetuar os 4 subtestes no período de 30 dias para obter uma recomendação. Caso contrário, terá de voltar a realizar os subtestes que já se encontram inválidos.

5.1.3.1 Validar as escolhas do subteste Esforço Auditivo

O subteste Esforço Auditivo sofreu algumas alterações na validação das respostas do utilizador. Inicialmente o utilizador teria de selecionar as 3 últimas palavras que ouviu, sendo que a ordem pela qual as selecionava era importante, o que significa

que seleccionar uma palavra correta, mas que não respondesse à ordem que foi tocada seria considerado errado. Posteriormente, não se deveria ter em conta a ordem pela qual o utilizador escolhia as palavras, sendo que seria apenas avaliado se tinham sido escolhidas as palavras corretas. Mais tarde, surgiu o requisito de não avaliar as 3 últimas palavras selecionadas, mas apenas a antepenúltima palavra selecionada, tendo sido reformulada toda a lógica de seleção de palavras, pois desta vez apenas seria necessário escolher uma palavra. Após alguns testes por parte da [ESTeSC-IPC](#), foi estipulado que o teste se encontrava muito difícil e foi apresentado um novo requisito. Desta vez, o utilizador teria de seleccionar novamente as 3 últimas palavras, mas o que realmente seria avaliado era se a antepenúltima palavra tocada se encontrava na lista de palavras escolhidas, sendo que a ordem da seleção não seria importante. O subteste Esforço Auditivo encontra-se implementado com este último requisito, tendo sido já avaliado pela [ESTeSC-IPC](#) e estar conforme o solicitado.

5.1.3.2 *Login automático após registo*

Aquando do registo de um novo utilizador, caso este fosse bem sucedido, era realizado automaticamente o *login* com a conta criada. Esta funcionalidade foi retirada, pois deve-se confirmar primeiramente o registo da nova conta antes de ser realizado o *Login*. Foi acrescentada essa informação após o sucesso da criação de conta.

5.2 TESTES DE USABILIDADE

Foi desenvolvido e partilhado um questionário com a [ESTeSC-IPC](#) que permitisse obter *feedback* dos utilizadores perante a utilização da aplicação. Inicialmente são retratadas as secções de perguntas do questionário (Secção 5.2.1), seguido das respostas obtidas pelos utilizadores (Secção 5.2.2).

5.2.1 *Secções*

O questionário encontra-se dividido em 5 secções:

- Descrição: A primeira secção (ver Figura 55) descreve o tema do questionário, referindo a anonimidade da recolha de dados e o seu uso exclusivo para o estudo das funcionalidades e fluxo da aplicação, permitindo avaliar e identi-

ficar oportunidades de melhoria. De igual forma, são identificados todos os responsáveis pelo desenvolvimento do projeto.

Secção 1 de 5

Questionário utilização aplicação *Hear App* iOS

O questionário que se segue pretende aferir a usabilidade da aplicação Hear App na versão iOS. Esta aplicação foi desenvolvida no âmbito do projeto Audiology4All/A4A ao abrigo do Programa Portugal 2020 através de consórcio liderado pela Evollu e desenvolvido em conjunto com os copromotores Escola Superior de Tecnologia da Saúde de Coimbra, Departamento Engenharia Informática e Departamento de Engenharia Eletrónica e de Computadores da Universidade de Coimbra.

A resposta às seguintes perguntas serve única e exclusivamente para o estudo das funcionalidades e fluxo da aplicação com vista à sua avaliação e identificação de oportunidades de melhoria. As respostas são obtidas anonimamente, o seu email e conta Google não farão parte das suas respostas.

Entidades responsáveis pelo desenvolvimento do projeto.



Figura 55: Questionário da secção 1

- Perfil utilizador: Esta secção (ver Figura 56) permite obter alguns dados sobre o perfil do utilizador. Encontram-se presentes as seguintes perguntas:
 - Faixa etária do utilizador: Criança (0 a 14 anos), Adolescente (15 a 19 anos), Adulto (20 a 64 anos) e Idoso (+65 anos).
 - Género: Masculino, Feminino e Prefiro não referir.
 - Localização geográfica: Norte de Portugal, Centro de Portugal, Sul de Portugal, Ilhas dos Açores, Ilhas da Madeira, Outro País, Prefiro não referir.

The image shows two parts of a questionnaire form. Part (a) is titled 'Perfil do utilizador' and contains two sections: 'Faixa etária do utilizador *' with four radio button options (Criança (0 a 14 anos), Adolescente (15 a 19 anos), Adulto (20 a 64 anos), Idoso (65 + anos)) and 'Género *' with four radio button options (Masculino, Feminino, Prefiro não referir, Outra: _____). Part (b) is titled 'Localização geográfica *' and contains six radio button options (Norte de Portugal, Centro de Portugal, Sul de Portugal, Ilhas dos Açores, Ilhas da Madeira, Outro País, Prefiro não referir).

Figura 56: Questionário da secção 2

De referir que os dados grupo etário são provenientes do instituto nacional de estatística [53].

- *System Usability Scale (SUS)*: O **SUS** é uma escala métrica de usabilidade que permite ajudar a medir a usabilidade de um produto. Trata-se de um dos métodos mais conhecidos e simples de utilização devido a ser cientificamente correto, mas não extremamente longo e monótono para o utilizador (ver Figura 57).

Permite avaliar os seguintes critérios:

- Efetividade: Os utilizadores conseguem concluir os objetivos?
- Eficiência: Quais são os esforços e recursos necessários para concluir os objetivos?
- Satisfação: A experiência de utilização foi satisfatória?

A escala é constituída por 10 perguntas, tendo estas sido adaptadas para o contexto do projeto e de resposta de escala linear 1 a 5 (Discordo totalmente a Concordo plenamente). De seguida, são enunciadas as perguntas utilizadas na secção 3 do questionário:

- Gostei da aplicação e estou a planear a sua utilização frequentemente;

- Achei a aplicação demasiado complexa;
- Achei a aplicação fácil e simples de utilizar;
- Acho que preciso de utilizar a aplicação com o auxílio de uma pessoa com conhecimentos técnicos sobre a mesma;
- Achei as funcionalidades implementadas uma mais-valia;
- A aplicação apresenta muitas inconsistências;
- Acho que os utilizadores facilmente conseguem interagir com a aplicação;
- Achei a aplicação demasiado complexa;
- Senti-me confortável e confiante a utilizar a aplicação;
- Necessitei de aprender novas informações para utilizar a aplicação.

Sistema de escala de usabilidade

Nesta secção é usado o sistema SUS (System Usability Scale), de modo a conseguir averiguar o nível de usabilidade da aplicação. Todas as perguntas desta secção apresentam escalas numéricas como opção de resposta.

Gostei da aplicação e estou a planear a sua utilização frequentemente. *

1 2 3 4 5

Discordo totalmente Concordo plenamente

Achei a aplicação demasiado complexa. *

1 2 3 4 5

Discordo totalmente Concordo plenamente

Achei a aplicação fácil e simples de utilizar. *

1 2 3 4 5

Discordo totalmente Concordo plenamente

Acho que preciso de utilizar a aplicação com o auxílio de uma pessoa com conhecimentos técnicos sobre a mesma. *

1 2 3 4 5

Discordo totalmente Concordo plenamente

Achei as funcionalidades implementadas uma mais-valia. *

1 2 3 4 5

Discordo totalmente Concordo plenamente

A aplicação apresenta muitas inconsistências. *

1 2 3 4 5

Discordo totalmente Concordo plenamente

Acho que os utilizadores facilmente conseguem interagir com a aplicação. *

1 2 3 4 5

Discordo totalmente Concordo plenamente

(a) Perguntas da secção 3

(b) Continuação das perguntas da secção 3

Achei a aplicação demasiado complexa. *

1 2 3 4 5

Discordo totalmente Concordo plenamente

Senti-me confortável e confiante a utilizar a aplicação. *

1 2 3 4 5

Discordo totalmente Concordo plenamente

Necessitei de aprender novas informações para utilizar a aplicação. *

1 2 3 4 5

Discordo totalmente Concordo plenamente

(c) Continuação das perguntas da secção 3

Figura 57: Questionário da secção 3

- Outras informações: Esta secção (ver Figura 58) permite conhecer a interação do utilizador com problemas de saúde auditivos. Encontram-se presentes as seguintes perguntas:
 - Através da utilização da aplicação descobri que tenho problemas auditivos: Sim, Não, Já tinha sido diagnosticado antes e a aplicação detetou esse problema, Já tinha sido diagnosticado antes, mas a aplicação não detetou esse problema.
 - Após a utilização da aplicação irei consultar um audiologista: Sim, Talvez, Não.

- Conheço familiares/amigos/outros que tenham problemas auditivos: Sim, Não.
- Vou aconselhar o uso da aplicação a amigos/familiares/outros: Sim, Talvez, Não.
- Utiliza outras aplicações ligadas à saúde?: Sim, Não.
- Se respondeu sim à pergunta anterior, por favor indique que aplicações utiliza: *SNS24, MySNS, Apple Saúde/Fitness, Huawei Health, Google Fit*, Outra opção. Esta pergunta não é de cariz obrigatório.
- Numa escala de 1 a 5, em que 1 significa “Nada importante” e 5 “Muito importante”, como classifica a importância da aplicação?
- Numa escala de 1 a 5, em que 1 significa “Nada satisfeito” e 5 “Muito satisfeito”, como classifica o seu nível de satisfação pelo estado geral da aplicação?

Outras perguntas

Esta secção permite conhecer a interação do utilizador com problemas de saúde auditivos.

Através da utilização da aplicação descobri que tenho problemas auditivos. *

Sim

Não

Já tinha sido diagnosticado antes e a aplicação detetou esse problema.

Já tinha sido diagnosticado antes, mas a aplicação não detetou esse problema.

Após a utilização da aplicação irei consultar um audiologista. *

Sim

Talvez

Não

Conheço familiares/amigos/outros que tenham problemas auditivos. *

Sim

Não

Vou aconselhar o uso da aplicação a amigos/familiares/outros. *

Sim

Talvez

Não

Utiliza outras aplicações ligadas à saúde? *

Sim

Não

Se respondeu sim à pergunta anterior, por favor indique que aplicações utiliza:

SNS 24

MySNS

Apple Saúde/Fitness

Huawei Health

Google Fit

Outra: _____

(a) Perguntas da secção 4

(b) Continuação das perguntas da secção 4

Numa escala de 1 a 5, em que 1 significa "Nada importante" e 5 "Muito importante", como classifica a importância da aplicação? *

1 2 3 4 5

Nada importante Muito importante

Numa escala de 1 a 5, em que 1 significa "Nada satisfeito" e 5 "Muito satisfeito", como classifica o seu nível de satisfação pelo estado geral da aplicação? *

1 2 3 4 5

Nada satisfeito Muito satisfeito

(c) Continuação das perguntas da secção 4

Figura 58: Questionário da secção 4

- **Sugestões/Melhorias/Problemas:** A última secção (ver Figura 59) permite ao utilizador identificar sugestões e/ou melhorias para melhorar o estado da aplicação ou problemas que ocorreram durante o seu uso.

Apresenta duas questões de texto livre de cariz não obrigatório:

- Novas funcionalidades e/ou melhorias na aplicação;
- Problemas encontrados.

The image shows a survey form with a blue header bar containing the text 'Sugestões, melhorias e problemas'. Below the header, there are three main sections, each with a light blue background and rounded corners. The first section contains the text 'Ajude-nos a melhorar, deixe em baixo as suas sugestões, opiniões e/ou problemas encontrados.' The second section is titled 'Novas funcionalidades e/ou melhorias na aplicação.' and includes a text input field with the placeholder 'A sua resposta'. The third section is titled 'Problemas encontrados' and also includes a text input field with the placeholder 'A sua resposta'.

Figura 59: Questionário da secção 5

5.2.2 Respostas

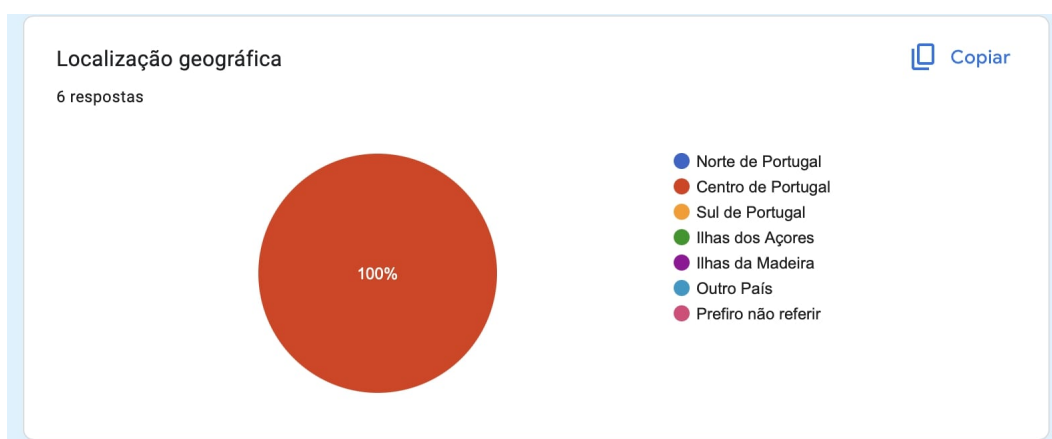
Infelizmente, não existiu uma grande afluência no preenchimento do questionário, tendo sido apenas registadas 6 respostas. De seguida, são analisados os dados das respostas obtidas:

Todas as respostas são provenientes de utilizadores do sexo feminino, adulto e da zona centro de Portugal (ver Figuras 60 e 61).



(a) Respostas da secção 2

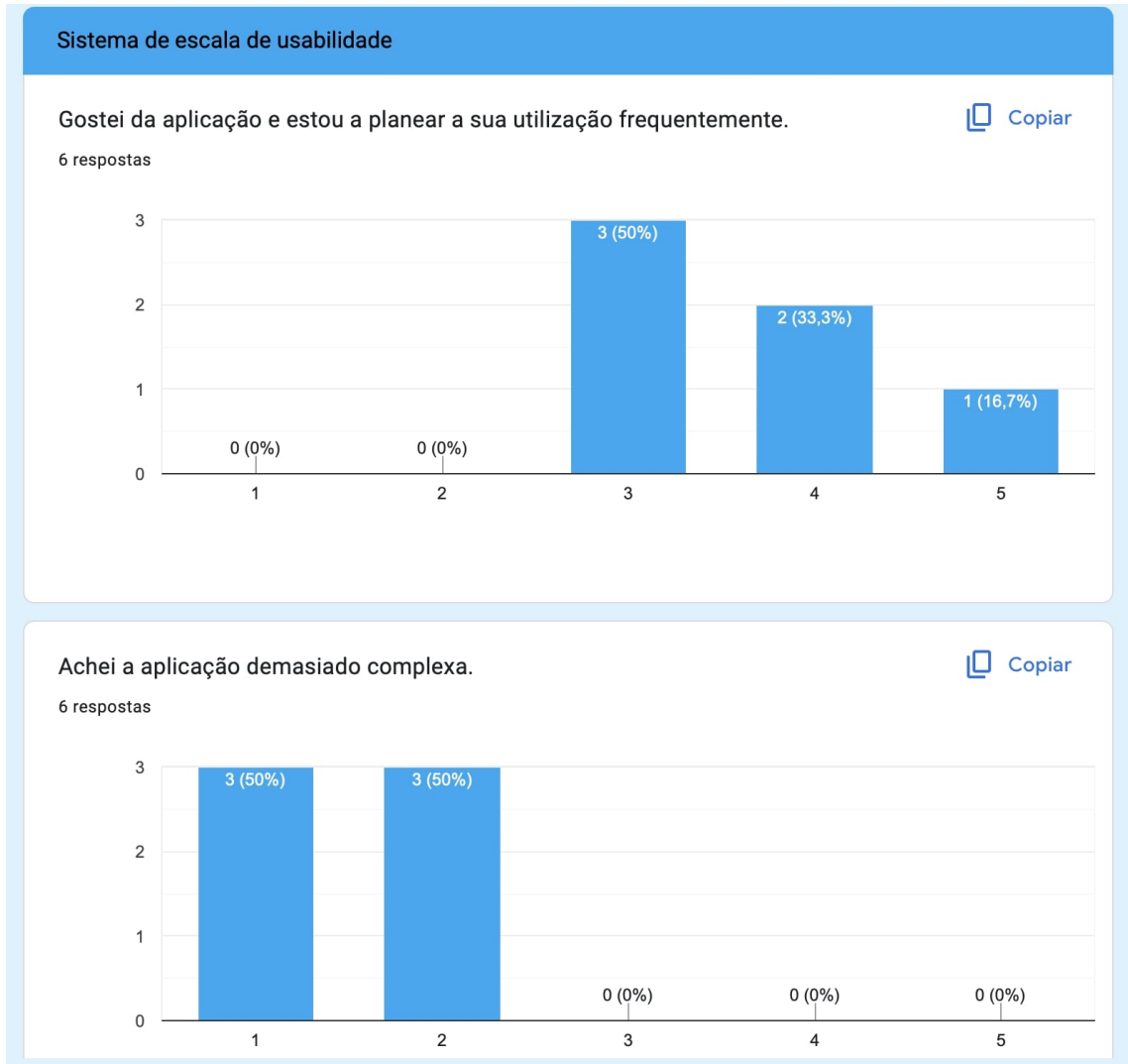
Figura 60: Respostas da secção 2



(a) Continuação de respostas da secção 2

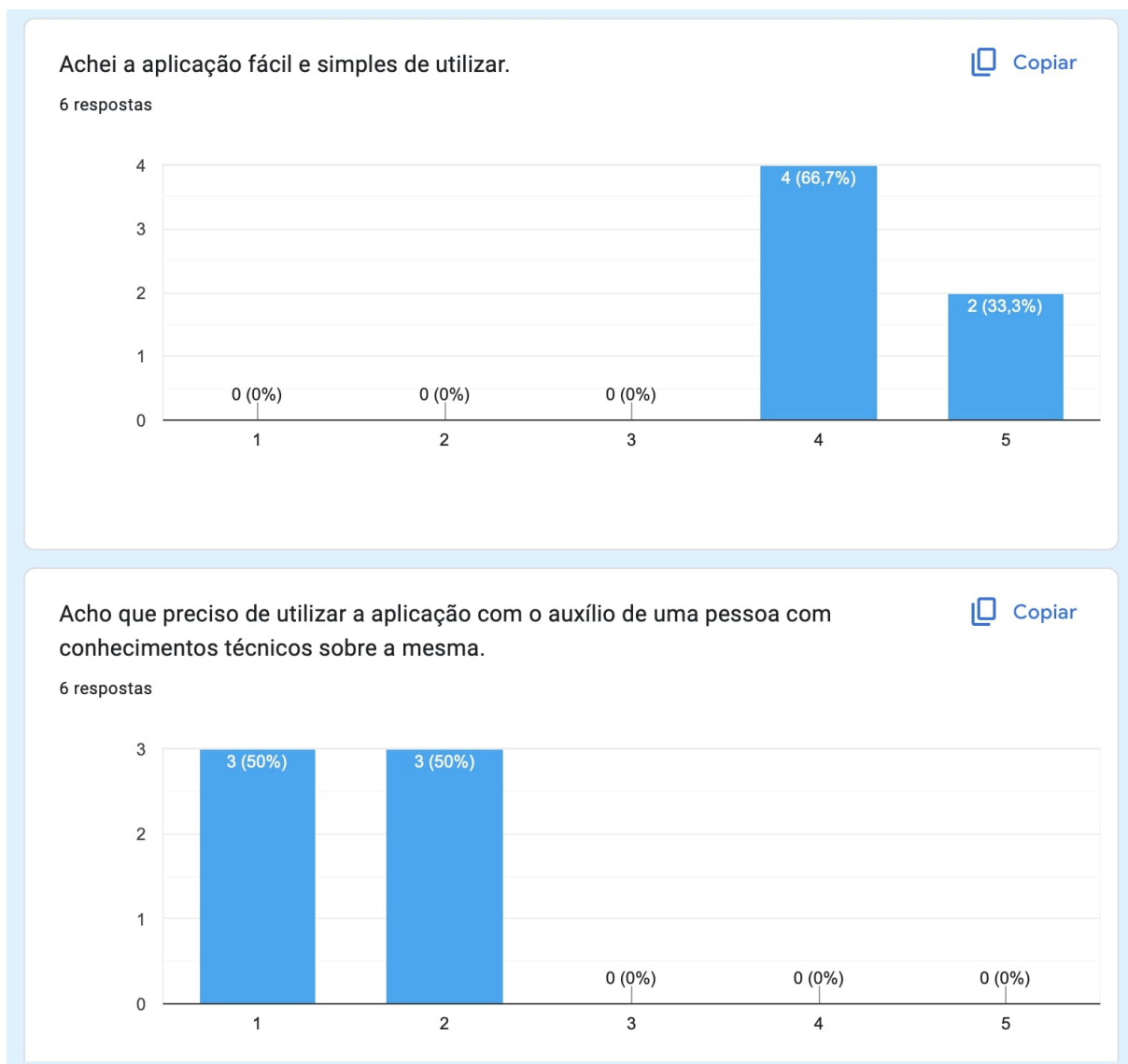
Figura 61: Continuação de respostas da secção 2

Apenas 1 utilizador planeia utilizar a aplicação frequentemente, 2 referem que é provável o seu uso frequente e 3 utilizadores absteram-se do uso frequente. No geral, os utilizadores não consideram a aplicação complexa (ver Figura 62) e consideram a mesma fácil de usar e interagir (ver Figura 64), não precisando de qualquer auxílio para a sua utilização (ver Figura 63).



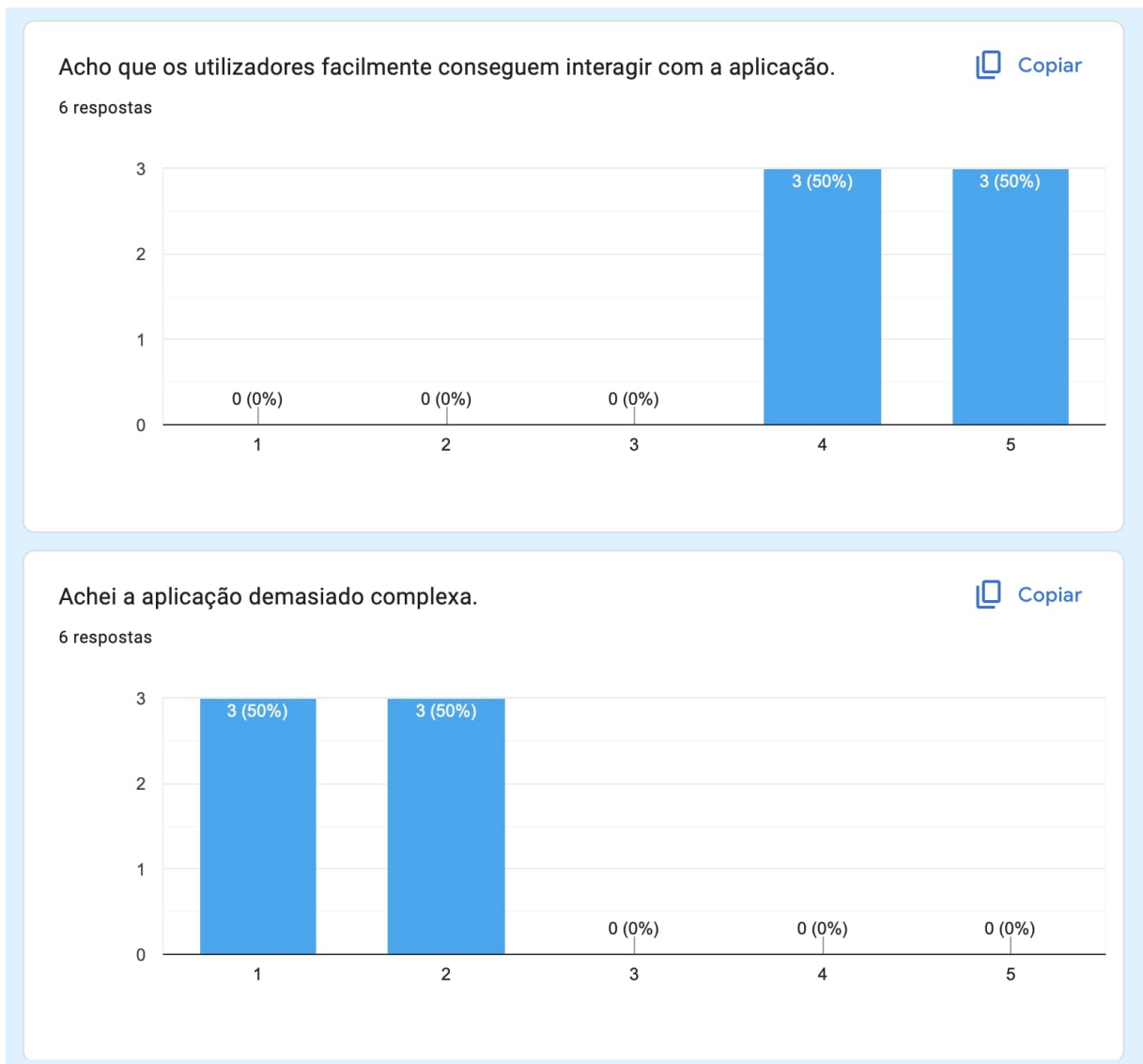
(a) Respostas da secção 3

Figura 62: Respostas da secção 3



(a) Continuação de respostas da secção 3

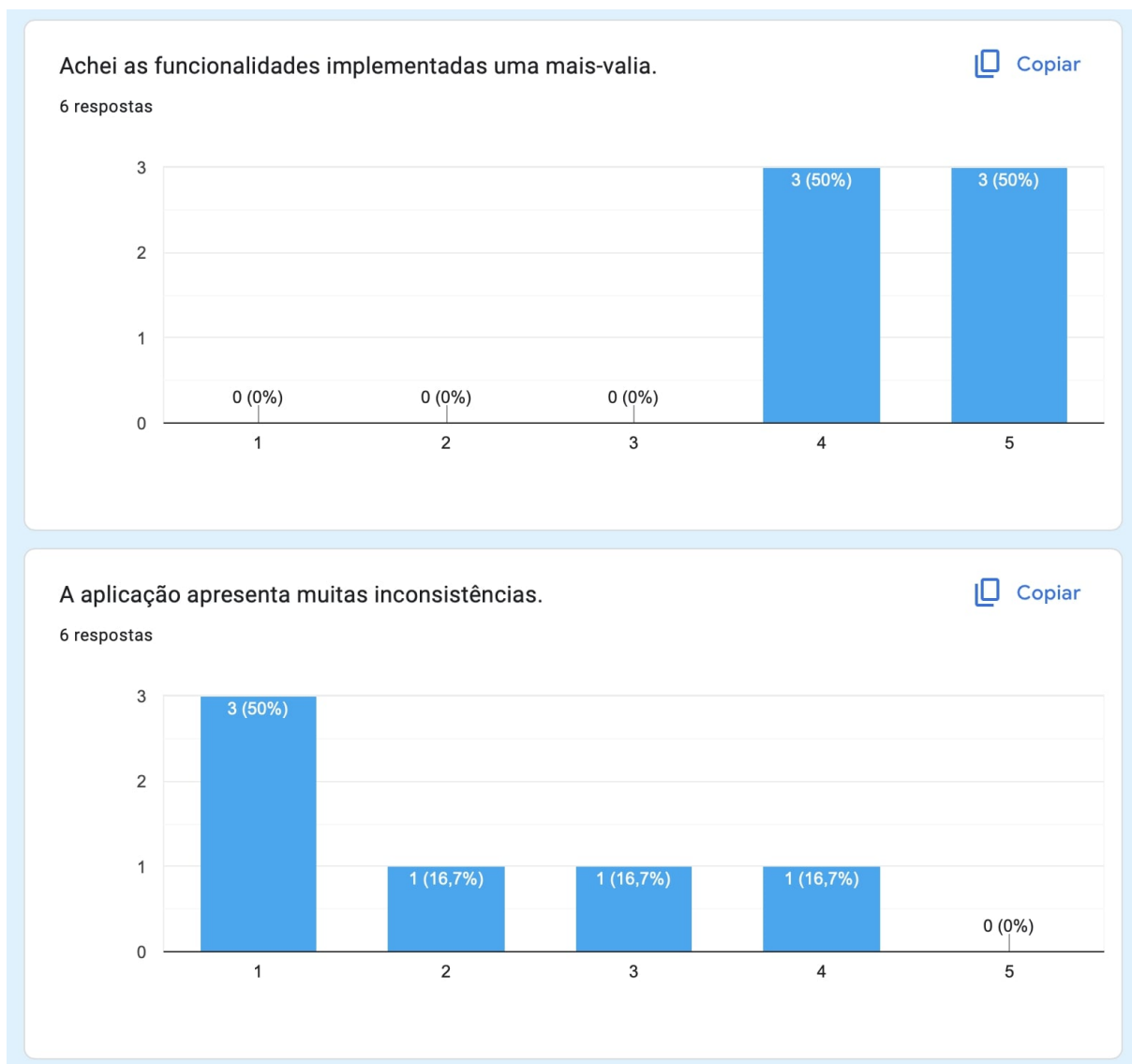
Figura 63: Respostas da secção 3



(a) Continuação de respostas da secção 3

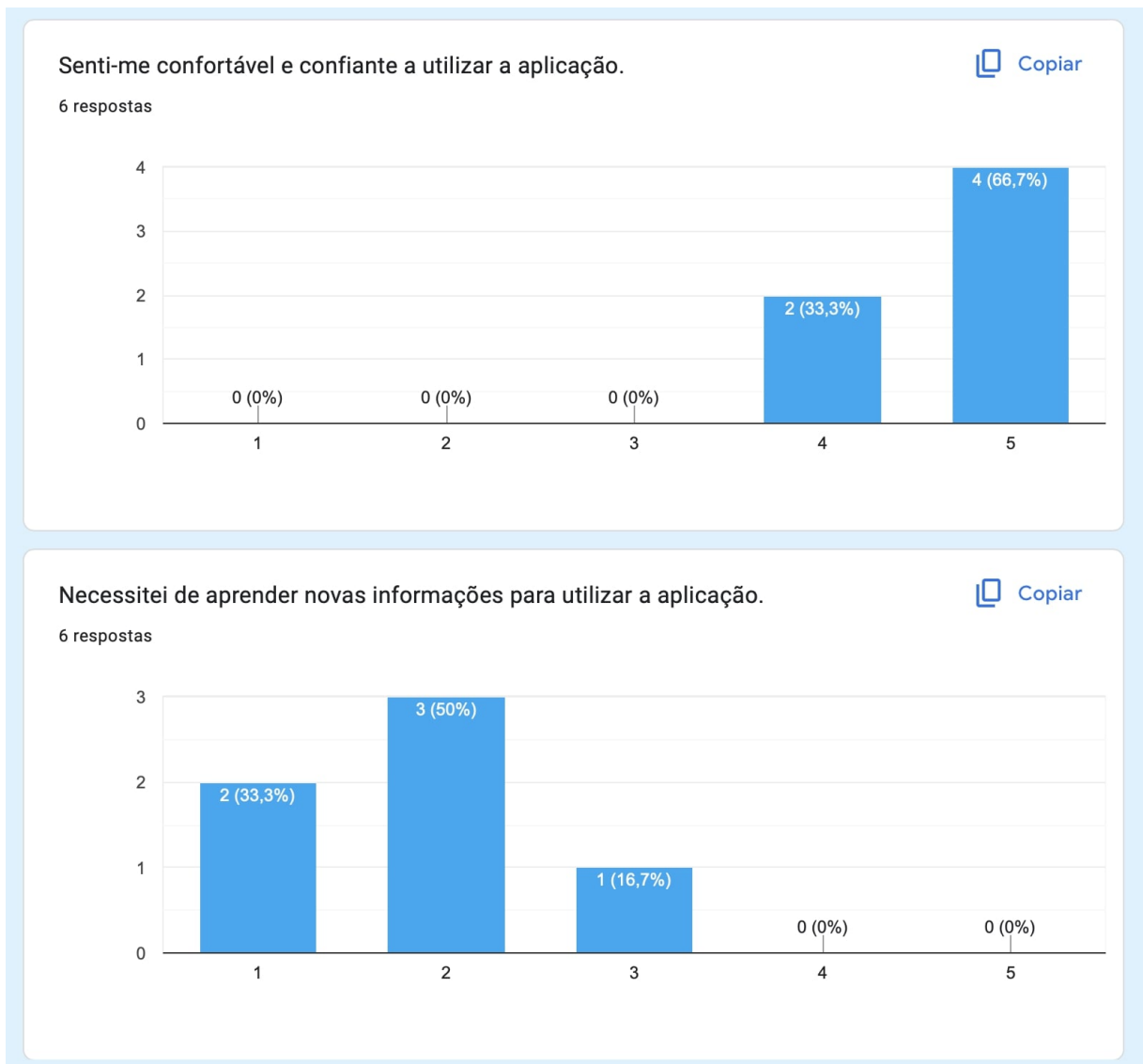
Figura 64: Continuação de respostas da secção 3

Todos os utilizadores apresentam a ideia de que as funcionalidades implementadas são importantes e, em média, que a aplicação apresenta poucas inconsistências (ver Figura 65). Sentem-se bastante confortáveis a utilizar a aplicação, sem existir a necessidade de aprender muitas novas informações para conseguir utilizar a mesma (ver Figura 66).



(a) Continuação de respostas da secção 3

Figura 65: Continuação de respostas da secção 3

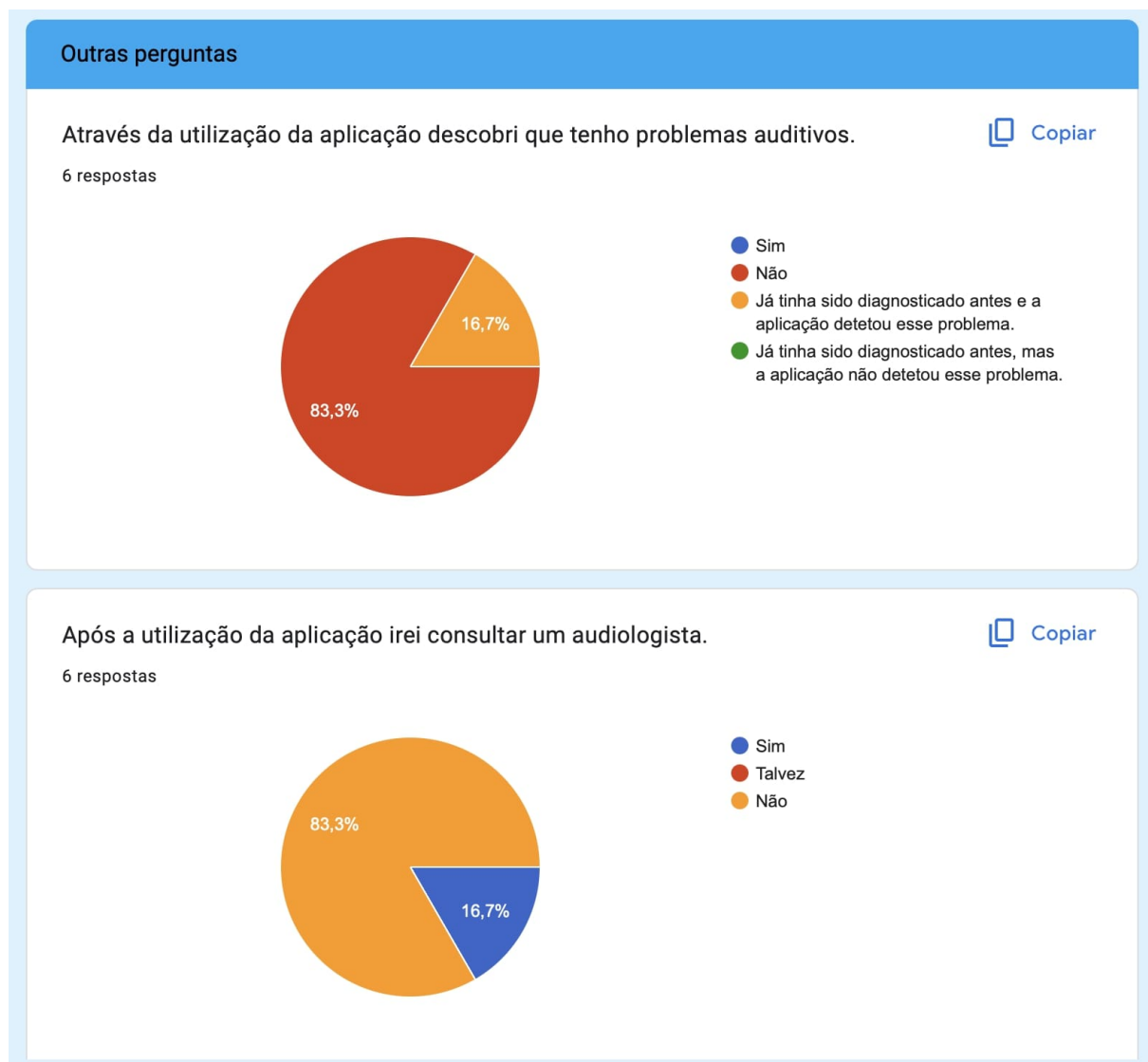


(a) Continuação de respostas da secção 3

Figura 66: Continuação de respostas da secção 3

5 utilizadores referiram que não descobriram a presença de problemas auditivos através do uso da aplicação. No entanto, 1 utilizador refere que a aplicação confirmou um problema auditivo, sendo que esse mesmo utilizador já tinha sido diagnosticado com um problema auditivo.

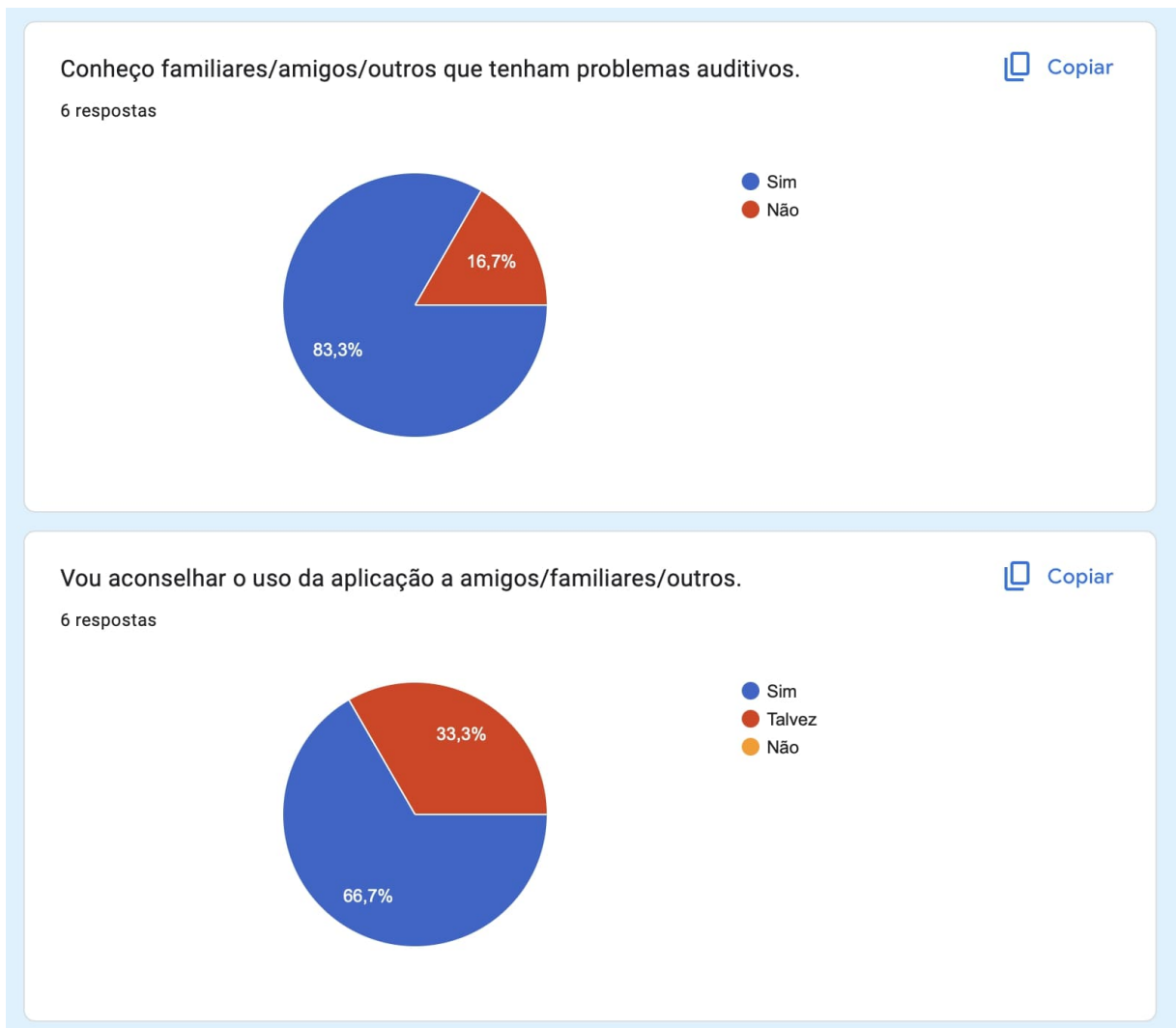
5 utilizadores não tencionam consultar um audiologista (o que é expectável, tendo em conta que não foi encontrado um problema auditivo através do uso da aplicação), mas 1 utilizador tenciona consultar um audiologista após o uso da aplicação (ver Figura 67). No entanto, é provável que existisse um maior número de utilizadores a considerar uma visita ao audiologista caso fosse encontrado um problema auditivo.



(a) Respostas da secção 4

Figura 67: Respostas da secção 4

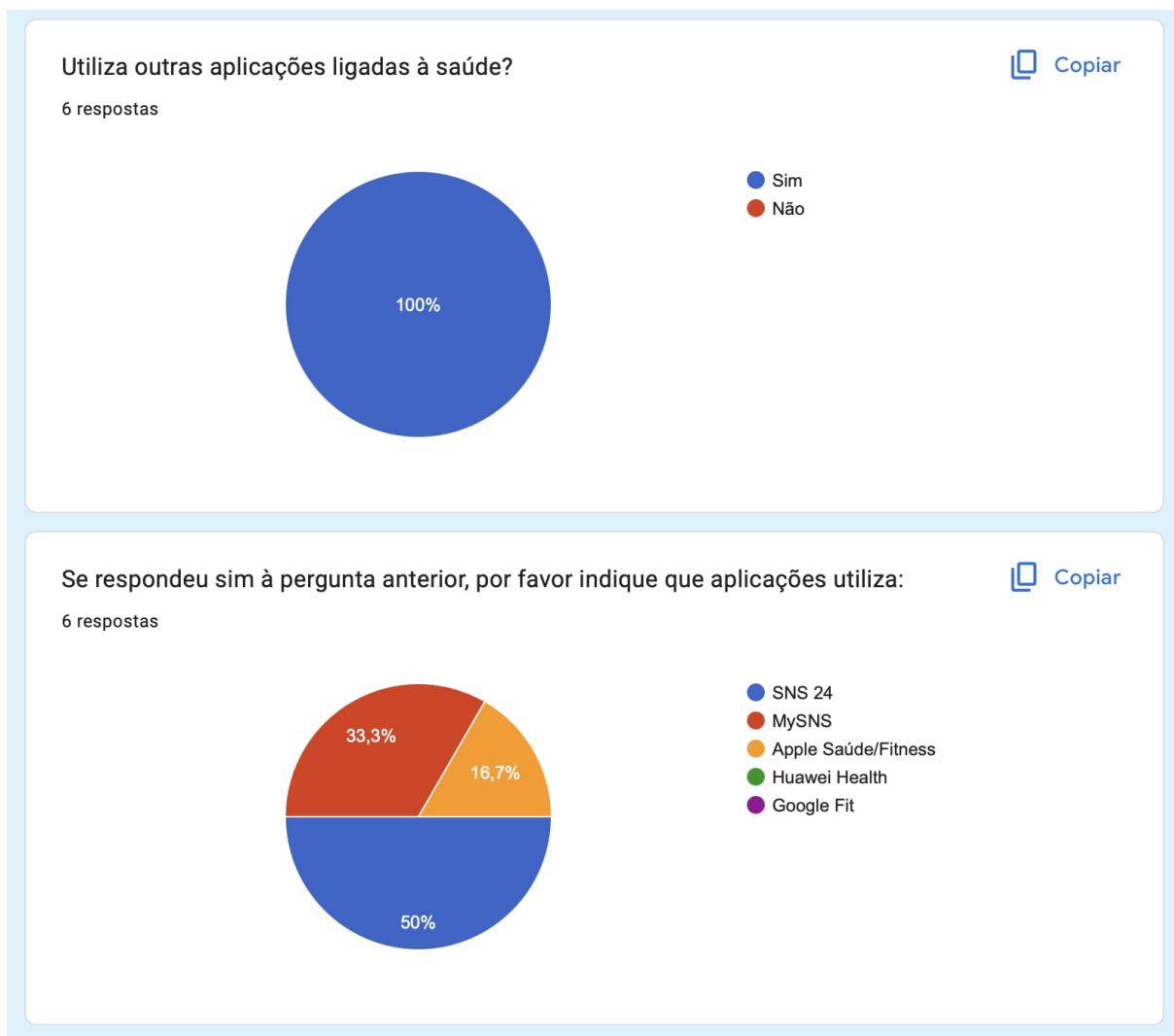
Quase todos os utilizadores conhecem alguém que apresenta problemas auditivos, e no geral, irão aconselhar a utilização da aplicação a amigos/familiares/outros (ver Figura 68).



(a) Continuação de respostas da secção 4

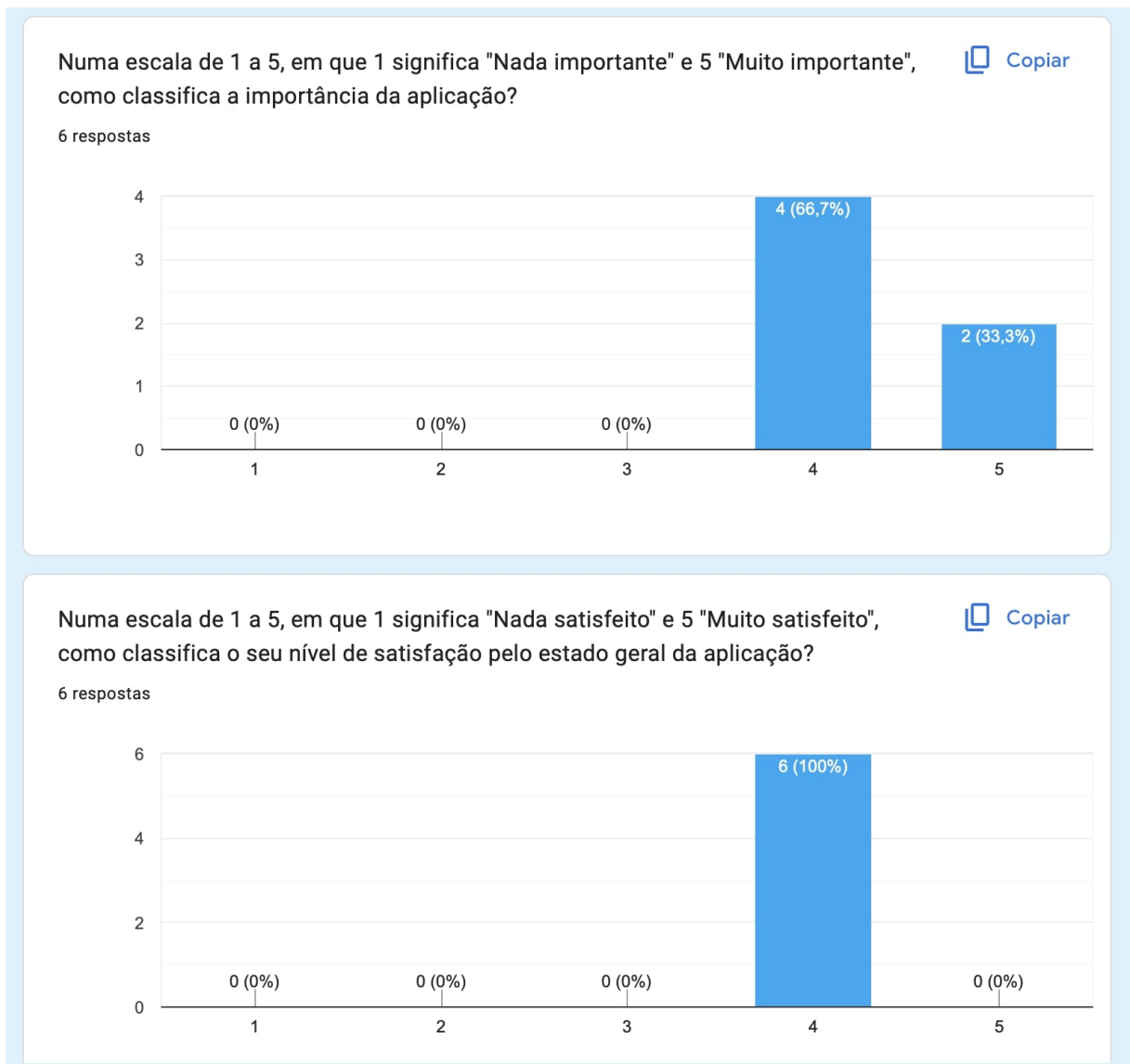
Figura 68: Continuação de respostas da secção 4

Todos os utilizadores referem utilizar outras aplicações ligadas à saúde, sendo que metade dos utilizadores utilizam a *Google Fit*, um terço utilizam o *MySNS* e uma pessoa utiliza a *Apple Saúde/Fitness* (ver Figura 69). De igual forma, todos os utilizadores referem que a aplicação é muito importante e encontram-se satisfeitos com o estado atual da mesma, como se pode verificar na Figura 70.



(a) Continuação de respostas da secção 4

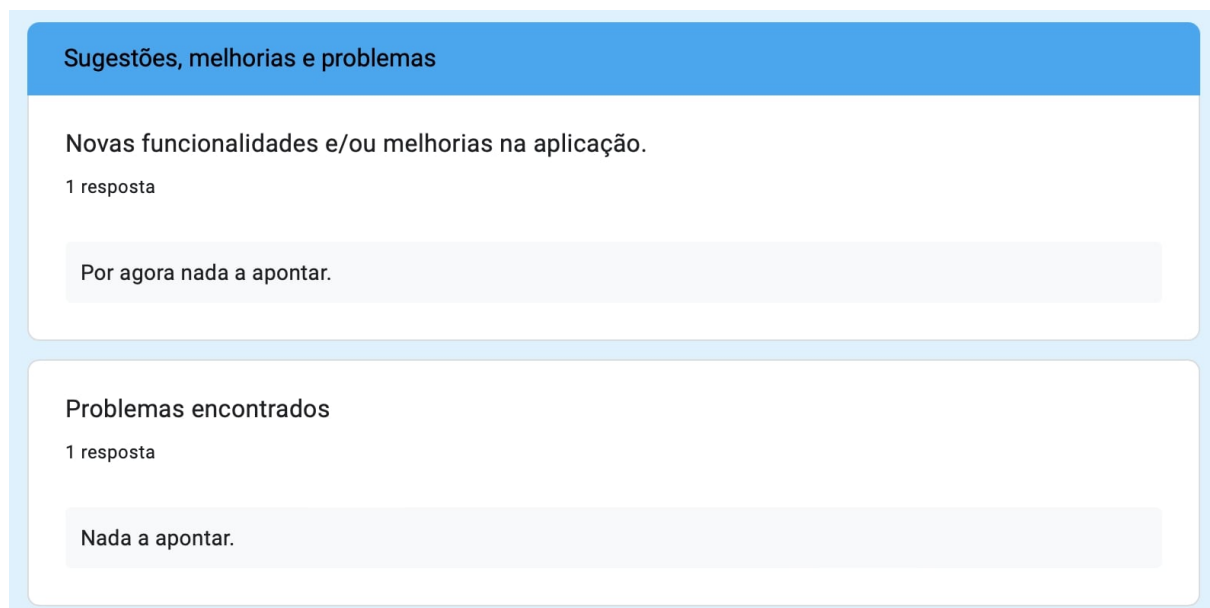
Figura 69: Continuação de respostas da secção 4



(a) Continuação de respostas da secção 4

Figura 70: Continuação de respostas da secção 4

Relativamente à secção de possíveis melhorias e/ou problemas encontrados, existe apenas uma resposta que não indica nenhum problema detetado ou possíveis melhorias (ver Figura 71).



(a) Respostas da secção 5

Figura 71: Respostas da secção 5

5.3 PRODUTO FINAL

Esta secção descreve todo o fluxo do estado final da aplicação aquando do término do estágio. A abertura da aplicação (Secção 5.3.1), a autenticação (Secção 5.3.2), a página inicial (Secção 5.3.3), as definições (Secção 5.3.4), o perfil (Secção 5.3.5 e os testes (Secção 5.3.6, 5.3.7, 5.3.8 e 5.3.9) encontram-se detalhados nesta secção.

5.3.1 Abertura da aplicação

Esta secção apresenta o ecrã inicial da aplicação, sendo que quando o utilizador abre a aplicação é apresentado um *SplashScreen* que apresenta o nome da aplicação e o *branding* da *Evollu* (ver Figura 72).

5.3.2 Autenticação

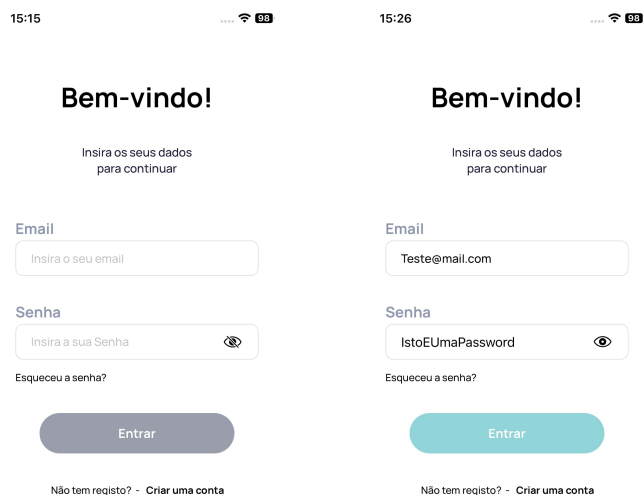
Esta secção apresenta o fluxo de autenticação da aplicação (secções 5.3.2.1, 5.3.2.2 e 5.3.2.3).



Figura 72: *Loading screen*

5.3.2.1 *Página de login*

Após o *Loading Screen* é apresentado ao utilizador a página para a realização da autenticação (ver Figura 73). Neste ecrã é possível realizar o *Login*, caso o utilizador já contenha uma conta de domínio *Evollu Auth-Clinical*, recuperar a palavra-passe ou criar uma nova conta. Esta página apresenta um botão “Entrar” que se encontra desabilitado (o clique no mesmo não realiza nenhuma ação) enquanto não for preenchido o campo de *Email* e Senha. O utilizador pode clicar no botão à direita presente no *card* da senha, para visualizar a palavra-passe que inseriu, pois por *default* a palavra-passe encontra-se protegida.

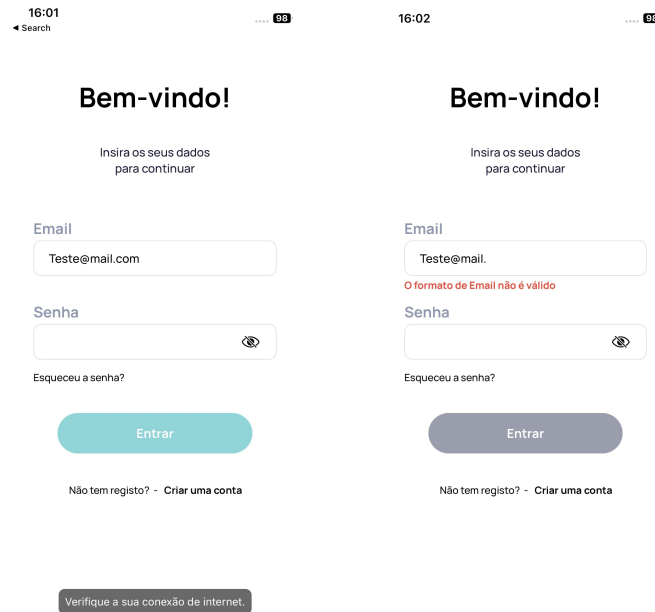


(a) Botão desabilitado

(b) Dados preenchidos e senha desprotegida

Figura 73: Página inicial de *login*

Nesta mesma página (ver Figura 74) são apresentadas algumas informações relativamente ao dispositivo não se encontrar ligado à *Internet* ou quando não existe nenhuma conta registada com os dados preenchidos pelo utilizador. Estas informações apenas são despoletadas quando o utilizador clica no botão “Entrar” (caso este esteja habilitado). Após o clique no botão “Entrar” e erro ao realizar o *Login*, o campo da palavra-passe é limpo por razões de segurança. De igual forma, é apresentada ao utilizador informação relativa aos dados de *Email*, nomeadamente se este se trata de um *Email* válido.

(a) Não está conectado à *Internet*(b) Formato errado de *Email*Figura 74: Erros na página inicial de *login*

5.3.2.2 *Página de recuperação de palavra-passe*

O utilizador pode aceder à página “Recuperar Palavra-Passe“ (ver Figura 75), de modo a definir uma nova palavra-passe para a sua conta. O acesso a esta página encontra-se na página inicial de *Login*, por baixo do *card* referente à senha, com o nome de “Esqueceu a senha;”. Este texto é na realidade um texto clicável que irá encaminhar o utilizador para a página de recuperação de palavra-passe. Este ecrã apresenta, da mesma forma, um botão que apenas será habilitado caso o *email* seja válido, assim como será apresentada uma mensagem de erro, se o *email* preenchido não estiver associado a nenhuma conta registada.

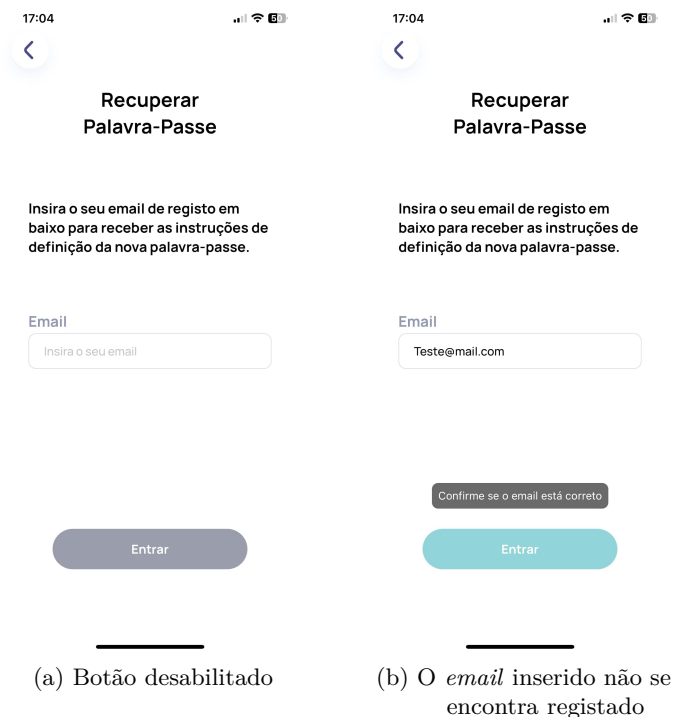


Figura 75: Página de recuperação da palavra-passe

De referir, que esta funcionalidade apenas funciona em ambiente de produção.

5.3.2.3 *Página de registo de nova conta*

Para utilizar a aplicação, o utilizador precisa de ter uma conta de domínio *Evolvu Auth-Clinical* e poderá criar uma conta clicando no texto “Criar uma conta“, presente na página inicial de *Login*. O utilizador será encaminhado para uma página de registo, onde terá que preencher todos os campos de informação, de modo a ativar o botão “Criar Conta“ (ver Figura 76). Estes campos são os seguintes:

- Nome: O utilizador deve indicar um nome que contenha pelo menos 3 caracteres.
- Data de Nascimento: O utilizador deve indicar a sua data de nascimento, que nunca poderá ser superior ao dia atual.
- País: Por *default*, o campo de país encontra-se automaticamente preenchido com o valor “Portugal“, mas poderá ser substituído por outro. Ao clicar no mesmo irá ser gerada uma lista com todos os países e o utilizador pode escolher a sua opção. De referir, que a escolha do país afeta diretamente o código

telefónico e este, por *default*, apresenta o valor de “+351” devido a ser o código telefónico para o país “Portugal”.

- *Email*: O utilizador deve indicar um *email* válido. Um *email* é considerado válido, caso coincida com os seguintes requisitos: “nome@entidade.domínio”.
- Senha: O utilizador deve formular uma palavra-passe que deve conter todos os seguintes requisitos:
 - 1 letra maiúscula;
 - 1 letra minúscula;
 - 1 carácter especial;
 - 1 número;
 - 8 caracteres.
- Termos e condições: O utilizador deve aceitar os termos e condições do serviço, que os seus dados pessoais sejam processados para efeitos de *marketing*, efeitos de pesquisa de mercado e envio de outras comunicações e ainda para marcação de consultas auditivas.

Os campos que necessitam de ser preenchidos com texto, apresentam mensagens de erro caso os dados não estejam de acordo com os requisitos. Outro fator importante consiste na informação apresentada ao utilizador, caso o *email* escolhido já se encontre registado na plataforma (não podem existir dois *emails* iguais registados na plataforma e foi implementado um *handling* de erros que permite realizar o *parsing* dos erros provenientes do servidor diretamente para a aplicação).

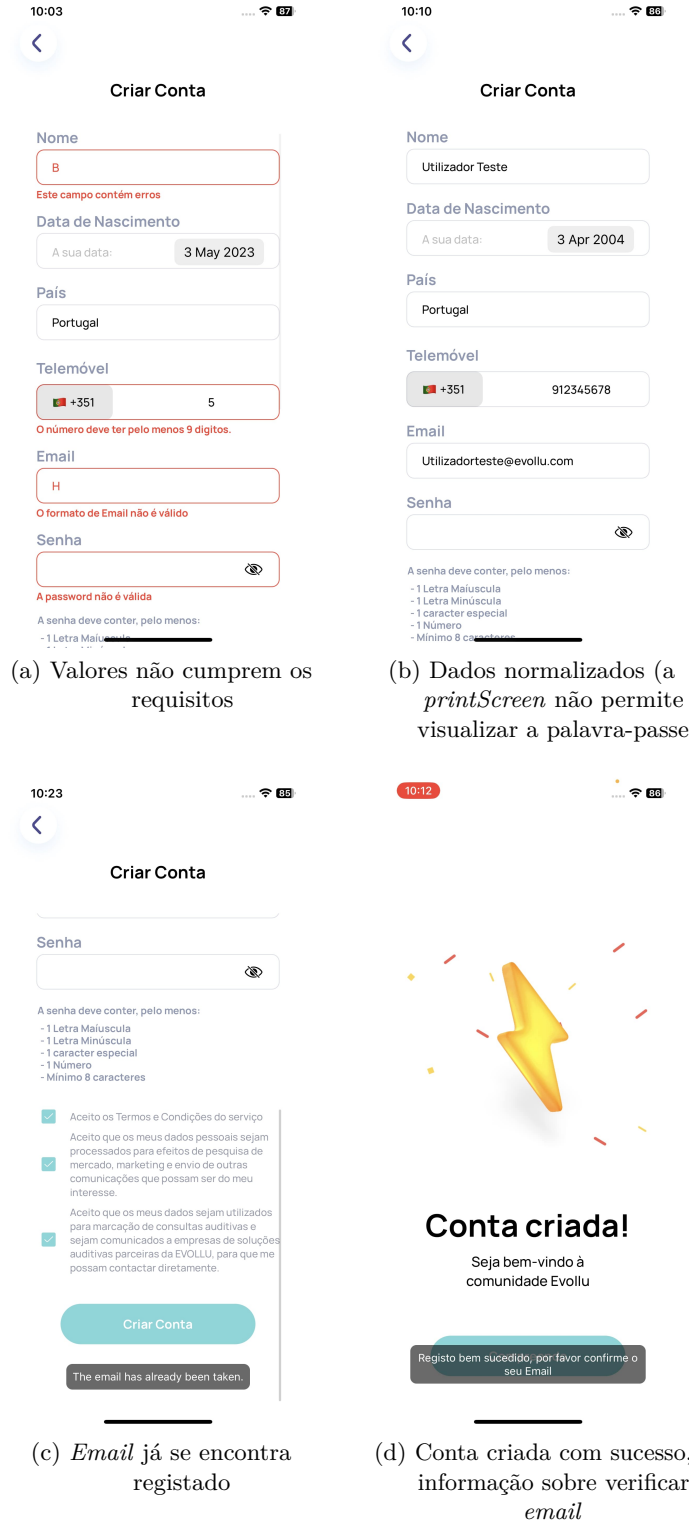


Figura 76: Processo de registar uma nova conta.

5.3.3 Página inicial

Após a realização do *Login*, o utilizador é encaminhado para a página inicial (ver Figura 77) onde poderá escolher o teste que deseja realizar, ou efetuar outras operações como aceder às definições, processamento de áudio, entre outros.

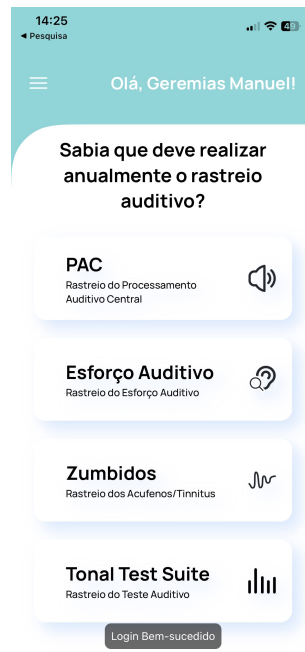


Figura 77: Página de seleção de testes

5.3.3.1 Escolha dos testes a realizar

O utilizador pode escolher o teste que pretende realizar e será encaminhado para esse mesmo teste.

5.3.3.2 Sidebar

A *Sidebar* (ver Figura 78) apresenta as diversas opções a que o utilizador tem acesso. É ainda apresentado o logótipo da *Evolu*, um ícone gerado automaticamente pelas primeiras letras do nome do utilizador (no caso de conter dois nomes) ou pela primeira letra do nome (no caso de conter apenas um nome), assim como o nome do utilizador e o seu *email*.

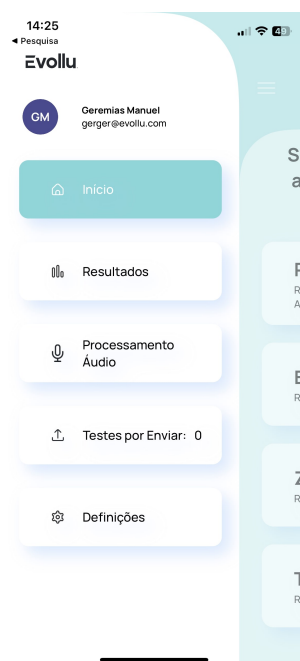


Figura 78: Side Bar

5.3.3.2.1 Início

Refere-se à página inicial onde pode seleccionar o teste que pretende efetuar.

5.3.3.2.2 Resultados

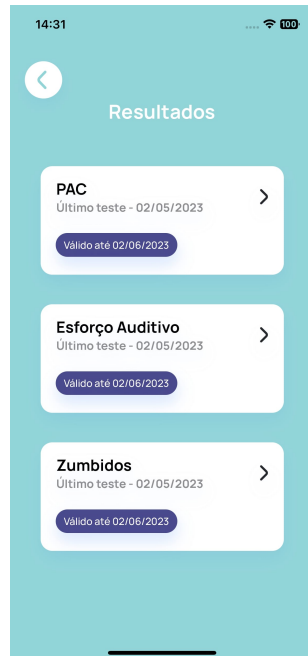
O utilizador poderá visualizar os testes que já efetuou anteriormente e que foram submetidos para o servidor.

Esta secção de resultados encontra-se dividida em 3 ecrãs (ver Figura 79), sendo que:

- O ecrã inicial é responsável por mostrar, para cada um dos testes, qual a data do último teste realizado e a sua validade.
- O segundo ecrã mostra todos os testes efetuados para aquele teste. Ou seja, se o utilizador seleccionou no primeiro ecrã o teste **PAC**, este ecrã irá mostrar todos os testes **PAC** que o utilizador já efetuou desde que criou a sua conta.
- O último ecrã mostra mais informações relativamente ao teste que foi escolhido para visualizar no ecrã anterior. Neste será possível verificar os subtestes que foram realizados, assim como a recomendação que foi obtida (caso tenham sido

realizados todos os requisitos para obter uma recomendação, caso contrário existe a informação sobre os requisitos que ainda se encontram em falta).

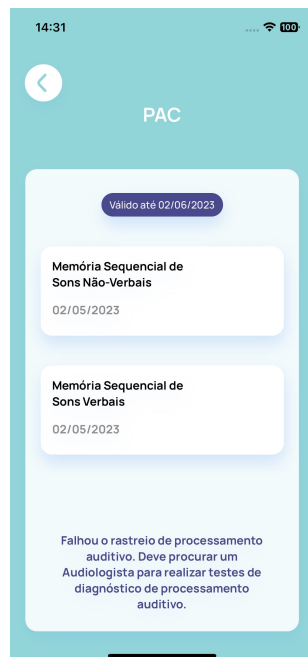
De seguida será demonstrado um exemplo para o teste PAC:



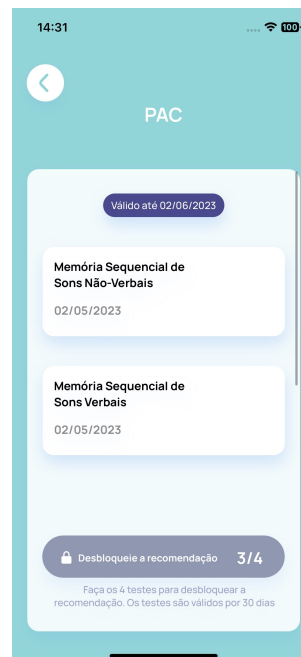
(a) Lista do último teste efetuado



(b) Lista de todos os testes efetuados



(c) Teste PAC recomendação desbloqueada



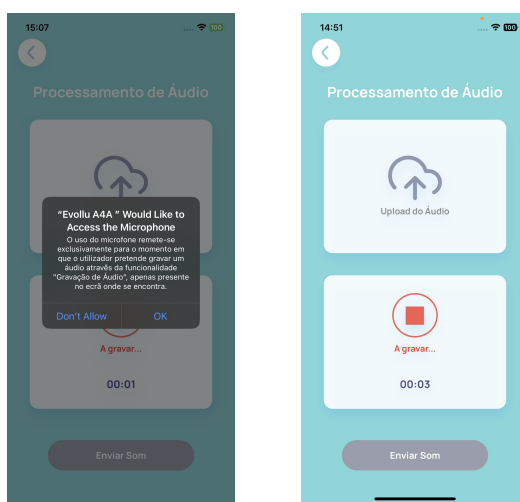
(d) Teste PAC sem recomendação desbloqueada

Figura 79: Ecrã de resultados.

5.3.3.2.3 Processamento de Áudio

O utilizador poderá comunicar com a [API do DEEC](#), de modo a processar o áudio através dessa mesma [API](#) (ver Figuras 80 e 81). Para tal, poderá realizar o carregamento de um ficheiro de áudio (nos formatos [.MPEG-1 Audio Layer-3 \(MP3\)](#), [.Waveform Audio \(WAV\)](#), [.Audio Interchange File Format \(AIFF\)](#) e [.mpeg4Audio](#) que se encontre disponível no seu dispositivo, ou poderá gravar um áudio na aplicação. Para realizar o carregamento de um ficheiro basta clicar na opção “[Upload do Áudio](#)” e escolher um áudio com um formato válido. Já para realizar a gravação de áudio basta clicar na opção “[Gravação do Áudio](#)” que irá começar automaticamente a gravar e o utilizador pode parar a gravação ao clicar novamente na *card*. Em ambos os casos é possível fazer o *play* do áudio ou a eliminação do áudio. De referir que sempre que o utilizador escolha uma opção e depois seleccione outra opção, o áudio atual será removido.

Outro fator importante, consiste na necessidade de aceitar a permissão para aceder ao microfone do dispositivo (aparece na primeira vez que o utilizador tentar gravar um áudio na aplicação), caso contrário, não será possível utilizar o microfone do dispositivo e não será gravado nenhum áudio.



(a) Permissão para gravar áudio

(b) A gravar áudio...

Figura 80: Ecrã de processamento de áudio.

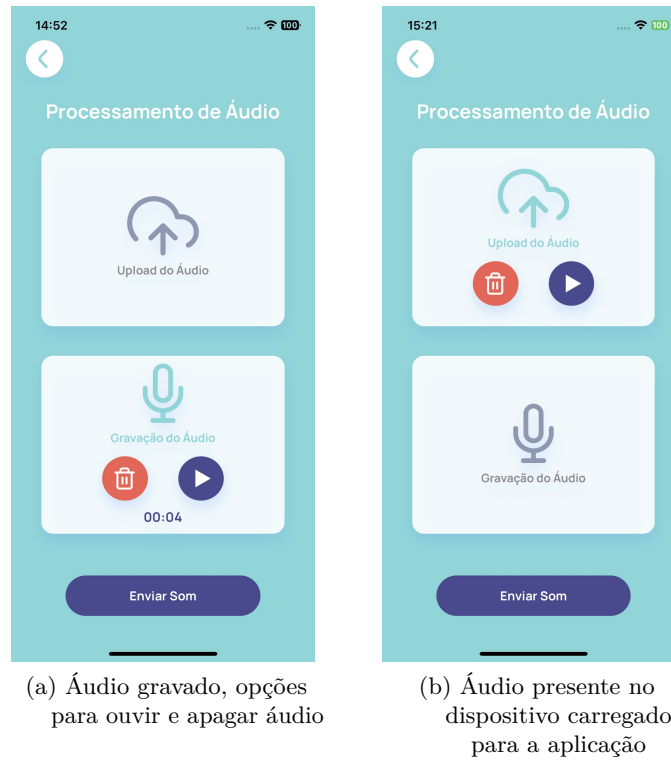
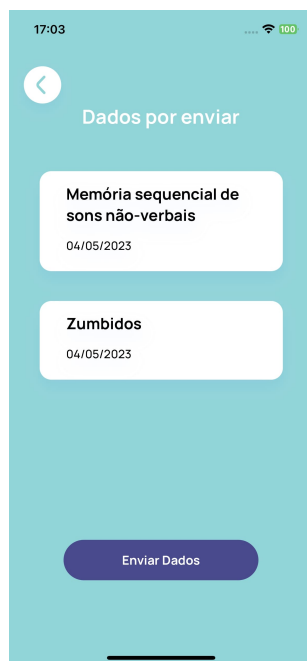


Figura 81: Continuação do ecrã de processamento de áudio.

5.3.3.2.4 Testes por enviar

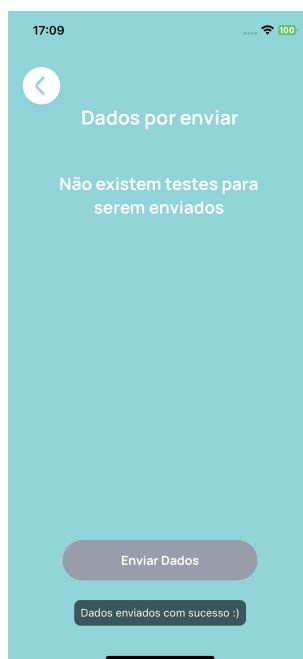
Este botão apresenta um valor que é apresentado a seguir ao texto “Testes por enviar” que é dinamicamente atualizado consoante o número de testes que existem para serem enviados. O utilizador poderá verificar neste ecrã se existem testes pendentes de envio para o servidor (ver Figura 82). Caso existam testes pendentes, o utilizador poderá tenta enviar todos os testes para o servidor ao clicar no botão “Enviar Dados”.



(a) Testes por enviar



(b) Atualização do número de testes por enviar



(c) Todos os testes enviados com sucesso

Figura 82: Ecrã de testes por enviar.

5.3.3.2.5 Definições

O utilizador poderá aceder às definições da aplicação que incluem o acesso aos dados do seu perfil, aos termos e condições do uso da aplicação, página de contacto e a realização do *logout* da conta.

5.3.4 Definições

Como referido anteriormente a aba das definições (ver Figura 83) permite ao utilizador aceder a diversas opções, sendo estas:

- Perfil;
- Termos e condições;
- Política de privacidade;
- Contactos;
- Partilhar *Hear App*;
- *Logout*.



Figura 83: Página de definições

5.3.4.1 *Perfil*

Após o clique nesta opção, o utilizador será reencaminhado para outro ecrã onde se encontram apresentadas outras opções associadas ao perfil.

5.3.4.2 *Termos e condições*

O utilizador poderá visualizar os termos e condições (ver Figura 84) relativos à aplicação. Atualmente, esta secção encontra-se preenchida com dados *dummy*, sendo posteriormente substituída pelos dados corretos.

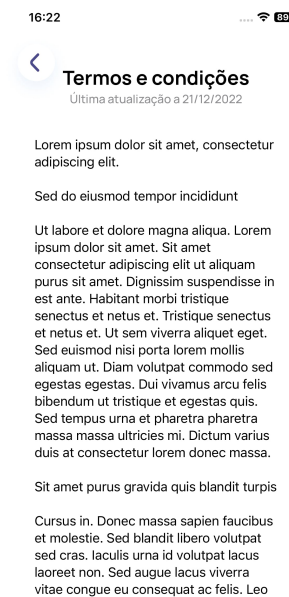


Figura 84: Página de termos e condições

5.3.4.3 *Política de privacidade*

O utilizador poderá visualizar a política de privacidade (ver Figura 85) da aplicação. Será reencaminhado para uma página *Web* com todas as informações.

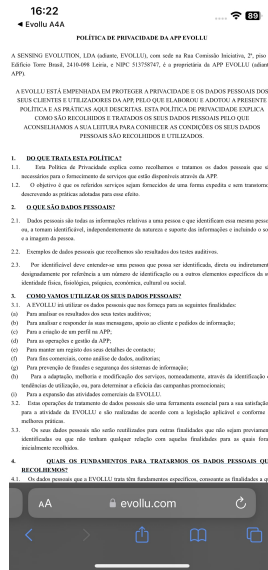
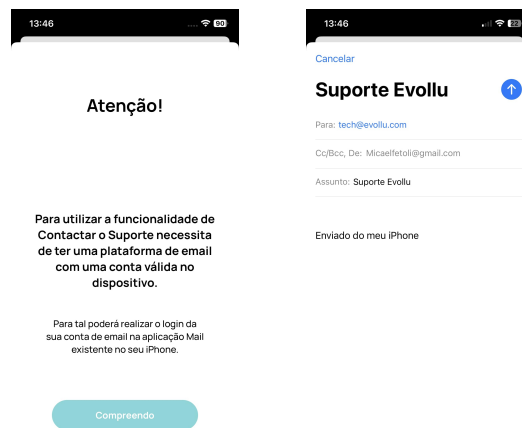


Figura 85: Página de política de privacidade

5.3.4.4 Contactos

Caso o utilizador utilize uma plataforma de *email* com uma conta válida no seu dispositivo, será aberta uma *view* para o contacto com o suporte da *Evollu* (ver Figura 86). Trata-se do *Mail view* presente no *Swift* e pode ser personalizado com o *email default* para o envio da mensagem, o assunto e a mensagem da mesma.



- (a) Não existe conta de *Email* válida no dispositivo
- (b) Conta de *Email* válida no dispositivo, opção de enviar *Email*

Figura 86: Enviar *email* para o suporte *Evollu*

5.3.4.5 Partilhar a Hear App

O utilizador poderá partilhar a ligação com outros utilizadores para que estes consigam instalar a aplicação e utilizar a mesma (ver Figura 87). Para a instalação da aplicação terão de instalar a aplicação *Testflight* (Secção 4.4.2 *TestFlight*).

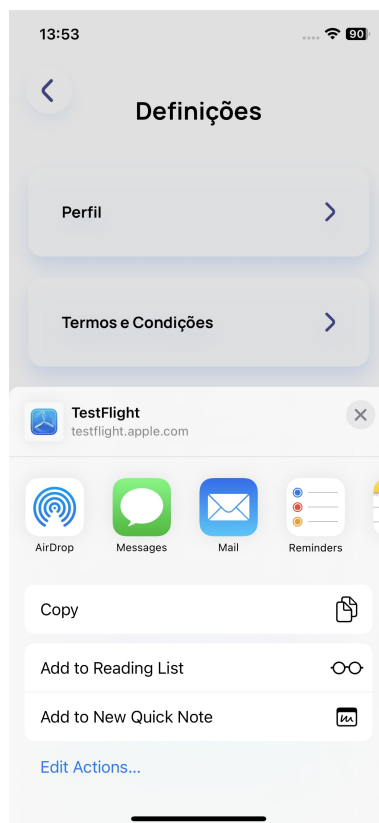


Figura 87: Partilhar a aplicação *Hear App*

5.3.4.6 Logout

Ao clicar no botão *Logout*, o utilizador será notificado se realmente deseja realizar o *logout* à sua conta e poderá confirmar essa intenção ao confirmar a notificação (ver Figura 88). De referir, que se existirem testes por enviar presentes na aplicação, o utilizador será notificado da presença de testes por enviar e que se realizar o *logout* esses testes serão perdidos.



Figura 88: Notificação para confirmar *logout*

5.3.5 Perfil

Esta secção permite ao utilizador (ver Figura 89):

- Visualizar o nome da sua conta e *email*;
- Editar perfil;
- Alterar palavra-passe;
- Notificações;
- Apagar conta.

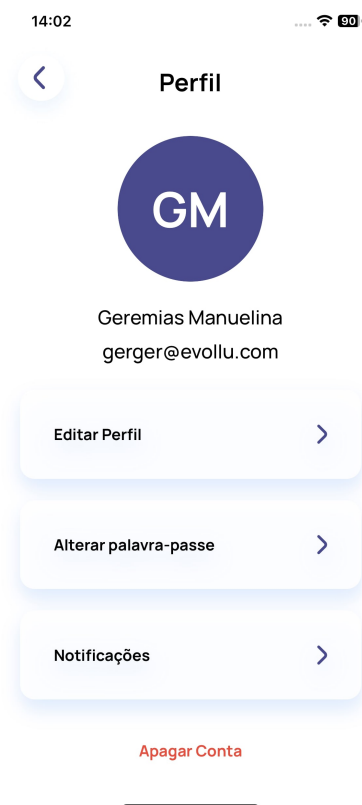


Figura 89: Página de perfil

5.3.5.1 *Editar o perfil*

O utilizador tem a liberdade de alterar alguns dos seus dados pessoais associados à sua conta, são estes (ver Figura 90):

- Nome;
- Data nascimento;
- Género;
- Número indicativo do país;
- Número de telemóvel.

14:02

.... 90

<

Editar Perfil

GM

Nome

Geremias Manuelina

Data de Nascimento

A sua data: 30 Dec 1977

Género

Não Especificado

Email

gerger@evollu.com

O email não pode ser editado na app

Telemóvel

+351 999455786

Atualizar

Figura 90: Página de editar os dados do utilizador

Por questões de segurança e otimização não é permitido a alteração do *email* através da aplicação.

Após o clique no botão “Atualizar” o utilizador será notificado sobre o sucesso dessa alteração.

5.3.5.2 *Alterar a palavra-passe*

Para conseguir alterar a palavra-passe quando o utilizador já se encontra com o *Login* efetuado, terá de indicar a palavra-passe atual e definir uma nova que seja coerente com a confirmação de palavra-passe (ver Figura 91). As regras estipuladas na Secção 5.3.2.3 (Página de registo de nova conta) são igualmente válidas e obrigatórias para este caso.



Figura 91: Alterar a palavra-passe

Após o clique no botão “Confirmar“ o utilizador será notificado sobre o sucesso dessa alteração.

5.3.5.3 Notificações

Esta secção encontra-se em *standby*, sendo que se encontra atualmente em avaliação com a *Chief technology (or technical) officer (CTO)* da *Evolvu*. O principal intuito da presença deste ecrã (ver Figura 92), é permitir a alteração das notificações que o utilizador deseja receber. Existiria assim a possibilidade de desativar o recebimento de todas as notificações, ou apenas das não pretendidas.

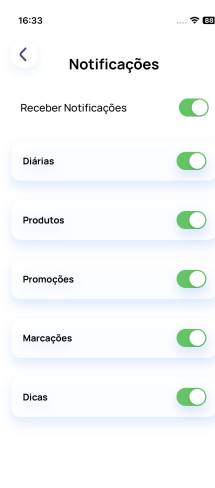


Figura 92: Modificar as notificações

5.3.5.4 Apagar a conta

O utilizador necessita de confirmar que pretende realmente apagar a sua conta e deve indicar uma razão para essa escolha. Após o clique no botão “Apagar conta“ o utilizador será notificado sobre o sucesso da eliminação (ver Figura 93).

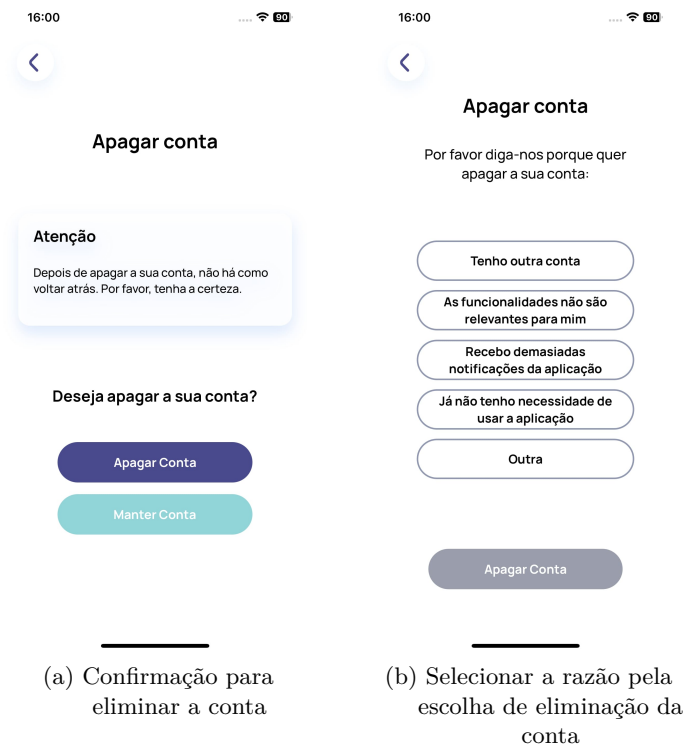


Figura 93: Eliminar a conta

5.3.6 Teste PAC

De seguida será apresentado o *flow* do teste PAC e dos subtestes associados.

5.3.6.1 Splash Screen

Após o utilizador escolher o teste PAC será mostrada uma *Splash Screen* (ver Figura 94) para reencaminhar o utilizador para a inicialização do teste.



Figura 94: *SplashScreen PAC*

5.3.6.2 *Página inicial*

A página inicial apresenta os 4 subtestes que se encontram disponíveis para realização por parte do utilizador. De igual forma, encontra-se presente a informação relativa ao desbloqueio da recomendação, sendo que a mesma apenas é desbloqueada com a realização de 4 subtestes diferentes e válidos (a diferença entre a data atual e a data de realização do teste é inferior a 30 dias).

A realização de um teste irá atualizar esta página inicial, colocando um “certo“ no subteste realizado e incrementar o número de subtestes efetuados (ver Figura 95). Esta página é atualizada igualmente com recurso ao servidor, sendo que é necessário realizar um pedido *get* para obter os subtestes que o utilizador já realizou e que se encontram válidos, pois se o utilizador encerrar a aplicação e voltar a iniciar a mesma, é necessário apresentar o “histórico“ dos subtestes que ainda se encontram válidos.

Este pedido ao servidor apenas é realizado nesta página inicial, no entanto, é necessário ir guardando e filtrando se o utilizador realizou 4 subtestes diferentes, pois caso o utilizador realize os 4 subtestes diferentes sem ter voltado à página inicial, terá de ser igualmente desbloqueada uma recomendação.

Assim, foram implementados métodos para verificar os parâmetros anteriormente descritos e para o *store* local de dados. Esta decisão deveu-se a não ser possível

contar apenas com a informação do servidor (seria necessário realizar mais pedidos, o que tornaria o processo mais penoso computacionalmente e dependente de ligação à *internet*).

Após a realização dos 4 subtestes diferentes, a realização de outro subteste irá realizar um *reset* ao grupo de subtestes realizados.

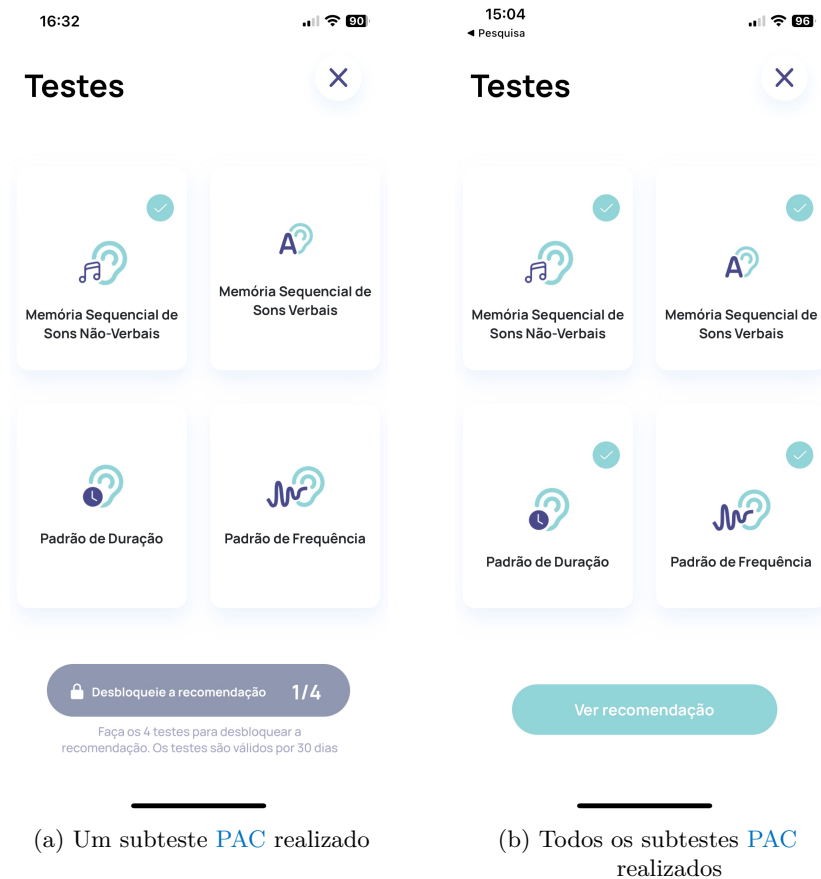


Figura 95: Página inicial do PAC

5.3.6.3 Ecrã informação do subteste

Após o utilizador escolher um subteste, será apresentada uma página com uma descrição relativa ao mesmo (ver Figura 96). O utilizador poderá clicar no botão “Começar” para avançar ou no botão “?” para visualizar as instruções relativas ao subteste.

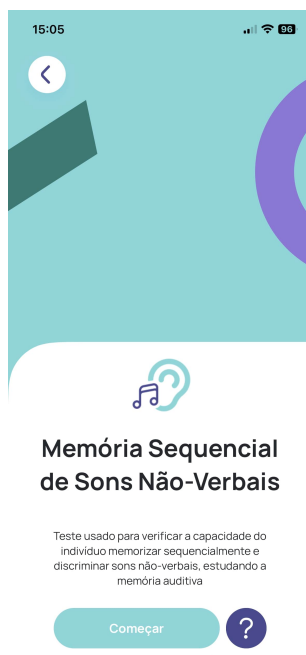


Figura 96: Descrição do subteste Memória Sequencial de sons não-verbais

5.3.6.4 Instruções do subteste

Cada subteste apresenta uma secção de instruções (ver Figura 97). O utilizador poderá visualizar todas as páginas de instruções e prosseguir para a página seguinte.



Figura 97: Ecrãs de instrução

5.3.6.5 *Opções do subteste*

A realização do subteste é alterada consoante as opções escolhidas pelo utilizador (ver Figura 98). O subteste “Memória sequencial de sons não-verbais” e o subteste “Memória sequencial de sons verbais” apenas permitem a escolha do dispositivo de *output* a ser utilizado (colunas ou auscultadores) e o nível de dificuldade (3 ou 4 sons). Já nos outros dois subtestes, “Padrão de duração” e “Padrão de frequência”, caso seja escolhido o *output* de auscultadores, é possível escolher entre testar os ouvidos em simultâneo ou separadamente.

De referir, que a escolha de *output* de auscultadores apenas permite avançar, caso se encontrem auscultadores conectados. De igual forma, não é permitido tentar realizar o subteste com *output* de colunas, caso os auscultadores se encontrarem conectados (ver Figura 99). O utilizador apenas poderá avançar, caso escolha uma opção, o que ativa a opção de clique no botão “Continuar”.

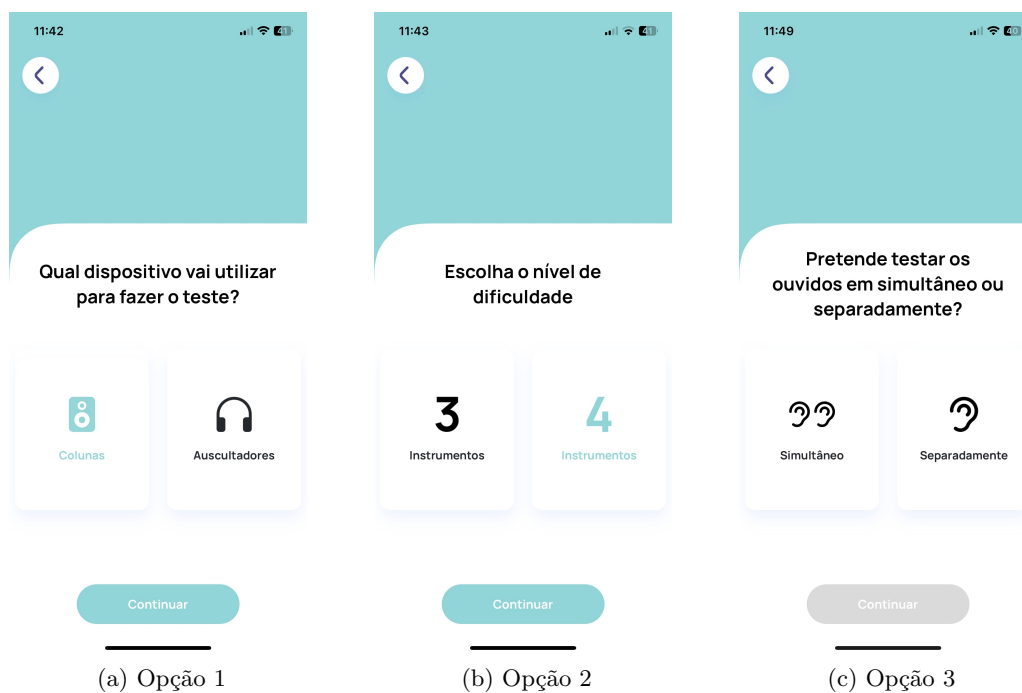
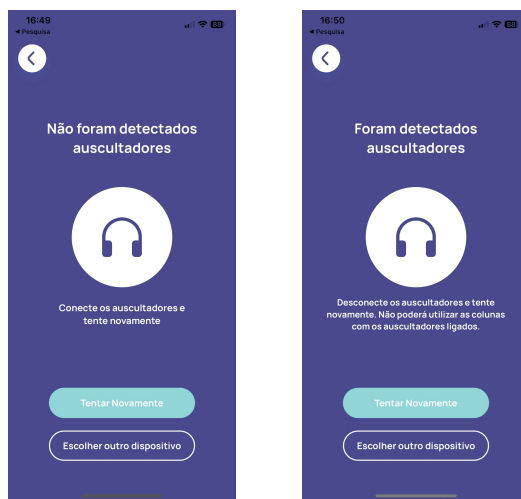


Figura 98: Opções dos subtestes do PAC



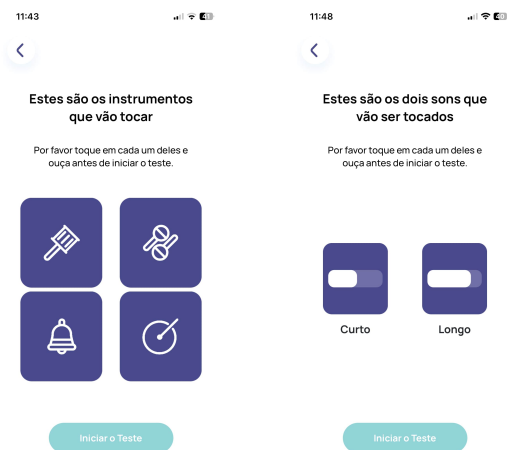
(a) Auscultadores não detetados quando escolhido opção “Auscultadores”

(b) Auscultadores detetados quando escolhido opção colunas

Figura 99: Auscultadores dos subtestes do PAC

5.3.6.6 *Ouvir previamente os sons*

Antes de iniciar o subteste, o utilizador terá de ouvir os sons que serão tocados ao longo do teste (ver Figura 100), de modo a conseguir posteriormente associar os mesmos.



(a) Memória Sons Não-Verbais

(b) Padrão Duração

Figura 100: Toque prévio dos sons

5.3.6.7 *Splash screen* pr vio ao in cio da sequ ncia

Previamente ao in cio de uma sequ ncia   mostrado um *Splash Screen* ao utilizador (ver Figura 101). Este serve para informar o utilizador para se preparar e para realizar a l gica necess ria para a execu o do teste.

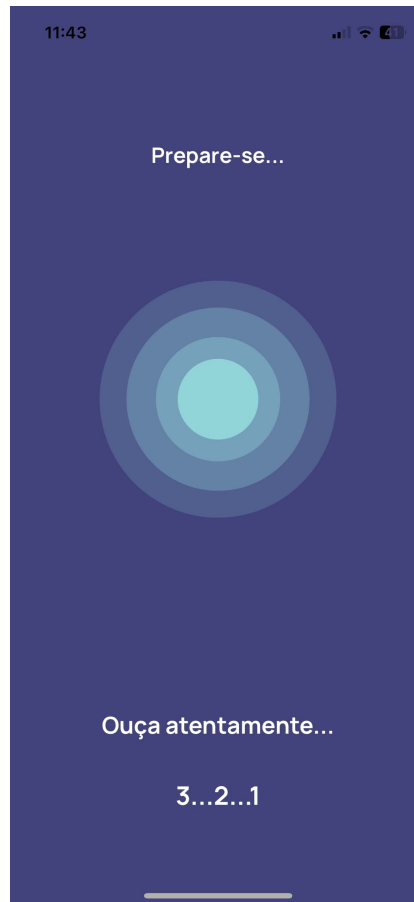


Figura 101: Ecr  antes de iniciar a sequ ncia

5.3.6.8 *Play da sequ ncia*

Ser o tocados 3 ou 4 sons dependendo da anterior escolha do utilizador (ver Figura 102). Os sons ser o tocados com um intervalo de 0.3 segundos entre si.

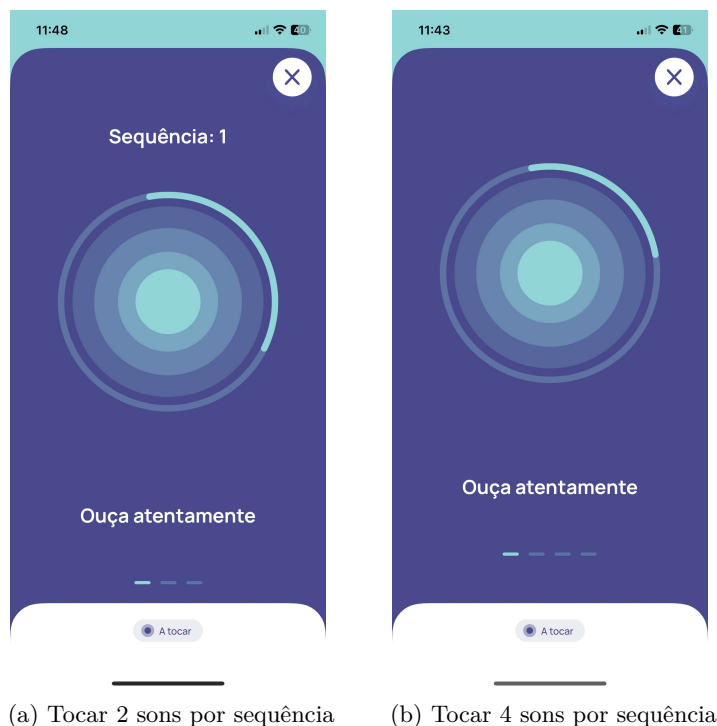


Figura 102: Toque dos sons da sequência

5.3.6.9 Escolha dos sons tocados

Após o fim da sequência, o utilizador terá de escolher os sons que ouviu pela ordem em que foram tocados (ver Figura 103). Caso os sons estejam corretos, mas a ordem selecionada não seja coerente com a ordem em que foram tocados é tido como errado. Caso tenham sido tocadas todas as sequências (3 para os subtestes “Memória sequencial de sons não-verbais“ e “Memória sequencial de sons verbais“ e 10 para o “Padrão de duração“ e “Padrão de frequência“) o utilizador ao clicar no botão “Continuar“ vai ser reencaminhado para o próximo ecrã.

De referir que serão tocadas 20 sequências, na escolha da opção “tocar em cada ouvido separadamente“ (apenas disponível nos testes de padrão).

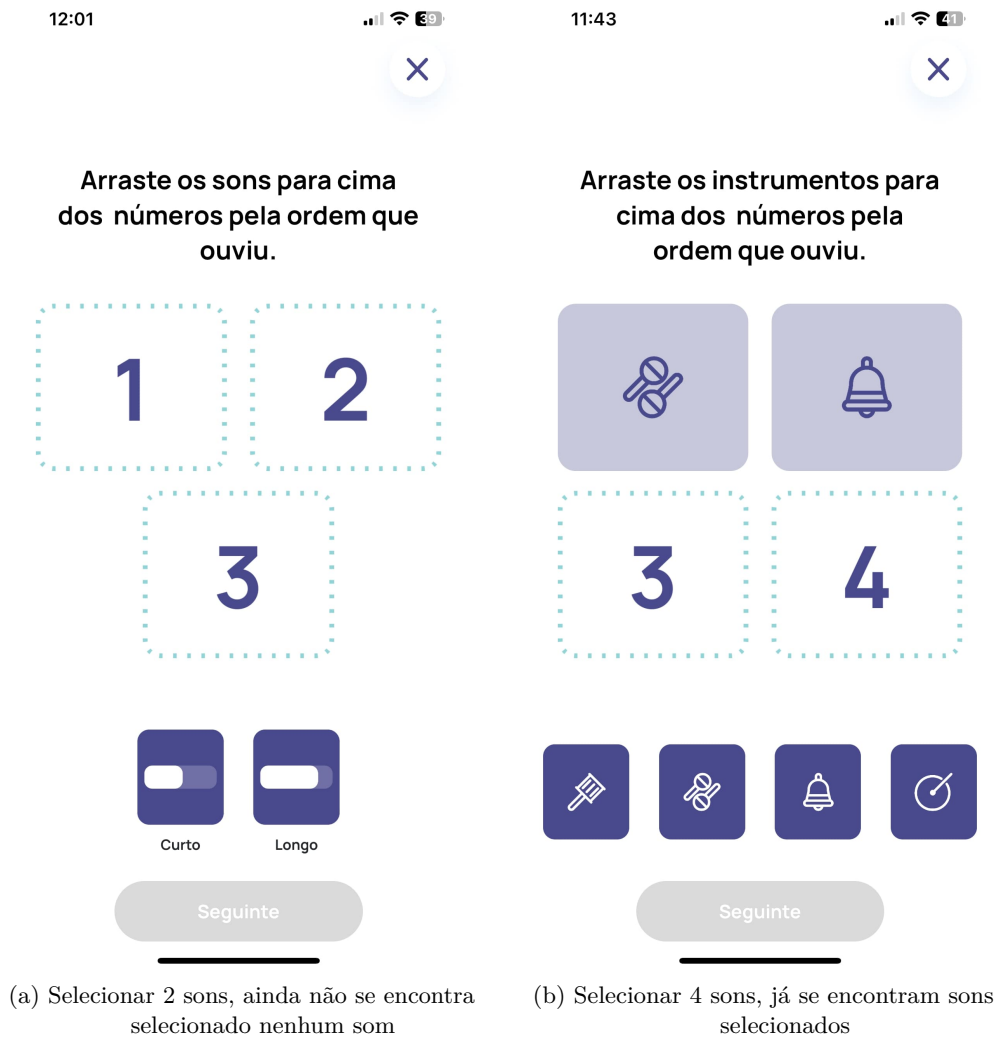
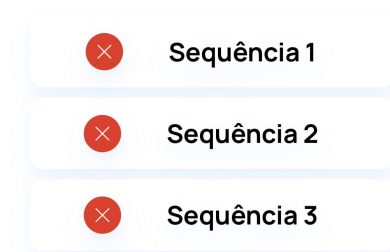


Figura 103: Selecionar os sons da sequência

5.3.6.10 Resultados

Serão apresentados os resultados (ver Figura 104), nomeadamente quais as sequências que o utilizador acertou.

Resultados



Acertou 0 de 3 sequências!



Figura 104: Resultados do subteste

5.3.6.11 *Escolher o próximo subteste*

Será apresentado ao utilizador o próximo subteste que se encontra disponível para realizar, sendo que o mesmo poderá decidir realizar o teste ou voltar à página inicial do PAC (ver Figura 105). Foi implementado um algoritmo que permite filtrar qual o próximo subteste que ainda não foi realizado e mostrá-lo ao utilizador, independentemente do utilizador ter começado pelo teste número 1, 2, 3 ou 4 e já ter realizado 1, 2, 3 ou 4 subtestes diferentes. Será realizado um pedido *Post* neste ecrã, para o envio do resultado do teste para o servidor.



Figura 105: Avançar para o próximo subteste

5.3.6.12 *Ecrã de desbloqueio de recomendação*

Página de gratificação após a conclusão de todos os subtestes (ver Figura 106).

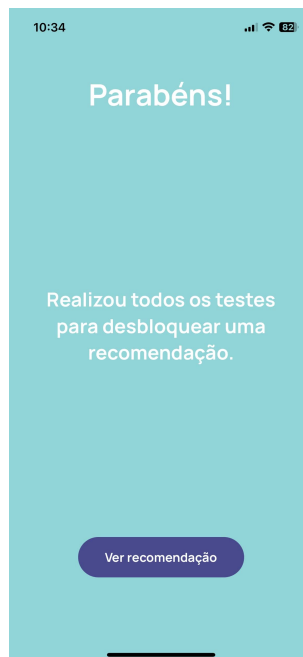


Figura 106: Desbloqueio da recomendação

5.3.6.13 *Ecrã de recomendação*

Página para mostrar a recomendação desbloqueada ao utilizador (ver Figura 107). A recomendação está dependente dos dados provenientes do servidor, sendo que é realizado um pedido *Get* para obter a mesma.



Figura 107: Recomendação do PAC

5.3.7 *Teste Esforço Auditivo*

De seguida será apresentado o *flow* do teste Esforço Auditivo.

5.3.7.1 *Splash Screen*

Ecrã (ver Figura 108) equivalente ao teste anterior, PAC.



Figura 108: *Splash Screen* do teste Esforço Auditivo

5.3.7.2 *Ecrã de informação do teste*

Uma página que apresenta uma breve descrição sobre do que se trata o teste Esforço Auditivo (ver Figura 109).

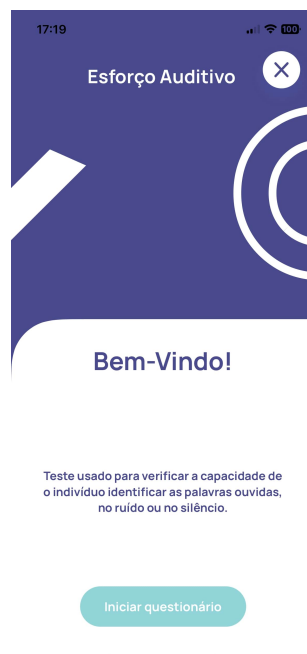


Figura 109: Descrição do teste Esforço Auditivo

5.3.7.3 *Questionário*

O utilizador terá de responder a um questionário com 7 perguntas, sendo que em 4 deverá seleccionar um valor num *slider* e nas restantes 3, seleccionar uma opção (ver Figura 110). De referir que na primeira pergunta “ O quão cansado se sente agora? “, caso seja respondido um valor maior ou igual a 7.0, o utilizador será reencaminhado para um ecrã informativo, que indica ao utilizador que tem um nível excessivo de cansaço para a realização do teste.



Figura 110: Questionário do teste Esforço Auditivo

5.3.7.4 *Teste de som*

Este ecrã permite ao utilizador ajustar o volume para um nível confortável, devido a ser constantemente tocado um som (ver Figura 111).

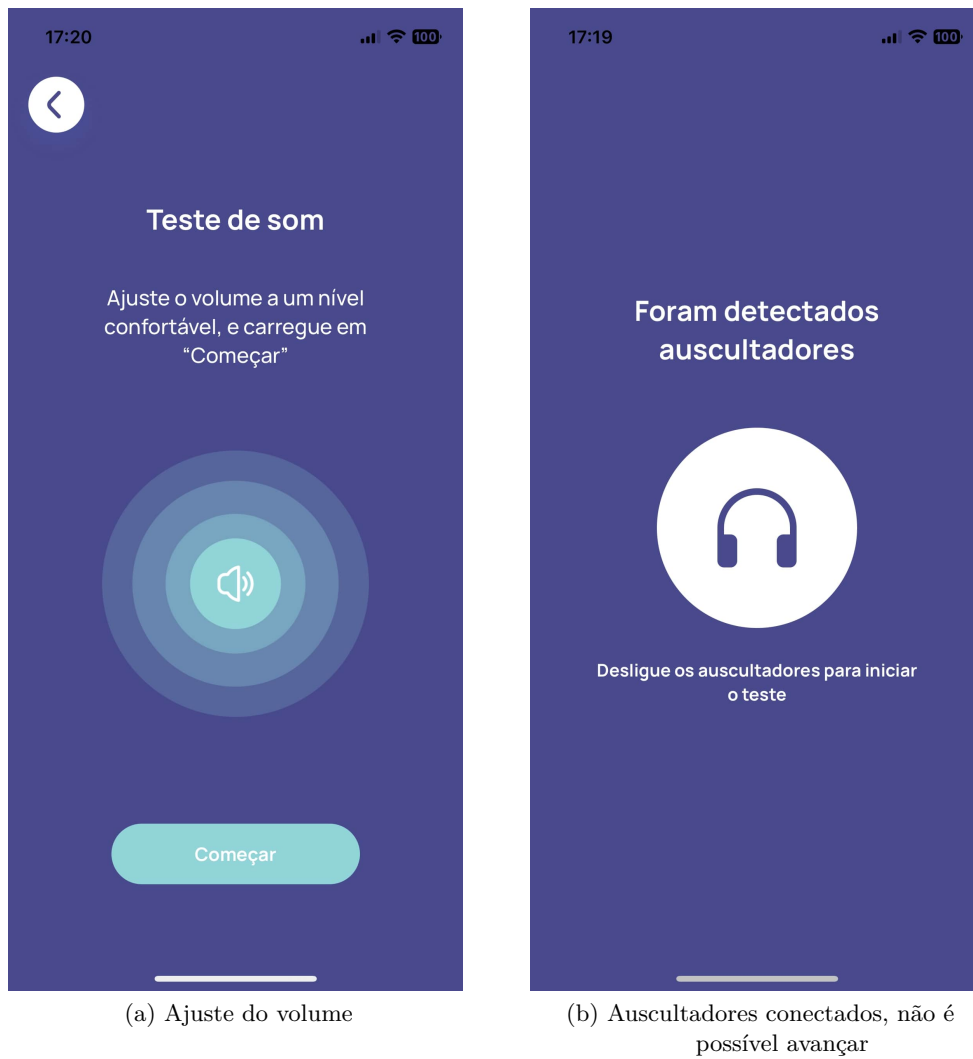


Figura 111: Ajuste de volume do teste Esforço Auditivo

5.3.7.5 Ecrã de informação do subteste

O teste de esforço auditivo encontra-se dividido em dois subtestes, que são realizados de forma sequencial, sendo estes o subteste de “Esforço Auditivo” e “Inteligibilidade“. Antes de ser iniciado um dos subtestes, é apresentado um ecrã com as informações relativas a esse subteste (ver Figura 112). Do mesmo modo, é possível visualizar as instruções do teste ao clicar no botão “ ? “.



Figura 112: Descrições dos subttestes do teste Esforço Auditivo

5.3.7.6 Instruções do subtteste

O utilizador pode visualizar as instruções do subtteste (ver Figura 113).



Figura 113: Instruções dos subttestes do teste Esforço Auditivo

5.3.7.7 *Splash Screen* prévio ao início da sequência

Ecrã equivalente ao teste PAC (ver Figura 101).

5.3.7.8 *Play* da sequência

Ecrã (ver Figuras 114 e 115) equivalente ao teste PAC. No subteste “Esforço Auditivo“, ao contrário do teste PAC, não são mostrados os traços relativamente a qual som está atualmente a ser tocado. Existe um intervalo de 5,6,9,10,12 sons a serem tocados e esse número é escolhido de forma aleatória. Já no subtteste “Inteligibilidade“ é apenas tocado um som a cada sequência.



Figura 114: Toque dos sons nos subttestes do teste Esforço Auditivo



Figura 115: Continuação do toque dos sons nos subtestes do teste Esforço Auditivo

5.3.7.9 Seleção dos sons

No caso do subteste “Esforço Auditivo“ o utilizador terá de seleccionar (entre 9 palavras) as 3 últimas palavras que ouviu (ver Figura 116), sendo que apenas é avaliado se o utilizador conseguiu seleccionar a antepenúltima palavra tocada. A ordem seleccionada não é relevante para a avaliação.

Já no teste de “Inteligibilidade“ o utilizador terá de escolher, entre 4 palavras, qual a palavra que ouviu.

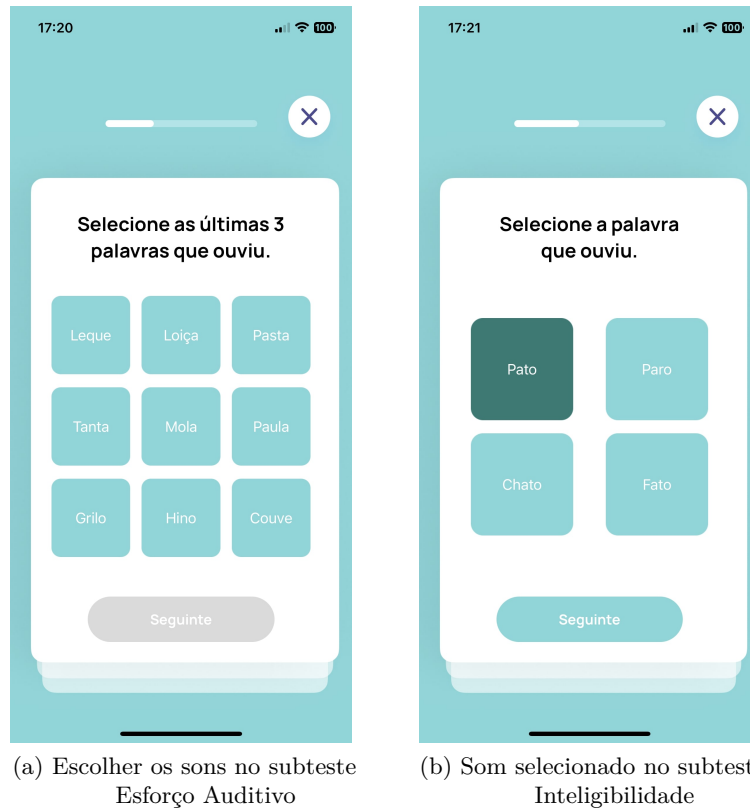


Figura 116: Escolher os sons nos subtestes do teste Esforço Auditivo

5.3.7.10 *Página de resultados*

Após o término de todas as sequências do subteste, será apresentada uma página com um gráfico que permite mostrar ao utilizador quantas sequências acertou (ver Figura 117). Este gráfico irá apresentar uma cor verde, caso seja obtido um resultado “positivo/dentro da média” ou vermelho caso seja obtido um resultado “negativo/abaixo da média“. Será realizado um pedido *Post* para o envio do resultado de ambos os subtestes para o servidor.



Figura 117: Resultado dos subtestes do teste Esforço Auditivo

5.3.7.11 *Ecrã de recomendação*

Página para mostrar a recomendação desbloqueada ao utilizador (ver Figura 118). A recomendação mostrada ao utilizador está dependente dos dados provenientes do servidor, sendo que é realizado um pedido *Get* para obter essa recomendação.



Figura 118: Recomendação do teste Esforço Auditivo

5.3.8 Teste Zumbidos/Tinnitus

De seguida será apresentado o *flow* do teste Zumbidos.

5.3.8.1 Splash Screen

Ecrã (ver Figura 119) equivalente aos outros testes PAC e Esforço Auditivo.



Figura 119: *SplashScreen* do teste Zumbidos

5.3.8.2 *Ecrã de informação do teste*

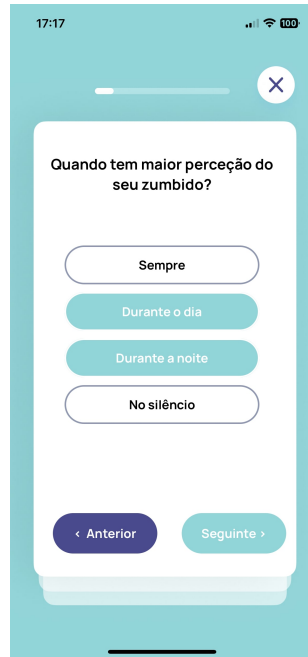
Uma página que apresenta uma breve descrição sobre do que trata o teste Zumbidos (ver Figura 120).



Figura 120: Descrição do teste Zumbidos

5.3.8.3 *Questionário*

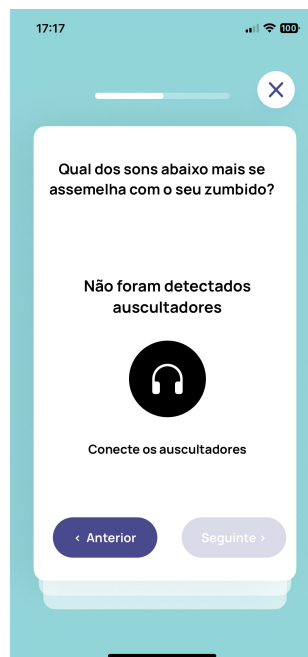
O teste Zumbidos foca-se maioritariamente num questionário com diversas opções de resposta. O utilizador terá de responder a um questionário com 15 perguntas, sendo que este encontra-se dividido em perguntas de resposta múltipla (de única resposta ou várias) e *sliders* (ver Figura 121). De referir, que na primeira pergunta “Qual a localização do seu zumbido?“, caso seja respondido “Ouvido Esquerdo“ ou “Ouvido Direito“, o utilizador será reencaminhado para um ecrã informativo que indica a necessidade de consultar um médico [Otorrinolaringologista \(ORL\)](#), uma vez que o ruído é apenas num ouvido.



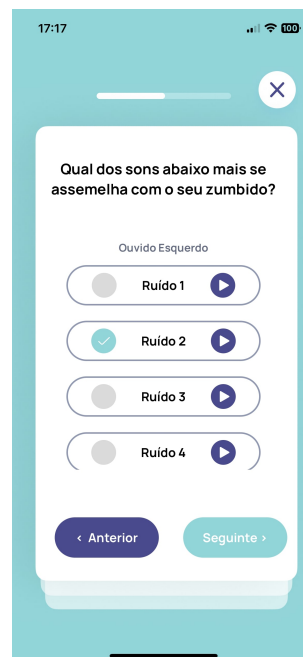
(a) Pergunta 4, resposta múltipla



(b) Pergunta 7, slider



(c) Pergunta 8, sem auscultadores conectados



(d) Pergunta 8, auscultadores conectados

Figura 121: Questionário do teste Zumbidos

5.3.8.4 *Ajuste Grosseiro*

Após a escolha do ruído, o utilizador pode alterar dinamicamente a frequência e a intensidade desse mesmo ruído através do ajuste grosseiro e fino. No ajuste grosseiro é possível alterar esses valores com menor nível de detalhe, fazendo jus ao nome grosseiro (ver Figura 122). De referir, que na última pergunta do questionário, é mencionado que será inicialmente tocado o ruído no ouvido contrário e apenas no ajuste fino é que será tocado no ouvido selecionado. O utilizador escolhe dois ruídos no questionário, mas apenas será tocado o ruído associado ao ouvido que o utilizador escolheu na última pergunta do questionário.



Figura 122: Ajuste Grosseiro do teste Zumbidos

5.3.8.5 *Ajuste Fino*

Este ecrã consiste na transcrição das definições que foram efetuadas ao ruído no ecrã de ajuste grosseiro, para um novo ecrã onde será realizado o ajuste fino, tendo em conta as definições previamente aplicadas (ver Figura 123).

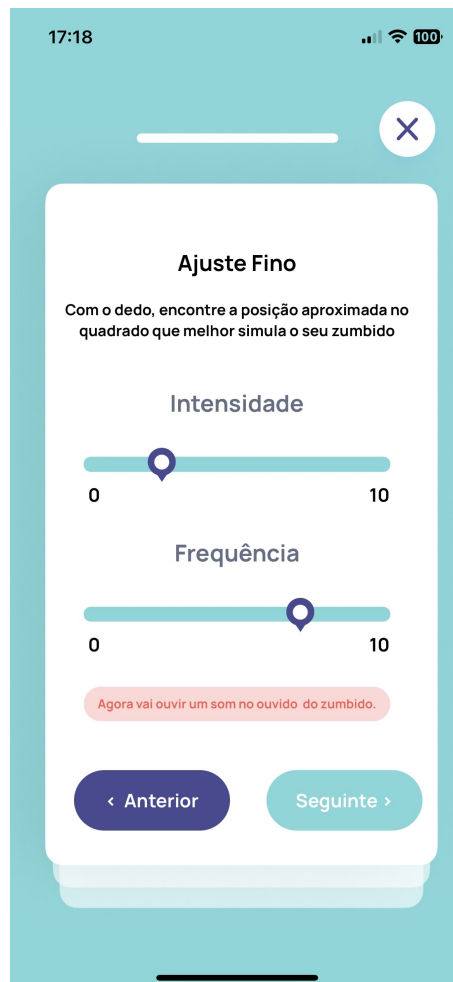
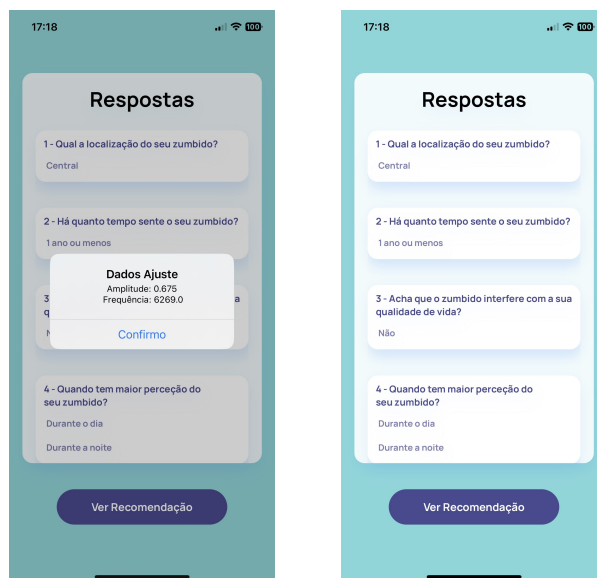


Figura 123: Ajuste Fino do teste Zumbidos

5.3.8.6 *Resultados*

São apresentadas as respostas do utilizador ao questionário, assim como uma *alert* que apresenta os dados da frequência e intensidade do ruído definidos pelo utilizador (Figura 124). Esta *alert* deve-se exclusivamente à ESTeSC-IPC necessitar de analisar esses dados antes de estes serem enviados e encontrarem-se disponíveis na plataforma *Web*.



(a) Alert com os resultados do ajuste anterior

(b) Respostas do questionário do teste Zumbidos

Figura 124: Resultados do teste Zumbidos

5.3.8.7 Ecrã de recomendação

Página para mostrar a recomendação desbloqueada ao utilizador (ver Figura 125). A recomendação mostrada ao utilizador está dependente dos dados provenientes do servidor, sendo que é realizado um pedido *Get* para obter essa recomendação.

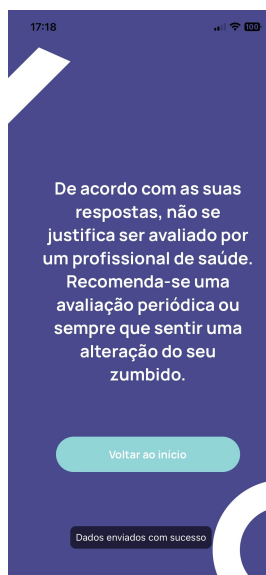


Figura 125: Recomendação do teste Zumbidos

5.3.9 *Teste Tonal*

De seguida será apresentado o *flow* do teste Tonal.

5.3.9.1 *Splash Screen*

Ecrã (ver Figura 126) equivalente aos outros testes PAC, Esforço Auditivo e Zumbidos.



Figura 126: *SplashScreen* do teste Tonal

5.3.9.2 *Ecrã de informação do teste*

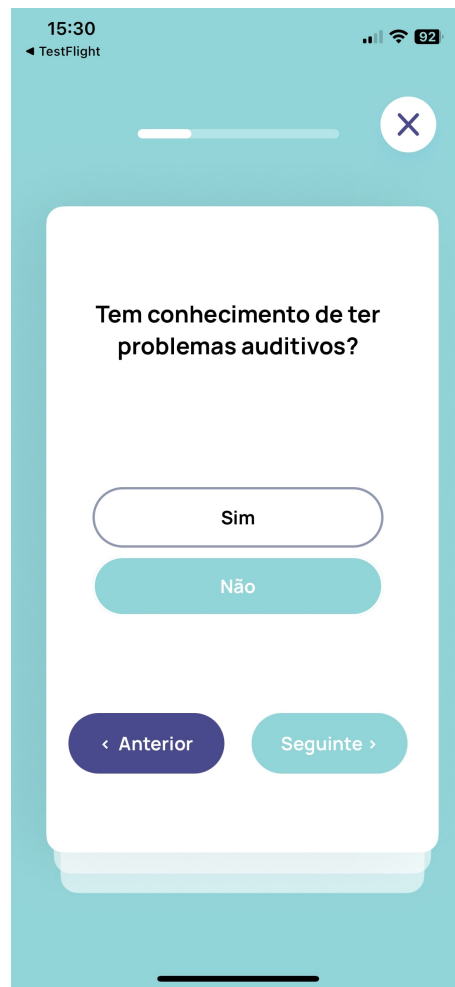
Uma página que apresenta uma breve descrição sobre do que trata o teste Tonal (ver Figura 127).



Figura 127: Descrição do teste Tonal

5.3.9.3 *Questionário*

O teste Tonal apresenta um mini questionário, constituído por 4 perguntas, apenas com opções de única resposta (ver Figura 128).



(a) Pergunta 1

Figura 128: Questionário do teste Tonal

5.3.9.4 Seleção de dispositivo

O teste tem de ser realizado com a utilização de auscultadores e o utilizador terá de escolher qual o tipo de auscultadores que vai utilizar. De referir, que a escolha de algum tipo de auscultadores apenas permite avançar, caso se encontrem auscultadores conectados (ver Figura 129).

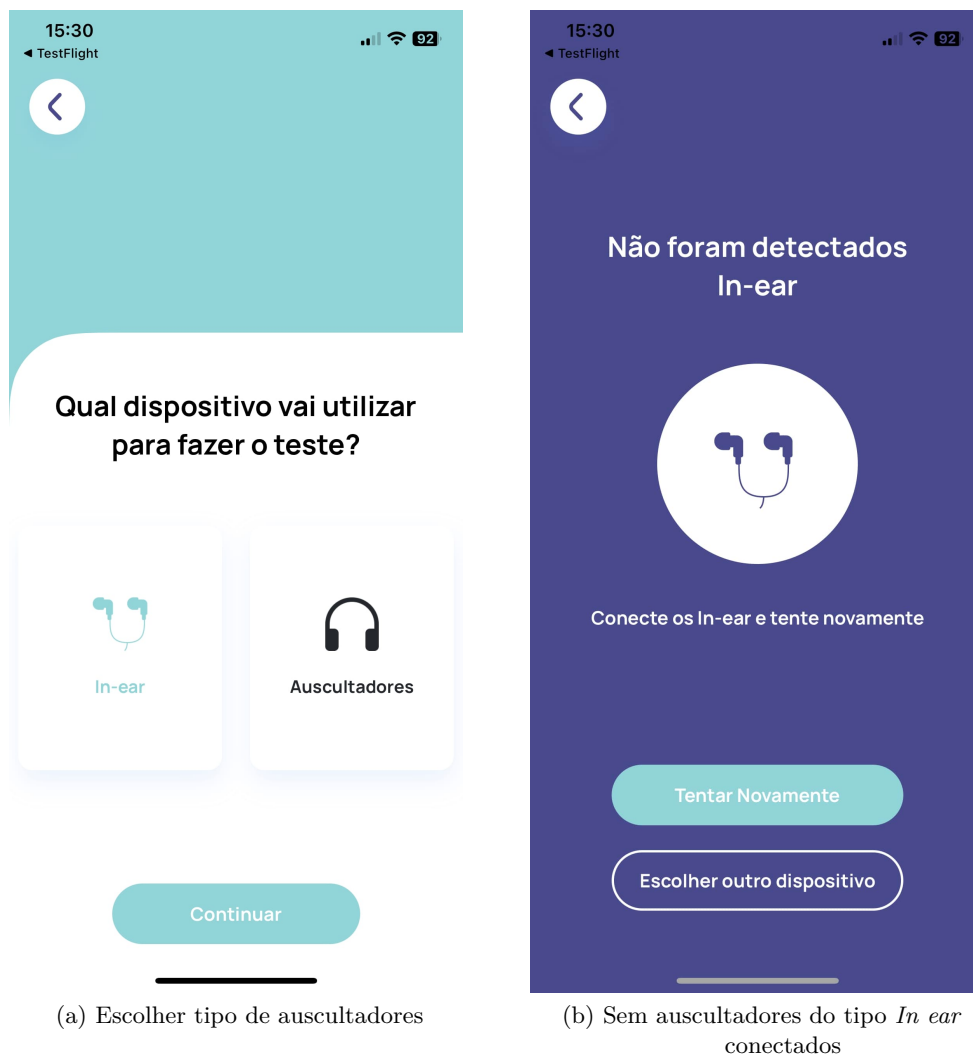


Figura 129: Escolher tipo de auscultador no teste Tonal

5.3.9.5 Instruções

São apresentadas instruções previamente ao início do teste (ver Figura 130). Estas são narradas com recurso ao *play* de sons.

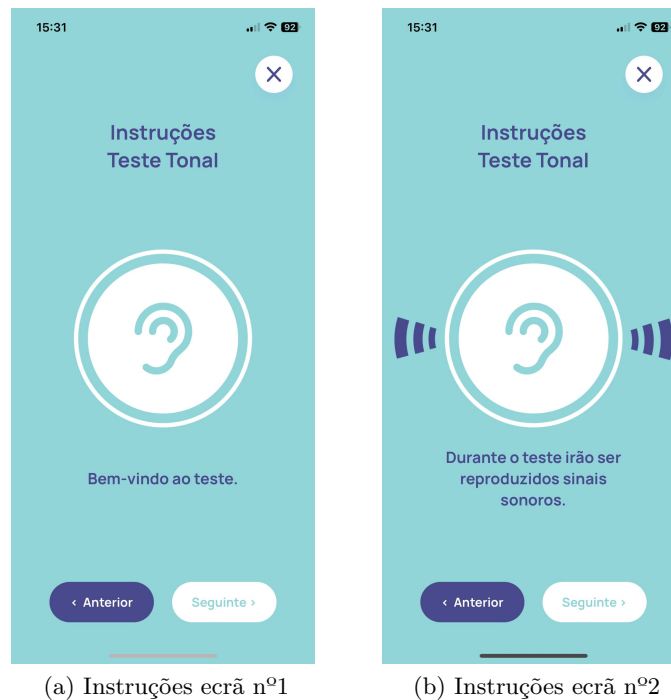


Figura 130: Instruções do teste Tonal

5.3.9.6 *Volume colocado no máximo*

O uso da aplicação *Evolllu* (Secção 2.2.3 *Evolllu App*) permitiu verificar que a mesma se encontra muito inconsistente em relação aos requisitos de manter o volume a 100% e a utilização obrigatória de auscultadores. Por vezes, ao tentar colocar o volume a 100%, a aplicação volta para a página inicial (comportamento errado) e mudar o volume na página de seleção de dispositivo não altera o comportamento da aplicação (sendo, por vezes, possível diminuir o volume antes do teste começar). Se diminuir o volume novamente enquanto o teste se encontra a decorrer, será então despoletado o estado de “nível de volume inválido“ e o teste é terminado.

Perante estas inconsistências e problemas, o objetivo seria:

- Antes do teste começar: O utilizador tem de colocar o volume no máximo para conseguir ter acesso à página onde pode escolher o dispositivo que está a utilizar.
- Teste a decorrer: Se diminuir o volume com um clique, vai ser despoletada uma *alert*/notificação para o utilizador aumentar o som para os 100%, mas o teste não termina. Caso o utilizador volte a clicar no botão para reduzir o som, vai ser terminado o teste.

Essas inconsistências foram tidas em conta no desenvolvimento do teste Tonal, de modo a conseguir apresentar o objetivo anteriormente referido. Deste modo, para ser possível avançar é necessário colocar o volume no máximo e assim ativar a ação de clique no botão “Compreendo“ (ver Figura 131).

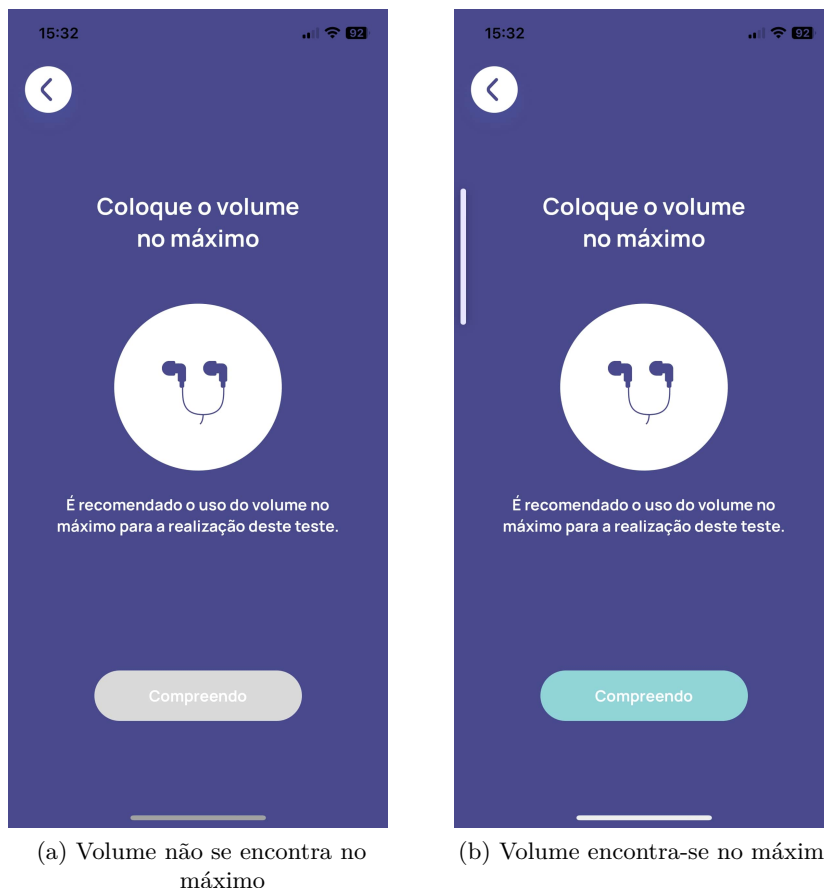


Figura 131: Colocar volume no máximo

5.3.9.7 *Splash Screen* prévio ao início da sequência

Ecrã equivalente ao teste [PAC](#) e Esforço Auditivo.

5.3.9.8 *Play do teste*

O utilizador consegue visualizar qual é o ouvido em que se encontram atualmente a serem tocados os sons, a percentagem de sequências que já foram tocadas e é mostrado um aviso se o volume não se encontrar no máximo (ver Figura 132).



Figura 132: Decorrer do teste Tonal

5.3.9.9 Escolher o ecrã de resultados

Após o término do teste, o utilizador pode visualizar 3 ecrãs de resultados, sendo que os poderá escolher através da página Escolher resultado (ver Figura 133).

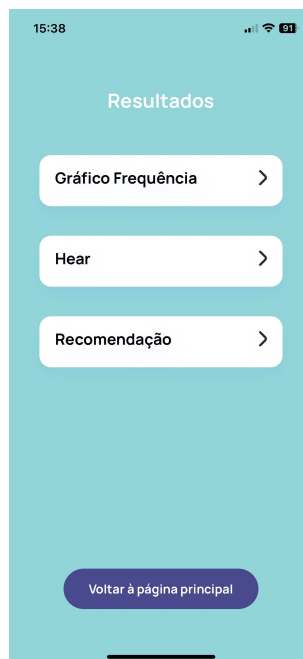


Figura 133: Ecrã de resultados do teste Tonal

De referir, que os próximos ecrãs das Secções 5.3.9.9.1, 5.3.9.9.2 e 5.3.9.9.3 não foram terminados, e possuem apenas informação estática.

5.3.9.9.1 Resultado *Hear*

Permite dar a conhecer a existência de problemas auditivos com um ouvido (ver Figura 134).

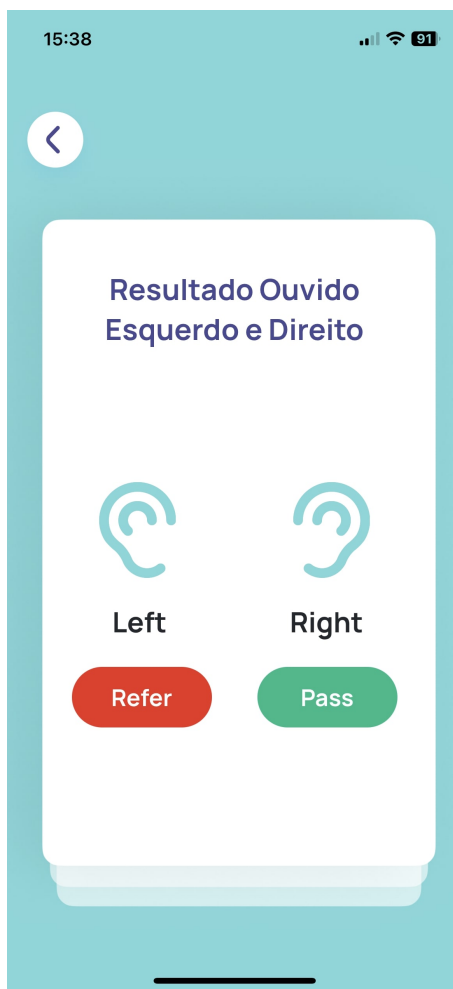


Figura 134: Resultados para cada ouvido

5.3.9.9.2 Resultado gráfico

Permite, graficamente, dar a conhecer a existência de perda auditiva (ver Figura 135).

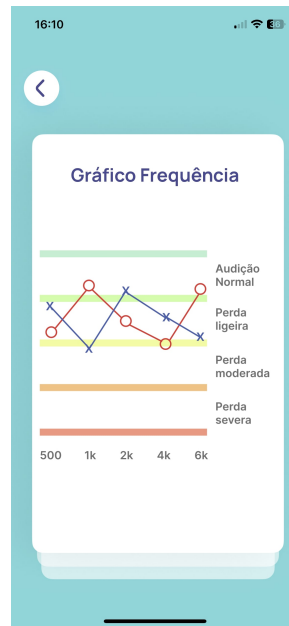


Figura 135: Gráfico sobre perda auditiva

5.3.9.9.3 Resultado recomendação

Recomendação após a realização do teste Tonal (ver Figura 136).

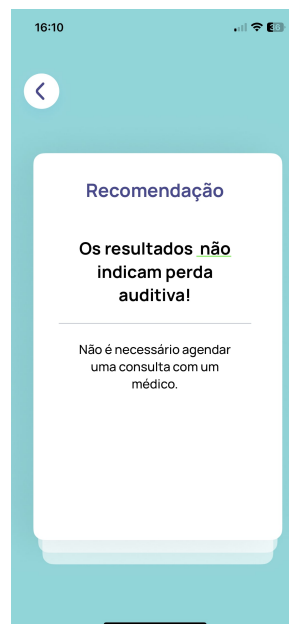


Figura 136: Recomendação do teste Tonal

De referir, que se encontra a ser desenvolvida uma plataforma *Web*, para a visualização de todos os dados provenientes dos testes da aplicação.

CONCLUSÕES

Este capítulo pretende concluir este relatório, ao destacar considerações sobre o desenvolvimento do projeto (Secção 6.1), melhorias (Secção 6.2) e trabalho futuro (Secção 6.3).

6.1 CONSIDERAÇÕES FINAIS

A saúde é um tema muito importante e discutível, pois existem diversos fatores que, direta ou indiretamente, afetam a saúde individual como um todo. A presença de mais e melhores meios tecnológicos, permite dar a conhecer opções que permitem facilitar o acesso a melhores condições de saúde.

O desenvolvimento deste projeto apresenta maioritariamente esse mesmo objetivo, conseguir disponibilizar aos utilizadores uma aplicação capaz de detetar e/ou verificar a presença de problemas auditivos nos utilizadores. A realização desta monitorização do perfil auditivo do utilizador, permite ajudar pessoas com perturbações na audição e democratizar o acesso aos serviços de saúde audiológica.

O desenvolvimento da aplicação consistiu, maioritariamente, no uso do *Xcode* em conjunto com a linguagem *Swift* e *framework SwiftUI*. Foram trabalhados alguns pontos importantes, tais como:

- *CoreData/UserDefaults*: A possibilidade de guardar dados localmente no dispositivo com a interação de *CoreData*, *UserDefaultls*.
- *Packages*: Utilização de *packages* externos e a criação de uma *library* para ser utilizada/integrada noutros projetos em *Swift*, através do *Swift Package Manager*.
- Notificações: Disponibilizar notificações locais para o utilizador perante determinada ação, *Local Notifications*.
- API: Consumir *RESTful* API, de modo a comunicar com o servidor para obter/enviar dados.

- *Threading*: Para permitir lidar com situações de *play* de sons e obter igualmente *input* do utilizador.

No decorrer do projeto, todos os objetivos principais foram cumpridos com sucesso. No entanto, devido ao foco no desenvolvimento de um novo teste e ao término do estágio, não foi possível iniciar o desenvolvimento das funcionalidades de marcar uma consulta na aplicação (ver Secção 6.3 Trabalho futuro, onde se discute o trabalho futuro).

O desenvolvimento em *Swift/SwiftUI* providenciou uma enorme satisfação e gosto, sendo algo que pretendo continuar a trabalhar e a conhecer. De igual forma, a possibilidade de desenvolver aplicações intuitivas, confiáveis e que melhorem a qualidade de vida dos utilizadores é muito gratificante. Deste modo, sinto-me atualmente confiante e entusiasmado para ingressar num projeto que me permita desenvolver as minhas capacidades e aprender coisas novas. Os conhecimentos adquiridos ao longo de todo o percurso académico, foram e são, extremamente importantes para conseguir atingir o objetivo referido.

6.2 MELHORIAS

Um projeto na área de informática, encontra-se sempre suscetível à realização e/ou reestruturação de novas funcionalidades que permitam melhorar de forma significativa a utilização do mesmo. De seguida, são apresentados alguns pontos passíveis de melhoria e/ou reestruturação, de modo a tornar a aplicação mais atualizada e rica em serviços/funcionalidades disponíveis:

- *Login* através de outras plataformas:

Foi inicializado o desenvolvimento do *Login* através de outras plataformas, mas devido a ser necessário interagir com diversos parâmetros definidos no servidor e à falta de tempo e disponibilidade por parte dos elementos da equipa responsáveis por esse requisito, este desenvolvimento foi deixado em *standby*. A sua continuação seria de extrema importância, sendo que é cada vez mais frequente a presença de diversas opções de *login* nas inúmeras aplicações disponíveis para *download* nas plataformas principais.

- Permitir *Login* com *email* validado:

Aquando do registo/criação de conta na aplicação é enviado um *email* de confirmação para o *email* registado. O correto seria apenas validar o *Login* na aplicação após o utilizador ter validado o *email* com a confirmação através

do *email* que recebeu aquando do registo. Devido a ser necessário efetuar alterações ao servidor e à falta de tempo e disponibilidade dos elementos da equipa responsáveis por esse requisito, este desenvolvimento foi deixado em *standby*.

- *Refresh Token* automático:

Como indicado anteriormente, o *token* apresenta uma validade de 24 horas, sendo que após esse período será necessário voltar a realizar o *login*. O objetivo será realizar automaticamente um pedido de *refresh token* (que nos devolva um *token* atualizado), assim que verificada a expiração da validade do *token* atual.

- Novas animações:

De modo a tornar a experiência de utilização da [App](#) o mais agradável possível, seria interessante substituir o *loading* de imagens em determinadas situações (como no *running* dos subtestes [PAC](#)) por animações.

6.3 TRABALHO FUTURO

- Marcação de uma consulta de rastreio:

Após a realização de um determinado teste e a obtenção de uma determinada recomendação, poderá existir a possibilidade de marcar uma consulta de rastreio diretamente na aplicação. O expectável seria obter uma lista com todas as farmácias parceiras e realizar a marcação para uma determinada data e horário.

BIBLIOGRAFIA

- [1] *Escola de Tecnologia da Saúde de Coimbra cria aplicações de rastreio e treino auditivo.* pt. Último acesso 31 de outubro 2022. URL: <https://www.saudemais.tv/noticia/33227-escola-de-tecnologia-da-saude-de-coimbra-cria-aplicacoes-de-rastreio-e-treino-auditivo>.
- [2] *O que é uma audiometria e quem/quando deve fazer?* pt. Último acesso 26 de junho 2023. Set. de 2022. URL: <https://www.minisom.pt/noticias/o-que-e-audiometria-e-quem-quando-deve-fazer>.
- [3] *Reabilitação auditiva | CUF.* pt-pt. Último acesso 10 de dezembro 2022. URL: <https://www.cuf.pt/saude-a-z/reabilitacao-auditiva>.
- [4] Conexa Saúde. *O que é saúde? Conceito, tipos e importância da saúde ocupacional.* pt-BR. Último acesso 6 de dezembro 2022. Ago. de 2019. URL: <https://www.conexasaude.com.br/blog/o-que-e-saude/>.
- [5] Carissa Etienne. «Relatório Mundial da saúde». pt. Em: *The World Health Report* (2010). Último acesso 6 de novembro 2022, pp. 2–3.
- [6] Carla Alessandra Scaranello. «REABILITAÇÃO AUDITIVA PÓS IMPLANTE COCLEAR». pt. Em: *Medicina (Ribeirão Preto)* 38.3/4 (dez. de 2005). Último acesso 6 de dezembro 2022, pp. 273–278. ISSN: 2176-7262, 0076-6046. DOI: [10.11606/issn.2176-7262.v38i3/4p273-278](https://doi.org/10.11606/issn.2176-7262.v38i3/4p273-278).
- [7] *Audiologia - Escola Superior de Saúde.* Último acesso 8 de dezembro 2022. URL: <https://www.ess.ipp.pt/clinica-pedagogica/servicos/audiologia>.
- [8] GoFox- www.gofox.pt. *Audiologia em Portugal.* pt. Último acesso 8 de dezembro 2022. Dez. de 2013. URL: <http://www.apta.org.pt/pt/audiologia-em-portugal>.
- [9] *Deficiência auditiva afeta um milhão em Portugal 2018.* pt. Último acesso 10 de dezembro 2022. Fev. de 2018. URL: <https://www.minisom.pt/noticias/deficiencia-auditiva-afeta-um-milhao-em-portugal>.
- [10] J. Lee. «Analysis of Predisposing Factors for Hearing Loss in Adults». Em: *Journal of Korean Medical Science* (2015).
- [11] *Perda de audição.* pt. Último acesso 22 de março 2023. URL: <https://www.minisom.pt/perda-de-audicao/causas>.

- [12] *FAQs Minisom*. pt. Último acesso 6 de dezembro 2022. URL: <https://www.minisom.pt/nossos-servicos/faqs>.
- [13] IBERDROLA CORPORATIVA. *O QUE É A METODOLOGIA AGILE?* pt-BR. Último acesso 9 de fevereiro 2023. URL: <https://www.iberdrola.com/inovacao/que-e-metodologia-agile>.
- [14] PMI. «Success Rates Rise». Em: *Pulse of the Profession* (2017), pp. 1–4.
- [15] Atlassian. *Scrum Sprints: Everything You Need to Know*. en. Último acesso 9 de fevereiro 2023. URL: <https://www.atlassian.com/agile/scrum/sprints>.
- [16] *What is Scrum?* en. Último acesso 9 de fevereiro 2023. URL: <https://www.scrum.org/learning-series/what-is-scrum>.
- [17] Atlassian. *Jira | Issue Project Tracking Software*. en. Último acesso 11 de fevereiro 2023. URL: <https://www.atlassian.com/software/jira>.
- [18] *Is it the right time to switch to SwiftUI?* en. Último acesso 16 de junho 2023. URL: <https://www.atipik.ch/en/blog/switch-to-swiftui-for-app-development>.
- [19] Apple Inc. *Demystify SwiftUI - WWDC21 - Videos*. en. Último acesso 16 de junho 2023. URL: <https://developer.apple.com/videos/play/wwdc2021/10022/>.
- [20] *Will Objective-C be deprecated*. Último acesso 8 de agosto 2023. URL: <https://developer.apple.com/forums/thread/97785>.
- [21] *Apple decided to deprecate Objective-C and AppKit*. Último acesso 8 de agosto 2023. URL: <https://news.ycombinator.com/item?id=30573379#:~:text=Mac%20apps...-Apple%20decided%20to%20deprecate%20Objective-C%20and%20AppKit.,all%20the%20way%20for%20everyone.&text=Neither%20of%20these%20things%20are,stable%20than%20any%20web%20framework..>
- [22] *SwiftUI vs UIKit in 2022, Which Framework Should Your App Use?* / BlueLabel. Último acesso 5 de março 2023. URL: <https://www.bluelabellabs.com/blog/swiftui-vs-uikit/>.
- [23] Ricardo Montemayor. *Why SwiftUI will make UIKit obsolete*. en. Último acesso 8 de agosto 2023. Ago. de 2022. URL: <https://medium.com/thedevproject/why-swiftui-will-make-uikit-obsolete-5819dc1c6ef>.
- [24] *SwiftUI Overview - Xcode - Apple Developer*. Último acesso 6 de março 2023. URL: <https://developer.apple.com/xcode/swiftui/>.

- [25] Matteo Manferdini. *The Strategic SwiftUI Data Flow Guide*. en-US. Último acesso 15 de junho 2023. Jan. de 2021. URL: <https://matteomanferdini.com/swiftui-data-flow/>.
- [26] *TypeORM - Quick Guide*. en-US. Último acesso 23 de junho 2023. URL: https://www.tutorialspoint.com/typeorm/typeorm_quick_guide.htm.
- [27] *SQL (TypeORM) | NestJS - A progressive Node.js framework*. Último acesso 23 de junho 2023. URL: <https://docs.nestjs.com/recipes/sql-typeorm>.
- [28] *Home - AudioKit*. Último acesso 1 de junho 2023. URL: <http://www.audiokit.io/>.
- [29] *Zumbido ou zunido no ouvido - Distúrbios do ouvido, nariz e garganta*. pt. Último acesso 21 de junho 2023. URL: <https://www.msmanuals.com/pt/casa/dist%C3%A9rbios-do-ouvido,-nariz-e-garganta/sintomas-das-doen%C3%A7as-dos-ouvidos/zumbido-ou-zunido-no-ouvido>.
- [30] *Xcode*. en-US. Último acesso 31 de outubro 2022. URL: <https://developer.apple.com/documentation/xcode>.
- [31] Apple Inc. *TestFlight*. en. Último acesso 31 de outubro 2022. URL: <https://developer.apple.com/testflight/>.
- [32] *The DevSecOps Platform*. en. Último acesso 2 de novembro 2022. URL: <https://about.gitlab.com/>.
- [33] *GitKraken Legendary Git Tools*. en-US. Último acesso 2 de novembro 2022. URL: <https://www.gitkraken.com>.
- [34] *See the companies that rely on the legendary GitKraken Git tools*. Último acesso 25 de março 2023. URL: <https://www.gitkraken.com/sample-customers>.
- [35] *Figma: the collaborative interface design tool*. Último acesso 3 de novembro 2022. URL: <https://www.figma.com/>.
- [36] *Request for Free Software | Balsamiq*. Último acesso 3 de novembro 2022. URL: <https://balsamiq.com/givingback/free/>.
- [37] *Pricing Miro*. en-US. Último acesso 3 de novembro 2022. URL: <https://miro.com/pricing/>.
- [38] *Pricing for Figma's Free, Professional, and Organization plans*. en. Último acesso 3 de novembro 2022. URL: <https://www.figma.com/pricing/>.
- [39] *HTTP request methods - HTTP | MDN*₂₀₂₃. en-US. Último acesso 25 de março 2023. Abr. de 2023. URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>.

- [40] Jason A. Donenfeld. *WireGuard: fast, modern, secure VPN tunnel*. en. Último acesso 25 de março 2023. URL: <https://www.wireguard.com/>.
- [41] *Mailtrap*. en-US. Último acesso 26 de março 2023. Dez. de 2022. URL: <https://mailtrap.io/>.
- [42] Beatriz Coelho. *Conheça o LaTeX: entenda para serve, suas vantagens e desvantagens*. pt-BR. Último acesso 21 de junho 2023. Set. de 2018. URL: <https://blog.mettzer.com/latex/>.
- [43] *O que é o formato Rich Text (RTF)?* Último acesso 21 de junho 2023. URL: <https://pt.theastrologypage.com/rich-text-format>.
- [44] *Overleaf - Digital Science*. Último acesso 21 de junho 2023. URL: <https://www.digital-science.com/product/overleaf/>.
- [45] *Generate and validate tokens*. en-US. Último acesso 12 de abril 2023. URL: https://developer.apple.com/documentation/sign_in_with_apple/generate_and_validate_tokens.
- [46] Apple Inc. *App Store Review Guidelines*. en. Último acesso 13 de abril 2023. URL: <https://developer.apple.com/app-store/review/guidelines/>.
- [47] *Apple GeometryReader*. en-US. Último acesso 23 de junho 2023. URL: <https://developer.apple.com/documentation/swiftui/geometryreader>.
- [48] *Apple CoreData*. en-US. Último acesso 10 de junho 2023. URL: <https://developer.apple.com/documentation/coredata>.
- [49] *Avoid using UserDefaults.standard*. Último acesso 07 de julho 2023. URL: <https://theinkedengineer.com/articles/avoid-using-standard-userdefaults>.
- [50] *Push Notifications in iOS SwiftlyRush₂₀₂₂*. en-GB. Último acesso 15 de abril 2023. Abr. de 2022. URL: <https://www.swiftlyrush.com/push-notifications-in-ios/>.
- [51] Paul Hudson @twostraws. *How to create views in a loop using ForEach - a free SwiftUI by Example tutorial*. en. Último acesso 10 de junho 2023. Mai. de 2022. URL: <https://www.hackingwithswift.com/quick-start/swiftui/how-to-create-views-in-a-loop-using-foreach>.
- [52] Paul Hudson @twostraws. *What is a singleton? - free Swift 5.4 example code and tips*. en. Último acesso 10 de junho 2023. Abr. de 2022. URL: <https://www.hackingwithswift.com/example-code/language/what-is-a-singleton>.

- [53] *Portal do INE*. Último acesso 2 de maio 2023. URL: https://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_estudos&ESTUDOSest_boui=284013038&ESTUDOSmodo=2&xlang=pt.

DECLARAÇÃO

Declaro, sob compromisso de honra, que o trabalho apresentado neste relatório de estágio, com o título “*Evollu Audiology for All, A4A*”, é original e foi realizado por Estudante Micael Lima Feteira Oliveira (2210613) sob orientação de Professora Doutora Anabela Moreira Bernardino (anabela.bernardino@ipleiria.pt).

Leiria, setembro de 2023



Estudante Micael Lima Feteira Oliveira