

DM

AQUA

Explorando a gamificação e ciência cidadã
na monitorização da biodiversidade de cetáceos

DISSERTAÇÃO DE MESTRADO

Diego Andrés da Silva Briceño

MESTRADO EM ENGENHARIA INFORMÁTICA



UNIVERSIDADE da MADEIRA

A Nossa Universidade

www.uma.pt

setembro | 2023

AQUA

Explorando a gamificação e ciência cidadã
na monitorização da biodiversidade de cetáceos

DISSERTAÇÃO DE MESTRADO

Diego Andrés da Silva Briceño

MESTRADO EM ENGENHARIA INFORMÁTICA

ORIENTAÇÃO

Mara Sofia Gomes Dionísio

COORIENTAÇÃO

Ana Karina Caldeira Caraban



FACULDADE DE CIÊNCIAS EXATAS E DA ENGENHARIA

MESTRADO EM ENGENHARIA INFORMÁTICA

AQUA: Explorando a Gamificação e Ciência Cidadã na Monitorização da Biodiversidade de Cetáceos

Diego Andrés da Silva Briceño

Orientado por:

Mara Sofia Gomes Dionísio

Ana Karina Caldeira Caraban

Constituição do júri de provas públicas:

Nome completo (categoria), Presidente

Nome completo (categoria), Vogal

Nome completo (categoria), Vogal

14 de setembro de 2023

Resumo

Ao longo dos anos, as atividades humanas têm ameaçado o ecossistema marinho, e os cetáceos têm sido uma das espécies mais afetadas. A recolha de informação é crucial para monitorizar estas espécies e, posteriormente, tomar medidas em relação às políticas e iniciativas de conservação. Recentemente, o potencial dos dados recolhidos através da ciência cidadã tem sido explorado como uma oportunidade rápida e económica para recolher informações sobre avistamentos de cetáceos. Em particular, nas ilhas da Macaronésia, iniciativas têm procurado envolver embarcações turístico-marítimas na recolha destes dados durante atividades de observação de cetáceos, estabelecendo uma parceria entre cientistas e empresas para promover o conhecimento sobre a biodiversidade marinha. No entanto, esta recolha de dados é feita em papel por biólogos que acompanham as embarcações, tornando difícil centralizar todas as informações e garantir a participação contínua.

Esta dissertação foca-se em melhorar o desenho e desenvolvimento de uma aplicação móvel de ciência cidadã projetada para facilitar a recolha e partilha de informações sobre avistamentos de cetáceos. Uma vez que a revisão de literatura destaca que um desafio comum na ciência cidadã é manter a motivação dos participantes ao longo do tempo, este trabalho integrou elementos de gamificação na aplicação.

Foi realizada uma avaliação do protótipo com profissionais durante viagens de observação de cetáceos e os nossos resultados indicam que, embora a gamificação não tenha aumentado a motivação dos participantes, a experiência de utilizador da aplicação foi muito positiva, com os participantes a destacar a facilidade de utilização, a simplicidade do processo de recolha de dados e a relevância da gamificação para entusiastas.

Além disso, este trabalho também contribuiu para o desenvolvimento de uma API que permite partilhar dados recolhidos a partir da aplicação móvel com entidades externas, contribuindo para a partilha dos dados das espécies. Este trabalho aponta possíveis caminhos para investigações futuras, considerando o potencial e a utilidade do sistema AQUA, que combina ciência cidadã, tecnologia e ciência na monitorização da biodiversidade de cetáceos.

Keywords: interação humano-computador · ciência cidadã · gamificação · monitorização marinha

Abstract

Over the years, human activities have threatened the marine ecosystem, and cetaceans have been one of the most affected species. Gathering information is crucial for monitoring these species and, ultimately, taking action regarding conservation policies and initiatives. Recently, the potential of data collected through citizen science has been explored as a rapid and cost-effective opportunity to gather data on cetacean sightings. Particularly, in the Macaronesia islands, initiatives have sought to involve tourist maritime vessels in collecting this data during whale-watching activities, establishing a partnership between scientists and companies to promote learning about marine biodiversity. However, this data collection is done in a paper-based format by biologists accompanying the vessels, making it difficult to centralize all the information and ensure continuous participation.

This dissertation focuses on refining the design and development of a citizen science mobile application designed to facilitate the collection and sharing of information about cetacean sightings. Since the literature review highlighted that a common challenge in citizen science is sustaining participants' motivation over time, this work integrated gamification elements into the application.

A prototype evaluation was conducted with professionals during whale-watching trips and our findings indicate that while the gamification did not increase participants' motivation, the application's user experience was very positive, with participants highlighting its ease of use, the seamless data collection process and the relevance of gamification for enthusiasts.

Furthermore, this work also contributed to the development of an API that allows for sharing data collected from the mobile application with external entities, contributing to the dissemination of the species data. This work points towards possible avenues for future research, considering the potential and usefulness of the AQUA system, which combines citizen science, technology, and science in marine biodiversity monitoring.

Keywords: human-computer interaction · citizen science · gamification · marine monitoring

Agradecimentos

Primeiramente, agradeço a todos os meus colegas e amigos por toda a ajuda que me deram e pela participação que tiveram ao longo de toda a minha jornada académica.

Obrigado a cada docente da Universidade da Madeira que me ensinou um pouco de tudo o que sei hoje, e que me fizeram capaz de realizar este trabalho da melhor forma possível.

A toda a equipa do ARDITI e ITI, pelas oportunidades de participar em vários projetos que me permitiram aprender muito, e por sempre terem sido acessíveis quando precisei.

Às minhas orientadoras, a Dra. Mara Dionísio e a Doutoranda Ana Caraban, pela disponibilidade que tiveram em me orientar e por terem acreditado sempre nas minhas competências.

Ao Marc Fernandez, por ter estado sempre disponível em relação a toda a parte que envolveu informação científica sobre as espécies e à colaboração com outros biólogos para testar a aplicação AQUA.

Ao João Gouveia, pela sua enorme disponibilidade em testar a aplicação e pela grande ajuda que me deu para melhorar o meu trabalho.

À Eva Íñiguez, ao Gonçalo Barros e ao Nuno Chelas, por terem tirado um pouco do seu tempo para testar a aplicação.

Aos biólogos Marc Tolosa, Camila Dávila e Sara Jardim, por todos os elogios e por todas as sugestões que me deram e que me fizeram compreender melhor o trabalho de um biólogo marinho.

Às empresas VMT Madeira, Lobosonda, H2O Madeira, Madeira Wild Blue, e Picos de Ventura, pela sua disponibilidade em participar nos testes realizados durante as suas viagens, mesmo sabendo que foi numa época de muito trabalho, e por todo o feedback que me forneceram e que me permitiu concluir o meu trabalho.

Agradeço imenso à minha mãe, ao meu pai e ao meu irmão por todo o apoio que me têm dado ao longo de toda a minha vida e por sempre acreditarem em mim. Têm sido anos difíceis, mas com o vosso apoio consegui.

Pai, sei que estarias muito orgulhoso, era um dos teus sonhos.

Conteúdo

Lista de Figuras	viii
Lista de Tabelas	xiii
1 Introdução.....	1
1.1 Objetivos e Questões de Investigação	2
1.2 Contribuições.....	3
1.3 Organização do Documento	3
2 Revisão de Literatura	5
2.1 Aplicações de Ciência Cidadã Existentes	5
2.2 Gamificação e o seu Potencial	9
2.3 Oportunidades de Contribuição	10
2.4 Revisão de Tecnologias de Desenvolvimento	11
2.4.1 Abordagens ao Desenvolvimento Móvel	12
2.4.2 <i>Frameworks</i> de Desenvolvimento Móvel Híbrido	12
2.4.3 Frameworks de Desenvolvimento <i>Back-end</i>	13
2.4.4 Sistemas Gestores de Bases de Dados	13
3 Design da Aplicação AQUA	14
3.1 Conceito da Aplicação	14
3.2 Protótipo de Alta Fidelidade.....	15
3.3 Testes de Usabilidade - 1ª Iteração	18
3.3.1 Objetivos e Protocolo	18
3.3.2 Participantes	19
3.3.3 Resultados	19
3.3.4 Discussão.....	20
3.4 Design Final	22
3.4.1 Reporte de Avistamentos e Incidentes	24
3.4.2 Gamificação	28
3.4.2.1 Coleção de Cartas de Espécies	29
3.4.2.2 Coleção de Distintivos	30
3.4.2.3 Ranking de Utilizadores.....	32
3.4.3 Perfil do Utilizador.....	34
3.4.4 Mapa de Avistamentos	34
3.4.5 Visualização de Informação sobre Espécies	36
3.4.6 <i>Tracking</i> das Rotas de Viagem	37
3.4.7 Histórico de Rotas de Viagem e Reportes	38
4 Modelação	39
4.1 Requisitos do Sistema	39
4.2 Casos de Uso	42
4.3 Atributos de Qualidade.....	45
4.3.1 Usabilidade.....	46
4.3.2 Desempenho	46

4.3.3	Disponibilidade	47
4.3.4	Modificabilidade	47
4.3.5	Segurança	48
4.4	Arquitetura do Sistema.....	49
4.4.1	Modelo C4 - Diagrama de Contexto	49
4.4.2	Modelo C4 - Diagrama de Containers.....	50
4.4.3	Modelo C4 - Diagrama de Componentes	51
5	Implementação	53
5.1	Desenvolvimento da Aplicação Móvel	53
5.1.1	Arquitetura MVVM	53
5.1.2	Boas Práticas	54
5.1.3	Estrutura do Projeto	54
5.1.4	Padrão Singleton.....	55
5.1.5	Gestão de Estados com Provider	56
5.1.6	Geolocalização	57
5.1.7	Visualização de Mapas	58
5.1.8	<i>Tracking</i> das Rotas de Viagem	59
5.1.9	Verificação de Conexão à Internet	60
5.1.10	Persistência de Dados Simples.....	61
5.1.11	Armazenamento de Imagens em Cache.....	62
5.1.12	<i>Text-To-Speech</i>	63
5.1.13	Armazenamento de Dados Local.....	63
5.2	Desenvolvimento da API	68
5.2.1	Arquitetura MVC.....	68
5.2.2	Endpoints	69
5.2.3	Tratamento de Erros	70
5.2.4	Testes	70
5.2.5	Documentação	71
5.2.6	<i>Data Transfer Objects</i>	72
5.2.7	Encriptação de <i>Passwords</i>	73
5.2.8	Json Web Token	75
5.3	Armazenamento de Dados <i>Online</i>	76
5.4	Testes Funcionais: Tracking da Rota da Viagem	78
5.4.1	Objetivos e Protocolo	78
5.4.2	Participantes	78
5.4.3	Resultados	78
5.4.4	Discussão.....	80
6	Avaliação do Protótipo	82
6.1	Avaliação com Especialistas em Desenho de Media Interativos	82
6.1.1	Objetivos e Protocolo	82
6.1.2	Participantes	83
6.1.3	Resultados	84
6.1.4	Impacto no Design Final.....	88
6.2	Avaliação com Profissionais em Observação de Cetáceos.....	89
6.2.1	Objetivos e Protocolo	89

6.2.2 Participantes	89
6.2.3 Resultados	90
7 Discussão	94
8 Limitações e Trabalho Futuro	101
9 Conclusão	104
Referências	105
A Apêndice - Tabelas	110
B Apêndice - Imagens	123
B.1 Aplicação AQUA - Design Final.....	123
B.2 Arquitetura do Sistema: Modelo C4	147
B.3 AQUA 1.0.....	150
B.4 Testes de Usabilidade	150
B.5 Diagramas de Casos de Uso.....	152
C Apêndice - Listas.....	157
C.1 Requisitos Funcionais	157
C.2 Distintivos.....	162
C.3 Citações dos Participantes	163
C.3.1 Testes de Usabilidade - 1ª Iteração	163
C.3.2 Testes de Usabilidade - 2ª Iteração	164
C.3.3 Estudo	166
D Apêndice - Detalhes do Desenvolvimento da API	169
D.1 Arquitetura MVC: Componentes	169
D.2 Boas Práticas	173
D.3 Endpoints	174
D.4 Entity Framework.....	183

Lista de Figuras

1	<i>Screenshots</i> de Aplicações: (1) Aplicação “ <i>Report Invasive Plants</i> ”, mapa com espécies reportadas; (2) Aplicação “ <i>Seek by iNaturalist</i> ”, visualização de distintivos obtidos e nível do utilizador; (3-4) Aplicação “ <i>IWDG Reporting App</i> ”, ecrã principal e ecrã formulário para reportar avistamentos.	6
2	<i>Screenshots</i> de Aplicações: (1-2) Aplicação “ <i>Sharktivity</i> ”, ecrã para reportar avistamentos e ecrã de mapa de avistamentos; (3-4) Aplicação “ <i>Whale and Dolphin Tracker</i> ”, ecrã com lista de opções de espécies (incluindo a opção “Unknown”), e ecrã com mapa de avistamentos; (5) Aplicação “ <i>Seafari</i> ”, ecrã com informação sobre uma espécie.	7
3	Protótipo de Alta Fidelidade: (1-2) Ecrãs de seleção de espécies avistadas; (3-4) Ecrãs para preenchimento de dados sobre o avistamento e as espécies avistadas;	16
4	Protótipo de Alta Fidelidade: (1) Coleção de cartas de espécies; (2) Ranking com cinco utilizadores com mais avistamentos reportados; (3) Cartas das espécies.	17
5	Abordagem adotada para os testes de usabilidade (1ª iteração)	19
6	Botão de acesso à coleção de espécies: (1) Versão da aplicação de teste; (2) Versão de melhoria sugerida.	21
7	Cartas das espécies com diferentes cores: (1) Carta de espécie comum; (2) Carta de espécie semirara; (3) Carta de espécie rara.	21
8	Protótipo de alta fidelidade: (1) Ecrã de coleção de distintivos; (2) Ecrã de desbloqueio de novo distintivo; (3) Ecrã de visualização de detalhes de um distintivo.	22
9	<i>Screenshots</i> da aplicação AQUA: (1) Splash screen; (2) Ecrã de login; (3) Ecrã de criação de conta; (4-5) Adição de informação ao perfil de utilizador.	23
10	<i>Screenshots</i> da aplicação AQUA: Introdução ao funcionamento da aplicação.	23
11	<i>Screenshots</i> da Aplicação AQUA: (1) Ecrã inicial incluindo contadores, botões e cinco espécies mais avistadas; (2) Ecrã inicial incluindo cinco espécies mais avistadas e mapa de avistamentos.	24
12	<i>Screenshots</i> da aplicação AQUA: (1) Ecrã de seleção de espécies; (2) Ecrã de pesquisa de espécies; (3) Formulário para adição de espécies desconhecida.	25
13	<i>Screenshots</i> da aplicação AQUA: (1) Ecrã de indicação do número de indivíduos das espécies; (2) Ecrã de indicação dos comportamentos das espécies; (3) Ecrã de preenchimento de informação sobre espécie desconhecida; (4) Ecrã para adição de comentário.	26
14	<i>Screenshots</i> da aplicação AQUA: (1-2) Ecrãs do sumário dos dados do avistamento; (3) Ecrã de seleção do tipo de incidente.	27
15	<i>Screenshots</i> da aplicação AQUA: (1) Ecrã de desbloqueio de novos distintivos; (2) Ecrã de desbloqueio de novas cartas de espécies.	28
16	<i>Screenshot</i> da aplicação AQUA: Botões para aceder à coleção de cartas e à coleção de distintivos destacados com brilho.	28
17	Cartas das espécies: (1) Carta de espécie comum; (2) Carta de espécie semirara; (3) Carta de espécie rara.	29

18	Screenshots da aplicação AQUA: (1) Ecrã da coleção de espécies sem filtros aplicados; (2) Ecrã da coleção de espécies com filtro aplicado; (3-4) Visualização de uma carta desbloqueada.	30
19	Ícones dos distintivos “Novice Spotter”, “Sighting Pro”, “Citizen Scientist” e “Ocean Guardian”	31
20	Ícones dos distintivos “Collector”, “Master Enthusiast”, “Marine Enthusiast” e “Ocean Explorer”	31
21	Ícones dos distintivos “Cetacean Expert”, “Dolphin Defender”, “Whale Watcher” e “Geographer”	32
22	Screenshots da aplicação AQUA: (1) Ecrã do ranking de utilizadores; (2) Visualização de perfil de um utilizador do ranking; (3) Mensagem ao tentar visualizar um perfil privado; (4) Filtros disponíveis para o ranking.	33
23	Screenshots da aplicação AQUA: (1) Ecrã de perfil do utilizador; (2) Lista de títulos desbloqueados pelo utilizador; (3) Alteração de privacidade do perfil (de público para privado).	34
24	Screenshots da aplicação AQUA: (1) Ecrã de mapa com avistamentos do último mês; (2) Detalhes do cluster de avistamentos.	35
25	(1) Sobreposição de áreas de incerteza de dois avistamentos no mapa; (2) Dois clusters de avistamentos, no mapa de avistamentos; (3) Agrupação de <i>clusters</i> próximos.	36
26	Screenshots da aplicação AQUA: (1) Ecrã de seleção do tipo de animal; (2) Ecrã de seleção da espécie; (3) Pesquisa de uma espécie pelo nome; (4-5) Ecrã com informação sobre a espécie.	37
27	Screenshots da aplicação AQUA: (1) Ecrã durante o tracking de uma rota de viagem; (2) Ecrã após terminar o tracking de uma rota de viagem.	37
28	Screenshots da aplicação AQUA: (1) Ecrã do histórico de reportes; (2) Detalhes de um avistamento submetido e (3) não submetido; (4) Ecrã do histórico de reportes, após submissão dos reportes.	38
29	Diagrama de Casos de Uso Geral.	43
30	Modelo C4: Diagrama de Contexto	49
31	Modelo C4: Diagrama de Containers	50
32	Modelo C4: Diagrama de Componentes (Aplicação Móvel AQUA)	51
33	Modelo C4: Diagrama de Componentes (API AQUA)	52
34	Arquitetura MVVM - Interação entre a Vista, Modelo de Vista, e Modelo	53
35	Implementação do MVVM: A Vista (1) solicita os dados dos utilizadores ao Modelo de Vista (método <i>getAllUsers</i>). O Modelo de Vista (2) solicita os dados ao Modelo, através da chamada ao serviço correspondente (3) e notifica a Vista sobre as alterações nos dados, através do método <i>notifyListeners</i>	54
36	Implementação do padrão ‘Singleton’ na classe ‘APIServices’	56
37	Definição do método ‘hasInternetConnection()’	61
38	Utilização do método ‘hasInternetConnection()’ no método ‘getAllAnimals()’ que permite obter os dados dos animais do sistema.	61
39	Apresentação de imagem de perfil em cache no <i>ranking</i> , sem conexão à Internet.	62
40	Inicialização da instância da classe ‘FlutterTts’ e definição dos métodos ‘textToSpeech(String text)’ e ‘stopTts()’	63
41	Implementação do modelo ‘User’ - Propriedades	65

42	Implementação do modelo ' <i>UserSQLite</i> ' - Propriedades	65
43	Implementação do método ' <i>getAllUsers()</i> ' para obtenção dos dados de todos os utilizadores	65
44	Implementação do método ' <i>User.fromJson()</i> ' responsável por mapear os dados de um ficheiro JSON num objeto da classe ' <i>User</i> '	65
45	Implementação do método ' <i>insertUser()</i> '	66
46	Implementação do método ' <i>UserSQLite.fromUser()</i> ' responsável por transformar um objeto da classe ' <i>User</i> ' num objeto da classe ' <i>UserSQLite</i> '	66
47	Exemplo do processo da cópia dos dados do utilizadores do PostgreSQL para SQLite ...	66
48	Arquitetura MVC - Interação entre os modelos, os controladores e as vistas	69
49	Teste do método GET ao <i>endpoint</i> ' <i>api/Animals</i> ' utilizando o <i>Postman</i>	71
50	Lista de <i>endpoints</i> desenvolvido para a coleção de animais, no <i>Swagger</i>	71
51	Detalhes do método GET do <i>endpoint</i> ' <i>api/Animals/{id}</i> ' no <i>Swagger</i>	72
52	Implementação da classe ' <i>User</i> '	72
53	Implementação da classe ' <i>UserDto</i> '	72
54	Implementação do método GET do endpoint ' <i>api/Users</i> '	73
55	Método POST - Aplicação de <i>password hash</i> e <i>password salt</i> no registo de um novo utilizador	74
56	Implementação do método ' <i>CreatePasswordHash</i> '	74
57	Parte da implementação do método POST, onde é feita a verificação da <i>password</i> do utilizador	75
58	Implementação do método " <i>getAllUsers()</i> ": Adição do token JWT ao cabeçalho "Authorization" de um pedido do cliente ao servidor.	76
59	Base de dados PostgreSQL: (1) Dados de três coordenadas de uma rota registadas automaticamente; (2) Dados das últimas duas coordenadas registadas da rota número 2.	80
60	Abordagens adotadas para os testes de usabilidade (2ª iteração)	83
61	Alterações no Design Final: (1) Novo ícone da barra de navegação; (2) Adição de animação; (3) Fonte de texto das cartas.	88
62	Screenshots da aplicação AQUA: Splash screen	123
63	Screenshots da aplicação AQUA: Ecrã de login	123
64	Screenshots da aplicação AQUA: Ecrã de criação de conta	124
65	Screenshots da aplicação AQUA: Ecrã de adição de informação ao perfil de utilizador - Parte 1	124
66	Screenshots da aplicação AQUA: Ecrã de adição de informação ao perfil de utilizador - Parte 2	125
67	Screenshots da aplicação AQUA: Introdução ao funcionamento da aplicação - Parte 1 ...	125
68	Screenshots da aplicação AQUA: Introdução ao funcionamento da aplicação - Parte 2 ...	126
69	Screenshots da aplicação AQUA: Introdução ao funcionamento da aplicação - Parte 3 ...	126
70	Screenshots da aplicação AQUA: Introdução ao funcionamento da aplicação - Parte 4 ...	127
71	Screenshots da aplicação AQUA: Ecrã inicial incluindo contadores, botões e cinco espécies mais avistadas	127
72	Screenshots da aplicação AQUA: Ecrã inicial incluindo cinco espécies mais avistadas e mapa de avistamentos	128
73	Screenshots da aplicação AQUA: Ecrã de seleção de espécies	128
74	Screenshots da aplicação AQUA: Ecrã de pesquisa de espécies	129

75 Screenshots da aplicação AQUA: Formulário para adição de espécies desconhecida	129
76 Screenshots da aplicação AQUA: Ecrã de indicação do número de indivíduos das espécies	130
77 Screenshots da aplicação AQUA: Ecrã de indicação dos comportamentos das espécies . . .	130
78 Screenshots da aplicação AQUA: Ecrã de preenchimento de informação sobre espécie desconhecida	131
79 Screenshots da aplicação AQUA: Ecrã para adição de comentário	131
80 Screenshots da aplicação AQUA: Ecrã do sumário dos dados do avistamento - Parte 1 . .	132
81 Screenshots da aplicação AQUA: Ecrã do sumário dos dados do avistamento - Parte 2 . .	132
82 Screenshots da aplicação AQUA: Ecrã de seleção do tipo de incidente	133
83 Screenshots da aplicação AQUA: Ecrã de desbloqueio de novos distintivos	133
84 Screenshots da aplicação AQUA: Ecrã de desbloqueio de novas cartas de espécies	134
85 Screenshots da aplicação AQUA: Ecrã da coleção de espécies sem filtros aplicados	134
86 Screenshots da aplicação AQUA: Ecrã da coleção de espécies com filtro aplicado	135
87 Screenshots da aplicação AQUA: Visualização de uma carta desbloqueada - Parte 1	135
88 Screenshots da aplicação AQUA: Visualização de uma carta desbloqueada - Parte 2	136
89 Screenshots da aplicação AQUA: Ecrã do ranking de utilizadores	136
90 Screenshots da aplicação AQUA: Visualização de perfil de um utilizador do ranking	137
91 Screenshots da aplicação AQUA: Mensagem ao tentar visualizar um perfil privado	137
92 Screenshots da aplicação AQUA: Filtros disponíveis para o ranking	138
93 Screenshots da aplicação AQUA: Ecrã de perfil do utilizador	138
94 Screenshots da aplicação AQUA: Lista de títulos desbloqueados pelo utilizador	139
95 Screenshots da aplicação AQUA: Alteração de privacidade do perfil (de público para privado)	139
96 Screenshots da aplicação AQUA: Ecrã de mapa com avistamentos do último mês	140
97 Screenshots da aplicação AQUA: Detalhes do cluster de avistamentos	140
98 Screenshots da aplicação AQUA: Ecrã de seleção do tipo de animal	141
99 Screenshots da aplicação AQUA: Ecrã de seleção da espécie	141
100 Screenshots da aplicação AQUA: Pesquisa de uma espécie pelo nome	142
101 Screenshots da aplicação AQUA: Ecrã com informação sobre a espécie - Parte 1	142
102 Screenshots da aplicação AQUA: Ecrã com informação sobre a espécie - Parte 2	143
103 Screenshots da aplicação AQUA: Ecrã durante o tracking de uma rota de viagem	143
104 Screenshots da aplicação AQUA: Ecrã após terminar o tracking de uma rota de viagem .	144
105 Screenshots da aplicação AQUA: Ecrã do histórico de reportes	144
106 Screenshots da aplicação AQUA: Detalhes de um avistamento submetido	145
107 Screenshots da aplicação AQUA: Detalhes de um avistamento não submetido	145
108 Screenshots da aplicação AQUA: Ecrã do histórico de reportes, após submissão dos reportes	146
109 Modelo C4: Diagrama de Contexto	147
110 Modelo C4: Diagrama de Containers	148
111 Modelo C4: Diagrama de Componentes (Aplicação Móvel AQUA)	148
112 Modelo C4: Diagrama de Componentes (API AQUA)	149
113 Screenshots da aplicação AQUA (Versão Anterior): (1-2) Ecrãs de seleção de espécies avistadas; (3-4) Ecrãs para preenchimento de dados sobre o avistamento e as espécies avistadas;	150

114	Jogo “This or That” - Seleção de versões da identificação de espécies avistadas: (A) Antes do <i>redesign</i> ; (B) Após o <i>redesign</i> da aplicação.	150
115	Jogo “This or That” - Seleção de versões da especificação dos detalhes das espécies avistadas e do avistamento: (A) Antes do <i>redesign</i> ; (B) Após o <i>redesign</i> da aplicação.	151
116	Jogo “This or That” - Seleção de versões das cartas de espécies: (A) Cartas com raridades diferentes com a mesma cor; (B) Cartas com raridades diferentes com cores distintas.	151
117	Diagrama de Caso de Uso: Registrar Utilizador.	152
118	Diagrama de Caso de Uso: Autenticar Utilizador.	152
119	Diagrama de Caso de Uso: Reportar Avistamento.	153
120	Diagrama de Caso de Uso: Reportar Incidente.	153
121	Diagrama de Caso de Uso: Iniciar Tracking da Rota de Viagem.	154
122	Diagrama de Caso de Uso: Visualizar Perfil de Utilizador.	154
123	Diagrama de Casos de Uso: Desbloquear Carta de Espécies; Desbloquear Distintivo.	154
124	Diagrama de Caso de Uso: Visualizar Ranking.	155
125	Diagrama de Casos de Uso: Visualizar Informação sobre Espécie; Ouvir Informação sobre Espécies.	155
126	Diagrama de Caso de Uso: Visualizar Histórico de Reportes.	156
127	Diagrama de Caso de Uso: Visualizar Histórico de Rotas de Viagem.	156
128	Propriedades do tipo 'DbSet' da classe 'ApiDbContext'	184
129	Utilização da anotação '[Table("user")]' na classe 'User'	184
130	Utilização da anotação '[ForeignKey("UserId")]' e '[ForeignKey("BadgeId")]' na classe 'UserEarnedBadge'	185
131	Utilização do método 'Add' para adicionar um novo distintivo à base de dados	186
132	Utilização do método 'FirstOrDefault' para obter os dados de uma coordenada específica da base de dados	186
133	Exemplo da operação 'Update' de um utilizador	186
134	Exemplo da operação 'Delete' de um animal	186

Lista de Tabelas

1	Comparação entre funcionalidades existentes e inclusão de gamificação em cada aplicação	7
2	Comparação entre as avaliações médias e número de downloads de cada aplicação	8
3	Comparação do Desenvolvimento de Aplicações Móveis Nativo e Híbrido	12
4	Comparação entre Frameworks de Desenvolvimento Back-end	13
5	Caraterísticas dos Participantes	19
6	Tabela de Brosseau - Priorização de Atributos de Qualidade	46
7	Cenário de Qualidade - Usabilidade	46
8	Cenário de Qualidade - Desempenho	47
9	Cenário de Qualidade - Disponibilidade	47
10	Cenário de Qualidade - Modificabilidade	48
11	Cenário de Qualidade - Segurança (Confidencialidade)	48
12	Cenário de Qualidade - Segurança (Autenticação)	49
13	Resultados Quantitativos dos Testes Funcionais do Tracking das Rotas de Viagem	79
14	Resultados da relevância média de cada distintivo da aplicação AQUA	84
15	Caraterização de empresas de observação de cetáceos	90
16	Dados demográficos dos participantes do estudo	90
17	Descrição Detalhada do Caso de Uso UC1	111
18	Descrição Detalhada do Caso de Uso UC2	112
19	Descrição Detalhada do Caso de Uso UC3	113
20	Descrição Detalhada do Caso de Uso UC4	114
21	Descrição Detalhada do Caso de Uso UC5	115
22	Descrição Detalhada do Caso de Uso UC6	116
23	Descrição Detalhada do Caso de Uso UC7	117
24	Descrição Detalhada do Caso de Uso UC8	117
25	Descrição Detalhada do Caso de Uso UC9	118
26	Descrição Detalhada do Caso de Uso UC10	118
27	Descrição Detalhada do Caso de Uso UC11	119
28	Descrição Detalhada do Caso de Uso UC12	119
29	Descrição Detalhada do Caso de Uso UC13	120
30	Descrição Detalhada do Caso de Uso UC14	121
31	Descrição Detalhada do Caso de Uso UC15	122
32	Propriedades do Modelo 'User'	169
33	Propriedades do Modelo 'Specie'	170
34	Propriedades do Modelo 'UserEarnedBadge'	170
35	Propriedades do Modelo 'UserEarnedSpecie'	171
36	Propriedades do Modelo 'TripRoute'	171
37	Propriedades do Modelo 'Report'	172
38	Propriedades do Modelo 'SightedSpecieInformation'	172
39	Métodos de Conveniência do ASP.NET Core e a respetiva resposta HTTP	174
40	Métodos HTTP desenvolvidos para o <i>endpoint</i> "api/Users"	174
41	Métodos HTTP desenvolvidos para o <i>endpoint</i> "api/Users/{id}"	175

42	Métodos HTTP desenvolvidos para o <i>endpoint</i> “api/Animals”	176
43	Métodos HTTP desenvolvidos para o <i>endpoint</i> “api/Animals/{id}”	176
44	Métodos HTTP desenvolvidos para o <i>endpoint</i> “api/Species”	177
45	Métodos HTTP desenvolvidos para o <i>endpoint</i> “api/Species/{id}”	177
46	Métodos HTTP desenvolvidos para o <i>endpoint</i> “api/Badges”	178
47	Métodos HTTP desenvolvidos para o <i>endpoint</i> “api/Badges/{id}”	178
48	Métodos HTTP desenvolvidos para o <i>endpoint</i> “api/Reports”	178
49	Métodos HTTP desenvolvidos para o <i>endpoint</i> “api/Reports/{id}”	179
50	Métodos HTTP desenvolvidos para o <i>endpoint</i> “api/SightedSpeciesInformation”	180
51	Métodos HTTP desenvolvidos para o <i>endpoint</i> “api/SightedSpeciesInformation/{id}”	180
52	Métodos HTTP desenvolvidos para o <i>endpoint</i> “api/Coordinates”	181
53	Métodos HTTP desenvolvidos para o <i>endpoint</i> “api/Coordinates/{id}”	181
54	Métodos HTTP desenvolvidos para o <i>endpoint</i> “api/UsersEarnedBadges”	182
55	Métodos HTTP desenvolvidos para o <i>endpoint</i> “api/UsersEarnedBadges/{id}”	182
56	Métodos HTTP desenvolvidos para o <i>endpoint</i> “api/UsersEarnedSpecies”	183
57	Métodos HTTP desenvolvidos para o <i>endpoint</i> “api/UsersEarnedSpecies/{id}”	183

Lista de Acrónimos

ACID Atomicidade, Consistência, Isolamento, Durabilidade

ACIF Associação Comercial e Industrial do Funchal

API Application Programming Interface

CRUD Create, Read, Update and Delete

GPS Global Positioning System

HTTP Hyper Text Transfer Protocol

HTTPS Hyper Text Transfer Protocol Secure

IST Instituto Superior Técnico

IUCN International Union for the Conservation of Nature

IWC International Whaling Commission

IWDG Irish Whale and Dolphin Group

JSON JavaScript Object Notation

JWT JSON Web Token

MVC Model-View-Controller

MVVM Model-View-ViewModel

SUS System Usability Scale

UC Use Case

UEQ User Experience Questionnaire

1 Introdução

A destruição, degradação e divisão de habitats naturais, resultado de atividades humanas, como as viagens não sustentáveis para observação de cetáceos e a pesca, têm vindo a colocar em ameaça espécies aquáticas selvagens, gerando a necessidade de criar meios que permitam minimizar esses danos. A identificação e monitorização de informação sobre as espécies marinhas têm permitido conhecer mais sobre as mesmas e tomar medidas de conservação que podem contribuir para a proteção das espécies ameaçadas [1].

Os cetáceos, que incluem baleias e golfinhos, desempenham um papel de extrema importância no nosso planeta [2]. Eles são responsáveis por modificar habitats, aumentando a biodiversidade das espécies e a produtividade do ecossistema marinho. Além disso, eles são indicadores importantes do estado do ambiente marinho, pois quando estes animais são ameaçados, todo o ecossistema é afetado, tornando-se essencial garantir a proteção destas espécies e dos seus habitats.

As baleias têm um papel fundamental na redução do dióxido de carbono (CO₂) na atmosfera [3]. Elas recolhem milhares de toneladas de carbono ao longo das suas vidas, agindo como reguladoras naturais do CO₂. Isso permite limitar o impacto dos gases do efeito de estufa e do aquecimento global, contribuindo para reduzir o impacto das alterações climáticas [2].

No entanto, é preocupante observar que os cetáceos estão a enfrentar ameaças significativas. De acordo com a *International Union for the Conservation of Nature* (IUCN), cinco espécies de cetáceos são classificadas como criticamente ameaçadas, doze como ameaçadas, sete como vulneráveis, e dez como quase ameaçadas [2].

O turismo atraído pelas viagens de observação destes animais apresenta uma grande oportunidade económica para muitas comunidades em todo o mundo, incluindo a Macaronésia [4]. Só na Ilha da Madeira, até ao final do ano 2022, era expectável que mais de 250 mil pessoas desfrutassem deste tipo de atividade disponibilizada por 87 operadores registados em toda a ilha [5].

A indústria de observação de cetáceos tem crescido significativamente devido a interesses económicos [4]. Em 2011, a *International Whaling Commission* (IWC) criou um plano estratégico para promover a observação responsável e sustentável de cetáceos, incluindo a recolha, monitorização e partilha de informação sobre essas espécies, envolvendo governos, cientistas e a indústria [6].

Embora esta atividade possa promover a preservação das baleias, golfinhos e outras espécies marinhas nos seus habitats naturais, também pode ter o potencial de causar distúrbios nos animais e impactar negativamente o ecossistema marinho, quando realizada de forma não sustentável [7]. Deste modo, é importante que os operadores turísticos e os responsáveis pela regulação do turismo de observação de cetáceos compreendam os benefícios e os potenciais impactos desta atividade.

Apesar dos interesses de cada operador turístico, a maioria tenta garantir uma observação de cetáceos sustentável, embora nem sempre sejam tomadas medidas para minimizar ou evitar esses impactos, levando a que algumas espécies migrem para outras regiões onde não são observadas, nem se sentem sob pressão, e onde podem alimentar-se e descansar em segurança [8].

O representante do setor de atividades marítimo-turísticas da Associação Comercial e Industrial do Funchal (ACIF) alertou para o facto da capacidade de observar cetáceos estar no limite e exigir mais fiscalização, principalmente nos períodos de grande procura deste tipo de atividade, como, por exemplo, o verão, onde acabam por haver entre 10 a 12 embarcações a rodear um mesmo grupo de animais [5].

Por outro lado, as viagens de observação de cetáceos, quando realizadas de forma sustentável, pode ser uma das melhores oportunidades para recolher dados que podem ser usados para proteger o ecossistema marinho. Uma viagem de uma embarcação específica para investigação pode ter um elevado custo económico e a sua preparação pode ser demorada. Logo, como as embarcações que praticam atividades turístico-marítimas estão frequentemente a realizar viagens, tornam-se um excelente meio para realizar uma recolha de dados sobre as espécies marinhas que, ao serem partilhadas com os cientistas, podem tornar-se valiosas contribuições para monitorizá-las [9].

Atualmente, algumas operadoras de viagens marítimas recorrem a um biólogo marinho que acompanha as embarcações, responsável por identificar as espécies avistadas, fornecer informações científicas aos passageiros, e por reportar os dados dos avistamentos de forma manual. Além de ser um processo demorado, o trabalho de observação e documentação das ocorrências torna-se difícil de gerir simultaneamente, devido à realização das restantes atividades a bordo.

Desta forma, para ajudar na identificação e monitorização das espécies marinhas, a ciência cidadã surge como oportunidade para envolver um grande número de pessoas na recolha e partilha de dados sobre as espécies [10], fornecendo um contributo para a ciência, enquanto promovem a aprendizagem e construção de conexões com a mesma [11].

No entanto, é importante destacar que a motivação dos utilizadores envolvidos na ciência cidadã pode ser um desafio, uma vez que a participação contínua e o interesse podem diminuir ao longo do tempo [12]. Neste contexto, surge a gamificação como possível solução. A introdução de elementos de gamificação, como distintivos e recompensas, podem tornar aplicações de ciência cidadã mais envolventes, mantendo os utilizadores motivados em recolher dados científicos [13].

1.1 Objetivos e Questões de Investigação

De forma a mitigar os problemas na recolha de dados para monitorização de cetáceos, foi proposto no contexto do projeto europeu INTERTAGUA [14], o desenvolvimento de uma aplicação móvel, intitulada de AQUA. Esta aplicação foi iniciada no âmbito da tese de Mestrado em Engenharia Informática e de Computadores de uma aluna do Instituto Superior Técnico da Universidade de Lisboa (IST) [15] e tem como principais funcionalidades a recolha de dados de avistamentos de cetáceos, visualização de informação sobre as espécies marinhas, a visualização da rota da viagem em tempo real, bem como de um mapa com a localização dos avistamentos reportados e ainda possibilidade de reportar incidentes.

A presente dissertação dá continuidade ao trabalho previamente desenvolvido, tendo como principais objetivos: 1) Melhorar a experiência do utilizador e desempenho da aplicação AQUA, através do redesenho da interação e interface de utilizador, incorporação de elementos de gamificação, e reimplementação técnica da aplicação; 2) O desenvolvimento de uma API RESTful que permita o acesso aos dados dos avistamentos recolhidos a outros sistemas externos; 3) A realização de uma avaliação do impacto da incorporação de elementos de gamificação no envolvimento dos utilizadores e na experiência proporcionada pela aplicação AQUA.

O processo de investigação será guiado pelas seguintes questões de investigação (QI):

QI1. Quais são as funcionalidades que mais influenciam a satisfação dos utilizadores durante a experiência com a aplicação móvel AQUA?

QI2. Quais os elementos de gamificação com potencial de envolver os utilizadores com a aplicação móvel AQUA?

QI3. De que forma podemos aumentar a partilha dos dados recolhidos a partir da aplicação móvel AQUA para a monitorização de espécies de cetáceos?

1.2 Contribuições

A principal contribuição desta dissertação reside no desenho e desenvolvimento de uma aplicação móvel de ciência cidadã dedicada à recolha de dados sobre avistamentos de cetáceos. Através desta aplicação, qualquer entusiasta de observação de cetáceos ou pessoa que realize atividades marítimas pode reportar avistamentos, contribuindo para a obtenção de informações valiosas sobre a presença e comportamento destas espécies marinhas. A aplicação é desenhada para oferecer uma experiência de utilização fácil e agradável, permitindo que qualquer pessoa possa colaborar com biólogos marinhos, cientistas e investigadores na recolha dos dados.

Outra contribuição desta dissertação é o desenvolvimento de uma API que permita a partilha dos dados recolhidos a partir da aplicação móvel com sistemas externos. A API permitirá que outras entidades e sistemas autorizados tenham acesso aos dados dos avistamentos e incidentes reportados na aplicação. Ao permitir o acesso aos dados, a API contribui para a monitorização destas espécies, permitindo realizar uma análise dos dados recolhidos e promover a proteção e conservação do ecossistema marinho.

Finalmente, outra contribuição é a realização de um estudo no contexto real para avaliar o impacto da incorporação de gamificação na aplicação móvel desenvolvida. Esta avaliação tem como objetivo investigar se a gamificação pode trazer benefícios no envolvimento dos utilizadores. Ao incluir gamificação, procura-se criar uma experiência de utilização envolvente e satisfatória, promovendo a motivação e o interesse dos utilizadores na participação ativa e contínua da aplicação.

1.3 Organização do Documento

A estrutura deste documento foi organizada de forma a fornecer uma compreensão lógica do desenho e desenvolvimento do sistema AQUA, bem como das avaliações realizadas ao longo do processo. No entanto, é importante mencionar que a organização dos capítulos não reflete a ordem cronológica do progresso do trabalho desenvolvido, pois este teve uma natureza muito iterativa.

O **Capítulo 2** é dedicado à revisão de literatura, onde são explorados tópicos relevantes para o contexto da dissertação. Ele inclui uma análise de aplicações de ciência cidadã existentes, incluindo a aplicação AQUA desenvolvida no âmbito da dissertação de outra aluna, e explora o potencial da gamificação como motivador para os utilizadores. Também, é abrangida uma revisão de tecnologias relevantes para o desenvolvimento do sistema.

O **Capítulo 3** inicia com uma introdução ao conceito da aplicação móvel AQUA. Posteriormente, detalha a evolução do *design* da aplicação desde a criação do protótipo de alta fidelidade até a apresentação do seu *design* final. Isto é essencial para compreender o estado inicial e o progresso da aplicação móvel, que é a principal contribuição desta dissertação. Além disso, é feita uma descrição dos primeiros testes de usabilidade e de como o *feedback* recolhido influenciou o redesenho da aplicação. Finalmente, o Capítulo termina com uma secção que descreve em detalhe o desenho e funcionalidades da versão final da aplicação.

No **Capítulo 4** é feita a especificação e descrição dos requisitos do sistema, dos casos de uso, dos atributos de qualidade considerados mais relevantes para o sistema e da sua arquitetura, que serviram de base para o desenvolvimento do sistema AQUA e que foram feitos em paralelo com o redesenho da interface e interação da aplicação móvel.

No **Capítulo 5** são destacados aspetos relevantes para o desenvolvimento da aplicação móvel AQUA, da API e do armazenamento de dados online. Além disso, são detalhados testes funcionais relacionados com a funcionalidade de *tracking* das rotas de viagem da aplicação móvel, realizados aquando a finalização da implementação da respetiva funcionalidade.

O **Capítulo 6** descreve a avaliação do protótipo em duas subsecções: A primeira, avaliou uma versão preliminar da versão do AQUA, que é descrita na subsecção 6.1. Esta avaliação foi feita com especialistas que trabalham na área do Desenho de Media Interativos e Interação Humano-Computador. É importante ressaltar que esta subsecção termina com a descrição das alterações feitas para a versão final, com base no *feedback* dos utilizadores; A segunda subsecção descreve a avaliação final realizada com profissionais de empresas de observação de cetáceos no contexto de uso real usando a versão final da aplicação, sendo esta descrita na subsecção 6.2.

No **Capítulo 7** é apresentada uma discussão que abrange todo o processo de desenvolvimento do sistema AQUA, destacando o impacto das avaliações no sistema e analisando os resultados obtidos em relação às QI propostas.

No **Capítulo 8** são identificadas e discutidas as limitações deste trabalho, juntamente com propostas para ultrapassá-las no futuro.

Por fim, o **Capítulo 9** finaliza o documento com as conclusões desta dissertação, destacando as suas contribuições.

2 Revisão de Literatura

Nesta secção são apresentadas algumas aplicações móveis de ciência cidadã que serviram de inspiração para o *design* da aplicação AQUA, incluindo uma análise comparativa entre elas. Também, é introduzido o conceito de gamificação, destacando os seus benefícios e como pode ser aplicado em aplicações de ciência cidadã para envolver os utilizadores e melhorar a recolha dos dados. Por fim, é feita uma comparação de várias tecnologias consideradas para o desenvolvimento do sistema, assim como a fundamentação das escolhas de seleção.

2.1 Aplicações de Ciência Cidadã Existentes

Atualmente, com o avanço tecnológico, são muitas as empresas que têm procurado envolver os cidadãos na contribuição para a ciência [16], através da utilização dos seus *smartphones*. Geralmente, estes sistemas têm como objetivo criar fortes conexões entre os cidadãos e os cientistas, enquanto concede aos utilizadores o enriquecimento de conhecimentos e uma melhor compreensão da ciência [17].

No contexto vegetal, a aplicação móvel “*Report Invasive Plants*” permite que vários utilizadores reportem avistamentos de plantas invasoras numa determinada área, através da identificação da espécie e da sua localização, e captura de imagens da planta observada. Esta informação ajuda a controlar a extensão das plantas invasoras e evitar que os habitats de animais selvagens sejam destruídos [18]. Além de permitir aos utilizadores aprender mais sobre as respetivas espécies, a aplicação permite a visualização de todas as espécies reportadas num mapa (ver *screenshot* 1 da figura 1).

“*Seek by iNaturalist*”, incluída no contexto vegetal e animal, fornece aos utilizadores uma ferramenta para a identificação de plantas e animais que o utilizador encontra, através de mecanismos de reconhecimento de imagem [19]. À medida que o utilizador explora e identifica novas espécies, são desbloqueados distintivos pela observação de espécies distintas (ver *screenshot* 2 da figura 1) e participação em desafios.

No contexto marinho, aplicações móveis como “*IWDG Reporting App*”, “*Sharktivity*”, “*Whale and Dolphin Tracker*”, e “*Seafari*”, têm servido de apoio aos cientistas e biólogos no reporte de dados de avistamentos de baleias, golfinhos e outras espécies marinhas.

“*IWDG Reporting App*” tem como objetivo envolver não apenas os cidadãos, como também investigadores, operadores de barcos, entusiastas da vida selvagem, e os observadores de baleias, na recolha de dados sobre avistamentos em terra ou durante uma viagem de barco nas águas irlandesas [20]. Os dados recolhidos são, posteriormente, validados para pesquisa, conservação e bem estar [21]. No *screenshot* 4 da figura 1, são apresentados alguns dados que são recolhidos e que são, posteriormente, validados para pesquisa, conservação e bem estar [21]. Para além do reporte de avistamentos, os utilizadores podem reportar incidentes, como, por exemplo, a observação de uma baleia morta [22] (ver *screenshot* 3 da figura 1).

Antes do desenvolvimento da aplicação móvel, a *Irish Whale and Dolphin Group* (IWDG) já possuía um *website* que permitia efetuar o reporte dos avistamentos e incidentes [22]. Contudo, a decisão de implementar uma aplicação móvel com as mesmas funcionalidades deveu-se ao facto de que a recolha dos dados pudesse ser feita em modo *offline* [22], e fosse uma ferramenta de rápido acesso [20], visto que os utilizadores possivelmente terão os seus *smartphones* por perto no

momento dos avistamentos. Quando não há ligação à *Internet*, a aplicação armazena os dados dos avistamentos localmente, e submete-os, automaticamente, quando a ligação é estabelecida [22].

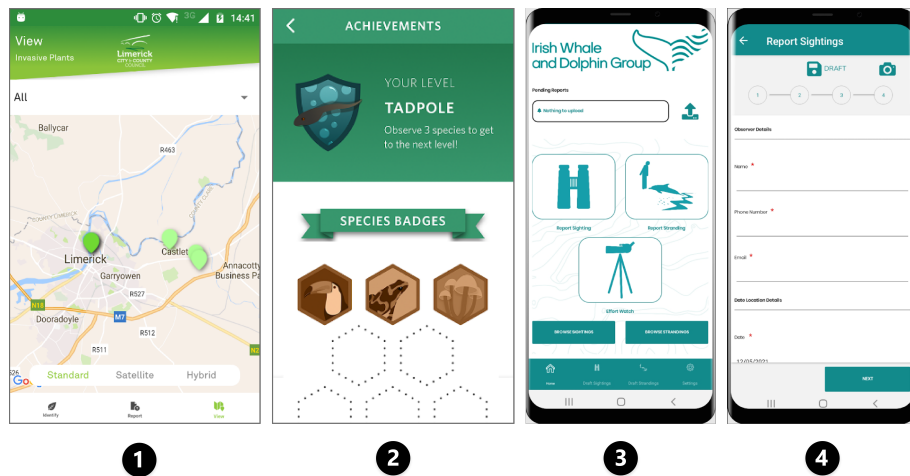


Figura 1. *Screenshots de Aplicações:* (1) Aplicação “*Report Invasive Plants*”, mapa com espécies reportadas; (2) Aplicação “*Seek by iNaturalist*”, visualização de distintivos obtidos e nível do utilizador; (3-4) Aplicação “*IWDG Reporting App*”, ecrã principal e ecrã formulário para reportar avistamentos.

“*Sharktivity*” é uma aplicação móvel desenvolvida para conscientizar a população em relação à presença de tubarões brancos nas proximidades. Esta aplicação tem como objetivos apoiar a investigação científica, melhorar a segurança pública, e educar a comunidade [23]. A aplicação apresenta diversas funcionalidades, sendo a principal, a visualização de avistamentos de tubarões a partir de um mapa [24] (ver *screenshot 2* da figura 2). Contudo, para a visualização dos dados dos avistamentos, é necessário que os utilizadores reportem a informação sobre os avistamentos de tubarões na costa. Para isso, permite que os utilizadores carreguem fotografias e adicionem informação sobre as suas observações [23] (ver *screenshot 1* da figura 2). A aplicação fornece informações sobre os movimentos e pesquisas dos tubarões brancos, aumentando a conscientização e facilitando a coexistência pacífica entre a população e esta espécie de tubarão, sendo os utilizadores notificados, por exemplo, quando uma praia apresenta perigo, com base nos avistamentos reportados [24].

À semelhança das aplicações anteriores, a aplicação “*Whale and Dolphin Tracker*” concede aos utilizadores uma ferramenta para contribuir com a ciência através do reporte de avistamentos de espécies marinhas em qualquer parte do mundo [25]. As contribuições dos utilizadores são adicionadas a uma base de dados global que ajuda os cientistas a monitorizar as espécies marinhas, determinar padrões em relação à maneira como se distribuem pelo mundo, e investigar a forma como elas se comportam no seu habitat natural [26]. Para além disso, permite a visualização de um mapa com todos os avistamentos reportados (ver *screenshot 4* da figura 2), e que os utilizadores indiquem a espécie observada como “*Unknown*”, caso não tenham conhecimento sobre a espécie ou estejam a uma distância que limite a sua distinção (ver *screenshot 3* da figura 2).

A aplicação móvel “*Seafari*” permite aos utilizadores registados reportar avistamentos de mamíferos marinhos em qualquer parte do mundo, e visualizar informação sobre as espécies (ver *screenshot 5* da figura 2), fotografias, e mapas com a distribuição de cada espécie marinha nas costas da África Oriental e Austral [27].

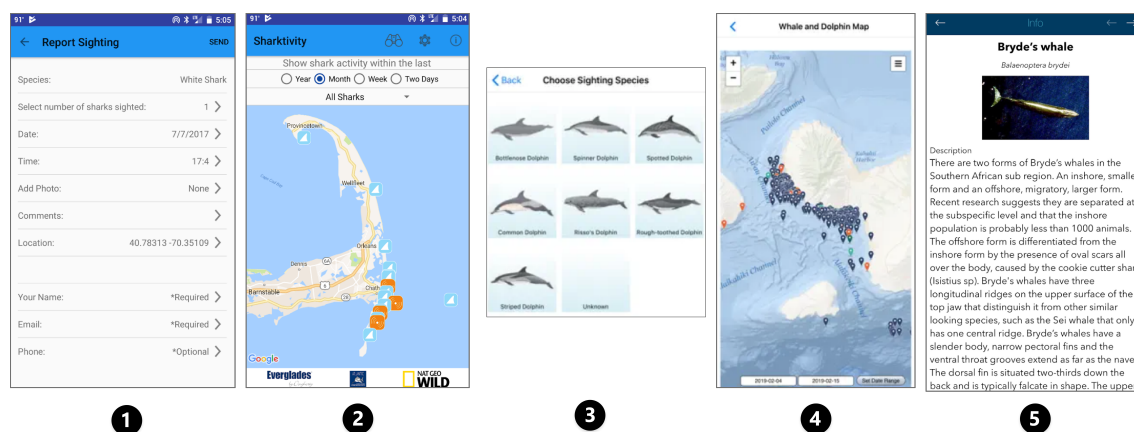


Figura 2. Screenshots de Aplicações: (1-2) Aplicação “Sharktivity”, ecrã para reportar avistamentos e ecrã de mapa de avistamentos; (3-4) Aplicação “Whale and Dolphin Tracker”, ecrã com lista de opções de espécies (incluindo a opção “Unknown”), e ecrã com mapa de avistamentos; (5) Aplicação “Seafari”, ecrã com informação sobre uma espécie.

Tabela 1. Comparação entre funcionalidades existentes e inclusão de gamificação em cada aplicação

Aplicação	Gamificação	Funcionalidades				
		F1	F2	F3	F4	F5
Report Invasive Plants	—	✓	—	✓	✓	—
Seek by iNaturalist	✓	✓	—	✓	✓	—
IWDG Reporting App	—	✓	✓	—	—	—
Sharktivity	—	✓	—	✓	—	—
Whale and Dolphin Tracker	—	✓	—	✓	—	✓
Seafari	—	✓	—	—	✓	—

F1: Reportar avistamentos; F2: Reportar incidentes; F3: Mapa com avistamentos; F4: Informação sobre espécies; F5: Ver rota da viagem; ✓: Contém; —: Não contém.

Na tabela 1 é apresentada uma comparação das aplicações relativamente às funcionalidades que cada uma possui, como, também, à inclusão de elementos de gamificação ou não. É possível observar que todas as aplicações permitem a recolha de dados dos avistamentos das espécies em questão, embora todas o façam de maneira distinta. Apesar das vantagens das mesmas, certas escolhas de *design* influenciam a sua eficácia na forma de reportar ou visualizar dados. Por exemplo, no caso das aplicações “IWDG Reporting App” e “Sharktivity”, a recolha de dados é feita através de formulários simples sem nenhuma apresentação de imagens, o que torna o sistema pouco informativo, visto que espera que os utilizadores já possuam algum conhecimento em relação às espécies marinhas. Desta forma, apenas utilizadores experientes conseguirão reportar avistamentos de forma acertada.

Na tabela 1, é possível observar que apenas a aplicação “Seek by iNaturalist” contém elementos de gamificação. Adicionalmente, com base nos dados da tabela 2, verifica-se que esta aplicação se destaca em comparação às restantes pelo elevado número de *downloads* e elevada avaliação média que apresenta. Por um lado, o número de *downloads* pode ser influenciado pela publicidade, contudo, e tendo em conta a avaliação média, o mesmo poderá ser um indicativo que a gamificação é um possível impulsionador do envolvimento dos utilizadores na utilização da aplicação.

Tabela 2. Comparação entre as avaliações médias e número de downloads de cada aplicação

Aplicação	Avaliação Média ^{ab}	Nº Downloads ^a
Report Invasive Plants	—	> 1000
Seek by iNaturalist	4.8	> 1.000.000
IWDG Reporting App	2.2	> 500
Sharktivity	3.9	> 100.000
Whale and Dolphin Tracker	2.3	> 5000
Seafari	—	> 1000

—: Dado não disponível;

^a Com base nos dados da Google Play Store.

^b Avaliação média dos utilizadores, numa escala de 0 a 5.

As aplicações “*Report Invasive Plants*” e “*Whale and Dolphin Tracker*” permitem visualizar um mapa com os avistamentos reportados na aplicação. Quando um utilizador visualiza o mapa, são apresentados marcadores representando a localização do utilizador quando o avistamento foi reportado. Porém, a localização do utilizador nem sempre corresponde à localização da espécie avistada, visto que a mesma poderá ter sido observada longe do utilizador. Como estas aplicações têm como objetivo envolver a comunidade, ao longo do tempo o número de avistamentos reportados será muito elevado. Dada situação, para além de dificultar a visualização dos marcadores no mapa devido à sobreposição de marcadores que poderá ocorrer, poderá causar uma diminuição no desempenho dos sistemas.

Além disso, quando uma espécie desconhecida é avistada, apenas a aplicação “*Whale and Dolphin Tracker*” fornece aos utilizadores a opção de indicar que é uma espécie desconhecida. Contudo, a indicação de espécies desconhecidas não significa que o utilizador não conheça algumas características sobre o animal.

Finalmente, é essencial referir como trabalho relacionado a aplicação AQUA desenvolvida no âmbito da dissertação da aluna do IST (que no contexto desta dissertação denominou-se por AQUA 1.0). Esta aplicação permitiu a realização de um estudo [28] que tinha como foco avaliar a consciencialização dos turistas em relação à biodiversidade marinha. Ao analisar e testar a aplicação desenvolvida para avaliar o funcionamento da mesma, e com base no estudo realizado pela aluna, foi possível identificar diversas limitações significativas que impactavam a sua funcionalidade, usabilidade e desempenho, nomeadamente:

- Os utilizadores sentiam-se desmotivados ao utilizar a aplicação móvel, visto que não eram recompensados ao reportar as suas contribuições;
- Os dados recolhidos a partir da aplicação móvel não podiam ser acedidos por nenhum sistema externo, devido à inexistência de uma API no sistema;
- O código desenvolvido apresentava documentação escassa e necessitava de otimização e de uma melhor organização;
- O tempo de inicialização da aplicação e das comunicações com o servidor, como ao obter os dados de todos os avistamentos ao aceder ao mapa de avistamentos, eram bastante longas (aproximadamente 5 minutos), acabando por afetar enormemente a experiência dos utilizadores;

- Durante as viagens de observação de cetáceos, onde a conexão à Internet é muito instável, havia perda de dados quando não havia conexão à Internet, o que resultava em perda de informação dos avistamentos reportados pelos utilizadores;
- Os utilizadores sentiam dificuldade em registar corretamente as informações sobre os avistamentos devido à complexidade do processo de reportar avistamentos na aplicação (ver figura 113 do Apêndice B.3);
- Os dados obtidos a partir do GPS do dispositivo eram muito imprecisos, o que comprometia a localização exata dos avistamentos;
- O mapa de avistamentos era muito lento e difícil de visualizar devido ao elevado número de avistamentos reportados nas mesmas zonas.

2.2 Gamificação e o seu Potencial

A gamificação corresponde à aplicação de elementos de jogos em contextos *non-gaming*, com o objetivo de alterar o comportamento das pessoas, aumentar o envolvimento dos utilizadores, promover a aprendizagem, ou resolver problemas específicos [13]. De acordo com Chen J. [29], a gamificação permite colocar os utilizadores dentro de uma “*flow zone*”, que representa a sensação de foco completo na realização de uma atividade com um elevado nível de satisfação, excluindo todo o aborrecimento e frustração que poderia surgir aquando a ausência da mesma.

Na ciência cidadã, onde a motivação dos voluntários é temporária e pode variar constantemente [12], o uso de gamificação tem demonstrado aumentar o envolvimento e a qualidade dos dados recolhidos [30], assim como aumentar a participação e a qualidade do trabalho colaborativo dos utilizadores [31]. Para que a gamificação seja bem-sucedida, é necessário ter em consideração dois tipos de motivação: a motivação extrínseca e a motivação intrínseca [32].

A motivação extrínseca está relacionada com o ambiente e com a interação com outros utilizadores. Na gamificação esta motivação pode ser explorada por meio de recompensas, que servem como reforço positivo para os utilizadores por determinados comportamentos [33].

Por outro lado, a motivação intrínseca está ligada à vontade interna do utilizador em realizar uma determinada tarefa, sem pensar nas dificuldades envolvidas. Esta motivação não depende do ambiente ou de outras pessoas, apenas das escolhas individuais do utilizador [32]. Portanto, é considerada mais duradoura ao longo do tempo, visto que não depende de fatores externos [34]. Além disso, é mais comum que as pessoas intrinsecamente motivadas realizem as tarefas com maior entusiasmo, devido à falta de pressão na forma como devem ser realizadas, permitindo a exploração de maneiras únicas de resolver os mesmos problemas.

Ao implementar gamificação numa aplicação de ciência cidadã é necessário criar um equilíbrio nestas duas formas de motivação, de forma a garantir que haja um aumento do envolvimento dos utilizadores e uma melhoria na qualidade das contribuições. Este equilíbrio pode ser alcançado através da adição de diversas mecânicas de jogo que tornam o sistema desafiante, divertido e satisfatório, como, por exemplo, pontos, níveis, desafios, bens virtuais, *leaderboards*, recompensas, entre outros [35].

Apesar da grande variedade de mecânicas de jogo, os pontos, níveis e *leaderboards* são consideradas as mais básicas, e as mais utilizadas em contextos de jogo e *non-gaming* [36]. De acordo com

Mekler *et al.* [37], a sua implementação é uma maneira fácil, viável e eficaz de motivar os utilizadores e, conseqüentemente, direcionar os seus comportamentos para os objetivos pretendidos.

Krath e Korfflesch [38] propuseram um conjunto de princípios de *design* para a gamificação e sistemas persuasivos. Estes princípios têm como objetivo alcançar resultados comportamentais tanto individuais quanto sociais, ao mesmo tempo que promovem experiências agradáveis, e servem de apoio na seleção de mecânicas apropriadas para alcançar os objetivos pretendidos.

Alguns princípios relacionados com o comportamento individual incluem fornecer *feedback* positivo imediato, introduzir incentivos comportamentais, fornecer dados para monitorização pessoal e visualização do progresso. Em relação aos princípios de comportamento social, destacam-se a possibilidade de comparação e competição social, bem como a conquista de reconhecimento social. Além disso, alguns princípios orientados pelo sistema, incluem personalização dos conteúdos e mecânicas do sistema, ajuste gradual da dificuldade ao longo do tempo e garantia de excitação contínua através de conteúdo novo ou oculto.

Existem diversas aplicações móveis que têm aplicado estes princípios e demonstrado um enorme impacto no envolvimento dos seus utilizadores, como, por exemplo, o “Duolingo” [39]. O “Duolingo” é uma aplicação móvel para o ensino de idiomas, com mais de 40 milhões de utilizadores ativos mensalmente [40]. Ele utiliza a gamificação para tornar o processo de aprendizagem mais envolvente. Os utilizadores ganham conquistas quando completam as tarefas corretamente, além de aprenderem um novo idioma. A aplicação inclui elementos de gamificação, como *rankings*, pontos de experiência, barras de progresso e “streaks” (sequência de sucesso contínua) para reforçar positivamente o progresso dos utilizadores. Além disso, utiliza aspetos de influência social para envolver os utilizadores e motivá-los a utilizar a aplicação e progredir juntamente com os seus amigos.

Outro exemplo de aplicação que utiliza elementos de gamificação é o Fitbit [41], uma aplicação móvel de saúde e condicionamento físico. Esta aplicação permite que os utilizadores monitorizem estatísticas básicas sobre o seu desempenho físico e mantenham a sua motivação. Ela inclui desafios, como atingir a meta de 8000 passos diários ou fazer exercício físico durante 15 minutos, além de permitir que os utilizadores compitam com os seus amigos, visualizando o desempenho de cada um. E, ainda, quando metas específicas são alcançadas, os utilizadores desbloqueiam distintivos.

2.3 Oportunidades de Contribuição

Com base nas limitações identificadas na aplicação AQUA 1.0, nas aplicações semelhantes e nas vantagens da gamificação apresentadas, não se verificou a existência de nenhuma aplicação de ciência cidadã no contexto marinho que inclua todas as funcionalidades apresentadas na tabela 1, nem elementos de gamificação. Desta forma, a aplicação AQUA permitirá, não apenas incluir todas as funcionalidades numa única aplicação, como, também, incluir a gamificação como impulsionador do envolvimento dos utilizadores, sendo o seu estudo um contributo expectável.

Além disso, as limitações do AQUA 1.0 permitiram propor algumas modificações no sistema com o objetivo de melhorar a funcionalidade, usabilidade e experiência dos utilizadores na aplicação, como também garantir que os dados reportados são preservados, nomeadamente:

- **Adição de Elementos de Gamificação:** Tendo em conta a falta de motivação demonstrada pelos utilizadores, propôs-se a adição de elementos de gamificação à aplicação móvel, como coleção de cartas de espécies e *ranking* de utilizadores, como possíveis impulsionadores do envolvimento dos utilizadores;

- **Desenvolvimento de uma API:** Com o objetivo de permitir o acesso aos dados recolhidos a partir da aplicação móvel a qualquer sistema externo, propôs-se o desenvolvimento de uma API dedicada ao sistema AQUA, permitindo o acesso autorizados aos dados dos avistamentos e incidentes por parte de outros sistemas;
- **Redesign da aplicação:** Sugeriu-se realizar um *redesign* de determinadas partes da aplicação, levando em consideração as dificuldades enfrentadas pelos utilizadores durante a sua utilização, com o objetivo de melhorar a usabilidade e experiência geral dos utilizadores;
- **Adição de *clusters* e filtros de tempo no mapa de avistamentos:** Com o objetivo de melhorar a visualização dos avistamentos no mapa e sobreposição de marcadores, foi proposta a implementação de *clusters*, que permite agrupar avistamentos próximos e apresentar apenas um único marcador no mapa para esse grupo. Além disso, sugeriu-se adicionar filtros de tempo, permitindo que o utilizador visualize apenas os avistamento dentro do intervalo selecionado;
- **Área do avistamento:** Em vez de apresentar um único ponto para a localização dos avistamentos, sugeriu-se criar uma área de incerteza à volta das coordenadas obtidas. Esta abordagem tem em consideração que as coordenadas obtidas se referem à localização do dispositivo e não necessariamente à localização exata da espécie avistada. Desta forma, a representação dos avistamentos nos mapas é mais realista e tem em conta a incerteza da localização das espécies;
- **Utilização do SQLite como base de dados local:** Para ultrapassar a perda de dados devido à falta de conexão à Internet em alto mar, foi proposta a implementação do SQLite como base de dados local na aplicação móvel, que permite armazenar os dados dos avistamentos, incidentes e rotas de viagem reportados sem conexão até o utilizador decidir enviá-los para o servidor. Assim, a falta de conexão à Internet não teria impacto na disponibilidade da aplicação, preservando os dados reportados pelos utilizadores.

Em relação à partilha de dados científicos, algumas aplicações, como a aplicação “*Whale and Dolphin Tracker*”, referem que as contribuições dos seus utilizadores são adicionadas a uma base de dados global de acesso para cientistas. Até à data, não se teve conhecimento sobre a existência de alguma API *open-source* para a partilha de dados de avistamentos de cetáceos. Desta forma, o desenvolvimento de uma API própria para o sistema AQUA será uma valiosa contribuição para a ciência e para a conscientização sobre a biodiversidade marinha. Além disso, ao desenvolver uma API própria, será possível ter controlo total da mesma, garantindo que os dados e as configurações são compatíveis com as necessidades do sistema AQUA, bem como a adição de camadas de segurança nos acessos à informação da base de dados e garantir a privacidade de dados sensíveis.

2.4 Revisão de Tecnologias de Desenvolvimento

Para desenvolver o sistema AQUA, foi necessário realizar uma revisão de tecnologias disponíveis, de forma a conseguir realizar uma seleção adequada entre as várias opções existentes, a fim de atender as necessidades específicas do projeto.

Assim, nesta subsecção é apresentada uma revisão das diferentes abordagens de desenvolvimento móvel, das *frameworks* de desenvolvimento móvel híbrido, das *frameworks* de desenvolvimento *back-end*, e alguns sistemas gestores de bases de dados considerados mais relevantes. Além disso, são apresentadas comparações das características de cada tecnologia, incluindo as vantagens e desvantagens, que permitiram realizar a seleção de acordo com os requisitos do sistema.

2.4.1 Abordagens ao Desenvolvimento Móvel

Na tabela 3 é apresentada uma comparação do desenvolvimento nativo e desenvolvimento híbrido de aplicações móveis, em termos de desempenho, velocidade e custo de desenvolvimento, e capacidade de garantir uma boa experiência aos utilizadores, com base em [42–45].

Tabela 3. Comparação do Desenvolvimento de Aplicações Móveis Nativo e Híbrido

	Desenvolvimento Nativo	Desenvolvimento Híbrido
Desempenho	Alto	Médio
Velocidade de Desenvolvimento	Baixo	Alto
Custo de Desenvolvimento	Alto	Baixo
Experiência de Utilizador	Alto	Alto

Apesar do desenvolvimento nativo apresentar um desempenho bastante elevado em comparação ao híbrido, a velocidade e custo de desenvolvimento tornam-se uma grande desvantagem, visto que exige o desenvolvimento da aplicação móvel para cada sistema operativo (por exemplo, Android e iOS). Desta forma, sendo a aplicação AQUA desenvolvida apenas para fins de investigação, escolheu-se seguir uma abordagem híbrida, pelo facto de apresentar um tempo e custo de desenvolvimento vantajoso em comparação à nativa, e, simultaneamente, ter capacidade de garantir uma boa usabilidade e experiência de utilizador, mesmo que não atinja o mesmo nível de desempenho do desenvolvimento nativo. Ainda qualquer modificação pode ser facilmente aplicada, visto que basta realizar as alterações num único código base comum entre todas as plataformas.

2.4.2 Frameworks de Desenvolvimento Móvel Híbrido

Apesar de haver diferentes opções de *frameworks* para o desenvolvimento híbrido, como o Flutter, o Ionic, React Native e Xamarin, a aplicação AQUA 1.0 encontrava-se numa fase avançada de desenvolvimento utilizando a *framework* Flutter.

O Flutter é uma *framework open-source* que permite o desenvolvimento de aplicações para o sistema operativo Android e iOS, compilados nativamente, utilizando um único código base [46]. Esta *framework* concede a capacidade de desenvolver aplicações com uma aparência e sensação semelhante às desenvolvidas utilizando uma abordagem de desenvolvimento nativa, com a vantagem de que o custo e tempo de desenvolvimento é muito reduzido. Além disso, o Flutter permite desenvolver aplicações para sistemas operativos diferentes simultaneamente.

No Flutter, o *layout* é construído principalmente através de *widgets* [47]. Os *widgets* são elementos fundamentais da interface de utilizador, sendo estes responsáveis por descrever a aparência e o comportamento dos diferentes elementos da interface. Quando estamos a criar um *layout*, estes *widgets* são organizados de forma hierárquica. Existem dois tipos principais de *widgets*, os *stateless*, que são usados para elementos da interface que não precisam mudar ou reagir a interações do utilizador, e os *stateful*, que podem mudar, responder a eventos e manter um estado interno que pode ser alterado para responder dinamicamente às interações dos utilizadores.

Dada a experiência prévia na manipulação desta *framework* e a existência de componentes já desenvolvidos, seleccionou-se o Flutter para a reimplementação da aplicação. Desta forma, foi

possível reutilizar alguns componentes já desenvolvidos, acelerando ainda mais o tempo de desenvolvimento.

2.4.3 Frameworks de Desenvolvimento *Back-end*

Na tabela 4 é apresentada uma comparação de quatro das mais conhecidas *frameworks* para desenvolvimento *back-end*, com base em [48–53].

Tabela 4. Comparação entre Frameworks de Desenvolvimento Back-end

	Node.js	ASP.NET Core	Django	Laravel
Linguagem	JavaScript	C#	Python	PHP
Desempenho	Alto	Alto	Alto	Médio
Open-Source	Sim	Sim	Baixo	Alto
Tempo de Desenvolvimento	Médio	Baixo	Baixo	Alto
Desenvolvimento API	Sim	Sim	Sim	Sim
Comunidade	Grande	Média	Grande	Pequena

Considerando o desempenho um atributo de qualidade muito importante para o sistema AQUA, a *framework* Laravel não é uma boa opção para o desenvolvimento do mesmo, uma vez que, para além de apresentar um desempenho inferior às restantes, requer um elevado tempo de desenvolvimento e apresenta uma comunidade muito pequena, o que dificultaria o processo de desenvolvimento. Por outro lado, apesar do Node.js, ASP.NET Core e Django apresentarem um alto desempenho, o Node.js distingue-se pelo facto de requerer um maior tempo de desenvolvimento, sendo uma desvantagem para o desenvolvimento do sistema AQUA. Ao comparar o ASP.NET Core com o Django, pode-se observar que ambos apresentam muitas semelhanças. Contudo, apesar da comunidade do ASP.NET Core ser mais reduzida, devido à experiência prévia com a linguagem de programação C#, escolheu-se esta *framework* para o desenvolvimento do *back-end*. Esta escolha permitiu garantir um alto desempenho e um tempo de desenvolvimento reduzido.

2.4.4 Sistemas Gestores de Bases de Dados

De acordo com [54], o ASP.NET Core pode realizar o acesso aos dados da base de dados utilizando o Entity Framework (EF), que utiliza uma técnica conhecida por *Object-relational mapping* (ORM). Esta técnica reduz a complexidade da programação orientada por objetos utilizando bases de dados relacionais, e aumenta a produtividade no desenvolvimento. Desta forma, considerando que o EF requer uma base de dados relacional, tem-se três sistemas gestores de base de dados *open-source* mais utilizados com o ASP.NET Core: MySQL, PostgreSQL, e SQLite [54].

Havendo necessidade de armazenar dados localmente, o SQLite permite que os dados sejam armazenados no dispositivo, sem necessidade de conexão à *Internet* pelo facto do mesmo não requerer um servidor. Por outro lado, há necessidade de armazenar dados numa base de dados *online*, permitindo que os mesmos possam ser acedidos a partir de qualquer dispositivo. O PostgreSQL permite o armazenamento *online* dos dados, e um desenvolvimento mais rápido, para além de ser mais completo, em comparação ao MySQL [55]. Ainda, o PostgreSQL possibilita o armazenamento de dados geospaciais através da extensão PostGIS. Assim, escolheu-se o SQLite para o armazenamento *offline*, e o PostgreSQL para o armazenamento *online*.

3 Design da Aplicação AQUA

Nesta secção, começa-se por introduzir o conceito da aplicação AQUA. Posteriormente, é apresentado o *design* inicial proposto para o redesenho da aplicação, com base nas propostas de melhorias identificadas para aplicação AQUA 1.0. Além disso, é fornecida uma descrição detalhada dos testes de usabilidade realizados para avaliar a usabilidade e aspetos relacionados com as melhorias efetuadas.

Na última subsecção, “Design Final”, é descrito o resultado final de todo o processo de desenho da aplicação móvel, sendo apresentado o *design* final da interface de utilizador, que inclui as alterações realizadas após a realização da avaliação descrita na subsecção 6.1 do Capítulo 6.

3.1 Conceito da Aplicação

A aplicação AQUA é uma aplicação móvel de ciência cidadã desenvolvida com o intuito de apoiar cientistas, biólogos marinhos e empresas que realizam viagens de observação de cetáceos na recolha de informação sobre avistamentos de espécies marinhas. Desta forma, é uma ferramenta para a recolha desta informação, que posteriormente pode ser acedida por profissionais, contribuindo para a ciência e para a monitorização do ecossistema marinho.

Tendo em conta o seu objetivo, a aplicação tem como utilizadores-alvo qualquer pessoa que realize atividades marítimas, como a observação de cetáceos e a pesca, considerando que são utilizadores que podem contribuir para a recolha desta informação mais facilmente.

A aplicação AQUA oferece aos seus utilizadores diversas funcionalidades, nomeadamente:

- **Autenticação:** Permite aos seus utilizadores a criação de uma conta para autenticar-se no sistema, garantindo-lhes acesso a todas as funcionalidades da aplicação, assim como às informações que ela disponibiliza.
- **Reportar Avistamentos:** Os utilizadores podem reportar os avistamentos das espécies, capturando imagens das suas observações e preenchendo a informação sobre as espécies avistadas, como o número de indivíduos e comportamentos de cada espécie identificada.
- **Reportar Incidentes:** A identificação de incidentes no ecossistema marinho, como avistamentos de espécies feridas ou mortas, também é algo muito relevante, permitindo tomar medidas de segurança e proteção para as espécies para reduzir os riscos de ocorrência de novos incidentes.
- **Tracking da Rota da Viagem:** Os utilizadores podem fazer o *tracking* das rotas das suas viagens. Isto permite melhorar a experiência dos utilizadores, envolvendo-os mais na atividade e permitindo que registem as suas viagens como memórias.
- **Visualização de Informação sobre Espécies:** A aplicação permite aos utilizadores visualizar informações detalhadas sobre cada espécie marinha, proporcionando uma oportunidade para aprender mais sobre elas. Isto facilita a identificação das espécies, fornecendo aos utilizadores novos conhecimentos que os permitam recolher informação mais precisa sobre os avistamentos.
- **Visualização de Mapa de Avistamentos:** Os avistamentos reportados na aplicação são representados por marcadores num mapa interativo. Isto permite ao utilizador explorar e visualizar informações detalhadas sobre cada avistamento, e fornece aos utilizadores uma perspetiva das áreas mais propícias à ocorrência de espécies marinhas.

- **Desbloqueio de Cartas de Espécies:** Quando os utilizadores reportam o avistamento de uma espécie pela primeira vez, é desbloqueada uma carta da respetiva espécie. Ao colecionar as cartas, os utilizadores podem visualizar as espécies que já avistaram ao longo das suas experiências. Além disso, permite incentiva-los a explorar novas espécies marinhas e a continuar a contribuir para a ciência.
- **Desbloqueio de Distintivos e Títulos:** A aplicação AQUA apresenta um conjunto de distintivos que podem ser desbloqueados pelos seus utilizadores. À medida que os utilizadores contribuem na recolha e partilha de dados sobre as espécies marinhas, desbloqueiam novos distintivos, tendo cada distintivo um título único. Ao desbloquear um distintivo, o respetivo título pode ser apresentado no perfil do utilizador, destacando-o em relação aos restantes utilizadores.
- **Visualização de *Ranking* de Utilizadores:** De forma a impulsionar o espírito competitivo, a aplicação disponibiliza *rankings* que refletem a participação e desempenho dos utilizadores, fornecendo a opção de visualizar um *ranking* global, que apresenta todos os utilizadores do sistema, ou um *ranking* local, que mostra os utilizadores da mesma região. Estes *rankings* são ordenados com base num critério específico selecionado pelo utilizador, como o número de avistamentos, número de cartas desbloqueadas, ou número de distintivos desbloqueados.

3.2 Protótipo de Alta Fidelidade

Nesta secção são apresentadas algumas tomadas de decisão consideradas mais relevantes em relação ao protótipo de alta fidelidade da aplicação móvel AQUA criado utilizando a ferramenta Adobe XD¹. O *redesign* da aplicação teve como objetivo resolver as limitações da aplicação AQUA 1.0, assim como melhorar a usabilidade e experiência de utilização da aplicação.

Considerando que a funcionalidade de reportar avistamentos é a mais importante da aplicação AQUA, uma vez que é a partir dos avistamentos que é possível contribuir para a ciência, realizou-se modificações significativas na interface relacionada a esse processo.

No *design* original, conforme apresentado no *screenshot* 1 da figura 113 do Apêndice B.3, havia um campo de texto para pesquisar espécies pelo nome, seguido de uma lista das cinco espécies mais avistadas. No entanto, essa lista era apresentada verticalmente, ocupando grande parte do ecrã e ocultando as restantes espécies na parte inferior. Além disso, no *screenshot* 2 da figura, é possível observar que havia a opção de visualizar a lista de espécies por tamanho, que também ocupava uma quantidade significativa de espaço no ecrã, dificultando a visualização de outros elementos.

Após a seleção das espécies, no *screenshot* 3 da figura, cada espécie selecionada tinha um formulário para preencher informações específicas da espécie avistada, além de dados gerais do avistamento, como as coordenadas, data e hora. Além disso, no *screenshot* 4 da figura, a versão anterior da aplicação permitia apenas a adição de fotografias, sem a possibilidade de visualizá-las ou removê-las posteriormente.

Desta forma, tendo em consideração as dificuldades enfrentadas pelos utilizadores, realizou-se um *redesign* da interface para reportar avistamentos. Dado que a captura de imagens das espécies avistadas é um passo crucial, pois as mesmas podem estar em constante movimento ou fugir rapidamente, decidiu-se que esse seria o primeiro passo do processo (ver *screenshot* 1 da figura 3).

¹<https://helpx.adobe.com/pt/xd/get-started.html>

Além disso, foi considerado relevante permitir aos utilizadores a opção de capturar tanto fotografias quanto vídeos das suas observações, visto que capturar fotografias das espécies pode ser desafiador. Após a captura das imagens, é solicitado ao utilizador a seleção das espécies avistadas (ver *screenshot 2* da figura 3). Para facilitar essa tarefa, a lista das espécies mais avistadas foi redesenhada com uma orientação horizontal, permitindo uma visão mais ampla dos elementos da interface. E, ainda, foram adicionados filtros que permitem aos utilizadores pesquisar espécies por tamanho e tipo de animal, evitando a apresentação de listas extensas e facilitando a seleção das espécies.

Em relação ao preenchimento das informações de cada espécie selecionada, o formulário foi dividido em etapas, permitindo que o utilizador compreenda quantos passos são necessários para concluir o processo. Primeiro, é solicitado ao utilizador que indique o número de indivíduos para cada espécie avistada (ver *screenshot 3* da figura 3). Em vez de fornecer uma lista de seleção com diversas opções, optou-se por apresentar seis botões com as diferentes opções, facilitando a seleção do número de indivíduos. Em seguida, é solicitada a indicação do comportamento de cada espécie avistada, fornecendo duas opções ao utilizador (ver *screenshot 4* da figura 3).

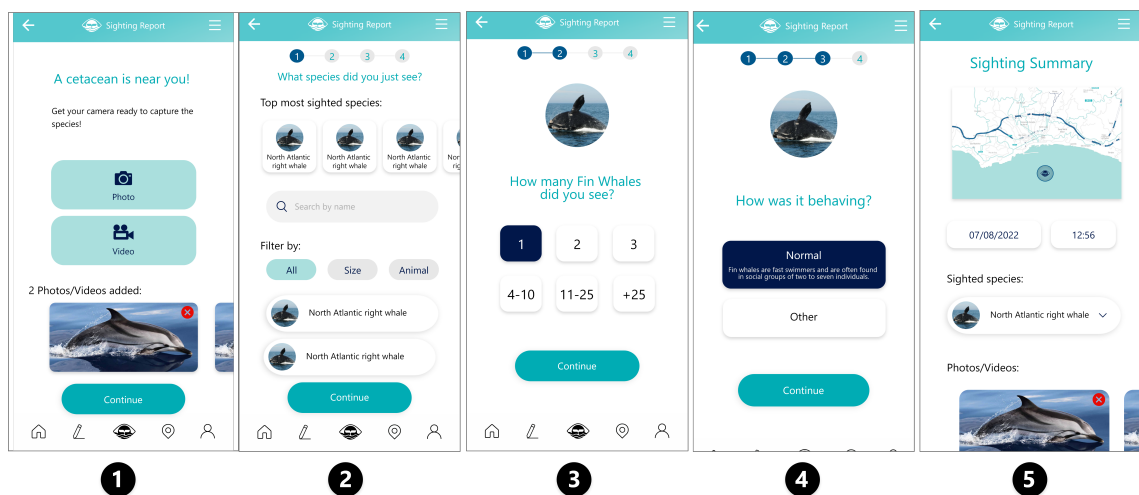


Figura 3. Protótipo de Alta Fidelidade: (1-2) Ecrãs de seleção de espécies avistadas; (3-4) Ecrãs para preenchimento de dados sobre o avistamento e as espécies avistadas;

Ao finalizar o preenchimento dos dados, é apresentado um sumário do avistamento, incluindo as informações preenchidas para as espécies avistadas (ver *screenshot 5* da figura 3). Ao contrário da versão anterior, o *redesign* teve como objetivo simplificar o formulário para reportar avistamentos, associando os dados comuns às espécies avistadas ao avistamento, em vez de repetir informações para cada espécie selecionada. Assim, as coordenadas do avistamento, a data, a hora e o comentário tornam-se dados gerais do avistamento, não sendo repetidos para cada espécie. No sumário do avistamento, é também apresentado um mapa com um marcador indicando a localização do avistamento e a respetiva área de incerteza à volta, fornecendo aos utilizadores uma noção clara do local onde as espécies foram avistadas, em vez de apresentar apenas as coordenadas em formato numérico.

Com o objetivo de promover o envolvimento dos utilizadores da aplicação AQUA, desenhou-se diversos elementos de gamificação, com base na revisão de literatura (ver detalhes na secção

3.4.2). Para recompensar os utilizadores que contribuem com o reporte dos avistamentos, criou-se uma coleção de cartas de espécies. Esta coleção permite que os utilizadores colecionem cartas de diferentes espécies que podem ser avistadas.

Além da coleção de cartas de espécies, incluiu-se um *ranking* para aumentar o espírito competitivo entre os utilizadores (ver *screenshot 2* da figura 4). O *ranking* apresenta os cinco utilizador com o maior número de avistamentos reportados, destacando-os em relação aos restantes utilizadores da aplicação. Este elemento visa aumentar a motivação dos utilizadores para reportar mais avistamentos e alcançar uma posição de destaque no *ranking*.

No ecrã da coleção de espécies, os utilizadores podem explorar as cartas das espécies que já desbloquearam, ou seja, as espécies que já avistaram, assim como aquelas que ainda estão por desbloquear. Inicialmente, o ecrã apresenta três contadores, um para golfinhos, um para baleias e um para outros animais que não se enquadram nessas categorias (ver *screenshot 1* da figura 4). Estes contadores contam o número de espécies desbloqueadas para cada tipo de animal. Desta forma, o utilizador consegue visualizar quantas espécies de cada tipo de animal faltam por desbloquear, motivando-o a completar a coleção.

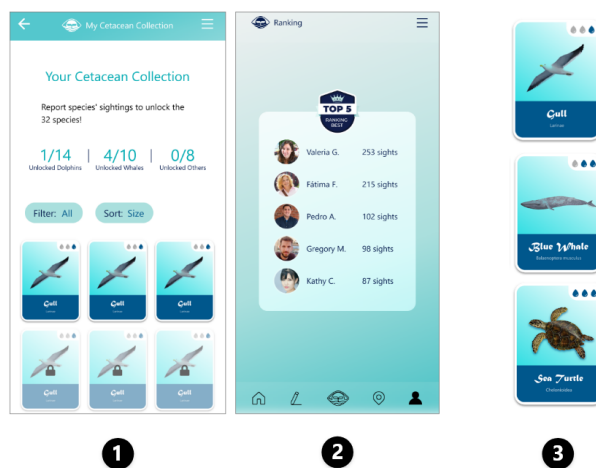


Figura 4. Protótipo de Alta Fidelidade: (1) Coleção de cartas de espécies; (2) Ranking com cinco utilizadores com mais avistamentos reportados; (3) Cartas das espécies.

A aplicação apresenta o conjunto de cartas desbloqueadas e, em seguida, as cartas bloqueadas, e oferece, ainda, opções de filtragem por tipo de animal e ordenação por tamanho ou data de desbloqueio, o que facilita a visualização das espécies.

Cada carta foi desenhada para apresentar uma ilustração da espécie, juntamente com o seu nome comum e nome científico, além de indicar a sua raridade de avistamento na ilha da Madeira (ver *screenshot 3* da figura 4). Essa raridade é representada na carta por meio de gotas de água, onde uma gota representa uma espécie comum, duas gotas representam uma espécie semirara, e três gotas representam uma espécie rara. Esta classificação das espécies por raridade tem como objetivo incentivar os utilizadores a desbloquear cartas mais raras, contribuindo para uma participação contínua da aplicação. Além disso, acredita-se que o desbloqueio de cartas de espécies raras pode ser satisfatório, melhorando a experiência de utilização da aplicação.

3.3 Testes de Usabilidade - 1ª Iteração

Nesta secção são descritos testes de usabilidade realizados para avaliar a aplicação AQUA após ter-se reimplementado todo o *front-end* em Flutter referente ao processo de reportar um avistamento, à coleção de espécies e ao *ranking*.

3.3.1 Objetivos e Protocolo

Os testes realizados tiveram como objetivos medir a usabilidade da aplicação AQUA durante o processo de reportar um avistamento, assim como verificar o impacto que a gamificação pode ter nos seus utilizadores, e, ainda, obter *feedback* em relação a possíveis melhorias no *design* da aplicação. Os mesmos foram realizados numa sala de aula na Universidade da Madeira, o que permitiu que a realização e recolha dos dados fosse feita de forma controlada.

A medição da usabilidade, assim como avaliar o impacto da gamificação numa fase inicial de desenvolvimento permitiria identificar possíveis problemas no *design* da aplicação e realizar alterações significativas para melhorar a experiência do utilizador e aumentar a sua motivação na utilização da aplicação AQUA.

Inicialmente, foi apresentado um vídeo sobre a experiência de uma viagem de barco para observação de cetáceos na ilha da Madeira, tendo sido pedido aos participantes para imaginarem que eram eles próprios a realizar essa viagem para, posteriormente, utilizarem a aplicação AQUA para reportar os avistamentos das espécies no vídeo. A ideia por trás do vídeo era simular um cenário real e tornar a experiência mais imersiva, permitindo obter dados mais precisos.

Durante a utilização da aplicação, observou-se a interação dos participantes com a mesma. Após a realização das tarefas solicitadas, pediu-se aos participantes para responder a um questionário com o objetivo de recolher dados demográficos, como também medir a usabilidade do sistema utilizando a *System Usability Scale* (SUS)².

Seguiu-se com a realização de um jogo “This or That”³. O objetivo do jogo era comparar duas versões distintas do *design* da aplicação AQUA, a versão do AQUA 1.0 e a versão redesenhada com base nas propostas de melhoria (ver figuras 114 e 115 e 116 do Apêndice B.4), com o intuito de compreender qual das opções era considerada melhor pelos participantes. Para isso, foram apresentadas imagens da interfaces das duas versões, solicitando aos participantes para fazer uma escolha entre as duas opções, indicando qual consideravam melhor e o motivo para a escolha. Desta forma, seria possível identificar se as alterações realizadas no *design* da aplicação foram consideradas como melhorias ou não, em relação à versão anterior.

Terminou-se o estudo com a realização de uma pequena entrevista com o objetivo de obter opiniões dos participantes em relação à sua experiência e motivação para reportar avistamentos. As questões abordavam aspetos específicos, como a clareza da seleção das espécies avistadas, possíveis dificuldades na procura de espécies, o impacto do desbloqueio de cartas na motivação para reportar mais avistamentos, a influência do *ranking* na vontade de se destacar em relação aos restantes utilizadores, a importância de ter um *ranking* global e local, e outras dificuldades encontradas durante a utilização da aplicação.

²<https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>

³Semelhante a um teste A/B, contudo as escolhas são mais subjetivas e baseadas apenas na preferência pessoal do participante.

Na figura 5 é apresentado um resumo da abordagem adotada para a realização dos testes.

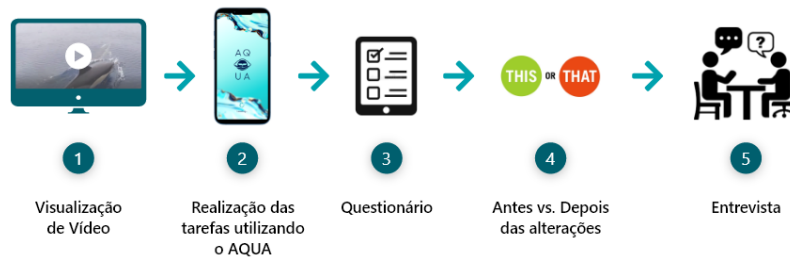


Figura 5. Abordagem adotada para os testes de usabilidade (1ª iteração)

3.3.2 Participantes

Na participação dos testes estiveram envolvidos 13 utilizadores. Os participantes seleccionados pertenciam a uma turma do 2º ciclo em Engenharia Informática. A tabela 5 sumariza os principais dados demográficos dos participantes.

Tabela 5. Características dos Participantes

Dado Demográfico	Grupo	Frequência	Percentagem
Género	Feminino	7	54%
	Masculino	6	46%
	Não indicado	0	0%
Idade	<18	4	30%
	18-24	6	46%
	25-34	1	8%
	35-44	1	8%
	45-54	1	8%
	+55	0	0%
Nacionalidade	Portuguesa (PT)	13	100%
	Outra	0	0%
Frequência da prática de observação de cetáceos	Muita	3	24%
	Pouca	5	38%
	Nenhuma	5	38%

3.3.3 Resultados

Nesta secção são apresentados os resultados dos testes realizados, dividido em duas subsecções: na primeira são apresentados os resultados quantitativos obtidos a partir do questionário SUS; na outra são apresentados os resultados qualitativos obtidos a partir da análise das observações e entrevistas.

– Resultados Quantitativos:

A partir do questionário SUS, calculou-se o *SUS Score* obtido para cada participante, e, de seguida, calculou-se o valor médio do sistema, seguindo as instruções de [56]. Obteve-se um resultado de 80.5, numa escala de 0 a 100, indicando que a aplicação era “excelente” [57] em termos de usabilidade.

– Resultados Qualitativos:

Para ilustrar os resultados, são citadas no texto algumas declarações dos participantes. As citações seguem o formato U<NúmeroDoParticipante>_<IDdaCitação>, e a lista completa de citações encontra-se no anexo C.3.1.

A partir da análise dos dados recolhidos durante as observações dos participantes aquando da realização das tarefas utilizando a aplicação AQUA, e dos dados recolhidos durante as entrevistas e do jogo “This or That”, identificou-se:

Dificuldades durante a interação com o AQUA: Durante a realização das tarefas, observou-se que vários participantes reportaram avistamentos distintos para cada espécie avistada, em vez de selecionar todas as espécies num único avistamento. Para além disso, verificou-se que todos os participantes tentaram selecionar múltiplos comportamentos para a mesma espécie, o que causou incertezas no comportamento a selecionar (U1_1, U2_1), visto que a aplicação apenas permitia selecionar um único comportamento para cada espécie.

Três participantes demonstraram dificuldade em encontrar a coleção de espécies, referindo que o ícone do botão utilizado (ícone do AQUA) já era utilizado noutros botões da interface (U3_1), o que dificultou a distinção entre eles.

Possíveis melhorias na interface do AQUA: A maioria dos participantes referiu que seria mais fácil utilizar cores distintas para as cartas de espécies de acordo com o seu grau de raridade (U1_2, U2_2, U3_2, U5_1). Ainda, dois participantes recomendaram aumentar o tamanho das imagens das espécies para facilitar a seleção (U4_1, U5_2).

Novos elementos de gamificação: A maioria dos participantes referiu que a adição de desbloqueio de troféus e distintivos para destacar os utilizadores que reportam mais avistamentos aumentaria a motivação em continuar a utilizar a aplicação (U1_3, U2_3, U4_2). A existência de um *ranking* global e de um *ranking* local (U1_4, U6_1), tal como o aumento do número de utilizadores apresentados seria algo desejável, visto que alguns consideraram que o *ranking* atual, que apenas apresentava cinco utilizadores, dificulta a visualização da posição atual do utilizador (U3_3, U5_3).

Preferências em relação às duas versões da aplicação: Todos os participantes preferiram a nova versão da aplicação AQUA com o processo de reportar avistamentos otimizado, pelo facto de considerarem que o processo é mais “simples” e permite visualizar “quantos passos faltam” até a sua conclusão (U1_5, U2_4, U3_4, U4_3, U6_2).

3.3.4 Discussão

Com base na análise dos resultados apresentados anteriormente, identificou-se vários problemas de usabilidade na interface da aplicação, apesar do *SUS Score* médio indicar que a aplicação era “excelente” [57] em termos de usabilidade.

Um dos principais problemas relatados pelos participantes foi a falta de clareza na seleção de espécies avistadas. Alguns participantes não perceberam que era possível selecionar várias espécies num único avistamento. Como solução, sugeriu-se adicionar um texto informativo que incentive os utilizadores a selecionar todas as espécies avistadas no mesmo avistamento.

Outra dificuldade identificada foi em relação à seleção dos comportamentos das espécies. No vídeo apresentado era possível visualizar que uma das espécies apresentava vários comporta-

mentos distintos. Os participantes tentaram reportar a espécie avistada e especificar os diversos comportamentos identificados, mas a aplicação apenas permitia selecionar um único comportamento por espécie. Para resolver este problema, recomendou-se adicionar a opção de permitir uma seleção múltipla de comportamentos para uma dada espécie, tendo em consideração a variedade de comportamentos que podem ser observados num conjunto de animais da mesma espécie.

Outro aspeto identificado pelos participantes foi a ambiguidade do ícone do AQUA. Este ícone era utilizado em vários botões com finalidades diferentes. Sugeriu-se, assim, a alteração do ícone do botão para aceder à coleção de espécies, com o objetivo de facilitar a identificação desta secção (ver figura 6).

Em relação à raridade das espécies, os participantes sugeriram alterar a cor das cartas de cada espécie de acordo com o nível de raridade, uma vez que nem todos compreenderam o significado das gotas de água no canto superior direito das cartas. Desta forma, a raridade da espécie passariam a ser identificada pela cor da carta e pelo número de gotas de água no canto superior direito. As espécies consideradas comuns seriam representadas por uma cor azul e uma única gota de água (ver 1 da figura 17). As espécies semiraras seriam identificadas por uma cor roxa e duas gotas de água (ver 2 da figura 17). E, por fim, as espécies raras seriam representadas por uma cor dourada e três gotas de água (ver 3 da figura 17).

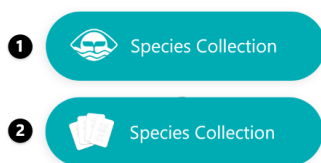


Figura 6. Botão de acesso à coleção de espécies: (1) Versão da aplicação de teste; (2) Versão de melhoria sugerida.



Figura 7. Cartas das espécies com diferentes cores: (1) Carta de espécie comum; (2) Carta de espécie semirara; (3) Carta de espécie rara.

Sugeriu-se, também, a adição de um novo elemento de gamificação que incluía o desbloqueio de distintivos para destacar e recompensar os utilizadores pelos seus comportamentos. Esta adição pode incentivá-los a continuar a utilizar a aplicação e a participar com mais frequência na observação e reporte de avistamentos. Assim, desenhou-se diversos distintivos, de forma a que cada um possuí-se um título associado que representa uma conquista específica (ver *screenshot* 2 da figura 8). A lista de distintivos propostos encontra-se no anexo C.2.

No ecrã da coleção de distintivos, sugeriu-se apresentar um contador do número de distintivos desbloqueados pelo utilizador, com o objetivo de motivá-lo a desbloquear todos os distintivos disponíveis (ver *screenshot* 1 da figura 8). Ao pressionar num distintivo desbloqueado, o utilizador pode visualizar informações detalhadas sobre ele, como o título, os requisitos para o desbloqueio e a data em que foi desbloqueado (ver *screenshot* 3 da figura 8). Além disso, recomendou-se separar o *ranking* num *ranking* global e um *ranking* local, sem limite de utilizadores apresentados, de forma a aumentar o espírito competitivo dos utilizadores.

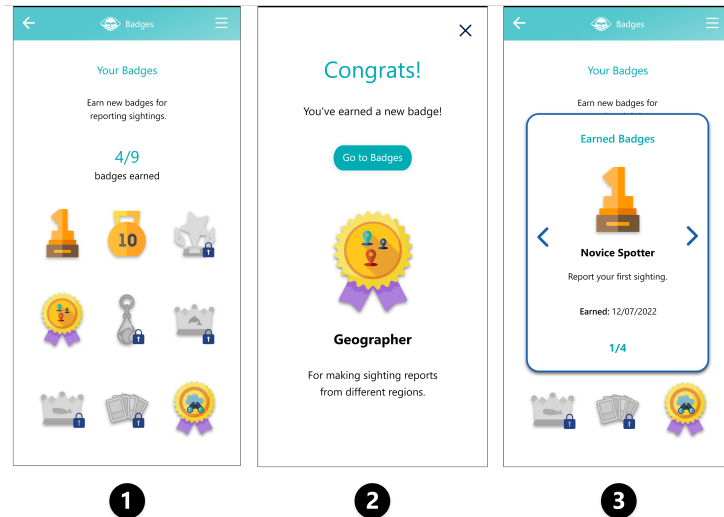


Figura 8. Protótipo de alta fidelidade: (1) Ecrã de coleção de distintivos; (2) Ecrã de desbloqueio de novo distintivo; (3) Ecrã de visualização de detalhes de um distintivo.

Os resultados do jogo “This or That” indicaram que as alterações feitas no *design* da aplicação foram consideradas pelos participantes como melhorias significativas na usabilidade.

3.4 Design Final

O *design* da aplicação AQUA seguiu um processo iterativo. Os resultados do estudo [28] com a aplicação AQUA 1.0, dos testes de usabilidade descritos anteriormente, e dos testes de usabilidade descritos na secção 6.1, permitiram identificar oportunidades de melhorias no *design* da aplicação ao longo do processo. Cada iteração ofereceu uma melhoria na usabilidade da aplicação, proporcionando uma experiência de utilização melhorada aos seus utilizadores. A aplicação foi desenhada de forma a atender às necessidades e expectativas dos seus utilizadores, garantindo a melhor experiência de utilização e uma recolha de dados eficiente. No Apêndice B.1 são apresentadas as imagens apresentadas ao longo desta secção numa maior resolução.

A aplicação AQUA conta com uma *splash screen* inicial que proporciona uma experiência visual agradável enquanto os dados armazenados *online* são carregados (ver *screenshot* 1 da figura 9). Durante esse processo, é apresentada uma mensagem dinâmica ao utilizador, informando sobre o tipo de informação que está sendo carregada naquele momento. Ao finalizar o carregamento da informação, é apresentado um ecrã onde o utilizador pode criar uma conta para ter acesso às funcionalidades e informações disponíveis na aplicação. Caso o utilizador já tenha uma conta, pode optar por fazer *login*, onde é solicitada a inserção do seu *email* e *password* (ver *screenshot* 2 da figura 9).

O processo de registo é dividido em três etapas. Na primeira (ver *screenshot* 3 da figura 9), os utilizadores preenchem as suas credenciais (*email*, *password* e confirmação de *password*). Em seguida, na segunda etapa (ver *screenshot* 4 da figura 9), são solicitados o nome do utilizador e a sua localização. Por fim, na terceira etapa, têm a opção de adicionar uma fotografia e uma descrição ao seu perfil (ver *screenshot* 5 da figura 9). No entanto, a aplicação oferece a flexibilidade de saltar esta última etapa, permitindo que criem a conta sem fornecer essas informações adicionais.

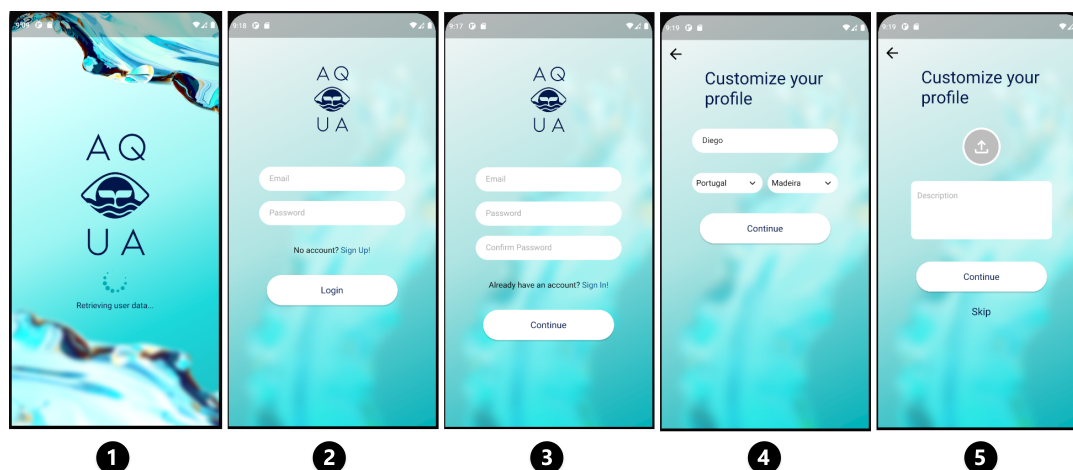


Figura 9. Screenshots da aplicação AQUA: (1) Splash screen; (2) Ecrã de login; (3) Ecrã de criação de conta; (4-5) Adição de informação ao perfil de utilizador.

Após a criação da conta, o utilizador é autenticado automaticamente e direcionado para o ecrã inicial da aplicação. No entanto, antes disso, é apresentada uma introdução que tem como objetivo explicar ao utilizador o funcionamento da aplicação e os benefícios de contribuir frequentemente (ver figura 10). Durante essa introdução, é destacado que ao reportar avistamentos de espécies marinhas, o utilizador contribui para a ciência. Além disso, é explicado que à medida que contribui, pode desbloquear recompensas, como distintivos e cartas de espécies, e tem a oportunidade de competir contra outros utilizadores, para motivar os utilizadores, fornecendo uma visão clara das recompensas que podem desbloquear ao contribuir com a recolha e partilha de informação científica.

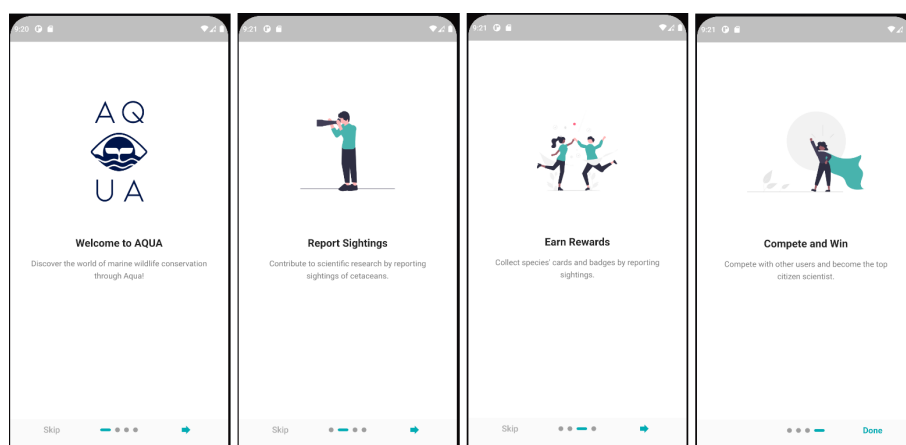


Figura 10. Screenshots da aplicação AQUA: Introdução ao funcionamento da aplicação.

O ecrã inicial da aplicação é composto por várias secções que fornecem uma visão geral das funcionalidades e informações importantes. Na primeira secção, caso o utilizador não tenha reportado nenhum avistamento, é apresentada uma mensagem incentivando-o a fazê-lo, juntamente com um botão que permite iniciar o processo de reporte de um avistamento rapidamente (ver *screenshot* 1 da figura 11).

Em seguida, há um conjunto de contadores que mostram ao utilizador quantos avistamentos já reportou, quantas rotas de viagem já registou e quantos reportes estão armazenados localmente no dispositivo e ainda não foram submetidos. Estes contadores ajudam o utilizador a ter uma noção clara das suas contribuições na aplicação. Este ecrã inclui também três botões que permitem ao utilizador aceder rapidamente a ecrãs referentes aos elementos de gamificação disponibilizados pela aplicação, nomeadamente, a coleção de cartas das espécies, a coleção de distintivos, e o *ranking*.

Outra secção relevante do ecrã inicial é a apresentação das cinco espécies mais avistadas pelos utilizadores da aplicação. Cada espécie é representada por um círculo contendo uma fotografia da espécie e um contador do número de vezes que a mesma foi avistada. O tamanho do círculo varia de acordo com o valor do contador, transmitindo uma visão da quantidade de ocorrências daquela espécie. Isto significa que a espécie mais avistada é apresentada num círculo maior, enquanto a menos avistada é apresentada num círculo menor. Ao pressionar em cada círculo, o utilizador pode aceder rapidamente à informação sobre a espécie em questão.

Por fim, é disponibilizado um pequeno mapa interativo onde são apresentados os marcadores dos avistamentos reportados pelos utilizadores (ver *screenshot 2* da figura 11). Caso o utilizador queira visualizar mais detalhes, a aplicação fornece uma opção de expandir o mapa para uma visualização mais detalhada em ecrã completo.

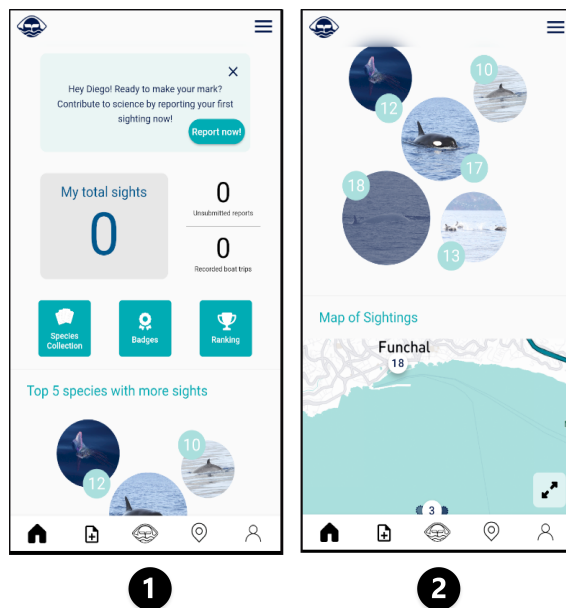


Figura 11. Screenshots da Aplicação AQUA: (1) Ecrã inicial incluindo contadores, botões e cinco espécies mais avistadas; (2) Ecrã inicial incluindo cinco espécies mais avistadas e mapa de avistamentos.

3.4.1 Reporte de Avistamentos e Incidentes

Ao pressionar o segundo botão da barra de navegação, o utilizador é redirecionado para o ecrã de reportes onde são apresentados dois botões distintos que permitem aos utilizadores reportar avistamentos ou reportar incidentes, respetivamente. Se um utilizador tiver reportes guardados localmente, ou seja, reportes que foram registados mas que ainda não foram submetidos, a aplicação apresenta uma mensagem indicando o número de reportes não submetidos. Além disso, é

apresentado um botão que permite ao utilizador aceder rapidamente ao seu histórico de reportes, onde podem rever e submetê-los.

Tal como referido em secções anteriores, o processo de reportar avistamentos é iniciado com a captura ou carregamento de fotografias e vídeos das observações a partir da galeria do dispositivo. Esta etapa é fundamental, pois permite apoiar os cientistas na validação da informação recolhida, principalmente na identificação das espécies. À medida que as imagens das observações são adicionadas, são apresentadas numa lista no ecrã da aplicação. Além disso, é disponibilizada a opção de eliminar permanentemente uma fotografia ou vídeo que tenha sido adicionado.

Ao clicar numa fotografia ou vídeo da lista, os utilizadores têm a possibilidade de visualizá-los em ecrã completo, permitindo uma visualização e análise mais detalhada das observações registadas, o que facilita a indicação das espécies avistadas. Após a captura das observações, o processo de reportar um avistamento é dividido em cinco etapas sequenciais para fornecer informações detalhadas sobre as espécies avistadas.

Na primeira etapa, o utilizador é solicitado a indicar todas as espécies que foram avistadas. Para facilitar este processo, são apresentados dois botões, um para procurar espécies pelo nome e outro para adicionar uma espécie desconhecida (ver *screenshot 1* da figura 12).

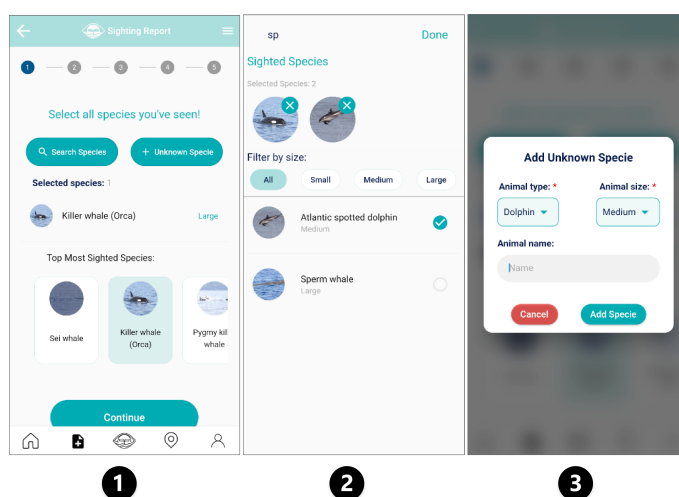


Figura 12. Screenshots da aplicação AQUA: (1) Ecrã de seleção de espécies; (2) Ecrã de pesquisa de espécies; (3) Formulário para adição de espécies desconhecida.

Ao pressionar o botão de pesquisa de espécies, o utilizador é direcionado para um ecrã onde são listadas todas as espécies existentes no sistema (ver *screenshot 2* da figura 12). Cada espécie inclui uma fotografia, nome e tamanho (*small*, *medium*, ou *large*). O utilizador pode pesquisar uma espécie específica digitando o seu nome num campo de texto no topo do ecrã. Conforme o utilizador digita, a lista de espécies é atualizada para apresentar apenas aquelas cujos nomes contenham a sequência de caracteres inserida. Para facilitar ainda mais a pesquisa, a aplicação disponibiliza filtros de tamanho, permitindo ao utilizador selecionar um determinado tamanho de espécie (por exemplo, “Large”) para apresentar apenas as espécies correspondentes. À medida que o utilizador seleciona as espécies avistadas, elas são adicionadas a uma lista no início do ecrã. Para

cada espécie adicionada, é apresentada a sua fotografia juntamente com um botão que permite remover a espécie selecionada, caso seja desejado.

É importante mencionar que a lista de espécies disponíveis na aplicação AQUA é limitada às espécies previamente avistadas na ilha da Madeira, com base nas informações fornecidas por um biólogo marinho. No entanto, devido à possibilidade de migração de espécies, novas espécies podem ser avistadas a qualquer momento. Desta forma, a aplicação permite aos utilizadores adicionar uma espécie desconhecida, solicitando informações como o tipo de animal avistado e o seu tamanho (ver *screenshot 3* da figura 12). Se o utilizador for experiente e souber o nome da espécie, pode especificá-lo preenchendo um campo de texto adicional.

Na segunda etapa, para cada espécie selecionada, são apresentados seis botões para especificar o número de indivíduos avistados da respetiva espécie. O utilizador pode apenas selecionar uma opção para cada espécie (ver *screenshot 1* da figura 13). De igual forma, na terceira etapa, para cada espécie selecionada, são apresentados quatro botões que permitem selecionar os comportamentos observados nos indivíduos da respetiva espécie. O utilizador pode selecionar várias opções de comportamento para cada espécie (ver *screenshot 2* da figura 13).

Caso o utilizador tenha selecionado espécies desconhecidas, a quarta etapa adicional é apresentada. Para cada espécie desconhecida, é solicitado ao utilizador a indicação do número de indivíduos avistados e dos comportamentos observados (ver *screenshot 3* da figura 13). Na última etapa, o utilizador tem a opção de preencher um campo de texto com comentários adicionais em relação ao avistamento, podendo o utilizador optar por saltar esta etapa, caso preferir (ver *screenshot 4* da figura 13).

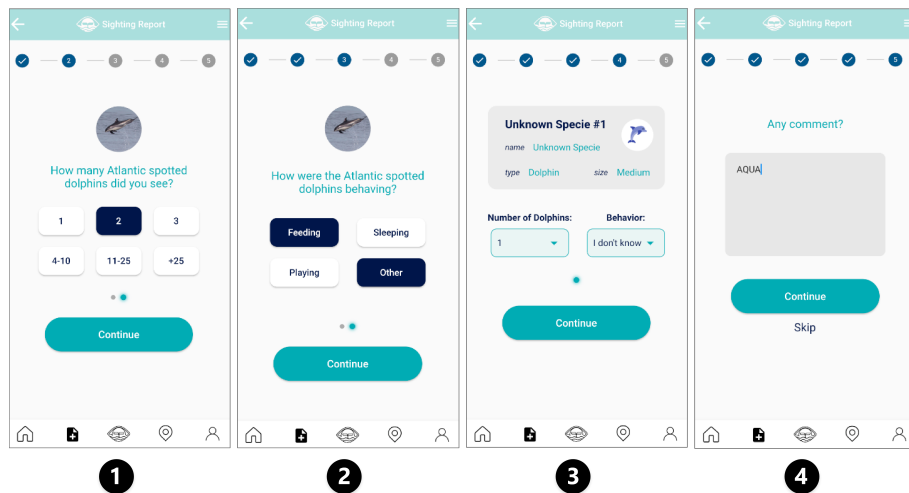


Figura 13. Screenshots da aplicação AQUA: (1) Ecrã de indicação do número de indivíduos das espécies; (2) Ecrã de indicação dos comportamentos das espécies; (3) Ecrã de preenchimento de informação sobre espécie desconhecida; (4) Ecrã para adição de comentário.

Para evitar o esquecimento do preenchimento de alguma informação sobre uma espécie, a aplicação AQUA não permite ao utilizador continuar para a etapa seguinte do processo de reportar o avistamento, caso a informação da etapa atual não estiver completamente preenchida. Contudo,

a mesma permite ao utilizador regressar para uma etapa anterior a qualquer instante, caso seja necessário realizar alguma modificação ou complementar a informação sobre o avistamento.

Após concluir o preenchimento de todas as informações sobre o avistamento, o utilizador é re-direcionado para um ecrã onde é apresentado um sumário completo do avistamento. Nesse sumário é apresentado um mapa que mostra a localização do avistamento, a data e hora em que o avistamento ocorreu, ou seja, em que o registo foi iniciado, assim como a lista das espécies avistadas indicadas pelo utilizador (ver *screenshot 1* da figura 14). Cada espécie pode ser pressionada para visualizar as informações preenchidas referentes à mesma. Caso o utilizador tenha adicionado um comentário sobre o avistamento, o mesmo é apresentado no sumário. Além disso, as fotografias e vídeos das observações são listadas, podendo o utilizador tocar numa fotografia ou vídeo para visualizá-lo em ecrã completo (ver *screenshot 2* da figura 14).

Após rever todas as informações no sumário do avistamento, o utilizador pode pressionar um botão para submeter o reporte do avistamento. Ao pressionar o botão, no caso do dispositivo não ter conexão à Internet no momento da submissão, o sistema informa que o reporte foi registado localmente para submissão posterior. Isto permite que o utilizador tenha a opção de submeter o avistamento assim que a conexão à Internet for retomada.

O processo de reportar um incidente é muito semelhante ao de reportar um avistamento, mas com algumas diferenças específicas, dependendo do tipo de incidente a ser reportado. Ao iniciar o reporte, é solicitado ao utilizador a especificação do tipo de incidente, podendo escolher entre três opções: animal morto, animal ferido, ou outro tipo de incidente (ver *screenshot 3* da figura 14).

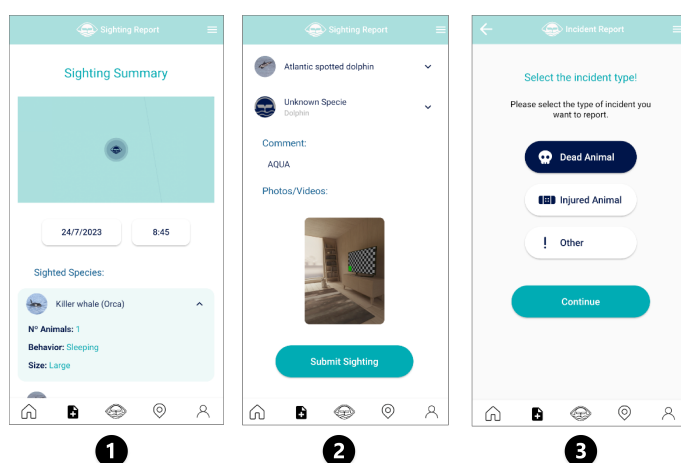


Figura 14. Screenshots da aplicação AQUA: (1-2) Ecrãs do sumário dos dados do avistamento; (3) Ecrã de seleção do tipo de incidente.

No caso do utilizador selecionar a opção de reportar um incidente que envolve animais, ou seja, um animal morto ou ferido, o processo de reportar o incidente é idêntico ao de reportar o avistamento, iniciando com a captura das observações e terminando com a adição de um comentário opcional. No caso do tipo de incidente ser diferente, apenas é solicitado ao utilizador a captura ou carregamento das fotografias e vídeos das suas observações e a adição de um comentário onde o utilizador deve descrever as suas observações, sendo este obrigatório. Esta diferenciação no processo

de reportar incidentes de categorias distintas permite recolher a informação relevante em cada situação específica, contribuindo com dados úteis para proteção do ecossistema marinho.

Ao submeter um avistamento, se o utilizador tiver reportado algumas espécies pela primeira vez, é apresentado um ecrã onde são atribuídas as cartas das espécies desbloqueadas. Neste ecrã o utilizador é felicitado pela conquista e as cartas são listadas. Além disso, o número de cartas espécies desbloqueadas é apresentado por raridade (por exemplo, “2 espécies comuns e 1 espécie rara”). Cada número é representado por uma cor específica: azul para comum, roxo para semirara, e dourado para rara (ver *screenshot* 2 da figura 15). Desta forma, é possível apoiar novos utilizador na compreensão e identificação das diferentes raridades das espécies.

De igual forma, caso o utilizador tenha cumprido os requisitos para desbloquear algum distintivo, após a submissão do avistamento, é apresentado um ecrã onde são atribuídos os respetivos distintivos. Nesse ecrã, além da imagem de cada distintivo, também é apresentado o título adquirido pelo utilizador e o requisito específico que foi cumprido para o seu desbloqueio (ver *screenshot* 1 da figura 15).

Após o utilizador desbloquear novas cartas de espécies ou distintivos, os botões que permitem aceder à coleção de cartas e à coleção de distintivos são destacados com um brilho dourado, atraindo a atenção do utilizador para as novas recompensas adquiridas. Este destaque tem como objetivo incentivar o utilizador a explorar e visualizar as suas conquistas com maior detalhe.

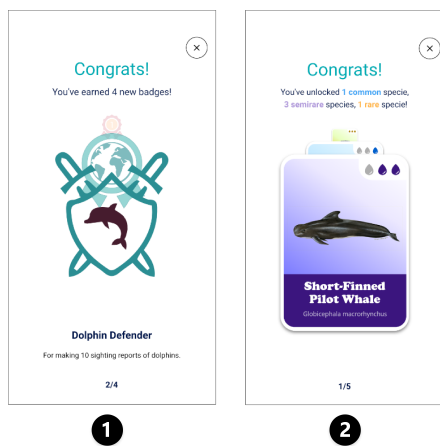


Figura 15. Screenshots da aplicação AQUA: (1) Ecrã de desbloqueio de novos distintivos; (2) Ecrã de desbloqueio de novas cartas de espécies.

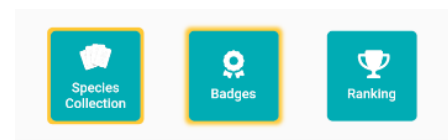


Figura 16. Screenshot da aplicação AQUA: Botões para aceder à coleção de cartas e à coleção de distintivos destacados com brilho.

3.4.2 Gamificação

Com base na revisão de literatura, seguiram-se princípios de gamificação para aumentar o envolvimento e melhorar a experiência dos utilizadores. Estes princípios têm como objetivo aumentar a motivação intrínseca e extrínseca dos utilizadores, proporcionando uma experiência mais envolvente e satisfatória.

Em relação aos princípios relacionados ao comportamento individual, tal como já referido anteriormente, introduziu-se na aplicação AQUA a coleção de espécies e a coleção de distintivos. Os utilizadores podem acompanhar e visualizar as suas contribuições e progressos através de diversos

contadores, incentivando-os a estabelecer objetivos pessoais, acompanhar as suas contribuições e sentirem-se satisfeitos ao adquirir novas recompensas.

No que diz respeito aos princípios de comportamento social, incluiu-se um *ranking* de utilizadores, permitindo que possam comparar o seu desempenho com os restantes. Desta forma, é criada uma competência social, motivando os utilizadores a participar ativamente no reporte das suas observações, na tentativa de destacar-se no *ranking*. Além disso, a possibilidade de desbloquear títulos para apresentá-los no perfil de utilizador tem como objetivo conquistar reconhecimento social em relação aos restantes utilizadores.

Por fim, em relação aos princípios orientados pelo sistema, introduziu-se o conceito de raridade nas cartas de espécies e tornou-se alguns distintivos mais desafiadores de obter, o que permite incentivar os utilizador a se esforçarem mais para desbloquear essas recompensas. Além disso, permite distinguir os utilizadores que contribuem mais para a ciência e destacá-los perante os outros utilizadores.

3.4.2.1 Coleção de Cartas de Espécies

Como já referido anteriormente, a raridade da espécie pode ser identificada pela cor da carta e pelo número de gotas de água no canto superior direito. As espécies consideradas comuns são representadas por uma cor azul e uma única gota de água (ver *screenshot 1* da figura 17). As espécies semiraras são identificadas por uma cor roxa e duas gotas de água (ver *screenshot 2* da figura 17). E, por fim, as espécies raras são representadas por uma cor dourada e três gotas de água (ver *screenshot 3* da figura 17). A seleção das cores das cartas foi feita com base numa escala de cores utilizada em jogos, que geralmente procura manter uma consistência visual [58, 59].



Figura 17. Cartas das espécies: (1) Carta de espécie comum; (2) Carta de espécie semirara; (3) Carta de espécie rara.

Quando uma carta desbloqueada ainda não foi visualizada com detalhe pelo utilizador, é exibida a indicação “NEW” sobre a carta, incentivando-o a pressionar nela. Ao pressionar na carta, esta é expandida para visualização com maior detalhe (ver *screenshot 3* da figura 18). O utilizador pode deslizar horizontalmente para virar a carta, sendo na parte posterior apresentada a data em que a espécie foi avistada e a carta desbloqueada (ver *screenshot 4* da figura 18). Além disso, é disponibilizado um botão que permite ao utilizador obter mais informações sobre aquela espécie em particular.

As cartas das espécies bloqueadas são apresentadas em tons de cinza e com um ícone de cadeado, transmitindo a ideia de que eles ainda não foram adquiridos pelo utilizador. Se o utilizador tentar clicar numa carta que ainda não desbloqueou, é apresentada uma mensagem indicando que a

espécies ainda não foi desbloqueada. Desta forma, a carta não é expandida, impedindo o utilizador de visualizá-la com detalhe.

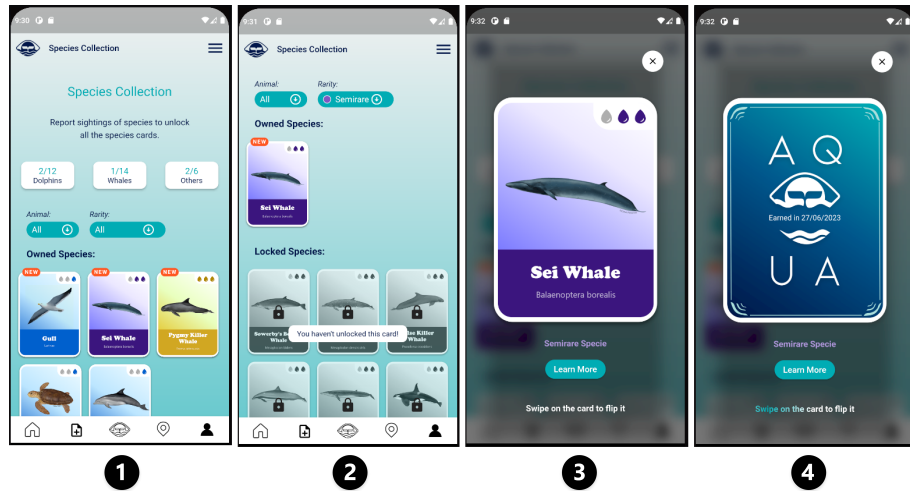


Figura 18. Screenshots da aplicação AQUA: (1) Ecrã da coleção de espécies sem filtros aplicados; (2) Ecrã da coleção de espécies com filtro aplicado; (3-4) Visualização de uma carta desbloqueada.

Esta funcionalidade permite que os utilizadores acompanhem as espécies que já avistaram, incentivando-os a desbloquear mais cartas e expandir a sua coleção. A apresentação das datas de desbloqueio permitem que o utilizador possa recordar a sua experiência, enquanto o acesso à informação sobre a espécie permite que o mesmo possa aprender mais sobre aquela espécie.

3.4.2.2 Coleção de Distintivos

Os distintivos, por sua vez, também desempenham um papel importante na aplicação AQUA, tendo como objetivo motivar os utilizadores a reportar avistamentos e incidentes, ao cumprir os requisitos necessários para adquirir cada distintivo.

Tal como nas cartas das espécies, os distintivos bloqueados são apresentados em tons de cinza e com um ícone de cadeado. Ao pressionar num distintivo bloqueado, é apresentada uma mensagem contendo o requisito necessário para desbloqueá-lo. Esta abordagem tem o intuito de criar uma sensação de mistério e despertar curiosidade dos utilizadores para desbloquear esses distintivos.

Inicialmente, os utilizadores podem desbloquear distintivos rapidamente, como ao reportar o primeiro avistamento e o primeiro incidente. No entanto, ao longo do tempo, a dificuldade dos distintivos aumenta para se ajustar à participação contínua do utilizador na aplicação. Alguns distintivos exigem, assim, um maior envolvimento e esforço por parte do utilizador, tornando-os mais difíceis de obter. Ao conquistar distintivos mais desafiantes, os utilizadores podem sentir orgulho, satisfação e obter reconhecimento dentro da comunidade da aplicação AQUA.

A seguir, são listados os distintivos existentes na aplicação, juntamente com os seus títulos e os requisitos necessários para desbloqueá-los:

- **Distintivo “Novice Spotter”:** Desbloqueado por reportar um primeiro avistamento.
- **Distintivo “Sighting Pro”:** Desbloqueado por reportar 10 avistamentos.

- **Distintivo “Citizen Scientist”**: Desbloqueado por reportar mais do que 25 avistamentos.

Estes distintivos estão relacionados com a quantidade de avistamentos reportados pelo utilizador, tendo como objetivo reconhecer o progresso do utilizador na sua participação contínua no reporte de avistamentos. Além disso, com o progresso gradual pretende-se motivar os utilizador a continuarem a contribuir com a ciência. Na figura 19 é possível observar os ícones dos respetivos distintivos.

- **Distintivo “Ocean Guardian”**: Desbloqueado por reportar um incidente.

Este distintivo (ver figura 19) está relacionado com o reporte de um incidente, com o objetivo de reconhecer a contribuição do utilizador na proteção do ecossistema marinho.



Figura 19. Ícones dos distintivos “Novice Spotter”, “Sighting Pro”, “Citizen Scientist” e “Ocean Guardian”

- **Distintivo “Collector”**: Desbloqueado por desbloquear pelo menos 10 cartas de espécies distintas.
- **Distintivo “Master Enthusiast”**: Desbloqueado por desbloquear uma espécie rara.
- **Distintivo “Marine Enthusiast”**: Desbloqueado por desbloquear pelo menos 3 cartas de espécies semiraras.
- **Distintivo “Ocean Explorer”**: Desbloqueado por desbloquear pelo menos 5 cartas de espécies comuns.

Estes distintivos, relacionados com a coleção de cartas de espécies, têm como objetivo recompensar os utilizadores que desbloqueiam um determinado número de cartas de espécies, incentivando-os a completar a sua coleção, além de gerar curiosidade em descobrir novas espécies. Na figura 20 é possível observar os ícones dos respetivos distintivos.

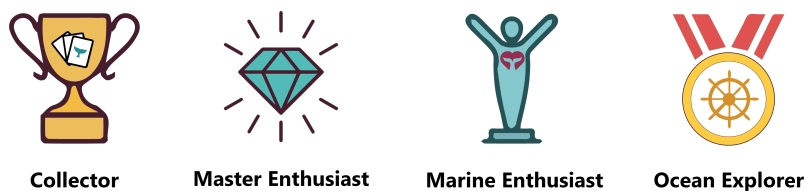


Figura 20. Ícones dos distintivos “Collector”, “Master Enthusiast”, “Marine Enthusiast” e “Ocean Explorer”

- **Distintivo “Cetacean Expert”**: Desbloqueado por identificar 10 espécies de cetáceos distintas.
- **Distintivo “Dolphin Defender”**: Desbloqueado por reportar 10 avistamentos de golfinhos.
- **Distintivo “Whale Watcher”**: Desbloqueado por reportar 5 avistamentos de baleias.

Estes distintivos estão relacionados com a identificação de espécies distintas. Eles têm como objetivo incentivar os utilizadores a contribuir e a explorar novas espécies, aprendendo mais sobre elas. Na figura 21 são apresentados os ícones dos respetivos distintivos.

- **Distintivo “Geographer”**: Desbloqueado por reportar avistamentos em diferentes regiões.

Por fim, este distintivo (ver figura 21) tem como objetivo reconhecer utilizadores que reportam avistamentos em diferentes regiões, incentivando a explorar e a partir na recolha e partilha das suas observações em várias zonas geográficas.

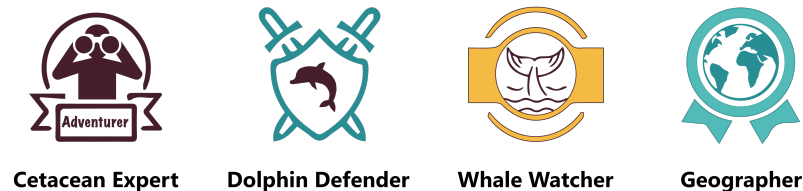


Figura 21. Ícones dos distintivos “Cetacean Expert”, “Dolphin Defender”, “Whale Watcher” e “Geographer”

3.4.2.3 Ranking de Utilizadores

Com o *ranking* de utilizadores, a aplicação oferece ao utilizador a oportunidade de visualizar e comparar as suas conquistas e participação com as dos restantes utilizadores (ver *screenshot* 1 da figura 22). Existem dois tipos de *rankings* disponíveis: o *ranking* global e o *ranking* local.

No ecrã do *ranking*, o utilizador tem a opção de selecionar o *ranking* que pretende visualizar. O *ranking* global inclui todos os utilizadores da aplicação AQUA, enquanto o *ranking* local inclui apenas os utilizadores com a mesma localização geográfica que o utilizador.

O *ranking* global tem como objetivo permitir uma visão mais ampla do desempenho de cada utilizador em relação a todos os outros, independentemente da sua localização. Por outro lado, o *ranking* local permite que o utilizador possa comparar a sua participação e desempenho com uma comunidade mais reduzida e próxima, com utilizadores da mesma região. Além da seleção do tipo de *ranking*, o mesmo pode ser filtrado por intervalo de tempo e por tipo de dado (número de avistamentos reportados, número de espécies desbloqueadas, ou número de distintivos desbloqueados) (ver *screenshot* 4 da figura 22).

A filtragem por intervalo de tempo tem como objetivo permitir ao utilizador visualizar o desempenho dos utilizadores num período de tempo específico. Por outro lado, a comparação por tipo de dado, como o número de avistamentos reportados ou o número de distintivos desbloqueados, permite ao utilizador ver como cada utilizador se destaca em diferentes partes da aplicação.

Por exemplo, ao comparar o número de avistamentos, o utilizador pode identificar quem está contribuindo mais para a recolha e partilha de informação para a ciência. Também, ao comparar o número de espécies e distintivos desbloqueados, é possível visualizar que utilizadores estão mais focados na exploração e na aquisição de novas recompensas.

Antes de apresentar a lista de utilizadores no *ranking*, é apresentado ao utilizador a sua posição atual com base nos filtros selecionados. Esta informação permite ao utilizador saber imediatamente onde ele se encontra em comparação aos outros utilizadores. Para facilitar a identificação da sua posição no *ranking*, especialmente quando há um grande número de utilizadores listados, a sua posição é destacada com uma cor diferentes em relação aos restantes utilizadores. No *ranking*, os utilizadores são listados por ordem decrescente, com base no filtro selecionado. Cada posição no *ranking* inclui a fotografia do utilizador, o nome, o título (se houver) e a quantidade correspondente ao filtro escolhido.

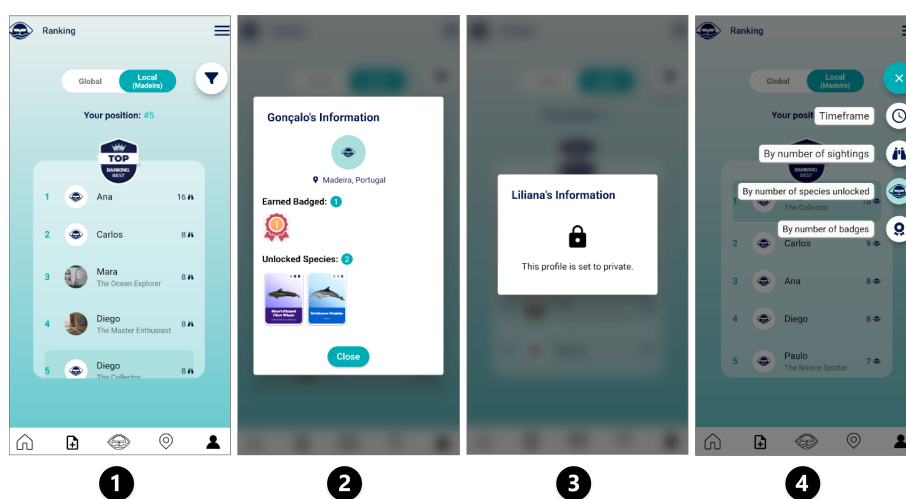


Figura 22. Screenshots da aplicação AQUA: (1) Ecrã do ranking de utilizadores; (2) Visualização de perfil de um utilizador do ranking; (3) Mensagem ao tentar visualizar um perfil privado; (4) Filtros disponíveis para o ranking.

Ao pressionar a posição de um utilizador, a aplicação apresenta informações detalhadas sobre o respetivo utilizador. Essas informações incluem o nome do utilizador, o título que ele selecionou (se houver), a localização, a descrição pessoal (se fornecida), e a lista de distintivos e cartas de espécies desbloqueadas pelo utilizador (ver *screenshot* 2 da figura 22). A visualização do perfil de um utilizador permite aos utilizadores podem conhecer melhor a pessoa, além de que, ao comparar recompensas, pode motivá-los a se esforçarem mais para obter recompensas semelhantes ou até mesmo melhores.

Para garantir a privacidade dos utilizadores, a aplicação AQUA respeita a configuração da privacidade de cada perfil. Se um utilizador tiver definido o seu perfil como privado, a aplicação não apresenta as informações desse utilizador no *ranking*. Quando um utilizador tenta aceder ao perfil de um utilizador cuja privacidade seja privada, a aplicação informa que a informação do perfil é privada, estando esta indisponível (ver *screenshot* 3 da figura 22).

Cada utilizador tem controlo da privacidade do seu perfil, podendo alterá-la em qualquer momento no ecrã do seu perfil (ver *screenshot 3* da figura 23). Isto permite que cada utilizador se sinta confortável e confiante ao utilizar a aplicação AQUA.

3.4.3 Perfil do Utilizador

No ecrã do perfil do utilizador, o utilizador pode visualizar informações associadas à sua conta e personalizar o seu título (ver *screenshot 1* da figura 23). Neste ecrã, são apresentadas as informações preenchidas durante a criação da conta, como a fotografia de perfil, o nome do utilizador, a descrição e a localização. Além disso, existem três contadores que fornecem um resumo da participação do utilizador na aplicação. Estes contadores mostram o número total de espécies desbloqueadas, o número de reportes (avistamentos e incidentes) submetidos, e o número de rotas de viagem registadas. Isto permite que o utilizador consiga ter uma visão geral da sua participação na aplicação.

No mesmo ecrã, há um botão específico que, ao ser pressionado, apresenta uma lista com todos os títulos disponíveis, ou seja, aqueles que foram adquiridos pelo utilizador (ver *screenshot 2* da figura 23). O utilizador pode escolher um único título para apresentá-lo no seu perfil, tendo também a opção de não seleccionar nenhum título. Ao seleccionar um, ele é apresentado no perfil do utilizador, permitindo que os outros utilizadores visualizem a sua conquista no *ranking* de utilizadores.

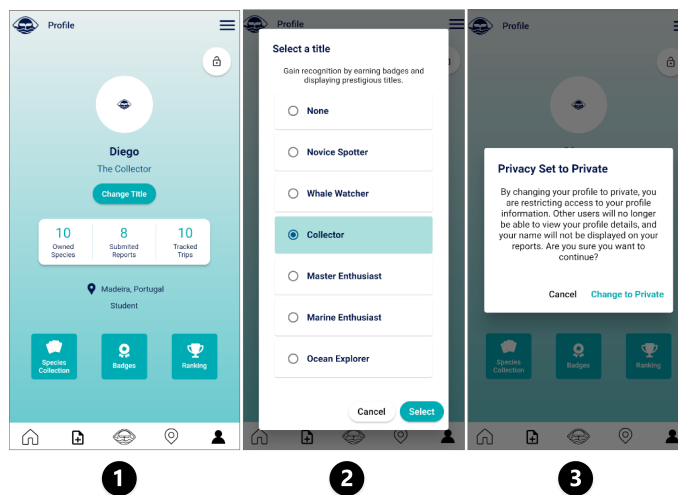


Figura 23. Screenshots da aplicação AQUA: (1) Ecrã de perfil do utilizador; (2) Lista de títulos desbloqueados pelo utilizador; (3) Alteração de privacidade do perfil (de público para privado).

3.4.4 Mapa de Avistamentos

Ao expandir o mapa de avistamentos do ecrã inicial, são apresentados os marcadores dos avistamentos reportados num mapa interativo (ver *screenshot 1* da figura 24). Tendo em conta que as espécies estão em constante movimento e que a localização obtida a partir do GPS se refere à localização do dispositivo e não da espécie avistada, é apresentado à volta de cada avistamento um círculo que representa uma área de incerteza da localização da espécie.

Cada área de incerteza de um avistamento é representada com uma opacidade baixa. Fez-se esta escolha para permitir a sobreposição de áreas, sendo que as áreas mais propensas a avistamentos são visualizadas com maior intensidade de cor. Quanto mais intensa for a cor de uma área, maior

é o número de avistamentos reportados naquela área específica. Isto permite que os utilizadores possam facilmente identificar as áreas mais relevantes e a compreender visualmente a distribuição dos avistamentos de espécies ao longo do mapa.

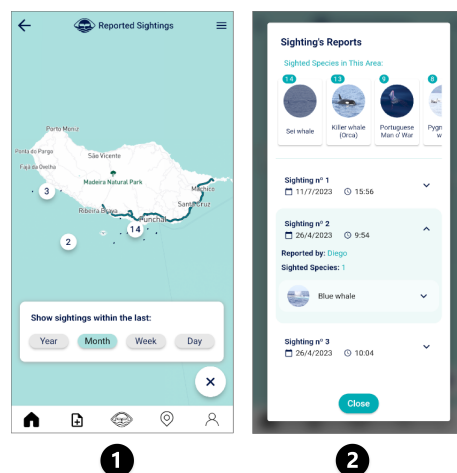


Figura 24. Screenshots da aplicação AQUA: (1) Ecrã de mapa com avistamentos do último mês; (2) Detalhes do cluster de avistamentos.

No *screenshot* 1 da figura 25 é apresentada uma sobreposição das áreas de incerteza de dois avistamentos reportados através da aplicação AQUA. É possível observar que a área de interseção entre essas duas áreas possui uma maior intensidade de cor, indicando que essa área é mais propícia para a ocorrência de espécies.

Ao longo do tempo, o número de avistamentos pode ser muito elevado, saturando o mapa com marcadores de avistamentos. Para facilitar a visualização do mapa, tornando-o mais limpo, os marcadores dos avistamentos são agrupados em *clusters*, caso estejam próximos. No *screenshot* 2 da figura 25 é possível observar dois *clusters*, cada um agrupando dois avistamentos, no mapa de avistamentos da aplicação AQUA. Ao fazer *zoom out* no mapa, os dois *clusters* são agrupados, criando um único *cluster* que agrupa os quatro avistamentos, tal como apresentado na *screenshot* 3 da figura 25.

A aplicação AQUA fornece aos utilizadores opções de filtragem dos avistamentos apresentados no mapa, podendo ser apresentados os avistamentos do último ano, mês, semana ou dia. Desta forma, apenas são apresentados no mapa os avistamentos registados durante esses períodos específicos. Esta funcionalidade de filtragem permite que possam obter informações específicas sobre os avistamentos mais recentes ou dentro de um período de tempo específico. Além disso, permite compreender melhor a ocorrência de algumas espécies ao longo do tempo.

Quando um utilizador clica num *cluster* no mapa de avistamentos, é apresentada uma lista com todas as espécies avistadas nos avistamentos agrupados naquela área (ver *screenshot* 2 da figura 24). As espécies são apresentadas por ordem decrescente, com base no número de vezes que foram avistadas, sendo esse número apresentado acima de cada espécie na lista. Isto permite que os utilizadores identifiquem facilmente que espécies foram avistadas com maior e menor frequência naquela região específica. Além disso, é apresentada uma lista com todos os avistamentos reportados naquela área, pertencentes ao *cluster*. Ao pressionar num avistamento da lista, é expandido e são

apresentados detalhes sobre o mesmo, incluindo a data e hora do avistamento, informações sobre as espécies avistadas e o nome do utilizador que reportou o avistamento, caso o mesmo tenha o seu perfil público.

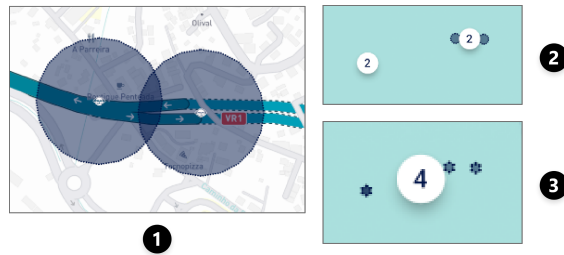


Figura 25. (1) Sobreposição de áreas de incerteza de dois avistamentos no mapa; (2) Dois clusters de avistamentos, no mapa de avistamentos; (3) Agrupação de *clusters* próximos

3.4.5 Visualização de Informação sobre Espécies

Tal como já foi referido anteriormente, a aplicação AQUA disponibiliza aos utilizadores a oportunidade de explorar informações sobre diferentes espécies marinhas. Através do terceiro botão da barra de navegação, o utilizador pode aceder a um ecrã onde é solicitado para selecionar entre três opções: golfinhos, baleias ou outros (ver *screenshot* 1 da figura 26). Ao escolher uma das opções, todas as espécies do tipo de animal selecionado são listados (ver *screenshot* 2 da figura 26). Além disso, a aplicação permite aos utilizadores filtrar a lista por tamanho (*small*, *medium*, ou *large*) e realizar uma pesquisa pelo nome (ver *screenshot* 3 da figura 26).

Ao selecionar uma espécie específica, os utilizadores são redirecionados para um ecrã que apresenta informações detalhadas sobre a espécie selecionada. Inicialmente, uma fotografia da espécie é apresentada, seguida de informações como o nome comum, o nome científico, uma descrição geral da espécie, características que permitem a sua identificação, e algumas curiosidades relevantes. Estas informações têm como objetivo proporcionar uma experiência de aprendizagem aos utilizadores, adquirindo conhecimentos sobre as espécies que podem ser avistadas, além de ajudá-los a realizar uma identificação mais precisa das espécies avistadas durante as suas viagens marítimas.

Com base no estudo realizado com o AQUA 1.0, foi possível identificar que a leitura das informações durante as viagens marítimas pode ser desafiadora e causar desconforto nalgumas pessoas. Assim, a aplicação AQUA permite que os utilizadores ouçam as informações sobre as espécies em vez de lê-las.

No ecrã das informações de uma espécie específica, são apresentados botões ao lado de cada informação relevante (ver *screenshots* 4 e 5 da figura 26). Ao pressionar um desses botões, a respetiva informação é narrada em voz alta, permitindo que os utilizadores estejam atentos ao ambiente ao seu redor enquanto conseguem obter informações importantes sobre as espécies marinhas.

Ao ouvir as informações em vez de lê-las, podem desfrutar da experiência da viagem, ao mesmo tempo que enriquecem o seu conhecimento sobre a biodiversidade de espécies marinhas. Além disso, ajuda a evitar desconfortos, como enjoos, que podem ser causados pela leitura em movimento, melhorando a experiência de utilização.

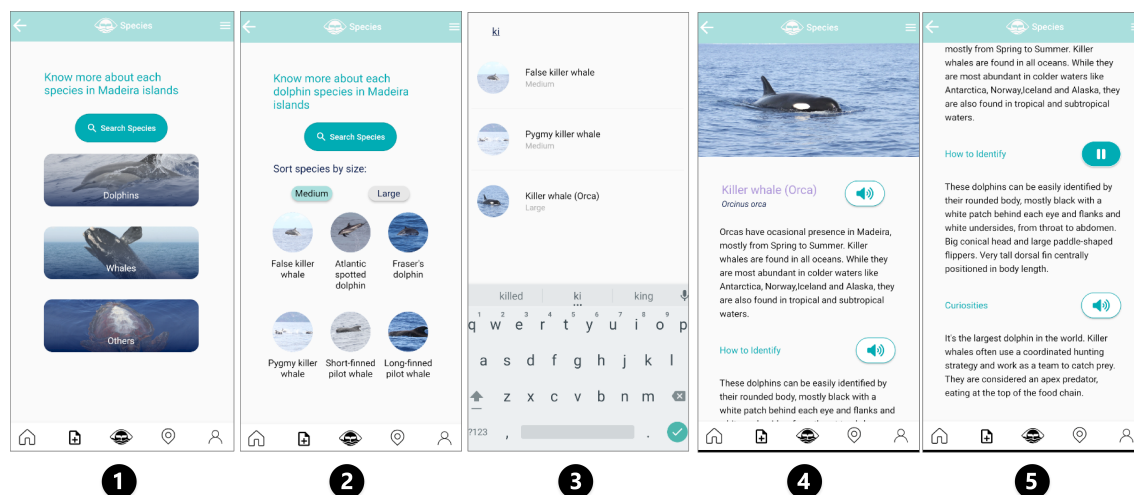


Figura 26. Screenshots da aplicação AQUA: (1) Ecrã de seleção do tipo de animal; (2) Ecrã de seleção da espécie; (3) Pesquisa de uma espécie pelo nome; (4-5) Ecrã com informação sobre a espécie.

3.4.6 Tracking das Rotas de Viagem

Ao pressionar o quarto botão da barra de navegação, o utilizador é redirecionado para um ecrã que apresenta um mapa interativo, e onde pode iniciar o *tracking* da rota de uma viagem. Ao iniciar uma rota de viagem, um temporizador é iniciado, contando e apresentando ao utilizador a duração da viagem a cada segundo. Além disso, um marcador verde é apresentado no mapa, representando o início e a localização atual do utilizador. A cada 5 minutos, um novo marcador azul é adicionado no mapa, conectando-se ao marcador anterior. Esta sequência de marcadores permite criar uma representação visual da rota percorrida durante a viagem (ver *screenshot 1* da figura 27).

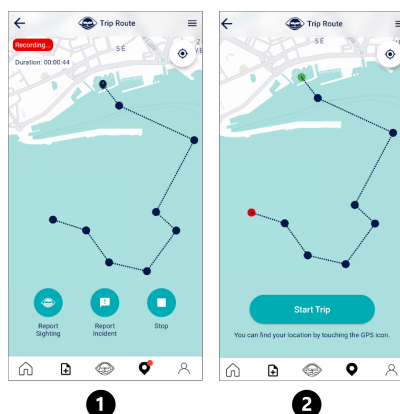


Figura 27. Screenshots da aplicação AQUA: (1) Ecrã durante o tracking de uma rota de viagem; (2) Ecrã após terminar o tracking de uma rota de viagem.

O utilizador pode terminar o *tracking* a qualquer momento, ao pressionar um botão que permite submeter a rota registada. Caso não haja conexão à Internet, a rota é armazenada localmente para submissão posterior. Ao finalizar a rota, é adicionado um último marcador vermelho no mapa, indicando o ponto final da viagem (ver *screenshot 2* da figura 27). Esta diferenciação de cores

nos marcadores permite que os utilizadores identifiquem facilmente onde a viagem foi iniciada e terminada. Além disso, permite ajudá-los a compreender a direção na qual o percurso foi realizado.

Durante o *tracking* da rota, é apresentada a indicação “Recording...” para informar o utilizador que uma rota está a ser registada. Além disso, é apresentado um círculo vermelho acima do botão correspondente na barra de navegação, informando ao utilizador que o *tracking* da rota está em curso, mesmo quando ele está a utilizar outras funcionalidades da aplicação AQUA.

3.4.7 Histórico de Rotas de Viagem e Reportes

As viagens registadas podem ser visualizadas pelo utilizador ao aceder ao seu histórico de rotas de viagem, que permite que os utilizadores revejam as suas viagens passadas, fornecendo uma maneira de recordar as suas experiências. Nesse ecrã são listadas todas as viagens registadas pelo utilizador, independentemente de terem sido submetidas ou não. Inicialmente, se existirem viagens por submeter, são listadas, juntamente com um botão que permite submeter todas as rotas armazenadas localmente. Se o utilizador estiver a fazer o *tracking* de uma rota naquele momento, é apresentada uma rota com a indicação de que está em andamento. Ao clicar nessa rota em andamento, o utilizador é redirecionado para o ecrã correspondente à funcionalidade do *tracking*.

Ao pressionar uma das rotas listadas, o utilizador tem a possibilidade de visualizar um pequeno mapa que apresenta a representação visual da rota da viagem com os respetivos marcadores. Além disso, são apresentadas informações como a data, a hora de início e fim da viagem, e a duração total. Caso o utilizador pressione uma rota que ainda não foi submetida, a aplicação disponibiliza a opção de eliminar essa rota, sendo a mesma removida permanentemente.

No que diz respeito ao histórico de reportes (ver *screenshots* 1 e 4 da figura 28), ao pressionar um reporte listado, são apresentadas informações relacionadas com o avistamento ou o incidente correspondente (ver *screenshots* 2 e 3 da figura 28). No caso de um avistamento, é apresentado uma mapa com um marcador que indica a localização do avistamento, a data e hora em que ocorreu, o comentário geral, se aplicável, e a lista de espécies avistadas, com as informações específicas sobre cada uma delas, enquanto no caso de um incidente é incluído o tipo de incidente.

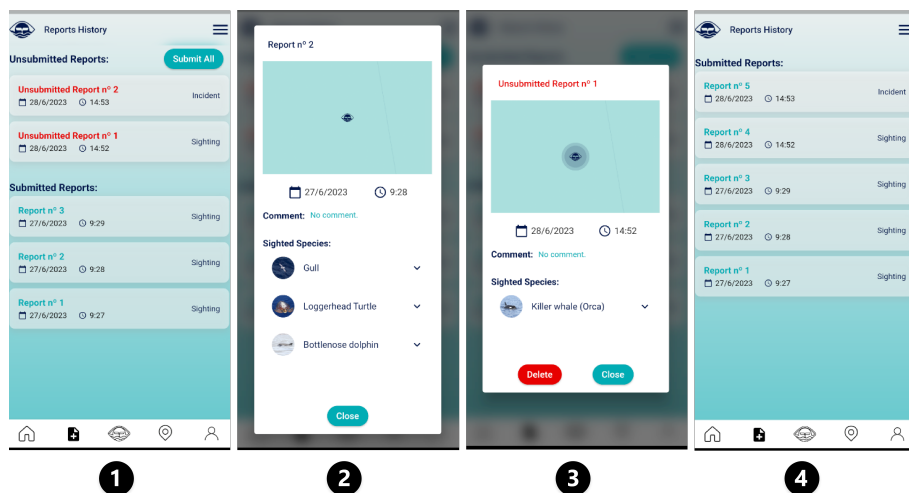


Figura 28. Screenshots da aplicação AQUA: (1) Ecrã do histórico de reportes; (2) Detalhes de um avistamento submetido e (3) não submetido; (4) Ecrã do histórico de reportes, após submissão dos reportes.

4 Modelação

Nesta secção são apresentados todos os elementos base para o desenvolvimento do sistema AQUA, incluindo os requisitos do sistema, os casos de uso, os atributos de qualidade, e a arquitetura do sistema.

4.1 Requisitos do Sistema

Os requisitos são essenciais para o desenvolvimento do sistema, visto que são eles que permitem compreender o funcionamento do mesmo e as necessidades que devem ser satisfeitas para assegurar o sucesso e qualidade do sistema. Os requisitos funcionais representam as funcionalidades esperadas no sistema, enquanto os requisitos não funcionais descrevem como o sistema deve executar uma ação, especificando atributos de qualidade.

A seguir, é apresentada uma lista dos principais requisitos funcionais da aplicação AQUA agrupados por funcionalidade. A lista completa de requisitos funcionais encontra-se no anexo C.1. Estes requisitos são a base para assegurar que o *software* atende às necessidades dos utilizadores e garantem uma experiência satisfatória e completa durante o uso da aplicação.

Para facilitar a organização e referência dos requisitos funcionais da aplicação AQUA, os mesmos serão listados seguindo uma nomenclatura específica para cada funcionalidade. Cada requisito será identificado pelo prefixo “RF” (Requisito Funcional), seguido das iniciais que representam a respetiva funcionalidade. Por exemplo, um requisito relacionado com o registo será identificado como “RF_R1”, representando o requisito funcional nº1 do registo.

Registo (R):

- **RF_R1:** O sistema deverá permitir que um novo utilizador se registe na aplicação através de um email, uma password e uma confirmação de password.
- **RF_R4:** O sistema deverá permitir ao utilizador adicionar o seu nome, país e cidade de residência após o registo.
- **RF_R5:** O sistema deverá permitir ao utilizador carregar uma fotografia de perfil caso pretenda, não sendo obrigatório.
- **RF_R6:** O sistema deverá permitir ao utilizador adicionar uma descrição do seu perfil caso pretenda, não sendo obrigatório.

Login (L):

- **RF_L1:** O sistema deverá permitir ao utilizador autenticar-se através do seu endereço de e-mail e password registados.
- **RF_L14:** O sistema deverá permitir ao utilizador efetuar logout da aplicação.

Excluir Conta (EC):

- **RF_EC1:** O sistema deverá permitir ao utilizador apagar a sua conta permanentemente.

- **RF_EC2:** O sistema deverá informar ao utilizador que, ao apagar a sua conta, perderá acesso à aplicação, mas que toda a informação reportada utilizando a aplicação será mantida.

Reportar Avistamento (RA):

- **RF_RA1:** O sistema deverá permitir ao utilizador capturar imagens ou vídeos utilizando a câmara do dispositivo.
- **RF_RA6:** O sistema deverá permitir ao utilizador pesquisar espécies específicas por nome.
- **RF_RA9:** O sistema deverá permitir ao utilizador identificar o número de indivíduos avistados para cada espécie (1, 2, 3, 4-10, 11-25, ou +25).
- **RF_RA11:** O sistema deverá permitir ao utilizador identificar o comportamento dos indivíduos avistados para cada espécie através das seguintes opções: feeding, playing, sleeping, other.
- **RF_RA18:** Após concluir a especificação da informação sobre o avistamento, o sistema deverá apresentar um sumário do mesmo, incluindo a localização num mapa, a data e hora do avistamento, uma lista de espécies avistadas com a informação preenchida para cada uma, o comentário geral, e as imagens e vídeos carregados.

Reportar Incidente (RI):

- **RF_RI1:** O sistema deverá permitir ao utilizador seleccionar o tipo de incidente a reportar, entre “dead animal”, “injured animal” e “other incident”.

Tracking da Viagem (TV):

- **RF_TV1:** O sistema deverá permitir ao utilizador iniciar o *tracking* da rota da viagem.

Perfil do Utilizador (PU):

- **RF_PU1:** O sistema deverá permitir ao utilizador visualizar o seu perfil.
- **RF_PU3:** O sistema deverá permitir ao utilizador seleccionar um “título” visível para outros utilizadores. As opções para o “título” devem ser todos os nomes dos distintivos desbloqueados pelo utilizador.

Coleção de Espécies (CE):

- **RF_CE1:** O sistema deverá permitir ao utilizador desbloquear a carta de uma espécie quando reporta o avistamento de uma espécie pela primeira vez e o avistamento for submetido com sucesso.
- **RF_CE2:** O sistema deverá permitir ao utilizador visualizar a sua coleção de cartas de espécies desbloqueadas, indicando o número total de golfinhos, baleias e outras espécies desbloqueadas.
- **RF_CE5:** O sistema deverá permitir ao utilizador tocar numa carta desbloqueada para visualizá-la com mais detalhe e interagir com ela para visualizar o outro lado da carta, onde é

possível visualizar a data de desbloqueio da mesma.

Distintivos (D):

- **RF_D1:** O sistema deverá permitir ao utilizador desbloquear um distintivo quando cumpre os requisitos específicos para cada distintivo.
- **RF_D2:** O sistema deverá permitir ao utilizador visualizar os distintivos desbloqueados, incluindo o número total de distintivos desbloqueados.
- **RF_D3:** O sistema deverá permitir ao utilizador tocar num distintivo desbloqueado para visualizar o distintivo com mais detalhe, incluindo o seu título, descrição e a data em que foi desbloqueado.

Ranking (R):

- **RF_R1:** O sistema deverá permitir ao utilizador visualizar o ranking global com todos os utilizadores do sistema.
- **RF_R2:** O sistema deverá permitir ao utilizador visualizar um ranking local com os utilizadores com a mesma localização que ele.
- **RF_R7:** Para cada utilizador no ranking, o sistema deverá apresentar a sua posição, a sua foto de perfil, o seu nome, o seu título (caso o tenha), e o número total de avistamento reportados, de espécies desbloqueadas ou de distintivos desbloqueados, dependendo do filtro selecionado pelo utilizador.

Mapa de Avistamentos (MA):

- **RF_MA1:** O sistema deverá apresentar um mapa com todos os avistamentos reportados pelos utilizadores na aplicação.
- **RF_MA2:** O sistema deverá apresentar os avistamentos em *clusters* com o número de avistamentos naquela zona.

Informação sobre Espécies (IE):

- **RF_IE3:** Ao selecionar uma espécie, o sistema deverá apresentar uma imagem da espécie, o seu nome comum, nome científico, descrição da espécie, uma descrição de como é possível identificar a espécie, e algumas curiosidades sobre a mesma.
- **RF_IE4:** O sistema deverá permitir ao utilizador ouvir a descrição da espécie, de como é possível identificá-la, e as curiosidades, em vez de ler o texto.

Histórico de Reportes (HR):

- **RF_HR1:** O sistema deverá permitir ao utilizador visualizar o seu histórico de reportes (avistamentos e incidentes) na aplicação, ordenados por data.

- **RF_HRV6:** O sistema deverá apresentar um mapa com as coordenadas do avistamento, a data e hora em que foi feito, o comentário geral e informações sobre cada espécie avistada.

Histórico de Rotas de Viagem (HRV):

- **RF_HRV1:** O sistema deverá permitir ao utilizador visualizar o seu histórico de rotas de viagem na aplicação, ordenadas por data.
- **RF_HRV4:** O sistema deverá permitir ao utilizador visualizar os detalhes de cada rota de viagem, sendo apresentado o mapa com o trajeto da viagem, a data, a duração, a hora de início e a hora de fim da viagem.

Tal como nos requisitos funcionais, cada requisito não funcional será identificado pelo prefixo “RNF” (Requisito Não Funcional). A seguir, é apresentada a lista dos requisitos não funcionais considerados mais relevantes:

RNF_1: O sistema deverá obter as coordenadas geográficas a partir do GPS em menos de 10 segundos.

RNF_2: Ao iniciar o tracking, o sistema deverá guardar localmente a localização do utilizador a cada 5 minutos, em *background*.

RNF_3: O sistema deverá apresentar um mapa indicando cada coordenada guardada, permitindo ao utilizador visualizar a rota da viagem em tempo real.

RNF_4: O utilizador deverá conseguir utilizar qualquer funcionalidade da aplicação sem conexão à Internet.

RNF_5: O utilizador não deverá demorar mais de 5 minutos para reportar um avistamento.

RNF_6: O sistema não deverá permitir o acesso à câmara sem autorização do utilizador.

RNF_7: O sistema deverá negar o acesso à aplicação a qualquer utilizador não autorizado.

RNF_8: A aplicação móvel AQUA deve ser compatível com vários dispositivos móveis e sistemas operativos.

4.2 Casos de Uso

Nesta secção serão apresentados os casos de uso (UC) ⁴ principais do sistema, mostrando como os requisitos funcionais do mesmo são realizados em termos de sequência de passos durante a interação entre os atores e o sistema. No Apêndice A encontram-se as descrições detalhadas dos casos de uso. A descrição de cada caso de uso principal é complementada com diagramas de casos de uso, sendo estes apresentados no Apêndice B.5. A seleção dos casos de uso principais teve em consideração os requisitos considerados mais relevantes para o funcionamento geral do sistema.

Na figura 29 é apresentado um diagrama de casos de uso geral, que representa todas as funcionalidades oferecidas pela aplicação AQUA a um utilizador.

⁴<https://www.usability.gov/how-to-and-tools/methods/use-cases.html>

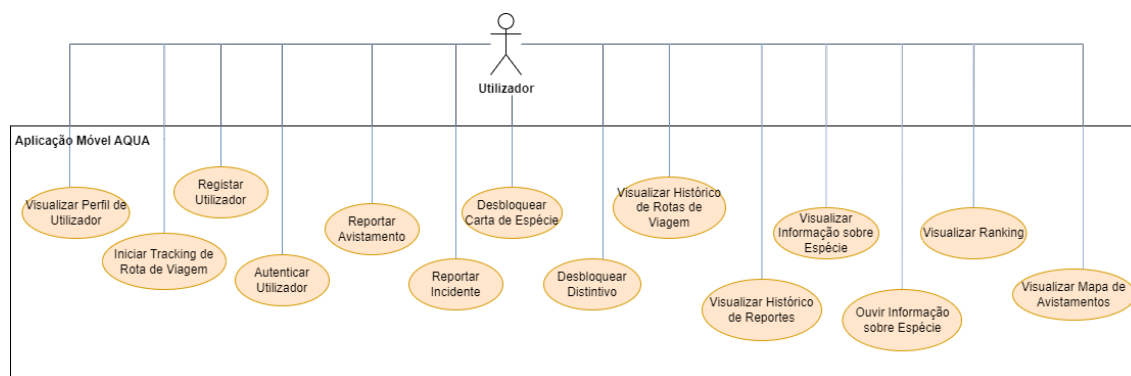


Figura 29. Diagrama de Casos de Uso Geral

A seguir, são listados os principais casos de usos identificados:

UC1 - Registrar Utilizador: O objetivo do caso de uso UC1 (ver figura 17 do Apêndice A) é permitir que um utilizador se registre na aplicação criando uma nova conta, fornecendo as informações solicitadas pelo sistema. Ao registrar-se, o utilizador poderá autenticar-se posteriormente para aceder às funcionais disponíveis na aplicação.

UC2 - Autenticar Utilizador: O caso de uso UC2 (ver figura 18 do Apêndice A) permite que um utilizador se autentique na aplicação e tenha acesso às funcionalidades disponíveis. Para autenticar-se na aplicação, o utilizador deve ter uma conta na aplicação. Desta forma, o utilizador poderá explorar as funcionalidades oferecidas pela aplicação.

UC3 - Excluir Conta: O caso de uso UC3 (ver figura 19 do Apêndice A) descreve a interação que permite que ao utilizador excluir a sua conta na aplicação. Ao excluir a conta, o utilizador perde o acesso à aplicação a partir das suas credenciais de login. Para realizar a exclusão, o sistema apresenta uma mensagem de confirmação indicando que a ação é irreversível.

UC4 - Reportar Avistamento: O objetivo do caso de uso UC4 (ver figura 20 do Apêndice A) é permitir que um utilizador reporte um avistamento de espécies na aplicação. Esta funcionalidade é a mais importante, visto que é essencial para a recolha de dados sobre as espécies, que posteriormente podem contribuir para estudos e monitorização das mesmas. Durante o processo de reporte, o utilizador tem a opção de capturar fotografias ou vídeos das espécies avistadas, identificar as espécies, assim como o número de indivíduos e comportamentos observados para cada uma. O avistamento reportado é enviado para o servidor quando há conexão à Internet. Caso contrário, o avistamento é armazenado localmente para envio posterior.

UC5 - Reportar Incidente do Tipo “Dead Animal” ou “Injured Animal”: O caso de uso UC5 (ver figura 21 do Apêndice A) descreve como o utilizador reporta um incidente envolvendo animais mortos ou feridos na aplicação. Esta funcionalidade permite identificar situações de emergência que podem ajudar a preservar a biodiversidade marinha. Durante o processo de reporte do incidente, o utilizador terá a opção de capturar fotografias ou vídeos do incidente, fornecer informações sobre as espécies envolvidas, tal como o número de indivíduos afetados e os comportamentos observados para cada uma. Tal como o avistamento, o incidente reportado é enviado para o servidor caso haja conexão à Internet. Caso contrário, o mesmo é armazenado localmente para envio posterior.

UC6 - Reportar Incidente do Tipo “Other”: O objetivo do caso de uso UC6 (ver figura 22 do Apêndice A) é permitir que o utilizador reporte um incidente distinta às categorias apresentadas na aplicação. Desta forma, o utilizador tem flexibilidade em relatar outros incidentes que considere relevantes. Durante o processo de reporte do incidente, o utilizador tem a opção de capturar fotografias e vídeos sobre o incidente e de descrever a situação detalhadamente. O incidente reportado é enviado para o servidor ou armazenado localmente para envio posterior, se houver conexão à Internet ou não, respetivamente.

UC7 - Iniciar Tracking da Rota da Viagem: O caso de uso UC7 (ver figura 23 do Apêndice A) descreve como o utilizador pode iniciar o *tracking* da rota de uma viagem. Esta funcionalidade utiliza o GPS do dispositivo para realizar o *tracking* da rota percorrida pelo utilizador durante a viagem. Ao iniciar o *tracking*, o utilizador tem acesso a um mapa onde pode visualizar as coordenadas iniciais da rota e, a cada 5 minutos, atualiza a localização atual do utilizador no mapa, traçando a rota em tempo real. O utilizador pode parar o *tracking* a qualquer momento para enviá-lo para o servidor. À semelhança do reporte de avistamentos e de incidentes, caso não haja conexão à Internet, a rota da viagem é armazenada localmente para envio posterior.

UC8 - Alterar Título: O caso de uso UC8 (ver figura 24 do Apêndice A) permite descrever como o utilizador pode alterar o título do seu perfil na aplicação. O título é uma pequena designação que o utilizador pode escolher para distinguir-se em relação aos restantes utilizadores da aplicação. Esta funcionalidade apenas está disponível para utilizadores que possuam pelo menos um distintivo desbloqueado.

UC9 - Desbloquear Carta de Espécie: O caso de uso UC9 (ver figura 25 do Apêndice A) descreve como o utilizador desbloqueia uma carta de uma espécie na aplicação. Esta funcionalidade está disponível para utilizadores que tenham efetuado login na aplicação, tenham acesso à Internet e tenham a carta da espécie bloqueada na sua coleção. Desta forma, o utilizador pode desbloquear uma nova carta de uma espécie e adicioná-la à sua coleção de espécies.

UC10 - Desbloquear Distintivo: À semelhança do caso de uso anterior, o caso de uso UC10 (ver figura 26 do Apêndice A) descreve como o utilizador desbloqueia um distintivo na aplicação. Esta funcionalidade está disponível para utilizadores que tenham efetuado login na aplicação, tenham acesso à Internet e tenham o distintivo bloqueado na sua coleção. Desta forma, o utilizador pode desbloquear um novo distintivo e adicioná-lo à sua coleção de distintivos. Após desbloquear um distintivo, o título do respetivo distintivo é adicionado como opção na lista de títulos disponíveis no perfil do utilizador.

UC11 - Visualizar Ranking: O objetivo do caso de uso UC11 (ver figura 27 do Apêndice A) é permitir que o utilizador visualize o *ranking* na aplicação. O *ranking* fornece informações sobre a classificação dos utilizador com base num dado específico, como o número de avistamentos reportados, espécies desbloqueadas e distintivos desbloqueados. O objetivo é permitir que o utilizador possa visualizar a sua posição em relação a outros utilizadores. Ao utilizar esta funcionalidade, o utilizador pode alternar entre o *ranking* "Global" ou "Local". Ao selecionar a opção "Local", o sistema atualiza o *ranking* e apresenta os utilizadores com a mesma localização do utilizador. Para além disso, o utilizador tem a opção de filtrar o *ranking* por intervalo de tempo ou tipo de dados específico. Ao aplicar um filtro, o sistema atualiza o *ranking* e organiza-o por ordem decrescente com base no tipo de dado selecionado.

UC12 - Visualizar Mapa de Avistamentos: O caso de uso UC12 (ver figura 28 do Apêndice A) descreve como o utilizador visualiza um mapa com todos os avistamentos reportados a partir da aplicação, com a capacidade de filtrar esses avistamentos por período. Ao aceder ao mapa de avistamentos, o sistema apresenta ao utilizador um mapa com *clusters* e áreas que representam todos os avistamentos reportados dentro do período selecionado. Durante a visualização do mapa, o utilizador pode selecionar um *cluster* para expandi-lo e visualizar os avistamentos individuais contidos nele. Ainda, o utilizador tem a opção de fazer *zoom in* e *zoom out* no mapa para explorar diferentes zonas com maior detalhe.

UC13 - Visualizar Informação sobre Espécie: O objetivo do caso de uso UC13 (ver figura 29 do Apêndice A) é descrever a funcionalidade que permite ao utilizador visualizar informações sobre uma espécie específica. O utilizador seleciona o tipo de animal, o tamanho, e o sistema apresenta uma lista de espécies com as características selecionadas. Posteriormente, o utilizador seleciona uma espécie e visualiza algumas informações sobre a mesma. Caso o utilizador pretenda, pode optar por ouvir a informação da espécie em áudio.

UC14 - Visualizar Histórico de Reportes: O caso de uso UC14 (ver figura 30 do Apêndice A) permite ao utilizador visualizar o histórico dos seus reportes submetidos e não submetidos. O sistema apresenta ao utilizador duas listas separadas por reportes submetidos e não submetidos, ordenados por data. O utilizador tem a opção de selecionar um reporte para ver detalhes sobre o mesmo, como as coordenadas, data e hora, espécies envolvidas e informações sobre as mesmas. O sistema apresenta ao utilizador a opção "Submit All" caso o utilizador tenha reportes não submetidos. Ao selecionar essa opção, todos os reportes não submetidos são enviados para o servidor caso haja conexão à Internet. Caso contrário, é apresentada uma mensagem de aviso. Para além disso, o utilizador pode excluir reportes não submetidos.

UC15 - Visualizar Histórico de Rotas de Viagem: O caso de uso UC15 (ver figura 31 do Apêndice A) permite ao utilizador visualizar o seu histórico de rotas de viagem. À semelhança do UC14, o sistema apresenta ao utilizador duas listas separadas por rotas de viagem submetidas e não submetidas, ordenados por data. O utilizador tem a opção de selecionar uma rota de viagem para ver detalhes sobre o mesmo, a rota num mapa, data, duração, e hora de início e fim da mesma. O sistema apresenta ao utilizador a opção "Submit All" caso o utilizador tenha rotas de viagem não submetidas. Ao selecionar essa opção, as mesmas são enviadas para o servidor caso haja conexão à Internet. Caso contrário, é apresentada uma mensagem de aviso. O utilizador pode excluir rotas de viagem não submetidas.

4.3 Atributos de Qualidade

Os atributos de qualidade são propriedades do sistema que permitem indicar a forma como o mesmo satisfaz necessidades dos utilizadores e das suas partes interessadas, e têm uma importância significativa na arquitetura do sistema. Sendo assim, fez-se uma priorização de alguns atributos de qualidade considerados mais importantes para o sistema a desenvolver, utilizando a tabela de *Brousseau* [60] (ver tabela 6). A partir da tabela é possível comparar e ordenar os *scores* obtidos para cada atributo, e, desta forma, podem ser identificados os atributos de qualidade mais prioritários a ter em conta durante o desenho da arquitetura do sistema e seleção das tecnologias a utilizar para os satisfazer.

Tabela 6. Tabela de Brosseau - Priorização de Atributos de Qualidade

Atributo	Score	Desempenho	Usabilidade	Disponibilidade	Modificabilidade	Segurança
Desempenho	3		^	<	<	<
Usabilidade	4			<	<	<
Disponibilidade	2				<	<
Modificabilidade	1					<
Segurança	0					

Nas próximas subseções, dedicadas a atributos de qualidade específicos, são apresentados cenários relacionados a cada atributo e discutida a sua relevância para o sistema. Esses cenários não apenas possibilitam análises mais detalhadas, mas também ajudam na identificação, descrição e teste dos mesmos.

4.3.1 Usabilidade

A seguir, é apresentado um cenário de qualidade relacionado com a usabilidade. Este cenário tem como objetivo permitir avaliar como a aplicação AQUA atende aos requisitos de facilidade de utilização. Para exemplificar a importância da usabilidade, na tabela abaixo é apresentado um cenário de qualidade relacionado com a procura de informações sobre espécies de baleias.

Tabela 7. Cenário de Qualidade - Usabilidade

Fonte: Utilizador
Estímulo: O utilizador procura informações sobre uma espécie de baleia.
Ambiente: A aplicação está em execução.
Artefatos: Interface da aplicação
Resposta: O utilizador encontra rapidamente a opção de procurar informações sobre espécies de baleias. A aplicação apresenta uma lista de espécies de baleias. O utilizador seleciona uma espécie e são apresentadas informações sobre a mesma.
Medição de Resposta: O tempo necessário para o utilizador encontrar as informações desejadas não deve exceder os 2 minutos.

Este cenário de qualidade ilustra a importância de desenhar uma interface intuitiva e de fácil utilização da aplicação AQUA. Um utilizador que pretenda obter informações sobre uma espécie deve conseguir encontrá-la rapidamente. Uma demora significativa em realizar uma determinada tarefa pode ser um indicativo de que a interface é de difícil compreensão, impactando negativamente a usabilidade do sistema. Desta forma, o tempo necessário para realizar a tarefa é uma métrica que permite avaliar o quão clara é a interação do utilizador com a aplicação.

4.3.2 Desempenho

O desempenho é um dos atributos de qualidade mais importantes para qualquer sistema e está relacionado diretamente com a satisfação dos utilizadores.

A tabela abaixo apresenta um cenário de qualidade relacionado com este atributo, cujo objetivo é verificar o tempo máximo necessário para o sistema obter as coordenadas geográficas do dispositivo através do GPS integrado. Este cenário avalia o desempenho do sistema quanto à rapidez

e eficiência na obtenção dessas coordenadas, com um limite de 10 segundos estabelecidos como métrica para garantir uma resposta satisfatória ao utilizador.

Tabela 8. Cenário de Qualidade - Desempenho

Fonte: Sistema
Estímulo: O sistema inicia a obtenção das coordenadas geográficas a partir GPS.
Ambiente: A aplicação está em execução.
Artefatos: Dispositivo móvel com GPS integrado.
Resposta: O sistema obtém as coordenadas geográficas do GPS do dispositivo com sucesso.
Medição de Resposta: O tempo máximo para obtenção das coordenadas geográficas é de 10 segundos.

4.3.3 Disponibilidade

A disponibilidade é um atributo de qualidade relacionado com a capacidade do sistema estar acessível aos utilizadores sempre que for necessário. Esta propriedade é essencial em sistemas dependentes da Internet, como aplicações móveis, principalmente quando a conectividade é instável ou inexistente. Nesse contexto, o sistema deve manter o seu funcionamento, sem afetar a experiência dos utilizadores.

O cenário seguinte avalia a capacidade do sistema em lidar com a falta de conexão à Internet e garantir o funcionamento essencial da aplicação para os utilizadores.

Tabela 9. Cenário de Qualidade - Disponibilidade

Fonte: Utilizador
Estímulo: O utilizador submete um avistamento.
Ambiente: A aplicação está em execução e o dispositivo não tem conexão à Internet.
Artefatos: Dispositivo móvel, aplicação móvel.
Resposta: A aplicação apresenta uma mensagem informando que o avistamento foi guardado localmente.
Medição de Resposta: O utilizador consegue reportar um avistamento mesmo sem conexão à Internet.

4.3.4 Modificabilidade

A modificabilidade é um atributo de qualidade que se refere à capacidade de realizar alterações e atualizações de forma eficaz.

Para garantir a modificabilidade é necessário criar uma arquitetura flexível e modular. Um sistema deve ser acoplado e coeso, permitindo que cada componente do sistema seja tratado de forma independente sem afetar os restantes. Na tabela abaixo é apresentado um cenário de qualidade relacionado com este atributo, em que o objetivo é que o desenvolvedor seja capaz de adicionar uma nova espécie ao sistema. Desta forma, o sistema é avaliado com a intenção de permitir que as modificações e atualizações nos dados seja feita de forma rápida e que não impactem negativamente o funcionamento do sistema.

Tabela 10. Cenário de Qualidade - Modificabilidade

Fonte: Desenvolvedor
Estímulo: O desenvolvedor deseja adicionar uma nova espécie ao sistema.
Ambiente: O sistema está em execução.
Artefatos: Base de dados
Resposta: O desenvolvedor consegue adiciona uma nova espécie ao sistema, inserindo todas as informações necessárias sobre a espécie.
Medição de Resposta: O tempo necessário para adicionar uma nova espécie não deve exceder 5 minutos.

4.3.5 Segurança

A segurança é um atributo de qualidade de extrema importância em qualquer sistema. Para além de garantir a segurança das informações pessoais e das ações realizadas, a conta do utilizador permite a distinção e identificação dos utilizadores num sistema, e permite personalizar a experiência de cada utilizador com base nas preferências e características de cada um.

Em sistemas que utilizam gamificação, especialmente envolvendo *rankings* e competições entre utilizadores, como é o caso da aplicação móvel AQUA, é essencial assegurar a confidencialidade e veracidade das informações apresentadas aos utilizadores. Quando se trata de *rankings*, é necessário garantir que as pontuações sejam justas e protegidas contra manipulações por parte dos utilizadores.

O conceito de segurança abrange várias propriedades. Entre essas propriedades pode-se destacar a confidencialidade, que preserva a privacidade de informação com mecanismos de encriptação e de autorização que impedem o acesso não autorizado às informações, e a integridade, que garante que as informações não são alteradas por entidades não autorizadas, usando propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade) na base de dados PostgreSQL no caso do sistema AQUA.

A seguir, são apresentados dois cenários que têm como objetivo garantir que a confidencialidade dos dados, assim como a segurança dos processos de interação do utilizador com o sistema, transmitindo ao utilizador uma experiência segura.

Tabela 11. Cenário de Qualidade - Segurança (Confidencialidade)

Fonte: Sistema
Estímulo: O sistema inicia a câmara para capturar uma fotografia.
Ambiente: A aplicação está em execução.
Artefatos: Dispositivo móvel, aplicação móvel.
Resposta: O sistema solicita ao utilizador a permissão para aceder à câmara do dispositivo.
Medição de Resposta: O sistema apresenta o pedido de permissão num tempo máximo de 2 segundos e aguarda a resposta do utilizador. Se o utilizador permitir o acesso, o sistema acede à câmara do dispositivo.

No cenário anterior, o sistema solicita permissão ao utilizador para aceder à câmara do dispositivo e, se concedida, permite a captura de fotografias, garantindo a privacidade das informações capturadas.

Tabela 12. Cenário de Qualidade - Segurança (Autenticação)

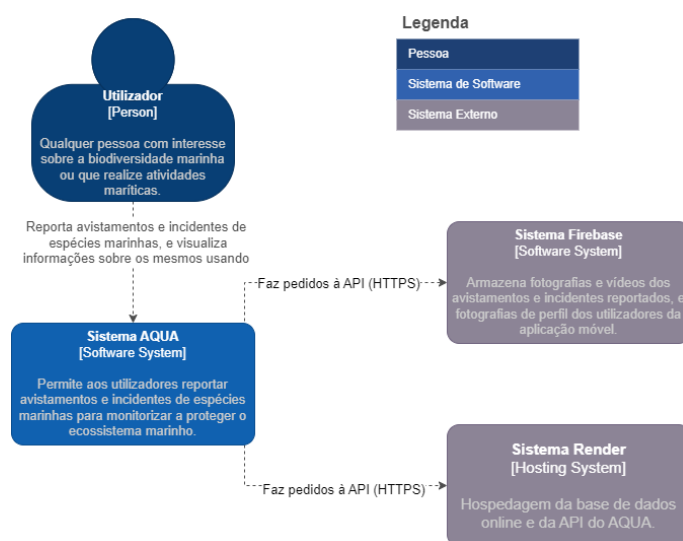
Fonte: Utilizador
Estímulo: O utilizador insere as suas credenciais de login na interface da aplicação móvel.
Ambiente: A aplicação está em execução.
Artefactos: Interface da aplicação, credenciais, servidor
Resposta: A aplicação recebe as credenciais inseridas pelo utilizador. As credenciais são enviadas para o servidor. O servidor verifica se as credenciais são válidas. Se forem válidas, o servidor envia uma resposta de sucesso para aplicação e o utilizador é autenticado. Caso contrário, é apresentada é enviada uma resposta de erro, indicando que as credenciais são inválidas e o acesso é negado.
Medição de Resposta: O tempo médio de autenticação.

Neste cenário o utilizador insere as suas credenciais na interface da aplicação, que são enviadas para validação no servidor. Se forem válidas, o servidor responde com sucesso, autenticando o utilizador. Caso contrário, o servidor nega a autenticação.

4.4 Arquitetura do Sistema

Neste capítulo é apresentada detalhadamente a arquitetura do sistema AQUA, através do Modelo C4, ⁵ este modelo é uma abordagem visual para descrever a arquitetura de *software*, fornecendo um visão clara e concisa das relações entre os diferentes componentes do sistema. O Modelo C4 é composto por 4 níveis de abstracção (Contexto, Container, Componente, Código), contudo, apenas serão apresentados os primeiros três níveis, devido ao facto do último nível apresentar um nível de abstracção muito baixo, descrevendo pontos de implementação que não são relevantes no presente contexto. No Apêndice B.2 é possível observar uma maior resolução das figuras apresentadas ao longo desta secção.

4.4.1 Modelo C4 - Diagrama de Contexto

**Figura 30.** Modelo C4: Diagrama de Contexto

⁵<https://c4model.com/>

O diagrama de contexto é uma representação visual com um elevado nível de abstração da arquitetura, apenas representando os sistemas externos existentes e os diferentes tipos de utilizadores, assim como a interação dos mesmos com o sistema. Na figura 30 é possível observar que o diagrama de contexto representa um nível pouco pormenorizado. No sistema AQUA existe um único tipo de utilizador que interage com o sistema, visto que todos os utilizadores têm os mesmos acessos e permissões. O sistema AQUA comunica com outros sistemas externos, como é o caso do sistema do Firebase, responsável pelo armazenamento de todas as fotografias e vídeos dos avistamentos e incidentes reportados na aplicação móvel, assim como todas as fotografias de perfil dos seus utilizadores. Ainda, o sistema comunica com o sistema de *hosting* Render⁶, na qual a API do AQUA e a base de dados *online* são hospedados.

4.4.2 Modelo C4 - Diagrama de Containers

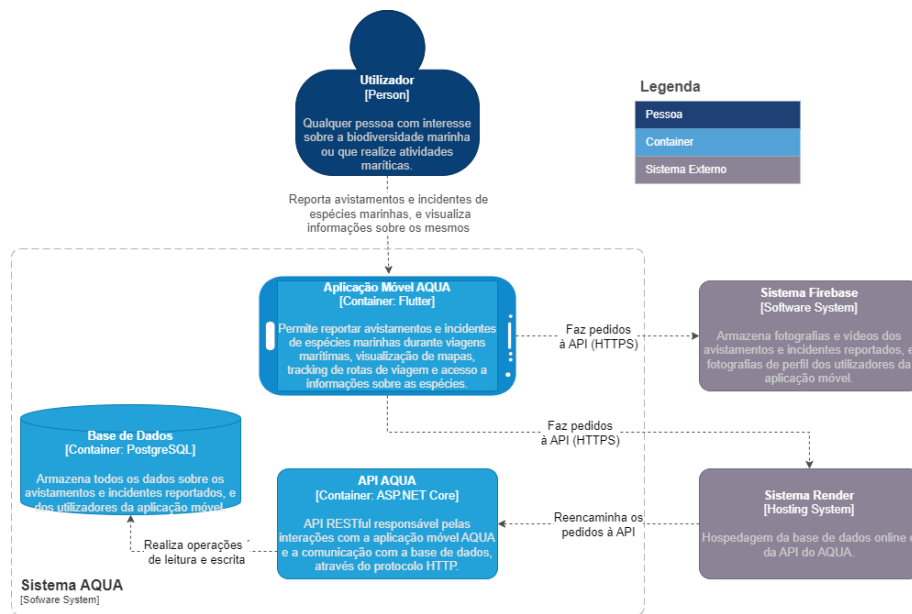


Figura 31. Modelo C4: Diagrama de Containers

O diagrama de *containers* do Modelo C4 é uma representação visual da decomposição de um sistema em distintos *containers* inter-relacionados, cada um representando uma aplicação ou micro-sistemas independentes, tais como uma aplicação para *end-users* ou uma base de dados. Na figura 31 é possível observar a interação entre os distintos *containers* do sistema AQUA. O *container* da Aplicação Móvel AQUA é o responsável pela interação do utilizador com o sistema. É este *container* que se comunica com outros sistemas externos, como o “Sistema Firebase” e o “Sistema Render”, através de pedidos HTTPS. Outro *container* do sistema AQUA é a “API AQUA”, uma API Restful que serve de intermediária da comunicação entre a aplicação móvel e a base de dados online. Contudo, a “API AQUA” e a “Base de Dados”, apesar de pertencerem ao sistema AQUA, os mesmos estão hospedados no “Sistema Render”. Desta forma, a “Aplicação Móvel AQUA” interage de forma indireta com estes *containers*, através de pedidos HTTPS ao “Sistema Render”. Este sistema, posteriormente, é responsável por reencaminhar os pedidos da aplicação móvel à “API

⁶<https://render.com/>

AQUA”. Dependendo do tipo de pedido, a “API AQUA” realiza operações de leitura ou escrita nas informações armazenadas na “Base de Dados”.

4.4.3 Modelo C4 - Diagrama de Componentes

O diagrama de componentes do Modelo C4 é uma representação visual da decomposição de um *container* em diferentes *componentes* inter-relacionados. Além disso, permite visualizar as relações desses componentes com outros *containers* ou outros sistemas externos. Neste contexto, a seguir, são apresentados dois diagramas de componentes: um para o *container* da aplicação móvel AQUA, e outro para o *container* da API.

– Diagrama de Componentes (Aplicação Móvel AQUA):

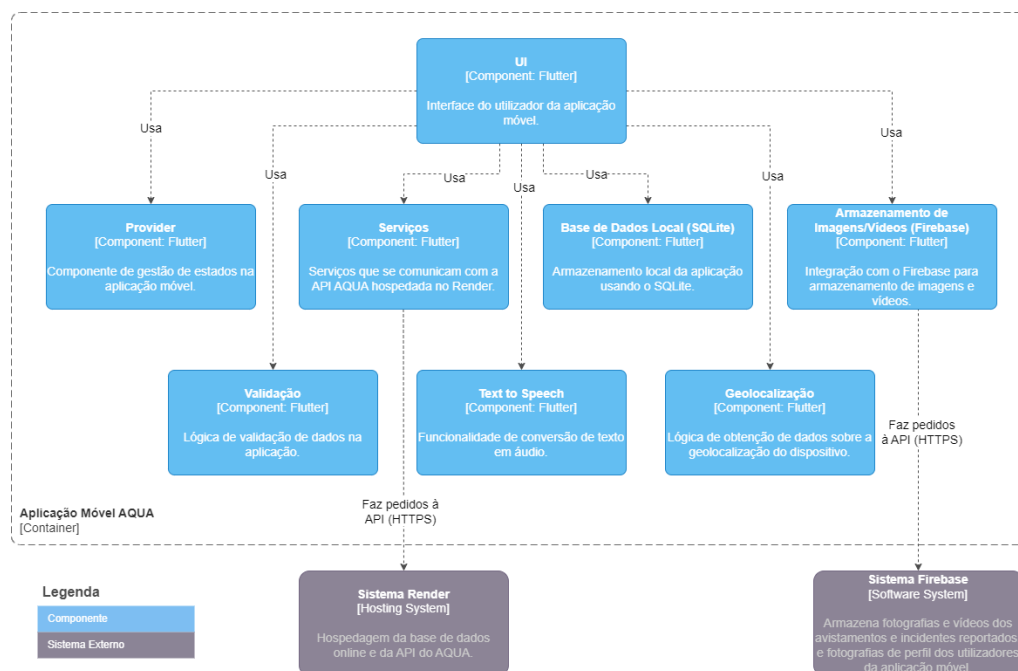


Figura 32. Modelo C4: Diagrama de Componentes (Aplicação Móvel AQUA)

Na figura 32 é apresentado o diagrama de componentes do *container* “Aplicação Móvel AQUA”. Este diagrama reflete a estrutura interna dos componentes que compõem a aplicação móvel, demonstrando como eles interagem para fornecer a funcionalidade completa aos seus utilizadores. O *container* “Aplicação Móvel AQUA” é composta pelos seguintes componentes:

- O componente “UI” é responsável pela interação direta com os utilizadores, representando a interface de utilizador. Este componente utiliza os restantes componentes do *container*.
- O componente “Provider” é responsável pela gestão de estados na aplicação, garantindo que a interface de utilizador reflete sempre os dados atualizados.
- O componente “Serviços” faz pedidos HTTPS ao “Sistema Render”. É a partir destes pedidos que a aplicação móvel consegue realizar operações nas informações da base de dados *online*.

- O componente “Validação” é um componente que permite garantir a integridade dos dados, verificando se as informações inseridas pelos utilizadores estão de acordo com as regras e formatos definidos.
- O componente “Text to Speech” é responsável pela conversão de texto em áudio, permitindo que as informações possam ser escutadas, melhorando a experiência dos utilizadores.
- O componente “Geolocalização” utiliza as capacidades do dispositivo móvel, como o GPS, para obter as coordenadas geográficas do utilizador.
- O componente “Base de Dados Local” é responsável por armazenar informações localmente no dispositivo, permitindo que os utilizadores tenham acesso aos dados mesmo em modo *offline*. Desta forma, a aplicação móvel mantém a sua funcionalidade, mesmo em locais com conectividade limitada.
- Por fim, o componente “Armazenamento de Imagens/Vídeos” realiza pedidos HTTPS ao “Sistema Firebase”, responsável pelo armazenamento de fotografias e vídeos carregados e apresentados na aplicação móvel.

– Diagrama de Componentes (API AQUA):

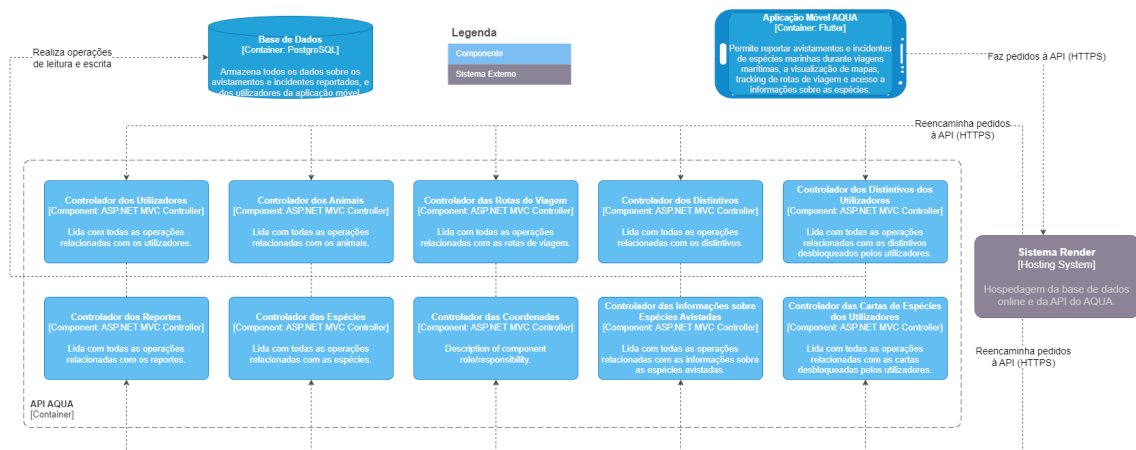


Figura 33. Modelo C4: Diagrama de Componentes (API AQUA)

De igual forma, na figura 33 é apresentado o diagrama de componentes do *container* “API AQUA”. Este diagrama reflete a estrutura interna dos componentes que compõem a “API AQUA” e a forma como interagem entre eles.

Na figura é possível observar que a “Aplicação Móvel AQUA”, quando faz pedidos HTTPS à API, os mesmos são enviados para o “Sistema Render”. Este sistema é um *proxy* responsável por reencaminhar os pedidos à “API AQUA”. Os pedidos enviados são redirecionados para os diferentes controladores, dependendo dos dados dos modelos que estão a ser acedidos. Por exemplo, ao enviar pedidos para operar sobre os dados dos utilizadores, os mesmos são redirecionados para o componente “Controlador dos Utilizadores”, responsável por lidar com todas as operações relacionados com os utilizadores. Cada controlador é responsável por interagir com o *container* “Base de Dados”, onde são realizadas as operações de leitura e escrita pretendidas nos respetivos dados.

5 Implementação

Nesta secção é descrita detalhadamente a implementação do sistema AQUA, incluindo tanto o desenvolvimento da aplicação móvel quanto da API, bem como o armazenamento dos dados *online*.

Ao longo desta secção, são explorados diversos aspetos, como a integração de recursos como a geolocalização e a visualização de mapas na aplicação móvel, e a implementação de boas práticas de desenvolvimento e de métodos utilizados para satisfazer os requisitos de todo o sistema. Para além disso, são apresentados testes funcionais realizados com o objetivo de testar o funcionamento da funcionalidade do *tracking* das rotas de viagem.

5.1 Desenvolvimento da Aplicação Móvel

Nesta subsecção são apresentados detalhes sobre a implementação da aplicação móvel AQUA desenvolvida em Flutter. São explorados aspetos técnicos e escolhas efetuadas durante o processo de desenvolvimento, abordando temas como a arquitetura da aplicação, a estrutura do projeto, a gestão de estados e integração com serviços externos. Além disso, são apresentados alguns desafios encontrados e as soluções adotadas para cada um.

5.1.1 Arquitetura MVVM

A aplicação móvel foi desenvolvida com base na arquitetura de software MVVM (*Model-View-ViewModel*). Esta arquitetura separa a lógica de negócio da interface do utilizador e dos dados em três componentes: Vista, Modelo de Vista e Modelo (ver figura 34).

A **Vista** é responsável por apresentar a interface do utilizador e interagir com o utilizador, refletindo o estado atual dos dados do Modelo de Vista. O **Modelo de Vista**, por sua vez, fornece dados e lógica de apresentação à Vista, bem como é responsável pela comunicação com o Modelo para obter dados. Além disso, ele notifica a Vista sobre mudanças nos dados. Por fim, o **Modelo** representa dados e lógica de acesso aos dados e fornece dados ao Modelo de Vista.

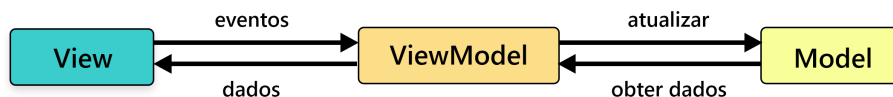


Figura 34. Arquitetura MVVM - Interação entre a Vista, Modelo de Vista, e Modelo

Na figura 35 é possível visualizar um exemplo de como a arquitetura é implementada na obtenção de dados do utilizador durante a inicialização da aplicação AQUA.



Figura 35. Implementação do MVVM: A Vista (1) solicita os dados dos utilizadores ao Modelo de Vista (método *getAllUsers*). O Modelo de Vista (2) solicita os dados ao Modelo, através da chamada ao serviço correspondente (3) e notifica a Vista sobre as alterações nos dados, através do método *notifyListeners*

5.1.2 Boas Práticas

Nesta secção são apresentadas algumas boas práticas adotadas durante o desenvolvimento da aplicação móvel AQUA. Entre essas boas práticas podemos destacar:

1. **Separação de responsabilidades:** Utilizou-se uma arquitetura MVVM (*Model-View-ViewModel*), que permitiu separar as responsabilidades do código e manter o código organizado.
2. **Organização do código:** Utilizou-se uma estrutura de diretórios clara e consistente para cada componente da aplicação, como os ecrãs, modelos, serviços e *widgets*.
3. **Gestão de estados:** Utilizou-se o Provider para a gestão de estados da aplicação AQUA, em vez do gestor de estados nativo “Set State” do Flutter.
4. **Reutilização de *widgets*:** Criou-se *widgets* personalizados que permitissem a sua reutilização, evitando duplicação de código.
5. **Utilização de *widgets stateless* sempre que possível:** Utilizou-se *widgets stateless* sempre que possível, pois estes *widgets* são mais simples e não são atualizados sempre que ocorrer uma atualização na interface do utilizador.
6. **Tratamento de erros e exceções:** Utilizou-se blocos *try-catch* para fazer o tratamento de possíveis exceções e apresentar mensagens de erro adequadas aos utilizadores.
7. **Objetos constantes:** Utilizou-se a palavra-chave “const” para tornar objetos constantes. Ao usar o “const”, consegue-se indicar que um objeto não deve ser modificado após a compilação, o que permite que hajam melhorias no desempenho.
8. **Gestão de versões:** A utilização do *GitHub* permitiu acompanhar detalhadamente as alterações, além de facilitar uma colaboração até com outros desenvolvedores que possam continuar o trabalho desenvolvido.

Ao longo da secção 5.1 são descritos detalhadamente algumas das práticas acima mencionadas.

5.1.3 Estrutura do Projeto

Nesta secção é apresentada uma visão geral da estrutura do projeto. A organização do projeto teve como objetivo facilitar a localização do código, simplificar a sua manutenção e escalabilidade,

e permitir a reutilização de código, para além de facilitar o desenvolvimento da aplicação. Os principais diretórios principais do projeto são:

Diretório “lib”: O diretório principal da aplicação móvel, onde a maioria do código-fonte se encontra. Este diretório é subdividido em:

- **Sub-diretório “constants”:** Armazena constantes da aplicação, como estilos de fonte de texto, cores, enumerações e mensagens de texto pré-definidas, permitindo que as mesmas sejam mapeadas e utilizadas como uma biblioteca interna, o que facilita o seu acesso e a alteração no futuro.
- **Sub-diretório “database”:** Gere todas as operações relacionadas com a base de dados SQLite.
- **Sub-diretório “models”:** Contém todas as definições das classes dos modelos utilizados na aplicação.
- **Sub-diretório “providers”:** Armazena todos os *Providers* utilizados para a gestão de estados da aplicação utilizando o *Provider*. Isto será descrito mais detalhadamente na secção 5.1.5.
- **Sub-diretório “screens”:** Armazena todas as classes para cada ecrã da aplicação.
- **Sub-diretório “services”:** Gere toda a lógica de negócios e a comunicação com a API REST.
- **Sub-diretório “utils”:** Armazena métodos e classes que são utilizadas em diferentes partes da aplicação, como classes e métodos para a criação de animações, validação de dados, verificação da Internet, obtenção de coordenadas, formatação de datas, e transformação de texto em áudio.
- **Sub-diretório “widgets”:** Armazena todos os *widgets* personalizados desenvolvidos de forma a serem reutilizados em vários ecrãs da aplicação.

Diretório “assets”: Utilizado para armazenar recursos como imagens e fontes.

Diretório “android” e “ios”: Contém os projetos nativos do sistema operativo Android e iOS, respetivamente, gerados automaticamente pelo Flutter.

Além dos diretórios apresentados, é importante destacar o **ficheiro “pubspec.yaml”** pela sua importância no projeto, visto que é ele que contém todas as dependências do projeto, bibliotecas utilizadas e configurações da aplicação, como o nome, a versão e as permissões.

5.1.4 Padrão Singleton

O padrão Singleton é um padrão de desenho de *software* usado para garantir uma única instância de uma classe, fornecendo um ponto de acesso global a essa instância. Isto é essencial para controlar a consistência dos dados e evitar problemas de múltiplas instâncias. A sua implementação envolveu um método estático que cria ou retorna a única instância existente da classe, assegurando que apenas uma instância é criada e reutilizada quando necessário.

Na aplicação AQUA, implementou-se uma classe chamada **‘APIServices’**, responsável pela gestão de todos os serviços de comunicação com o servidor através da API. Sendo esta classe utilizada em todo o sistema, para evitar a criação de várias instâncias sempre que um serviço fosse chamado, aplicou-se o padrão Singleton. Este padrão garante que a classe **‘APIServices’** possui apenas uma única instância e fornece um ponto de acesso global para todo o sistema, ou seja, a partir de qualquer parte do código a mesma instância dessa classe pode ser utilizada.

Na figura 36 é possível observar a implementação do padrão Singleton na classe '**APIServices**'. Ao chamar um serviço, a classe verifica se já existe uma instância criada. Caso existir, essa instância é retornada. Caso contrário, uma nova instância é criada.

```
class APIServices {
  static APIServices? _instance;

  factory APIServices() {
    _instance ??= APIServices._();
    return _instance!;
  }

  APIServices._();
}
```

Figura 36. Implementação do padrão 'Singleton' na classe 'APIServices'

De igual forma, foi implementada a classe '**DatabaseHelper**' para fornecer um ponto de acesso global ao armazenamento local utilizando o SQLite. Esta classe é responsável por criar e executar as operações CRUD (Create, Read, Update and Delete) na base de dados local. De forma a evitar a criação de várias bases de dados *offline*, foi implementado o padrão Singleton na classe '**DatabaseHelper**'.

Ao aplicar este padrão, garantiu-se que apenas uma instância da classe é criada, sendo todos os dados armazenados e lidos a partir da mesma base de dados, independentemente da parte do código que estiver sendo acedida. Assim, é possível fazer uma melhor gestão dos dados e evitar problemas de inconsistência que poderiam surgir caso fossem criadas várias bases de dados.

5.1.5 Gestão de Estados com Provider

Após criar um ecrã da aplicação no Flutter, surge a necessidade de gerir os estados da mesma. O Flutter desenha os elementos da interface apenas uma vez para evitar sobrecarregamento do processador. No entanto, quando ocorrem mudanças nas informações apresentadas, é preciso redesenhar a interface para apresentar os dados atualizados.

Nativamente, o Flutter possui o "**Set State**", uma opção para notificar as mudanças e permitir redesenhar parte do ecrã. Embora seja simples de usar, esta abordagem pode impactar o desempenho da aplicação. Ao chamar o "**Set State**", todos os *widgets* que compõem o *stateful widget* são redesenhadas, mesmo que apenas uma informação tenha sido atualizada, resultando num desperdício de recursos e num baixo desempenho da aplicação.

Existem várias opções para gerir os estados em aplicações desenvolvidas em Flutter, como o GetX, BLoC e Provider. Para a gestão dos estados da aplicação AQUA, escolheu-se utilizar o Provider. O Provider é uma solução simples e fácil de usar, oficialmente mantida pela equipa do Flutter. Sendo atualizado regularmente, ele oferece estabilidade ao longo do tempo. Além disso, o Provider permite gerir estados globais e locais específicos de partes da aplicação.

Estas soluções têm como objetivo resolver os impactos negativos no desempenho ao usar o gestor de estados nativo do Flutter. Desta forma, o Provider garante alto desempenho e eficiência, evitando a necessidade de redesenhar *widgets* quando ocorre uma alteração no estado da aplicação.

Para implementar a gestão de estados, criou-se diversos **'Providers'** para cada estado específico da aplicação. Dentro de cada **'Provider'**, definiu-se variáveis que representam o estado a ser gerido. Estas variáveis podem ser acedidas e atualizadas pelos *widgets* conectados ao respetivo **'Provider'**.

Sempre que ocorre uma alteração no estado, utilizou-se o método **'notifyListeners()'** para notificar os *widgets* conectados ao respetivo **'Provider'**. Desta forma, os *widgets* conectados são atualizados automaticamente com os novos dados.

Para que um *widget* aceda e acompanhe as alterações do estado gerido pelo **'Provider'**, utilizou-se o *widget* **'Consumer'** ou o **'Provider.of'** no Flutter. Além disso, utilizou-se o *widget* **'Selector'** para atualizar apenas *widgets* específicos. Os **'Selectors'** permitem definir quais partes do estado devem ser observadas por um determinado *widget*, evitando atualizações desnecessárias.

No desenvolvimento da aplicação AQUA criou-se vários **'Providers'** para manter a aplicação sempre atualizada. Estes permitem que os utilizadores visualizem as informações do estado da aplicação em tempo real. Quando ocorre uma alteração nos dados, apenas os *widgets* relacionados são redesenhados, garantindo um bom desempenho da aplicação e uma melhor experiência de utilização. Entre esses **'Providers'** pode-se destacar os seguintes:

- **Provider Geral:** Este Provider é responsável por gerir estados relacionados com a aplicação em geral, ou seja, os estados globais que são comuns em diversas partes da aplicação, como a indicação na barra de navegação de que uma rota de viagem está a ser gravada.
- **Provider dos Avistamentos:** Este Provider é responsável por gerir estados relacionados com os dados do avistamento a ser reportado, como a data e hora, coordenadas, e informação sobre as espécies avistadas.
- **Provider dos Incidentes:** À semelhança do Provider anterior, este Provider é responsável por gerir estados relacionados com os dados do incidente a ser reportado.
- **Provider dos Dados do Utilizador:** Este Provider é responsável por gerir estados relacionados com os dados preenchidos pelo utilizador durante a criação da sua conta, como o seu *email*, nome, descrição e localização.

5.1.6 Geolocalização

A geolocalização desempenha um papel fundamental na aplicação AQUA. Um exemplo claro disso é a funcionalidade de reportar avistamentos e incidentes. Ao identificar a localização do dispositivo, os utilizadores podem fornecer informações precisas sobre a presença das espécies, fornecendo aos cientistas informação relevante sobre as áreas onde estas espécies são encontradas e ajudando-os a identificar possíveis ameaças no ecossistema. Além disso, a aplicação AQUA permite aos utilizadores fazer o *tracking* das suas rotas de viagem. Assim, é necessário registar periodicamente a localização do dispositivo durante a viagem, permitindo aos utilizadores acompanhar e visualizar o percurso da viagem.

Para a obtenção da geolocalização do dispositivo na aplicação AQUA utilizou-se a biblioteca **"geolocator"** que fornece as coordenadas geográficas precisas do dispositivo, como a latitude e longitude. Esta biblioteca suporta diferentes provedores de localização, como o GPS, rede móvel e Wi-Fi, o que permite que a aplicação consiga obter informações sobre a localização de maneira flexível, dependendo da disponibilidade do provedor disponível no dispositivo, quando for necessário.

5.1.7 Visualização de Mapas

Sendo que é fundamental apresentar informação sobre a localização dos avistamentos, incidentes e viagens num mapa interativo na aplicação AQUA, utilizou-se a biblioteca “*flutter_map*”, baseada no Leaflet⁷, para a implementação dos mapas. O Leaflet é uma biblioteca JavaScript reconhecido pela sua simplicidade, desempenho e usabilidade. Ao utilizar o “*flutter_map*”, são herdadas as características que o Leaflet apresenta. Esta biblioteca fornece recursos avançados que permitem apresentar mapas em aplicações móveis, e a personalização da sua aparência e interação.

Para personalizar o mapa e torná-lo mais semelhante ao *design* da aplicação AQUA, utilizou-se a plataforma MapBox⁸, um serviço de mapas e geolocalização que fornece diversos estilos de mapas e ferramentas de personalização. Através desta plataforma, foi possível personalizar a aparência do mapa de forma a manter uma consistência com o restante *design* da mesma.

Através da biblioteca “*flutter_map*” e do MapBox, foi possível criar mapas interativos que permitem aos utilizadores visualizar a localização dos avistamentos das espécies, identificar as áreas onde ocorrem mais incidentes e acompanhar e visualizar as suas rotas de viagem.

- **Clusters de Avistamentos:** Ao longo do tempo o número de avistamentos pode aumentar, afetando o desempenho e a visualização dos avistamentos no mapa interativo, devido à sobreposição de marcadores. Desta forma, foi proposto a implementação de *clusters* que permitissem agrupar avistamentos próximos num único marcador. À medida que o utilizador faz *zoom in*, os marcadores que fazem parte do respetivo *cluster* são apresentados.

A biblioteca “*flutter_map_marker_cluster*” é uma extensão do “*flutter_map*” que permite agrupar marcadores no mapa, melhorando o desempenho e usabilidade quando há um grande número de elementos a serem apresentados.

Ao utilizar esta biblioteca em conjunto com o “*flutter_map*”, é possível agrupar os marcadores próximos num único marcador (*cluster*), que inclui um contador que representa o número de marcadores agrupados naquela zona. Para além de melhorar o desempenho devido à redução de elementos a serem renderizados no mapa, ele garante que a visualização do mapa é mais organizada e limpa.

No desenvolvimento da aplicação AQUA, estas bibliotecas foram utilizadas para agrupar avistamentos próximos no mapa de avistamentos, facilitando a identificação de zonas mais propensas a ocorrências de espécies. Os avistamentos próximos são agrupados num único marcador, representando um *cluster*. Este marcador, por sua vez, inclui um contador que indica a quantidade de avistamentos agrupados naquela zona.

- **Área do Avistamento:** Outra proposta de melhoria na aplicação AQUA foi a criação de uma área de incerteza à volta das coordenadas de cada avistamento. Tal como já foi referido anteriormente, as coordenadas obtidas referem-se à localização do dispositivo e não precisamente à localização exata das espécies avistadas.

Desta forma, utilizou-se a fórmula de *haversine* [61] para calcular a latitude e a longitude de pontos num círculo à volta de uma determinada coordenada, com base num raio especificado em metros. Esta fórmula considera a curvatura da Terra para calcular com precisão as coordenadas

⁷<https://leafletjs.com/>

⁸<https://www.mapbox.com/>

dos pontos num círculo. Para o cálculo da latitude e longitude dos pontos no círculo, ele baseia-se na latitude e longitude do centro, no raio em metros e o número de pontos desejados no círculo.

A implementação desta fórmula permitiu criar um conjunto de coordenadas para cada ponto à volta da coordenada obtida para cada avistamento. Com base nessas coordenadas, utilizou-se os recursos disponibilizados pela biblioteca “*flutter_map*” para a criação de polígonos que posteriormente são apresentados no mapa de avistamentos. Cada polígono representa a área de incerteza da localização das espécies avistadas.

Para a representação das áreas de incerteza considerou-se um raio de 100 metros, levando em conta a consulta realizada com um biólogo marinho. Dentro de um intervalo de 50 a 100 metros, a precisão da localização não é considerada crítica, sendo aceitável para a finalidade de identificar zonas propensas a avistamentos de espécies. Para o número de pontos por círculo, considerou-se um valor de 20 pontos, com o objetivo de manter a forma do polígono o mais próxima possível de um círculo. Caso fosse utilizado um número menor de pontos, o polígono resultante apresentaria uma forma irregular.

- **Rota da Viagem:** Para a apresentação das rotas de viagem nos mapas, utilizou-se recursos disponibilizados pela mesma biblioteca, a “*flutter_map*”. A partir das coordenadas de uma rota, é possível criar uma polilinha (*Polyline*), que consiste num conjunto de múltiplos segmentos de linha ligados, formando uma sequência contínua de pontos. Cada coordenada é ligada à coordenada seguinte, e assim sucessivamente, até criar a polilinha completa. Esta polilinha é posteriormente renderizada no mapa, fornecendo uma representação visual da rota da viagem. Desta forma, os utilizadores podem visualizar o percurso percorrido ao longo de uma viagem, ou acompanhar o percurso da viagem em tempo real, caso esteja a realizá-la.

5.1.8 *Tracking* das Rotas de Viagem

Para implementar a funcionalidade do *tracking* das rotas de viagem, utilizou-se a biblioteca “**geolocator**” e a biblioteca “**background_location**”. A biblioteca “**geolocator**” é responsável por obter a geolocalização do dispositivo, enquanto a biblioteca “**background_location**” permite realizar o *tracking* em segundo plano.

Ao iniciar o tracking da rota da viagem, é utilizado o método ‘`getCurrentPosition`’ da biblioteca “**geolocator**” para obter a localização atual do dispositivo. Em seguida, é criada uma instância da classe ‘**CoordinateSQLite**’ com as coordenadas obtidas, juntamente com a data e hora do registo. Esta instância é armazenada localmente e um objeto do tipo **Marker** é adicionado a uma lista de **Markers** que representa a rota de viagem no mapa.

Também, é criada uma instância da classe ‘**TripRouteSQLite**’, responsável por armazenar informações sobre a rota da viagem. Inicialmente, é definida a data e hora de início da rota, o ID do utilizador que iniciou a rota, e é adicionada a coordenada inicial à lista de coordenadas da rota de viagem. Esta instância também é armazenada localmente.

Em seguida, utilizou-se os métodos ‘`startLocationService`’ e ‘`getLocationUpdates`’ da biblioteca “**background_location**” para iniciar o *tracking* em segundo plano. Desta forma, o utilizador não precisa manter a aplicação móvel aberta. O método ‘`startLocationService`’ é responsável por inicializar o serviço em segundo plano, enquanto o método ‘`getLocationUpdates`’ permite definir as operações a serem executadas sempre que a localização do utilizador for alterada.

Para evitar registos excessivos de coordenadas enquanto o utilizador está em constante movimento, é feita uma comparação do *timestamp* da última coordenada registada com o *timestamp* da coordenada atual. Apenas é registada uma nova coordenada se a diferença entre os *timestamps* for superior a 5 minutos, sendo criada uma instância da classe **'CoordinateSQLite'** e um objeto **'Marker'** para a nova coordenada, que são armazenados localmente.

O utilizador pode parar o tracking da rota da viagem a qualquer instante, pressionando um botão. Ao fazer isso, é chamado o método **'stopLocationService'** da biblioteca **"background_location"** para terminar o serviço em segundo plano. É criada uma instância da classe **'CoordinateSQLite'** com as informações sobre a última coordenada registada no momento de pressionar o botão, que é armazenada localmente. A informação da instância da classe **'TripRouteSQLite'** é atualizada com a lista completa de coordenadas registadas e a hora atual, que representa o fim da viagem.

De seguida, é feita a verificação da conexão à Internet. Se houver conexão, as informações sobre a rota da viagem e das suas coordenadas são enviadas para o servidor. Caso contrário, o utilizador é informado que a rota da viagem foi armazenada localmente, podendo ser submetida posteriormente, acedendo ao seu histórico de rotas de viagem.

Para evitar que o utilizador esqueça de terminar o *tracking* de uma rota, enquanto o *tracking* em segundo plano está ativo, é feita a verificação da distância entre a coordenada inicial e a coordenada atualmente registada. Se a distância for inferior a 100 metros, indicando que o utilizador regressou ao ponto de início, o *tracking* é terminado automaticamente, seguindo o mesmo procedimento que ocorre quando o utilizador pressiona o botão para terminar o *tracking*. Para calcular a distância entre as coordenadas, utilizou-se o método **'distanceBetween'** da biblioteca **"geolocator"**, que retorna a distância em metros entre duas coordenadas.

É importante destacar que, em determinadas situações, em que o local de destino difere do local de partida, não é possível terminar o *tracking* automaticamente, sendo nestes casos necessário que o utilizador o faça manualmente.

5.1.9 Verificação de Conexão à Internet

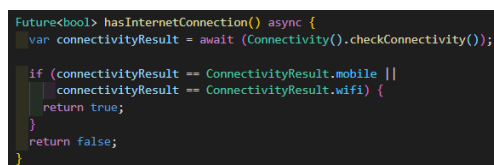
Para garantir um bom funcionamento da aplicação AQUA em ambientes onde a conexão à Internet é instável ou inexistente, é essencial verificar constantemente a disponibilidade da conexão no dispositivo do utilizador. Antes de qualquer comunicação com o servidor, é realizada uma verificação da conexão à Internet, evitando assim tentativas de comunicação que falhariam devido à falta de conexão.

Para implementar essa verificação, foi utilizada a biblioteca **'connectivity_plus'** do Flutter. Esta biblioteca permite que as aplicações desenvolvidas em Flutter verifiquem a disponibilidade da conexão à Internet, seja através de dados móveis ou de conexão Wi-Fi.

Na figura 37 é possível observar a implementação do método **hasInternetConnection()**. Este método é responsável por verificar a conexão à Internet. A variável **'connectivityResult'** recebe como resultado a verificação de conexão realizada pela biblioteca **'connectivity_plus'**. A função **'checkConnectivity()'** da classe **'Connectivity'** retorna um objeto do tipo **'ConnectivityResult'**, que indica o tipo de conexão que está a ser utilizado no dispositivo, podendo ser dados móveis (**mobile**), Wi-Fi (**wifi**), ou nenhum (**none**).

De seguida, é feita uma verificação para determinar se o resultado da conexão é **mobile** ou **wifi**, ou seja, se o dispositivo está conectado à Internet através dos dados móveis ou Wi-Fi, respetivamente. Caso seja a condição seja verificada, o método retorna **'true'**, indicando que há conexão à Internet. Caso contrário, o método retorna **'false'**.

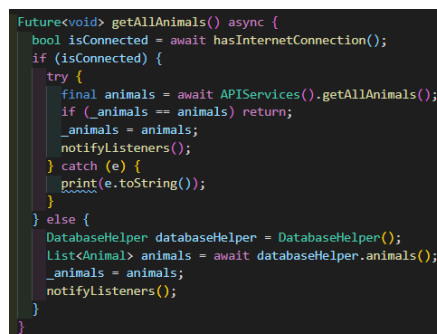
Na figura 38 é apresentado um exemplo da utilização do método **'hasInternetConnection()'** na tentativa de obter os dados de todos os animais do sistema. Se o resultado do método for **'true'**, indicando que há conexão à Internet disponível, a aplicação faz uma chamada ao serviço responsável pela comunicação com o servidor para obter a informação sobre os animais na base de dados *online*.



```
Future<bool> hasInternetConnection() async {
  var connectivityResult = await (Connectivity().checkConnectivity());

  if (connectivityResult == ConnectivityResult.mobile ||
      connectivityResult == ConnectivityResult.wifi) {
    return true;
  }
  return false;
}
```

Figura 37. Definição do método **'hasInternetConnection()'**



```
Future<void> getAllAnimals() async {
  bool isConnected = await hasInternetConnection();
  if (isConnected) {
    try {
      final animals = await APIServices().getAllAnimals();
      if (_animals == animals) return;
      _animals = animals;
      notifyListeners();
    } catch (e) {
      print(e.toString());
    }
  } else {
    DatabaseHelper databaseHelper = DatabaseHelper();
    List<Animal> animals = await databaseHelper.animals();
    _animals = animals;
    notifyListeners();
  }
}
```

Figura 38. Utilização do método **'hasInternetConnection()'** no método **'getAllAnimals()'** que permite obter os dados dos animais do sistema.

No caso de não haver conexão à Internet, o sistema acede diretamente aos dados dos animais armazenados localmente na base de dados SQLite. Para além de evitar uma tentativa de comunicação falhada com o servidor, é garantido que o sistema irá apresentar a informação ao utilizador.

Embora possam haver diferenças nos dados locais com os dados disponíveis na base de dados *online*, é garantido que o sistema está sempre disponível, mesmo quando não há conexão à Internet, permitindo que o utilizador visualize as informações a qualquer momento sem afetar a experiência de utilização da aplicação. Esta abordagem é aplicada em todas as interações que requerem conexão à Internet na aplicação AQUA, incluindo todas as comunicações com o servidor.

5.1.10 Persistência de Dados Simples

No desenvolvimento da aplicação AQUA utilizou-se a biblioteca **'shared_preferences'** do Flutter para o armazenamento de dados simples. Esta biblioteca é muito utilizada e permite armazenar e aceder a dados persistentes de forma rápida e fácil. Através do **'shared_preferences'**, é possível armazenar dados que são mantidos mesmo após fechar a aplicação móvel.

Além de ser simples de utilizar, devido à facilidade de armazenar e recuperar dados utilizando os métodos fornecidos, não é necessário preocupar-se com a forma como o armazenamento é feito. Por exemplo, na aplicação AQUA alguns dados armazenados utilizando esta biblioteca são o ID do utilizador com *login* efetuado, a lista de distintivos novos, e a lista de cartas das espécies novas.

Para garantir a persistência do utilizador com *login* efetuado na aplicação, armazena-se de forma persistente o ID do utilizador no momento em que o *login* é realizado com sucesso. Desta

forma, mesmo quando a aplicação móvel é fechada, é possível identificar o utilizador e manter a sessão ativa. Se o utilizador eliminar a sua conta ou efetuar *logout*, o ID armazenado é removido, garantindo que o sistema não mantém referência ao último utilizador com *login* efetuado.

Quando o utilizador desbloqueia um novo distintivo ou uma nova carta de uma espécie, o ID do distintivo ou da espécie é adicionado à respetiva lista que contém todos os IDs da recompensa em questão. Ao visualizar uma recompensa, o ID correspondente é removido da lista. Assim, é possível indicar quais recompensas são novas, apresentando um “New” sobre a recompensa. Esta indicação tem como objetivo incentivar o utilizador a visualizar as suas novas recompensas com maior detalhe.

5.1.11 Armazenamento de Imagens em Cache

De forma a garantir que os utilizadores têm acesso às imagens apresentadas na aplicação AQUA mesmo quando estão sem conexão à Internet, utilizou-se a biblioteca `'cached_network_image'` do Flutter. Esta biblioteca permite o armazenamento em cache de imagens que são carregadas pela primeira vez quando o dispositivo está conectado à Internet.

Quando o utilizador tem conexão disponível e a imagem é carregada pela primeira vez, a biblioteca `'cached_network_image'` é responsável por armazenar em cache a imagem obtida do servidor. Isto significa que a imagem é armazenada localmente no dispositivo, estando disponível mesmo sem conexão à Internet. Desta forma, quando o utilizador perde a conexão, a aplicação consegue aceder à imagem armazenada em cache, sem a necessidade de realizar um novo pedido ao servidor, que resultaria numa falha na apresentação da imagem.

No entanto, se a imagem ainda não estiver em cache e o utilizador estiver sem conexão à Internet, a imagem não estará disponível. Nestas situações, a aplicação apresentará um ícone da aplicação AQUA em vez da imagem, evitando assim a apresentação de erros ou espaços em branco. Esta abordagem tem como objetivo garantir que a experiência dos utilizadores é consistente e agradável, mesmo em modo *offline*.

Na figura 39 é apresentado um exemplo de uma situação em que o utilizador acedeu ao *ranking* de utilizadores, sem conexão à Internet. Nesta situação, a imagem do utilizador “Mara” foi carregada com sucesso, pois estava armazenada em cache, sendo apresentada sem problemas. No entanto, as imagens dos restantes utilizadores, “Ana” e “Marc”, não estavam em cache, ou seja, não estavam disponíveis localmente no dispositivo, pois nunca foram acedidas anteriormente. Devido à falta de conexão à Internet, a aplicação apresentou o ícone da aplicação AQUA no lugar de cada imagem dos utilizadores cuja imagem estava indisponível.



Figura 39. Apresentação de imagem de perfil em cache no *ranking*, sem conexão à Internet.

5.1.12 *Text-To-Speech*

Uma das funcionalidades desenvolvidas na aplicação AQUA é a capacidade de permitir aos utilizadores ouvir as informações sobre as espécies através de áudio. Esta funcionalidade foi desenvolvida para resolver um problema identificado no estudo do AQUA 1.0, onde vários participantes relataram sentir enjoos ao ler as informações sobre as espécies durante a viagem marítima. Embora não seja algo comum entre os utilizadores, considerou-se importante permitir que os utilizadores pudessem ouvir as informações das espécies, para desfrutarem da viagem enquanto ouvem a informação, sem necessidade de visualizar a interface da aplicação.

Para implementar esta funcionalidade, utilizou-se a biblioteca `'flutter_tts'` do Flutter, que permite converter texto em áudio. O sistema recebe um texto como *input*, e utiliza a biblioteca `'flutter_tts'` para processá-lo e gerar um ficheiro de áudio com a informação recebida. De seguida, o áudio pode ser reproduzido e ouvido pelos utilizadores. Esta biblioteca fornece várias opções de configuração, como a velocidade de reprodução, o volume do áudio, o idioma e diversas opções de voz.

Na figura 40 é possível observar a inicialização de uma instância da classe `'FlutterTts'`, responsável por realizar a conversão do texto em áudio. Esta instância é utilizada para chamar todos os métodos responsáveis pela configuração, reproduzir e parar o áudio. O método `'textToSpeech(String text)'`, é o método responsável por reproduzir um texto recebido como parâmetro. Ele recebe o texto como *input* e utiliza a instância da classe `'FlutterTts'` para reproduzir o áudio com base no texto recebido, usando o método `'speak()'` definido na classe `'FlutterTts'`.

```
final FlutterTts flutterTts = FlutterTts();

Future textToSpeech(String text) async {
  await flutterTts.speak(text);
}

Future stopTts() async {
  await flutterTts.stop();
}
```

Figura 40. Inicialização da instância da classe `'FlutterTts'` e definição dos métodos `'textToSpeech(String text)'` e `'stopTts()'`

De forma a permitir que o áudio reproduzido possa ser interrompido a qualquer momento pelo utilizador, o método `'stopTts()'` chama o método `'stop()'` da instância `'flutterTts'`, que interrompe a reprodução do áudio que estiver a ser reproduzido, seja ele qual for.

Para evitar a reprodução de vários áudios em simultâneo, criou-se uma única instância da classe `'FlutterTts'`. Desta forma, é sempre utilizada a mesma instância, evitando a criação de múltiplas instância para cada informação de cada espécie e evitando conflitos na reprodução dos áudios. Caso o sistema esteja a reproduzir um áudio, e o utilizador clique em reproduzir outro, o áudio anterior é interrompido e o novo áudio é iniciado.

5.1.13 *Armazenamento de Dados Local*

A aplicação AQUA é uma aplicação móvel desenvolvida para funcionar principalmente durante viagens marítimas, onde a conexão à Internet é instável ou inexistente. Nestes cenários, a comunicação com o servidor para a obtenção de informação da base de dados *online* não estaria disponível,

impossibilitando que os utilizadores pudessem utilizar a aplicação móvel, devido à falta de informação que seria apresentada. Para resolver este problema, foi implementado um armazenamento de dados local, conhecido como base de dados *offline*, utilizando o SQLite. Sendo a aplicação AQUA desenvolvida utilizando a *framework* Flutter, utilizou-se a biblioteca '**sqflite**'. Esta biblioteca permite que as operações nos dados sejam feitas de forma mais rápida, comparada a outras soluções, como o '**shared_preferences**'.

– Cópia de Dados do Servidor Localmente:

Ao abrir a aplicação móvel, após o carregamento de informação do servidor *online*, é feita uma cópia dos dados da base de dados *online* na base de dados *offline*, quando o utilizador tem ligação à Internet. Isto garante que os dados sobre os animais, espécies, utilizadores, rotas de viagem, recompensas, avistamentos e incidentes estão sempre disponíveis na aplicação móvel. Quando o utilizador acede à aplicação sem conexão à Internet, a mesma acede aos dados armazenados localmente.

Como ambas as bases de dados são relacionais SQL, não houve muitos problemas na implementação da base de dados *offline*, visto que as tabelas seriam semelhantes. No entanto, apesar das semelhanças, o SQLite é uma base de dados *light-weight* e tem várias limitações em relação aos tipos de dados que permite armazenar. Por exemplo, não é possível armazenar dados do tipo '**DateTime**', '**Boolean**' e '**List**'. Estes tipos de dados são muito utilizados nos modelos da base de dados *online*, o que dificultou a cópia dos dados entre as bases de dados.

Para realizar a cópia dos dados, foi necessário desenvolver modelos distintos para trabalhar com os modelos do SQLite. Por exemplo, no modelo '**TripRoute**', foi necessário desenvolver um modelo semelhante chamado '**TripRouteSQLite**', mas com tipos de dados compatíveis com o SQLite, visto que o modelo '**TripRoute**' possui como propriedades uma lista de inteiros com os IDs dos objetos das coordenadas pertencentes à rota e dois **DateTime** que representam a data e a hora em que a rota foi iniciada e concluída.

Apesar de ter sido necessário desenvolver modelos distintos, alguns modelos originais não apresentavam tipos de dados incompatíveis com o SQLite, o que evitou a necessidade de criar modelos complementares para efetuar a cópia dos dados.

No momento em que a aplicação é iniciada e há conexão à Internet, é feita a comunicação com o servidor para obter os dados armazenados *online*. Ao chamar um determinado serviço, a aplicação móvel recebe um ficheiro **JSON** que contém os dados correspondentes, e que posteriormente é decodificado com base nos modelos desenvolvidos, utilizando o método '**fromJson**' de cada modelo.

Devido à incompatibilidade dos tipos de dados '**List**' e '**DateTime**' com o SQLite, foi necessário transformar esses dados no formato '**String**'. Nas figuras 41 e 42, pode observa-se a implementação dos modelos '**User**' e '**UserSQLite**', respetivamente, e onde se pode observar as diferenças nos tipos de dados das propriedades de cada modelo.

```

class User with ChangeNotifier {
  final int? id;
  final String name;
  final String email;
  final String role;
  final String description;
  final String location;
  final String imageUrl;
  final List<int> tripRoutes;
  final List<int> unlockedSpecies;
  final List<int> unlockedBadges;
  final bool active;
  final int? badgeTitleId;
}

```

Figura 41. Implementação do modelo 'User' - Propriedades

```

class UserSQLite {
  final int? id;
  final String name;
  final String email;
  final String role;
  final String description;
  final String location;
  final String imageUrl;
  final String tripRoutes;
  final String unlockedSpecies;
  final String unlockedBadges;
  final int active;
  final int? badgeTitleId;
}

```

Figura 42. Implementação do modelo 'UserSQLite' - Propriedades

Ao chamar o serviço para obter os dados dos utilizadores, o ficheiro **JSON** recebido como resposta é decodificado utilizando o método 'fromJson' do modelo 'User' (ver figura 43). Esse método permite fazer o mapeamento dos dados do ficheiro **JSON** e criar um objeto do tipo 'User' com as propriedades recebidas (ver figura 44). Após a decodificação do ficheiro **JSON** e a criação dos objetos do modelo em questão, é feita a cópia dos dados para a base de dados SQLite.

```

Future<List<User>> getAllUsers() async {
  try {
    final response = await http.get(Uri.parse("${_baseUrl}/$usersEndpoint"));
    if (response.statusCode == 200) {
      final List<dynamic> decodedJson = json.decode(response.body);
      return decodedJson
        .map((usersJson) => User.fromJson(usersJson))
        .toList();
    } else {
      throw Exception('Failed to load users');
    }
  } on TimeoutException catch (e) {
    throw Exception('Failed to load users');
  } catch (error) {
    rethrow;
  }
}

```

Figura 43. Implementação do método 'getAllUsers()' para obtenção dos dados de todos os utilizadores

```

factory User.fromJson(Map<String, dynamic> json) {
  final tripRoutes =
    (json['tripRoutes'] as List<dynamic>).map((e) => e as int).toList();
  final unlockedSpecies = (json['unlockedSpecies'] as List<dynamic>)
    .map((e) => e as int)
    .toList();
  final unlockedBadges =
    (json['unlockedBadges'] as List<dynamic>).map((e) => e as int).toList();

  return User(
    id: json['id'],
    name: json['name'],
    email: json['email'],
    role: json['role'],
    description: json['description'],
    location: json['location'],
    imageUrl: json['imageUrl'],
    tripRoutes: tripRoutes,
    unlockedSpecies: unlockedSpecies,
    unlockedBadges: unlockedBadges,
    badgeTitleId: json['badgeTitleId'],
    active: json['active']);
}

```

Figura 44. Implementação do método 'User.fromJson()' responsável por mapear os dados de um ficheiro JSON num objeto da classe 'User'

Para efetuar a cópia, é feita a transformação do modelo criado no respetivo modelo compatível com o SQLite. Na figura 45 é possível observar que para a inserção dos dados de um objeto do modelo 'User', é criado um objeto do modelo 'UserSQLite' utilizando o método 'fromUser' definido na classe 'UserSQLite'. Este método recebe como parâmetro o objeto 'User', transforma os seus dados incompatíveis, e retorna um objeto do modelo 'UserSQLite' idêntico mas com os tipos de dados compatíveis com o SQLite.

Como se pode observar na figura 46, as propriedades cujo tipo de dado era 'List' foram transformadas numa 'String' através do método 'join()', e a propriedade 'active' do tipo 'bool' foi transformada num inteiro, sendo '1' ou '0', no caso de ser 'true' ou 'false', respetivamente.

Após as transformações realizadas, o objeto compatível é inserido na base de dados *offline*, a partir da operação 'insert' disponibilizada pela biblioteca 'sqlite'.

```
Future<void> insertUser(User user) async {
  var dbClient = await db;

  UserSQLite userSQLite = UserSQLite.fromUser(user);
  await dbClient.insert(
    'user',
    userSQLite.toJson(),
    conflictAlgorithm: ConflictAlgorithm.replace,
  );
}
```

Figura 45. Implementação do método 'insertUser()'

```
factory UserSQLite.fromUser(User user) => UserSQLite(
  id: user.id,
  name: user.name,
  email: user.email,
  role: user.role,
  description: user.description,
  location: user.location,
  imageUrl: user.imageUrl,
  tripRoutes: user.tripRoutes.join(','),
  unlockedSpecies: user.unlockedSpecies.join(','),
  unlockedBadges: user.unlockedBadges.join(','),
  active: user.active ? 1 : 0,
  badgeTitleId: user.badgeTitleId,
);
```

Figura 46. Implementação do método 'UserSQLite.fromUser()' responsável por transformar um objeto da classe 'User' num objeto da classe 'UserSQLite'

Na figura 47 é ilustrado o processo de transformações efetuadas para copiar os dados dos utilizadores da base de dados *online* (PostgreSQL) para a base de dados *offline* (SQLite), descrito anteriormente.

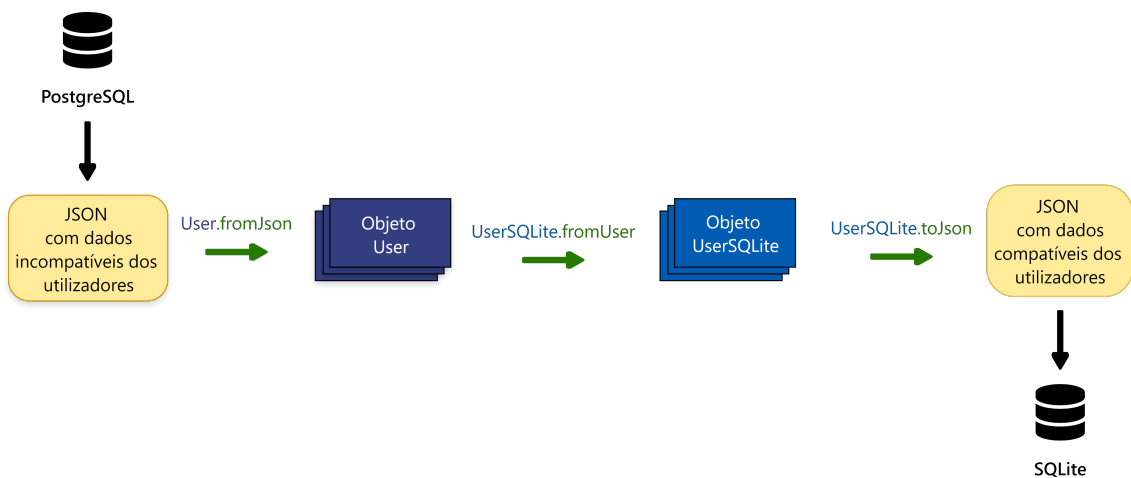


Figura 47. Exemplo do processo da cópia dos dados dos utilizadores do PostgreSQL para SQLite

– Armazenamento de Reportes *Offline*:

Durante uma viagem marítima, os utilizadores da aplicação AQUA podem reportar avistamentos de espécies marinhas. No entanto, a falta de conexão à Internet no momento de submissão de um avistamento pode levar a uma má experiência de utilização, visto que a mensagem de erro impediria a submissão da informação. Embora o utilizador pudesse optar por aguardar até que a conexão estivesse disponível para submeter o avistamento, não seria possível reportar novos avistamentos ou utilizar qualquer funcionalidade da aplicação até que o avistamento anterior fosse submetido.

Para evitar esta situação, a aplicação AQUA possui um mecanismo que permite armazenar localmente os avistamentos reportados quando não há conexão à Internet. Esses avistamentos são armazenados na base de dados *offline* (SQLite) no momento em que o utilizador tenta submetê-los, mas não há conexão disponível. Desta forma, mesmo sem conexão, os utilizadores podem continuar a reportar avistamentos e ter a garantia de que as informações são guardadas localmente, evitando a perda de informação sobre os avistamentos e permitindo que os utilizadores possam reportar múltiplos avistamentos mesmo em situações em que a conexão à Internet é limitada.

Da mesma forma que é feita a cópia dos dados do servidor para a base de dados localmente, o mesmo é aplicado para os avistamentos e incidentes que são armazenados localmente. Para garantir a compatibilidade dos dados com o SQLite, foi necessário criar dois modelos complementares, o '**ReportSQLite**' e o '**SightedSpecieInfoSQLite**'.

Quando o utilizador decide submeter um avistamento ou incidente, é criado um objeto da classe '**Report**' para armazenar todas as informações, como data, hora e coordenadas. Além disso, são criados objetos da classe '**SightedSpecieInformation**' para cada espécie avistada, contendo os detalhes sobre a espécie, como o número de indivíduos e comportamentos avistados.

Se houver conexão à Internet disponível, todas as informações do avistamento, incluindo as espécies avistadas, são enviadas ao servidor para a submissão dos dados. No entanto, se não houver conexão, o objeto '**Report**' é convertido num objeto da classe '**ReportSQLite**' utilizando o método '**TripRouteSQLite.fromTripRoute()**'. Este método permite converter os dados do avistamento num formato compatível com o SQLite, possibilitando o armazenamento dos dados localmente.

Uma vez que o objeto '**ReportSQLite**' é criado, o método '**TripRouteSQLite.toJson()**' é utilizado para colocar os dados num ficheiro JSON, que é posteriormente armazenado na base de dados local usando a operação '**insert**' fornecida pela biblioteca '**sqflite**'. Da mesma forma, todos os objetos da classe '**SightedSpecieInformation**' são transformados em objetos da classe '**SightedSpecieInfoSQLite**' para garantir a compatibilidade dos dados e o armazenamento local até a submissão dos avistamentos e incidentes.

Os utilizadores têm acesso aos seus históricos de reportes, onde todos os avistamentos e incidentes reportados são listados, através da aplicação AQUA. Este histórico inclui aqueles que ainda não foram submetidos e que estão armazenados localmente. Quando a conexão à Internet estiver disponível, o utilizador pode pressionar um botão para submeter todos os reportes armazenados localmente na base de dados *online*. Ao submeter um reporte, ele é removido do armazenamento local automaticamente, usando a operação '**delete**' da biblioteca '**sqflite**'.

– Armazenamento de Rotas de Viagem *Offline*:

Assim como os reportes, as rotas de viagem registadas também são armazenadas localmente até serem submetidas na base de dados *online*. No entanto, há uma diferença no armazenamento das coordenadas das rotas. Enquanto o *tracking* da viagem está a ser feito, as coordenadas são registadas periodicamente a cada 5 minutos e armazenadas localmente.

Ao armazenar todas as coordenadas localmente durante a viagem, é garantido que os dados são preservados, sendo, apenas no momento da submissão da rota, feita a transformação dos

dados armazenados localmente. Assim, todas as coordenadas são enviadas para o servidor ao mesmo tempo.

Quando o utilizador decide parar o *tracking* da rota da viagem, é verificado se há conexão disponível. Se houver, a última coordenada é registada e a informação completa da rota é submetida na base de dados *online*. No entanto, se não houver conexão, a última coordenada é armazenada localmente, para submissão futura.

5.2 Desenvolvimento da API

Nesta secção são apresentados detalhes sobre a implementação da API RESTful utilizando a *framework* ASP.NET Core. Serão apresentados diversos conceitos e práticas utilizadas na criação da API, incluindo a utilização de recursos oferecidos pela *framework*.

Sendo a aplicação AQUA uma aplicação de ciência cidadã, é de enorme importância garantir que é possível partilhar os dados dos avistamentos reportados na aplicação com outros sistemas externos. Desta forma, a API desenvolvida serve como uma ponte de ligação para possibilitar a partilha dos dados com outros sistemas, promovendo o acesso à informação que pode ser muito valiosa para a ciência e para a conservação do ecossistema marinho.

Além disso, ao desenvolver uma API própria é possível ter controlo total sobre as funcionalidades e configurações, permitindo adaptá-la às necessidades do sistema AQUA. E, ainda, é possível criar mecanismos para controlar o acesso aos dados, para evitar os acessos ou manipulações indevidas nos dados.

Devido às restrições de número de páginas deste documento, uma descrição mais detalhada do desenvolvimento da API do sistema pode ser encontrada no Apêndice D.

5.2.1 Arquitetura MVC

Ao optar pelo desenvolvimento da API utilizando a *framework* ASP.NET Core, foi possível implementar uma arquitetura MVC (*Model-View-Controller*) de forma nativa. Esta arquitetura é composta por três componentes distintos: os modelos (*models*), as vistas (*views*) e os controladores (*controllers*). Os controladores são responsáveis por receber os pedidos HTTP dos clientes, processar a lógica de negócio e retornar as respostas apropriadas. Por outro lado, os modelos representam as entidades e a lógica de dados da aplicação. No caso do ASP.NET Core, as vistas geralmente são implementadas utilizando tecnologias de *front-end* e são processadas no lado do cliente, como é o caso da aplicação móvel AQUA.

Na figura 48 é ilustrada a arquitetura MVC implementada, assim como a interação dos principais componentes: o **Cliente** (Aplicação móvel AQUA) e a **API**. No **Cliente**, as **Vistas** são implementadas como componentes responsáveis pela apresentação visual dos dados e pela interação com o utilizador. Estas **Vistas** incluem toda a interface do utilizador. Na **API**, são implementados os **Controladores** e estes atuam como intermediários entre os pedidos do **Cliente** e a manipulação dos dados nos **Modelos**. Os **Modelos** também são implementados na **API** e incluem todas as entidades do sistema, como utilizadores, espécies, animais, rotas de viagem, reportes e recompensas.

De forma resumida, o **Cliente** interage com os **Controladores** da **API** através de pedidos HTTP. Os **Controladores** processam a lógica de negócio e utilizam os **Modelos** para aceder e manipular os dados na base de dados. Os **Modelos** interagem diretamente com a base de dados

para realizar operações nos dados. De seguida, os **Controladores** retornam a resposta para as **Vistas**, que são responsáveis por atualizar as informações apresentadas ao **Cliente**.

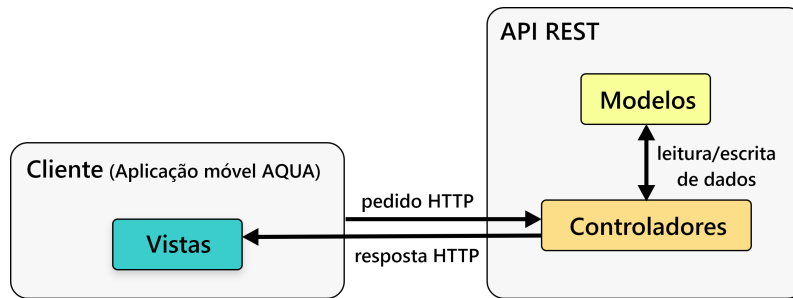


Figura 48. Arquitetura MVC - Interação entre os modelos, os controladores e as vistas

Para o desenvolvimento da API foram desenvolvidos vários modelos e controladores. No Apêndice D.1 encontra-se a lista completa dos componentes desenvolvidos, assim como os detalhes sobre cada um.

5.2.2 Endpoints

Nesta subsecção são apresentados os *endpoints* desenvolvidos na API do sistema AQUA. Os *endpoints* representam os pontos de entrada da API e permitem que os clientes interajam com o sistema e realizem as operações pretendidas.

No total foram desenvolvidos 18 *endpoints* para atender às diversas funcionalidades do sistema. A seguir, é apresentada a lista dos principais *endpoints* desenvolvidos, sendo no Apêndice D.3 apresentada a lista completa de *endpoints* desenvolvidos e uma descrição detalhada de cada um individualmente.

- **Endpoint “api/Users”**, para a gestão dos dados de todos os utilizadores registados no sistema;
- **Endpoint “api/Users/Login”**, para autenticar um utilizador no sistema;
- **Endpoint “api/Animals”**, para a gestão dos dados dos animais do sistema;
- **Endpoint “api/Species”**, para a gestão dos dados de todas as espécies do sistema;
- **Endpoint “api/Badges”**, para a gestão dos dados de todos os distintivos do sistema;
- **Endpoint “api/Reports”**, para a gestão dos dados de todos os reportes (avistamentos e incidentes) do sistema;
- **Endpoint “api/SightedSpeciesInformation”**, para a gestão dos dados de todas as espécies associadas a algum report do sistema;
- **Endpoint “api/Coordinates”**, para a gestão dos dados de todas as coordenadas das rotas de viagem do sistema;
- **Endpoint “api/UsersEarnedBadges”**, para a gestão dos dados dos distintivos desbloqueados pelos utilizadores do sistema;
- **Endpoint “api/UsersEarnedSpecies”**, para a gestão dos dados das cartas de espécies desbloqueadas pelos utilizadores do sistema.

5.2.3 Tratamento de Erros

No desenvolvimento da API foram utilizados os métodos de conveniência fornecidos pelo ASP.NET Core para retornar as respostas HTTP comuns. Estes métodos facilitam a forma como é feito o tratamento de erros, permitindo retornar uma resposta adequada em diferentes cenários [62]. Alguns dos métodos mais utilizados foram o método '**Ok**', '**BadRequest**' e '**NotFound**'.

O método '**Ok**' retorna o *status* HTTP 200 (Ok), e indica que o pedido foi processado com sucesso. Já o método '**BadRequest**' permite retornar uma resposta com o *status* 400 (Bad Request), sendo utilizado quando ocorrem erros de validação ou quando o pedido não corresponde aos critérios esperados. Por outro lado, o método '**NotFound**' permite retornar o *status* 404 (NotFound) indicando que o recurso solicitado não foi encontrado.

Além disso, estes métodos de conveniência permitem enviar mensagens mais detalhadas ao cliente, garantindo uma melhor compreensão do erro ocorrido. Por exemplo, na figura 132 é possível observar que, ao receber um pedido para obter uma coordenada com um determinado ID, se a mesma não for encontrada, é utilizado o método '**NotFound**' para reportar o *status* 404 e é enviada uma mensagem personalizada informando que a coordenada com o ID especificado não foi encontrada. Desta forma, as respostas de erro da API desenvolvida fornecem informações úteis e específicas que ajudam o cliente a compreender o motivo do erro e a tomar as ações necessárias para o resolver.

5.2.4 Testes

Para garantir a qualidade e o correto funcionamento da API, foram realizados testes nos *endpoints* desenvolvidos utilizando o *Postman*⁹. O *Postman* é uma ferramenta que permite construir e testar APIs de forma eficiente.

Cada *endpoint* desenvolvido passou por testes manuais, onde cada funcionalidade da API foi verificada de forma isolada. Para isso, foram enviados pedidos HTTP para cada *endpoint*, utilizando os métodos GET, POST, PUT ou DELETE, e incluindo os dados necessários no corpo da mensagem, quando aplicável. Após o envio dos pedidos, foi analisada a resposta da API, tendo em conta o *status* HTTP retornado, assim como qualquer mensagem associada a ele.

Por exemplo, na figura 49 é possível observar um teste realizado utilizando o Postman. Neste teste foi feito um pedido GET ao *endpoint* '*api/Animals*'. Na figura é possível visualizar o sucesso da operação, onde a resposta obtida foi um *status* "200 OK", incluindo no corpo da resposta ("Body") a informação sobre os animais do sistema.

Estes testes tiveram como objetivo avaliar o correto funcionamento da API em diferentes situações. Além de testar as respostas quando os dados inseridos eram válidos, também foram realizados testes com dados inválidos, com o intuito de avaliar o comportamento da API nesses casos, onde as respostas devem ser mensagens de erro apropriadas.

⁹<https://www.postman.com/>

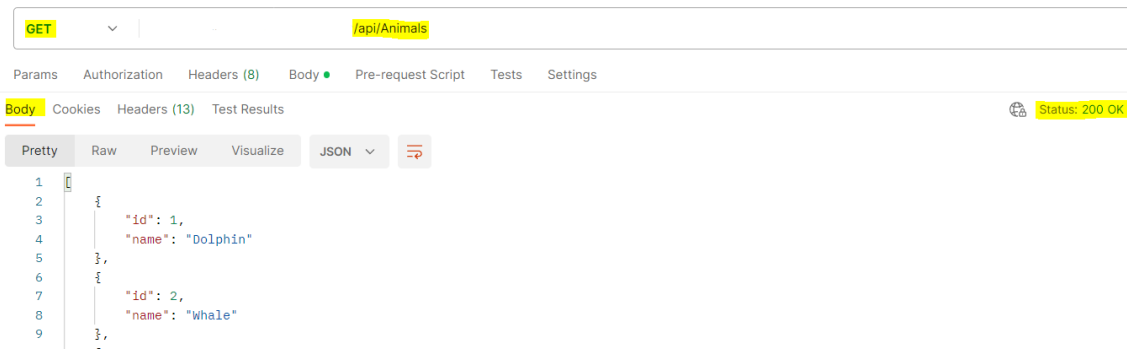


Figura 49. Teste do método GET ao *endpoint* 'api/Animals' utilizando o *Postman*

Além disso, permitiram identificar *endpoints* mal implementados, principalmente no tratamentos de dados inválidos. Ao identificar estes problemas, foram realizadas as correções necessárias em cada *endpoint*. De seguida, os testes foram executados novamente para confirmar a correção do problema e avaliar o funcionamento da API nessas situações.

5.2.5 Documentação

A documentação da API tem um papel fundamental pelo facto de fornecer uma visão clara do funcionamento da API. Para documentar a API desenvolvida, utilizou-se o *Swagger*¹⁰. O *Swagger* é uma ferramenta que permite documentar APIs de forma automática, e que oferece uma interface interativa que permite visualizar todos os *endpoints* desenvolvidos, assim como os parâmetros, respostas e outras informações relevantes da API.

No ASP.NET Core, a integração com o *Swagger* foi facilitada graças aos pacotes *NuGet* disponíveis. A própria framework oferece suporte nativo para o *Swagger*, o que permitiu tornar a criação e atualização da documentação da API uma tarefa simples e rápida.

Tal como já foi referido, o *Swagger* consegue gerar a documentação automaticamente, e, desta forma, não é necessário criar e manter a documentação manualmente. Isto permite que qualquer consumidor da API consiga integrá-la facilmente, além de facilitar a manutenção contínua do sistema.

Na figura 50 pode-se observar uma parte da interface do *Swagger*, onde são listados os cinco *endpoints* desenvolvidos para a coleção de animais.

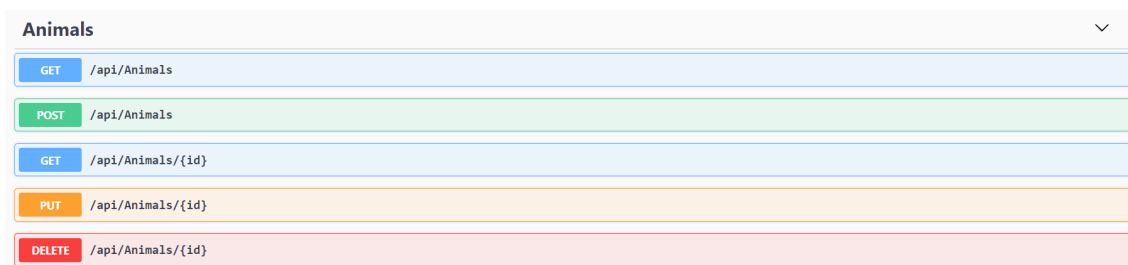


Figura 50. Lista de *endpoints* desenvolvido para a coleção de animais, no *Swagger*

¹⁰<https://swagger.io/>

Ao clicar num *endpoint*, como mostrado na figura 51, é possível visualizar detalhes sobre o mesmo, incluindo os parâmetros necessários e as respostas possíveis, assim como os códigos de *status* HTTP correspondentes.

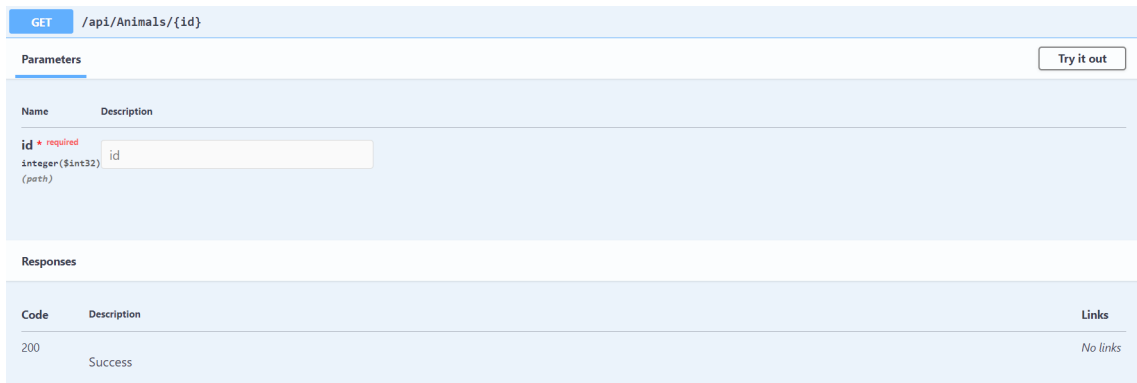


Figura 51. Detalhes do método GET do *endpoint* `'api/Animals/{id}'` no *Swagger*

Tal como apresentado para a coleção de animais, o *Swagger* gerou a documentação completa dos restantes *endpoints* desenvolvidos na API, fornecendo detalhes sobre cada um deles.

5.2.6 Data Transfer Objects

No desenvolvimento da API, utilizou-se *Data Transfer Objects* (DTOs) para separar as informações sensíveis dos dados que são transferidos entre o cliente e o servidor.

No caso do sistema AQUA, a classe `'User'` contém todos as propriedades relacionadas com o utilizador, incluindo informações sensíveis como o *password hash* e o *password salt* (ver figura 52). Por outro lado, a classe `'UserDto'` possui apenas os campos não considerados sensíveis e que são relevantes para serem transferidos e apresentados ao cliente (ver figura 53).

```
[Table("user")]
4 referências
public class User
{
    [Key]
    1 referências
    public int Id { get; set; }
    [Required, StringLength(255, ErrorMessage = "Name can't be longer than 255 characters.")]
    3 referências
    public string Name { get; set; }
    [Required, StringLength(255, ErrorMessage = "Email can't be longer than 255 characters."), EmailAddress(255)]
    9 referências
    public string Email { get; set; }
    [Required]
    2 referências
    public byte[] PasswordHash { get; set; } = new byte[0];
    [Required]
    2 referências
    public byte[] PasswordSalt { get; set; } = new byte[0];
    4 referências
    public bool Active { get; set; }
    [Required, StringLength(255, ErrorMessage = "Role can't be longer than 255 characters.")]
    3 referências
    public string Role { get; set; }
    [StringLength(255, ErrorMessage = "Location can't be longer than 255 characters.")]
    2 referências
    public string Location { get; set; }
```

Figura 52. Implementação da classe `'User'`

```
3 referências
public class UserDto
{
    2 referências
    public int Id { get; set; }
    3 referências
    public string Name { get; set; }
    5 referências
    public string Email { get; set; }
    2 referências
    public string Role { get; set; }
    2 referências
    public bool Active { get; set; }
    3 referências
    public string Location { get; set; }
    3 referências
    public string Description { get; set; }
    3 referências
    public string ImageUrl { get; set; }
    3 referências
    public List<int> TripRoutes { get; set; } = new List<int> { };
    4 referências
    public List<int> UnlockedSpecies { get; set; } = new List<int> { };
    4 referências
    public List<int> UnlockedBadges { get; set; } = new List<int> { };
    3 referências
    public int? BadgeTitleId { get; set; }
```

Figura 53. Implementação da classe `'UserDto'`

A utilização de DTOs garante que as informações sensíveis do utilizador não são expostas. Isto permite aumentar a segurança da aplicação, pois evita que dados confidenciais sejam enviados pela

rede. Além disso, também permite controlar o tipo de informação que é compartilhado com o cliente, garantindo a privacidade de cada utilizador.

Quando a API recebe um pedido de um cliente, em vez de utilizar a classe 'User', a classe 'UserDto' é utilizada para retornar as informações necessárias para o cliente, sem incluir dados sensíveis, como é possível observar na implementação do *endpoint* para obter a informação de todos os utilizadores do sistema, através do método HTTP GET (ver figura 54).

```
// GET: api/Users
[HttpGet]
[Authorize]
0 referências
public IActionResult Get()
{
    var users = _apiDbContext.Users
        .Select(u => new UserDto
        {
            Id = u.Id,
            Name = u.Name,
            Email = u.Email,
            Role = u.Role,
            Active = u.Active,
            Location = u.Location,
            Description = u.Description,
            ImageUrl = u.ImageUrl,
            TripRoutes = u.TripRoutes,
            UnlockedSpecies = u.UnlockedSpecies,
            UnlockedBadges = u.UnlockedBadges,
            BadgeTitleId = u.BadgeTitleId,
            PrivateProfile = u.PrivateProfile
        });
    return Ok(users);
}
```

Figura 54. Implementação do método GET do endpoint 'api/Users

Por exemplo, na implementação do método GET é feito o acesso à base de dados para obter os utilizadores ('Users'). A partir deste conjunto de utilizadores, é aplicado um 'Select()' que permite transformar os objetos 'User' obtidos a partir da base de dados para objetos 'UserDto', especificando quais propriedades devem ser incluídas. Como se pode observar na figura, apenas as propriedades não sensíveis são incluídas. Desta forma, ao aceder a este *endpoint* através de um pedido GET, será retornado um 'JSON' com os utilizadores no formato 'UserDto', evitando a exposição de dados confidenciais dos utilizadores.

5.2.7 Encriptação de Passwords

Qualquer sistema que envolve autenticação de utilizadores deve garantir a segurança das *passwords* dos utilizadores. No sistema AQUA, são utilizadas técnicas de encriptação que permitem proteger as *passwords* dos seus utilizadores de forma segura, como o *password hash* e o *password salt*.

O *password hash* é uma transformação que é aplicada à *password* original do utilizador através de um algoritmo. Como resultado da aplicação do algoritmo, é obtido um valor único e irreversível, conhecido como *hash*, que representa a *password* do utilizador. Contudo, se dois utilizadores tiverem *passwords* idênticas, o resultado da *password hash* será igual.

Desta forma, com o intuito de aumentar ainda mais a segurança, utilizou-se o *password salt*. O *salt* é um valor aleatório único gerado para cada utilizador. Este valor é adicionado à *password* antes de aplicar o algoritmo para obter a *password hash*. Assim, é possível garantir que utilizadores com *passwords* iguais terão como resultado *password hashes* distintas.

Ao combinar ambas abordagens, *password hash* e *password salt*, o sistema AQUA garante que as *passwords* dos seus utilizadores são armazenadas de forma segura e protegidas contra ataques. Mesmo que alguém consiga aceder à base de dados, não conseguirá visualizar as *password* originais, e, caso tente obtê-las, terá dificuldade.

Na figura 55 é possível observar a implementação do método POST que permite o registo de um novo utilizador. Inicialmente é feita a verificação se o *email* utilizado pelo novo utilizador já está a registado no sistema, e se a estrutura da mensagem enviada pelo cliente é inválida. Caso não se verifique nenhuma destas situações, é utilizado o método '**CreatePasswordHash**' para a criação do *hash* e do *salt* da password do novo utilizador.

```
[HttpPost]
0 referências
public IActionResult Post([FromBody] User userObj)
{
    if (_apiDbContext.Users.Any(x => x.Email == userObj.Email))
    {
        return BadRequest("A user with the same email already exists.");
    }

    if (!ModelState.IsValid)
    {
        var errors = ModelState.Values.SelectMany(v => v.Errors).Select(e => e.ErrorMessage);
        return BadRequest(errors);
    }

    byte[] passwordHash, passwordSalt;
    CreatePasswordHash(userObj.Password, out passwordHash, out passwordSalt);

    userObj.PasswordHash = passwordHash;
    userObj.PasswordSalt = passwordSalt;

    _apiDbContext.Users.Add(userObj);
    _apiDbContext.SaveChanges();

    return StatusCode(StatusCode.Status201Created);
}
```

Figura 55. Método POST - Aplicação de *password hash* e *password salt* no registo de um novo utilizador

No método '**CreatePasswordHash**' (ver figura 56) é utilizado o algoritmo '**HMAC-SHA512**' para criar o *hash* da *password* de forma segura. Este algoritmo gera automaticamente uma chave aleatória, utilizada como o *salt* da *password*. De seguida, é utilizado o método '**ComputeHash**', responsável por gerar o *hash* da *password* recebida usando o algoritmo '**HMAC-SHA512**'. Por fim, os valores do *salt* e do *hash* são retornados como *output* do método, permitindo que sejam utilizados posteriormente para armazenar a password do novo utilizador de forma segura na base de dados.

```
1 referência
private void CreatePasswordHash(string password, out byte[] passwordHash, out byte[] passwordSalt)
{
    using (var hmac = new System.Security.Cryptography.HMACSHA512())
    {
        passwordSalt = hmac.Key;
        passwordHash = hmac.ComputeHash(Encoding.UTF8.GetBytes(password));
    }
}
```

Figura 56. Implementação do método '**CreatePasswordHash**'

No momento em que um utilizador tenta autenticar-se no sistema, é chamado o método POST responsável por validar as credenciais recebidas e permitir ou negar o acesso do utilizador ao sistema (ver figura 57).

```
[HttpPost("[action]")]
0 referências
public IActionResult Login([FromBody] LoginUserData userData)
{
    var user = _apiDbContext.Users.FirstOrDefault(x => x.Email == userData.Email);

    if (user == null)
    {
        return NotFound("User not found.");
    }

    if (!user.Active)
    {
        return Unauthorized("Your account is not active.");
    }

    if (!VerifyPassword(userData.Password, user.PasswordHash, user.PasswordSalt))
    {
        return Unauthorized("Incorrect email or password.");
    }
}
```

Figura 57. Parte da implementação do método POST, onde é feita a verificação da *password* do utilizador

No método POST é possível observar que é utilizado o método **'VerifyPassword'** para verificar se a *password* fornecida pelo utilizador no momento de *login* corresponde ao *hash* e o *salt* armazenados previamente. Este método recebe a *password* inserida, e o *hash* e o *salt* armazenado. Da mesma forma que na criação da *password* no momento de registo, é utilizado o algoritmo HMAC-SHA512. A instância da classe **'HMACSHA512'** é criada com o *salt* armazenado na base de dados. Desta forma, o objeto é configurado com o mesmo *salt* que foi utilizado durante a criação do *hash* da *password*.

Posteriormente, o método **'ComputeHash'** é utilizado para gerar o *hash* da *password* inserida com o mesmo *salt* utilizado anteriormente. Isto permite que seja feita uma comparação *byte* a *byte* entre o *hash* gerado e o *hash* armazenado na base de dados. Se algum *byte* for diferente, isso significa que a *password* inserida não é válida, retornando **'false'**.

Caso todas as comparações de *bytes* sejam verificadas com sucesso e não se verifique nenhuma diferença entre os *hashes*, o método retorna **'true'**, indicando que a *password* inserida pelo utilizador é válida, permitindo que o mesmo seja autenticado no sistema.

5.2.8 Json Web Token

O Json Web Token (JWT)¹¹ é um padrão aberto utilizado para partilhar informação de forma segura entre o cliente e o servidor. Ele oferece uma forma segura de autenticar utilizadores e partilhar informações, sendo constituído por três partes, separadas por pontos ("."), o cabeçalho, o *payload* e a assinatura.

No sistema AQUA, utilizou-se esta abordagem para garantir segurança. No desenvolvimento da API, no método POST do *endpoint* "api/Users/Login", inicialmente é verificado se o utilizador existe na base de dados e se as credenciais fornecidas estão corretas. De seguida, é feita a criação de um novo JWT, que inclui declarações relevantes do utilizador, como o ID, *email* e papel de utilizador. O *token* JWT gerado é retornado para o cliente como resposta ao pedido de login.

¹¹<https://www.akana.com/blog/what-is-jwt>

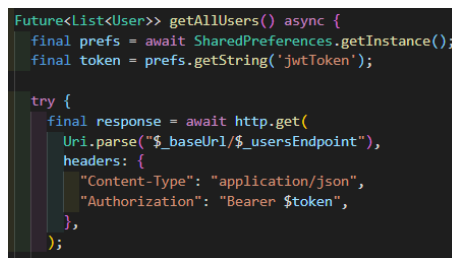
No lado do cliente, após fazer um pedido de login para o servidor, o *token* JWT retornado é armazenado localmente no dispositivo utilizando a biblioteca “**SharedPreferences**”, caso a autenticação seja feita com sucesso.

Quando o utilizador inicia a aplicação móvel, é verificado se existe algum *token* JWT armazenado localmente. Se houver, é feita a validação do *token*. Caso seja válido, o utilizador é automaticamente redirecionado para o ecrã inicial da aplicação, sem a necessidade de realizar login novamente. Isso permite manter a sessão do utilizador ativa entre diferentes sessões.

No entanto, se o *token* armazenado localmente não foi válido ou se não houver nenhum *token* armazenado (por exemplo, quando o utilizador efetua *logout* ou elimina a sua conta), o utilizador é solicitado a realizar login novamente para obter um novo *token* válido. Esta abordagem de validação do token no início da aplicação assegura que apenas utilizadores autenticados tenham acesso às funcionalidades e dados da aplicação.

Além de garantir segurança, a aplicação fornece uma experiência mais fluída para o utilizador, evitando a necessidade de inserir credenciais de acesso sempre que iniciar a aplicação móvel.

Em relação aos pedidos subsequentes do cliente ao servidor, o *token* JWT do utilizador autenticado é adicionado aos cabeçalhos dos pedidos HTTP. Na figura 58 é possível observar que é o *token* é adicionado ao cabeçalho “Authorization” com o prefixo “Bearer”.



```
Future<List<User>> getAllUsers() async {
    final prefs = await SharedPreferences.getInstance();
    final token = prefs.getString('jwtToken');

    try {
        final response = await http.get(
            Uri.parse("$_baseUrl/$_usersEndpoint"),
            headers: {
                "Content-Type": "application/json",
                "Authorization": "Bearer $token",
            },
        );
    }
}
```

Figura 58. Implementação do método “getAllUsers()”: Adição do token JWT ao cabeçalho “Authorization” de um pedido do cliente ao servidor.

No servidor, utilizou-se o atributo “[Authorize]” para proteger os *endpoints* implementados que requerem autenticação, tal como se pode observar na figura 54. Desta forma, o ASP.NET Core é responsável por validar o *token* JWT automaticamente, antes de permitir o acesso ao *endpoint*. Se o *token* foi válido e autenticado, o acesso ao *endpoint* é permitido. Caso contrário, o acesso é negado e o servidor retorna um erro de autorização.

Esta abordagem de autenticação e validação de *tokens* JWT no servidor ajuda a aumentar significativamente a segurança do sistema AQUA, adicionando uma camada adicional de proteção aos dados do sistema e apenas atribuindo o acesso aos mesmos a utilizadores autenticados e autorizados. Desta forma, é proporcionada uma camada que garante a proteção dos dados dos utilizadores e evita que os mesmos possam ser alterados por terceiros não autorizados.

5.3 Armazenamento de Dados *Online*

Ao disponibilizar os dados remotamente, é possível assegurar que as informações dos avistamentos e dos utilizadores podem ser acedidas e partilhadas de forma segura com outros utilizadores ou

sistemas externos, contribuindo para a construção de uma base de dados com informação valiosa sobre as espécies marinhas para investigadores, cientistas e qualquer entusiasta de cetáceos.

Tal como já referido anteriormente, a segurança é um atributo de qualidade muito importante para o sistema AQUA. Assim, optou-se por utilizar o PostgreSQL como sistema gestor de base de dados para o armazenamento e gestão dos dados *online*. O PostgreSQL proporciona as propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade), desempenhando um papel fundamental para preservar a integridade e consistência dos dados [63].

A atomicidade do PostgreSQL garante que as operações realizadas na base de dados sejam tratadas como uma só. Desta forma, caso ocorra algum erro durante o processamento de uma transação, todas as operações são revertidas, evitando perda ou alterações incompletas nos dados.

A consistência garante que a base de dados esteja sempre num estado válido a qualquer momento, ou seja, as transações realizadas não podem violar as restrições definidas na estrutura dos dados, garantindo que apenas são armazenados dados válidos e consistentes.

Por outro lado, o isolamento diz respeito à capacidade do sistema gerir várias transações concorrentes de forma independente. Esta propriedade é particularmente relevante em sistemas com múltiplos utilizadores, como é o caso do sistema AQUA, onde várias pessoas podem submeter e aceder dados de avistamentos simultaneamente.

A durabilidade assegura que as alterações realizadas na base de dados sejam permanentemente armazenadas, evitando perda de dados em caso de falha do sistema. Isto garante que as informações recolhidas no sistema AQUA são preservadas ao longo do tempo.

Além disso, sendo o PostgreSQL um sistema gestor de base de dados relacional, o mesmo permitiu criar uma estrutura de dados organizada em tabelas relacionadas, facilitando o mapeamento com os modelos de dados do *Entity Framework* (EF) e, conseqüentemente, simplificando as operações CRUD nos dados.

– Armazenamento de Imagens:

Durante o desenvolvimento da aplicação AQUA, enfrentou-se um desafio em relação ao armazenamento de imagens e vídeos. O PostgreSQL, a base de dados utilizada para o armazenamento dos dados remotos do sistema AQUA, possui algumas limitações em relação ao armazenamento direto deste tipo de dados, o que levou a procurar outra solução que permitisse o armazenamento deste tipo de dados.

Para resolver esta limitação, optou-se por utilizar o Firebase¹². O Firebase é uma plataforma de desenvolvimento móvel criada pelo Google, e que oferece um conjunto de serviços para o desenvolvimento de aplicações, incluindo um serviço de armazenamento de ficheiros na nuvem conhecido como Firebase Storage.

O Firebase Storage permite armazenar e gerir facilmente imagens e vídeos na nuvem. Desta forma, ao utilizar este serviço, as imagens e vídeos carregados na aplicação AQUA podem ser enviados diretamente para o Firebase Storage, sendo estes armazenados de forma segura e confiável. Estes ficheiros podem posteriormente ser acedidos pela aplicação móvel.

A utilização do Firebase para o armazenamento destes ficheiros foi fundamental para garantir o armazenamento das imagens e vídeos sobre os avistamentos e incidentes reportados pelos

¹²<https://firebase.google.com/>

utilizadores na aplicação AQUA. Além disso, este serviço permitiu armazenar as fotografias de perfil dos utilizadores, carregadas no momento de criação de perfil.

Para integrar o Firebase no Flutter, utilizou-se as bibliotecas “**firebase_core**” e “**firebase_storage**”. A biblioteca “**firebase_core**” é responsável pela inicialização e configuração do Firebase na aplicação móvel, enquanto a biblioteca “**firebase_storage**” fornece recursos necessários para interagir com a Firebase Storage, ou seja, para realizar o envio, acesso e gestão das imagens e vídeos.

5.4 Testes Funcionais: Tracking da Rota da Viagem

Nesta secção são descritos os testes funcionais realizados para avaliar o funcionamento da funcionalidade de *tracking* das rotas de viagem.

5.4.1 Objetivos e Protocolo

O objetivo dos testes realizados é verificar se a aplicação AQUA regista corretamente a rota das viagens em diferentes situações, tanto com uma conexão à Internet estável, quanto em condições onde a conexão é instável ou inexistente.

Para a realização dos testes forneceu-se a aplicação aos participantes, solicitando que registassem diversas rotas de viagem ao longo de um dia. Além de registar as rotas de viagem, também solicitou-se aos participantes que registassem qualquer problema identificado durante a utilização da aplicação. Após a realização dos testes, realizou-se uma entrevista individual com cada participante para compreender melhor as dificuldades relatadas e avaliar o impacto da funcionalidade no desempenho do dispositivo, incluindo o aquecimento e consumo de recursos. Desta forma, é possível identificar falhas e melhorar a funcionalidade antes da realização do estudo final no contexto real.

5.4.2 Participantes

Para a realização dos testes recrutou-se 4 participantes. Três participantes foram selecionados para testar a funcionalidade durante viagens de barco, onde a conexão à Internet é instável, enquanto que um participante foi selecionado para realizar os testes durante viagens de carro, onde a conexão é estável.

5.4.3 Resultados

A análise dos dados realizou-se com base em duas abordagens. A primeira consiste em analisar as coordenadas registadas durante as viagens dos participantes. Sendo as coordenadas registadas a cada 5 minutos após iniciar o *tracking* utilizando a aplicação, espera-se que, por exemplo, numa viagem com duração de 23 minutos, sejam obtidas: uma coordenada registada ao iniciar o *tracking* da rota, quatro coordenadas registadas automaticamente a cada 5 minutos (aos 5, 10, 15 e 20 minutos), e uma coordenada registada ao terminar o *tracking* da rota (aos 23 minutos).

Posteriormente, a taxa de sucesso do *tracking* é calculada comparando o número de coordenadas registadas na base de dados com o número esperado, avaliando, assim, a precisão e efetividade da funcionalidade de *tracking* da rota das viagens.

A segunda abordagem é baseada no *feedback* dos participantes em relação às dificuldades experienciadas e ao aquecimento e consumo de bateria do dispositivo durante a utilização da funcio-

nalidade. Ao analisar estas informações é possível identificar possíveis problemas de otimização ou falta de eficiência na utilização de recursos do dispositivo.

– Resultados Quantitativos:

Nesta secção são apresentados os resultados quantitativos obtidos a partir da análise de um total de 11 rotas de viagens registadas pelos participantes do estudo. A duração de cada rota foi calculada com base na diferença de tempo entre o *timestamp* da primeira coordenada registada no momento de início do *tracking* e o *timestamp* da última coordenada registada no momento de paragem do *tracking*. Os participantes com IDs 1, 2 e 3 realizaram o *tracking* durante viagens em alto mar, enquanto que o participante com o ID 4 utilizou o *tracking* durante viagens de carro. Na tabela seguinte são apresentados os resultados obtidos.

Tabela 13. Resultados Quantitativos dos Testes Funcionais do Tracking das Rotas de Viagem

ID Participante	Nº Rota	Conexão à Internet	Duração da viagem (hh:mm:ss)	Nº de Coordenadas Registadas	Nº de Coordenadas Esperadas	Taxa de Sucesso (%)
1	1	Instável	03:48:21	38	45	84,44%
2	2	Instável	03:48:25	24	45	53,33%
3	3	Instável	01:05:40	11	15	73,33%
4	4	Estável	00:47:56	11	11	100%
	5	Estável	00:46:09	8	11	72,73%
	6	Estável	00:40:01	7	10	70%
	7	Estável	00:39:26	8	9	88,89%
	8	Estável	00:44:08	9	11	81,82%
	9	Estável	00:25:44	7	7	100%
	10	Estável	01:31:39	14	20	70%
	11	Inexistente	00:47:22	11	11	100%

Com base na tabela apresentada calculou-se a média da taxa de sucesso para as rotas registadas em alto mar e para as rotas registadas durante as viagens de carro. Para as rotas em alto mar observou-se, aproximadamente, uma taxa de sucesso média de 70.37%, enquanto que para as rotas de carro obteve-se uma taxa de sucesso média de, aproximadamente, 85.43%.

De acordo com os dados das coordenadas registadas na base de dados PostgreSQL após a submissão de uma rota de viagem (ver tabela 1 da figura 59), é possível verificar que, apesar da funcionalidade ter sido implementada para registar coordenadas de 5 em 5 minutos, há alguns atrasos no registo das coordenadas. Por exemplo, a coordenada com ID 556 foi registada automaticamente, seguida pela coordenada com ID 557 que foi registada 5 minutos e 57 segundos depois (atraso de 57 segundos). Da mesma forma, a coordenada com ID 558 foi registada 6 minutos e 37 segundos após a coordenada com ID 557 (atraso de 1 minuto e 37 segundos).

Ainda, ao analisar as coordenadas de cada rota registada individualmente, foi possível identificar uma perda significativa de coordenadas na rota número 2, que apresentou uma taxa de sucesso de apenas 53.33%. Ao comparar a última coordenada registada automaticamente com a coordenada registada no momento em que o *tracking* foi interrompido, observou-se um intervalo de tempo de 1 hora e 41 minutos sem nenhum registo de coordenadas (ver tabela 2 da figura 59), e onde era esperado o registo automático de 20 coordenadas.

1	Id [PK] integer	Latitude double precision	Longitude double precision	Timestamp timestamp with time zone
164	556	32.6993603	-16.7354697	2023-05-21 17:54:11.50677+00
165	557	32.6999124	-16.7382608	2023-05-21 18:00:08.471874+00
166	558	32.6997335	-16.7402177	2023-05-21 18:06:45.454134+00

2	Id [PK] integer	Latitude double precision	Longitude double precision	Timestamp timestamp with time zone
113	503	32.5770391	-17.0101707	2023-05-17 12:02:43.238165+00
114	504	32.6457307	-16.9116507	2023-05-17 13:43:42.7428+00

Figura 59. Base de dados PostgreSQL: (1) Dados de três coordenadas de uma rota registradas automaticamente; (2) Dados das últimas duas coordenadas registradas da rota número 2.

– Resultados Qualitativos:

Os participantes com ID 3 e 4 relataram que durante a utilização da funcionalidade não observaram uma diferença significativa no consumo da bateria em comparação com o uso normal do dispositivo. Da mesma forma, não observaram o aquecimento do dispositivo, mesmo quando utilizavam outras aplicações em simultâneo, enquanto o *tracking* era executado em segundo plano.

O participante com ID 4 referiu ter parado o *tracking* da rota de uma viagem e que a mesma tinha sido armazenada localmente com sucesso, sendo posteriormente enviada para o servidor sem problemas. Também, referiu que ao tentar terminar a rota de uma viagem, foi apresentada uma mensagem de erro indicando que ocorreu um problema ao guardar a rota e que a mesma acabou por se perder.

Para além de problemas funcionais, o participante com ID 4 revelou uma dificuldade relacionada com as mensagens apresentadas, que desapareciam rapidamente, não permitindo tempo suficiente para a leitura completa da mensagem, apontando para um possível problema de usabilidade. Ainda, para além da tarefa solicitada, o mesmo participante decidiu explorar a aplicação, referindo que a voz do áudio utilizada para reproduzir as informações sobre as espécies marinhas era “irritante” devido à sua semelhança com um “robot”.

5.4.4 Discussão

Devido à dificuldade encontrada em recrutar participantes para testar a funcionalidade durante viagens marítimas, e sendo estas pouco frequentes, tornou-se difícil recolher uma amostra de dados suficientes que permitissem tirar conclusões concretas em relação ao funcionamento do *tracking* das rotas de viagem em alto mar, onde a conexão à Internet é instável. Contudo, os dados recolhidos no *tracking* das viagens de carro permitem complementar a informação recolhida em relação à funcionalidade, embora não representem um contexto tão semelhante ao real, das viagens marítimas.

Com base nos resultados apresentados na tabela 13, é possível concluir que a funcionalidade de *tracking* das rotas de viagem de carro apresentou uma taxa de sucesso média superior às viagens realizadas no mar. Esta diferença poderia indicar que a instabilidade da conexão à Internet em alto mar afeta a capacidade da aplicação registar as coordenadas com a mesma frequência e precisão em comparação com contexto das viagens de carro, resultando numa taxa de sucesso inferior.

Contudo, ao analisar a rota número 11, realizada sem conexão à Internet, verificou-se uma taxa de sucesso de 100%. O participante que realizou a viagem relatou que durante esta viagem não utilizou o dispositivo, devido à falta de conexão. Desta forma, acredita-se que a utilização do dispositivo com outras aplicações enquanto o *tracking* é executado em segundo plano, pode afetar o desempenho da funcionalidade. Estas aplicações podem ter utilizado recursos do dispositivo, como o processador e o GPS, impactando a precisão e a frequência do registo das coordenadas das rotas de viagem, como observado na tabela 1 da figura 59.

Ao analisar as coordenadas da rota número 2, referente a uma viagem marítima, foi identificado um intervalo de 1 hora e 41 minutos sem registo de coordenadas. Durante o período em que o *tracking* esteve ativo, verificou-se que as coordenadas foram registadas automaticamente a aproximadamente cada 5 minutos. No total, foram registadas 24 coordenadas, enquanto outras 20 coordenadas esperadas não foram registadas. Somando esses valores, obtém-se um total de 44 coordenadas. Considerando que o número esperado de coordenadas era 45, é possível concluir que o *tracking* da rota foi bastante eficiente por aproximadamente 2 horas e 7 minutos. A paragem no registo automático das coordenadas pode ter sido causada por problemas na conexão à Internet ou nos recursos do dispositivo, como o GPS ou processador. No entanto, não é possível tirar conclusões concretas em relação ao problema identificado, devido à falta de informação sobre o mesmo.

No entanto, é importante ressaltar que os resultados obtidos são baseados numa amostra de dados muito limitada, o que afeta a generalização dos resultados e, consequentemente, as conclusões dos testes. Para obter melhores conclusões, seria necessário recolher uma amostra de dados superior, principalmente no número de viagens marítimas.

Com base no *feedback* dos participantes, não foi relatado nenhum aquecimento excessivo do dispositivo, nem de descarregamento da bateria além do que eles consideravam normal, indicando que a funcionalidade foi desenvolvida de forma eficiente, garantindo uma utilização satisfatória e discreta sem afetar negativamente o desempenho do dispositivo.

No caso da rota da viagem perdida pelo participante com ID 4, identificou-se um problema relacionado com o armazenamento das rotas quando o utilizador tem conexão à Internet mas ocorre uma falha na comunicação com o servidor. A aplicação AQUA apenas armazena as rotas de viagem localmente quando a conexão à Internet está indisponível. Desta forma, a rota acaba sendo perdida no caso de haver conexão mas ocorrer um problema na comunicação com o servidor. Para evitar a perda de informação nestas situações, recomenda-se armazenar as rotas de viagem localmente quando ocorrer uma falha na comunicação com o servidor. Assim, a rota da viagem será armazenada localmente e estará disponível para submissão posterior.

Apesar de não ter sido um objetivo dos testes, foi identificado um problema de usabilidade relacionado com a apresentação de mensagens de sucesso e de erro durante a submissão das rotas. As mensagens desapareciam rapidamente, não permitindo uma leitura completa da mensagem e dificultando a compreensão do resultado da ação do utilizador. Com base neste problema, recomenda-se aumentar o tempo de apresentação das mensagens para garantir que os utilizadores têm tempo suficiente para ler e compreender a mensagem apresentada.

6 Avaliação do Protótipo

Nesta secção são descritas duas avaliações realizadas com a aplicação AQUA: a primeira, realizada com especialistas em desenho e desenvolvimento de jogos e outros sistemas interativos, com o objetivo de identificar melhorias no desempenho e usabilidade aplicação antes da realização da avaliação no contexto real, e a segunda, realizada com profissionais pertencentes a empresas de observação de cetáceos, com o objetivo de avaliar o impacto da gamificação no envolvimento e na experiência dos utilizadores, assim como o funcionamento da aplicação no contexto real. A discussão dos resultados de ambas as avaliações será apresentada em conjunto com a discussão de todo o processo seguido nesta dissertação, no Capítulo 7.

6.1 Avaliação com Especialistas em Desenho de Media Interativos

Nesta secção são descritos testes de usabilidade realizados para avaliar o desempenho funcional e a usabilidade da aplicação AQUA. Estes testes desempenharam um papel crucial na preparação da aplicação para a avaliação final, identificando possíveis problemas e melhorias necessárias para garantir o sucesso do sistema e a realização da avaliação no contexto real.

6.1.1 Objetivos e Protocolo

O objetivo principal destes testes foi verificar se a aplicação AQUA estava a funcionar corretamente em termos de funcionalidades principais, com o registo de avistamentos, visualização de mapas e informação sobre as espécies, assim como a interação com os elementos de gamificação incluídos.

Os testes realizados foram feito com especialistas em desenho e desenvolvimento de medias interativos, de forma a identificar melhorias que pudessem garantir que a aplicação fosse mais intuitiva e fácil de utilizar, e oferecesse uma experiência mais satisfatória para os seus utilizadores.

Para a realização das sessões de teste, decidiu-se realizar as mesmas de forma individualizada. Desta maneira, a observação e análise das interações de cada participante seria facilitada e de forma mais detalhada. Ao realizar os testes de forma individual, foi possível dedicar mais atenção a cada participante e recolher dados mais completos, apesar de requerer mais tempo.

Inicialmente, foi apresentado o objetivo dos testes aos participantes e fornecido um *smartphone* com a aplicação AQUA previamente instalada. Foram solicitados a realizar diversas tarefas, desde o registo na aplicação até o reporte de um avistamento, visualização de recompensas desbloqueadas e exploração de restantes funcionalidades da aplicação.

Para facilitar o reporte do avistamento, foi fornecida uma tabela com informações sobre as espécies que deveriam ser reportadas. As espécies foram selecionadas de forma a desafiar os participantes a procurá-las, adicionar uma espécie desconhecida, bem como desbloquear cartas de uma espécie considerada rara e de uma semirara, e dois distintivos. Desta forma, seria possível recolher informação sobre os elementos de gamificação experienciados pelos participantes.

Após a conclusão das tarefas, solicitou-se aos participantes o preenchimento de um questionário que tinha como objetivo recolher dados demográficos, e avaliar a usabilidade da aplicação e a experiência de utilização da mesma, utilizando o questionário SUS e o *User Experience Questionnaire* (UEQ)¹³, respetivamente. O questionário SUS permite obter uma pontuação geral em

¹³<https://www.ueq-online.org/>

relação à usabilidade do sistema, identificando necessidades de melhorias no *design* da aplicação, enquanto o UEQ permite complementar essa informação através da medição da experiência do utilizador em termos das suas perceções e emoções durante a interação com a aplicação. Além disso, o questionário incluía a lista de distintivos disponíveis na aplicação AQUA, sendo solicitado aos participantes que avaliassem cada distintivo numa escala de 1 a 5, de forma a compreender a relevância desses distintivos no contexto da aplicação e, também, recolher sugestões de novos distintivos que poderiam ser adicionados no futuro.

Por fim, realizou-se uma entrevista com os participantes, contendo perguntas sobre as suas dificuldades durante a utilização da aplicação, incluindo *bugs* e problemas de usabilidade, e ainda para compreender melhor a experiência e as suas reações perante os elementos de gamificação apresentados.

Com a realização dos testes de usabilidade individuais com alguns participantes iniciais, foi possível observar padrões de problemas e inconsistências que se repetiram em todos os participantes. Desta forma, com o intuito de obter *feedback* mais aprofundado sobre a versão atual da aplicação e identificar possíveis melhorias e novos elementos de gamificação, adotou-se uma abordagem diferente com os restantes participantes. Realizou-se entrevistas com dois grupos, cada um composto por dois participantes. Durante as entrevistas, apresentou-se a aplicação aos participantes, permitindo que eles explorassem a interface, testassem as funcionalidades e fornecessem *feedback* específico.

As entrevistas permitiram abordar diferentes aspetos, como a experiência de utilização, a clareza da interface, e compreensão das funcionalidades. Além disso, discutiu-se a gamificação na aplicação, na tentativa de compreender que elementos os participantes consideravam mais atraentes e motivadores, assim como entender que novos elementos poderiam ser adicionados para aumentar o envolvimento dos utilizadores da aplicação AQUA.

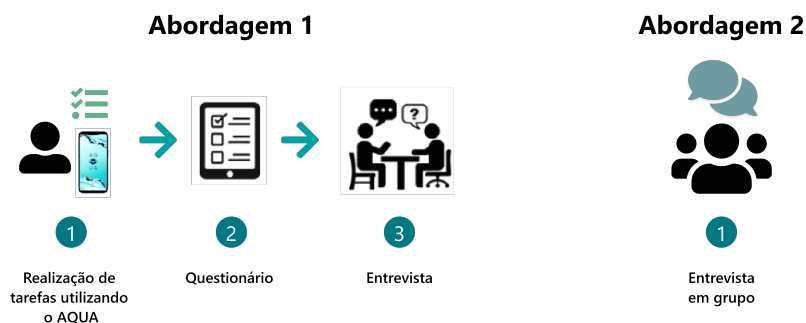


Figura 60. Abordagens adotadas para os testes de usabilidade (2ª iteração)

6.1.2 Participantes

Nos testes estiveram envolvidos 9 participantes. Entre os participantes recrutados, 2 deles já tinham tido experiência prévia com a aplicação AQUA 1.0. Além disso, outros 2 participantes já tinham testado a versão anterior da aplicação AQUA na primeira interação dos testes de usabilidade.

É importante destacar que 7 participantes eram profissionais com experiência em *design* e desenvolvimento de *software*, proporcionando uma visão especializada em relação ao funcionamento da aplicação e permitindo obter *feedback* específico para melhorias no *design* da aplicação.

6.1.3 Resultados

Nesta subsecção são apresentados os resultados quantitativos e os resultados qualitativos obtidos a partir da realização dos testes anteriormente descritos.

– Resultados Quantitativos:

- **Resultados do SUS:** A partir do questionário SUS, calculou-se o SUS Score obtido para cada participante, e, de seguida, calculou-se o valor médio do sistema, seguindo as instruções de [56]. Obteve-se um resultado de 82.5, numa escala de 0 a 100, indicando que a aplicação era “excelente” [50] em termos de usabilidade.
- **Resultados do UEQ:** A partir do questionário UEQ, obteve-se valores médios dos adjetivos que avaliam a qualidade pragmática e qualidade hedônica da aplicação. Em relação à qualidade pragmática, a média indica que a maioria considerou a aplicação como “conduutora”, “fácil de usar”, “eficiente” e “evidente”. Já em relação à qualidade hedônica, a média indica que a maioria dos participantes considerou a aplicação “excitante”, “interessante”, “original” e “vanguardista”.

O UEQ considera uma escala de valores entre -3 e +3. Desta forma, considera-se que valores acima de “0.8” indicam uma avaliação positiva nas escalas. Os valores médios obtidos para cada escala avaliada — qualidade pragmática, qualidade hedônica, geral — foram: 1.375, 1.250 e 1.313, respetivamente.

- **Relevância dos distintivos:** Tal como referido anteriormente, solicitou-se aos participantes para a avaliar a relevância dos distintivos existentes na aplicação AQUA. A relevância dos distintivos foi medida numa escala de 1 (pouca relevância) a 5 (muita relevância).

Na tabela seguinte são apresentados os resultados obtidos com base nas respostas dos participantes. São apresentados os títulos de cada distintivo avaliado, juntamente com a média da relevância atribuída pelos participantes.

Tabela 14. Resultados da relevância média de cada distintivo da aplicação AQUA

Distintivo	Relevância Média
Novice Spotter	5
Sighting Pro	4
Citizen Scientist	4
Geographer	4.5
Collector	4.75
Master Enthusiast	4
Marine Enthusiast	3.75
Ocean Explorer	2.25
Whale Watcher	2.25
Dolphin Defender	2.5
Marine Life Adventurer	3.75
Ocean Guardian	4.75

– Resultados Qualitativos:

Para ilustrar os resultados, são citadas no texto algumas declarações dos participantes. As citações seguem o formato U<NúmeroDoParticipante>_<IDdaCitação>, e a lista completa de citações encontra-se no anexo C.3.2.

Durante os testes de usabilidade com os quatro participantes iniciais, foram identificados padrões nos problemas encontrados (problemas de usabilidade e *bugs*). Os quatro participantes, ao criar uma conta, tentaram continuar sem adicionar uma foto e adicionando apenas uma descrição ao seu perfil. No entanto, ao tentar continuar sem adicionar a foto, a aplicação ficava num estado de carregamento permanente e a conta não era criada. Os participantes optaram, então, por pressionar o botão “Skip”.

Ao pressionar o botão “Skip”, os participantes ficaram confusos, pois não receberam nenhum tipo de *feedback* em relação à ação realizada. Isso deu a impressão de que a aplicação tinha parado de funcionar e que a conta não estava sendo criada.

Um participante mencionou que não percebeu que era possível carregar uma foto de perfil, pois o título apresentado no ecrã era “Tell us about yourself”, o que o levou a entender que era apenas necessário escrever algo sobre si na descrição do perfil (U1_1).

Além disso, um participante tentou criar uma conta usando um *email* fictício que já estava registado na aplicação. A aplicação apresentava uma mensagem de erro indicando que algo de errado havia acontecido, mas não era especificado o tipo de problema. Isso levou o participante a alterar todos os dados preenchidos, na tentativa de resolver o problema (U2_1).

Durante o processo de reportar avistamentos, não foram identificados problemas no funcionamento da aplicação. No entanto, ao ser solicitado que corrigissem o nome da espécie avistada, um participante pressionou um botão na barra de navegação com o ícone de um lápis. O participante mencionou que esse ícone normalmente indica a edição de informações, e por isso pressionou esse botão (U3_1).

Na pesquisa das espécies avistadas, nenhum problema foi identificado. Contudo, um participante sugeriu que, ao selecionar uma espécie após a pesquisa pelo nome, o campo de texto da pesquisa fosse limpo automaticamente, facilitando a pesquisa de outras espécies de forma mais rápida (U2_2).

Todos os participantes conseguiram submeter o avistamento armazenado localmente ao aceder ao histórico de reportes. Dois participantes tentaram clicar no avistamento para enviá-lo individualmente, mas a aplicação AQUA permitia apenas o envio de todos os reportes armazenados localmente de uma vez. Apesar disso, os participantes perceberam e concluíram a tarefa com sucesso.

Um participante sugeriu a criação de reportes colaborativos, onde vários utilizadores poderiam colaborar entre si para fazer um reporte de avistamentos. Desta forma, os membros da mesma equipa de uma empresa de observação de cetáceos poderiam trabalhar juntos para recolher informações sobre os avistamentos e competir com outras empresas (U4_1).

No ecrã de desbloqueio de recompensas (cartas e distintivos), os participantes demonstraram dificuldade em interagir com as recompensas desbloqueadas, pois era necessário fazer *scroll* verticalmente. Todos tentaram fazer *scroll* horizontalmente, referindo estarem acostumados a

navegar entre páginas dessa maneira (U1_2, U2_3, U3_2). Eles sugeriram que fosse fornecida alguma indicação de como interagir com os elementos ao visualizá-los pela primeira vez (U2_4, U4_2).

Vários participantes sugeriram a criação de uma animação para destacar quando um utilizador desbloqueia uma carta de uma espécie rara, tornando essa recompensa mais destacada em relação às outras (U5_1, U6_1). Na visualização das cartas das espécies, dois participantes sugeriram alterar a fonte de texto utilizada para o nome das espécies, pelo facto de ser difícil de ler (U1_3, U7_1). Na identificação do nível de raridade das cartas das espécies, não se verificou dificuldades. Todos consideraram que as cores das cartas estavam adequadas para cada nível de raridade correspondente (U1_4, U2_5, U4_3, U7_2).

Ao visualizar a coleção de distintivos, todos os participantes compreenderam que era possível visualizar a explicação de como cada distintivo pode ser desbloqueado, ao clicar nos distintivos bloqueados. Um dos participantes sugeriu que ao clicar num distintivo bloqueado, fosse apresentado o título do mesmo, permitindo aos utilizadores compreender qual seria o título desbloqueado caso obtivessem esse distintivo (U6_2).

Durante a visualização das informações sobre uma espécie marinha, um participante mencionou ter pensado que o botão para ouvir o áudio das informações era para ouvir o som produzido pela espécie, devido à posição do botão na interface (U7_3).

Quanto à localização do mapa de avistamentos, houve algumas dificuldades. Alguns participantes não perceberam que o mapa de avistamento estava localizado no ecrã inicial da aplicação, pois o título apresentado era “Reported Sightings” (U1_5, U5_2). No entanto, todos sugeriram que, ao clicar nos marcadores dos avistamentos no mapa, fosse possível visualizar detalhes sobre eles, como as espécies que foram reportadas nesse local (U1_6, U2_6, U3_3, U4_4, U6_3).

Dois participantes sugeriram que, no ecrã inicial, fosse possível visualizar dados gerais da aplicação, como as 5 espécies mais avistadas pelos utilizadores, assim como os dados pessoais do próprio utilizador, ou seja, as suas 5 espécies mais avistadas. Eles referiram que, inicialmente, nesse ecrã são apresentados contadores com dados pessoais do utilizador, mas depois são apresentados dados gerais de todos os utilizadores do sistema, o que gera uma inconsistência (U6_4, U7_4).

Além disso, um participante mencionou que ficou confuso em relação aos números apresentados no ecrã inicial. Inicialmente, era apresentado que o utilizador tinha dois avistamentos reportados, mas que tinha quatro espécies desbloqueadas. Essa divergência gerou confusão, pois ele considerava que cada espécie desbloqueada correspondia a um avistamento distinto, mesmo que fossem reportados no mesmo reporte (U4_5).

Ao visualizar o *ranking* de utilizadores, todos os participantes tentaram clicar num utilizador para ver os seus dados, incluindo as recompensas desbloqueadas (U1_7, U2_7). A maioria dos participantes gostou da opção de seleccionar um título para se destacar em relação aos restantes (U3_4, U4_6). Três participantes sugeriram destacar a posição atual do utilizador no *ranking*, alterando a cor do utilizador na lista para facilitar a identificação da sua posição (U5_3, U6_5, U7_5).

Quanto ao *design* da aplicação, todos os participantes mencionaram que gostaram das cores selecionadas e que a interação com o sistema era simples e intuitiva (U1_8, U2_8, U3_5, U5_4). Eles apreciaram o *design* das cartas, mas acharam que os ícones dos distintivos eram muito diferentes dos restantes elementos da interface (U6_6).

Em relação aos elementos de gamificação existentes na aplicação AQUA, todos os participantes os consideraram relevantes e motivadores para reportar mais avistamentos e competir com outros utilizadores.

No entanto, um participante mencionou que, embora tenha gostado das cartas, não considerou o desbloqueio delas muito relevante. Ele sugeriu que, ao desbloquear cartas, a parte posterior delas pudesse fazer parte de um *puzzle*. À medida que o utilizador desbloqueasse várias cartas, poderia uni-las e criar um *puzzle* completo, desbloqueando assim novas recompensas (U1_9). Além disso, um participante sugeriu adicionar um aquário virtual como novo elemento de gamificação. Nesse aquário, o utilizador poderia visualizar as espécies que já desbloqueou e interagir com elas (U4_7).

Outro participante sugeriu adicionar um sistema de níveis à aplicação, onde os utilizadores poderiam avançar de nível à medida que ganhassem pontos (U6_7). Esta subida de níveis permitiria obter recompensas ao longo do progresso. Cada vez que um utilizador desbloqueasse uma nova espécie, ele ganharia pontos com base na raridade da espécie. Por exemplo, uma espécie comum atribuiria 1 ponto, enquanto uma espécie rara atribuiria 3 pontos (U6_8).

Quanto aos distintivos, alguns participantes sugeriram criar distintivos progressivos, ou seja, distintivos semelhantes que estão relacionados, em vez de serem considerados distintivos diferentes (U6_9, U7_6). Isso significa que alguns distintivos apenas poderiam ser desbloqueados após outros terem sido adquiridos pelo utilizador. Como sugestão de novos distintivos sugeriu-se adicionar um por estar no top 10 do *ranking*, por desbloquear várias espécies no mesmo dia, por reportar avistamentos em diferentes horas do dia (manhã e tarde) e por desbloquear espécies de tamanhos diferentes.

Em relação à utilização da aplicação AQUA, os participantes mencionaram que cidadãos locais não utilizariam a aplicação com frequência, já que reportar avistamentos destas espécies exigiria viagens pagas ou uma embarcação própria (U1_10, U2_9, U6_5). No entanto, consideraram a aplicação útil para biólogos marinhos que monitorizam as espécies marinhas (U3_6, U4_8). Também, sugeriram que a aplicação poderia ser útil para empresas que realizam viagens de observação de cetáceos, permitindo que competissem entre si e destacando aquelas que contribuísssem mais para a ciência através da recolha de informações utilizando a aplicação AQUA (U7_7). Além disso, consideram que as empresas poderiam visualizar onde as espécies geralmente são avistadas, facilitando o caminho até elas (U6_10). Foi mencionado que, se a aplicação fosse utilizada por cidadãos locais, os dados poderiam ser manipulados apenas para obter as recompensas, comprometendo a credibilidade das informações (U4_9, U7_8). Portanto, consideram necessário validar esses dados para garantir a sua veracidade.

Alguns participantes sugeriram que para a validação dos dados reportados poderia ser utilizada inteligência artificial. Através da inteligência artificial poderia ser verificado se as imagens capturadas correspondem às espécies identificadas pelos utilizadores (U4_10, U6_6). Outros mencionaram que, se a aplicação fosse utilizada apenas por profissionais, não seria validar os

dados reportados, uma vez que o objetivo deles seria contribuir para a ciência e não partilhar dados incorretos (U6_11, U7_9).

É importante destacar que os participantes que testaram a aplicação na primeira iteração dos testes de usabilidade consideraram que a aplicação AQUA teve uma melhoria significativa em termos de usabilidade e funcionalidade (U4_11). Além disso, os participantes que testaram a versão desenvolvida pela aluna do IST também notaram uma grande melhoria em termos de desempenho, referindo que a experiência de utilização era fluida, ao contrário do que acontecia anteriormente (U1_11, U2_10).

6.1.4 Impacto no Design Final

Os testes realizados permitiram identificar diversos problemas que afetavam a usabilidade da aplicação, levando à necessidade de implementar várias modificações no *design* da aplicação AQUA. Estas alterações tiveram como objetivo melhorar a experiência dos utilizadores, estando estas incluídas no *design* apresentado na secção 3.4.

Entre essas alterações, destacam-se a adição de uma animação indicando o estado de carregamento ao pressionar o botão “Skip” durante a criação do perfil. O título “Tell us about yourself” foi alterado para “Customize your profile”. Quando um *email* já está registado durante a criação de uma conta, apresenta-se uma mensagem específica explicando o motivo do erro.

Para evitar confusão na barra de navegação, o ícone do lápis foi substituído por um ícone que sinaliza a criação de um novo reporte, seja um avistamento ou um incidente, em vez de edição de informações (ver 1 da figura 61).

Na secção de reportes, o botão para iniciar uma rota de viagem foi removido, pois essa funcionalidade não está relacionada com os reportes. Para agilizar a pesquisa de espécies por nome, implementou-se a limpeza automática do campo de texto após a seleção de uma espécie.

No ecrã de desbloqueio de recompensas, adicionou-se uma animação para indicar ao utilizador que pode interagir com as recompensas desbloqueadas ao fazer *scroll* vertical, especialmente na primeira vez que desbloqueia uma recompensa (ver 2 da figura 61).

As cartas das espécies tiveram a fonte de texto alterada para melhorar a legibilidade (ver 3 da figura 61). No ecrã inicial, o título “Map of Sightings” substituiu o título anterior do mapa de avistamentos, facilitando a compreensão. Além disso, a descrição do contador foi modificada para evitar más interpretações pelos utilizadores. No mapa de avistamentos adicionou-se a opção de visualizar detalhes sobre os avistamentos ao clicar nos marcadores no mapa, tornando esta funcionalidade mais interativa.



Figura 61. Alterações no Design Final: (1) Novo ícone da barra de navegação; (2) Adição de animação; (3) Fonte de texto das cartas.

6.2 Avaliação com Profissionais em Observação de Cetáceos

Nesta secção é descrita a avaliação do protótipo final realizado para avaliar o impacto da gamificação no envolvimento e na experiência dos utilizadores e o funcionamento da aplicação AQUA no contexto real de uso, ou seja, durante viagens marítimas.

6.2.1 Objetivos e Protocolo

O objetivo do estudo consistiu em avaliar o impacto da gamificação no envolvimento dos participantes, assim como destacar a relevância de cada elemento de gamificação implementado como mecanismo para aumentar essa motivação e o envolvimento dos utilizadores. Além disso, o mesmo teve como objetivo avaliar o comportamento da aplicação AQUA em alto mar e num contexto mais aproximado ao real.

Desta forma, e devido à complexidade de envolver cidadãos locais na recolha de dados de avistamentos de cetáceos e às limitações temporais, optou-se por recrutar empresas de observação de cetáceos para participar no estudo, visto que estas empresas realizam várias viagens marítimas quase todos os dias. Esta abordagem permite testar a aplicação móvel AQUA num período mais longo, o que é crucial para avaliar o impacto da gamificação na motivação dos utilizadores. Avaliar esse impacto exige um uso contínuo durante algum tempo, uma vez que participantes que utilizariam a aplicação apenas uma vez não proporcionariam resultados tão abrangentes e significativos.

Para o recrutamento, foram contactadas dez empresas de observação de cetáceos, sendo nove localizadas na ilha da Madeira e uma na ilha de São Miguel, Açores. No entanto, apenas cinco empresas mostraram disponibilidade em participar no estudo. Após o contacto inicial, foi agendada uma reunião individual com cada uma dessas empresas, com o intuito de apresentar e explicar os detalhes do estudo, assim como esclarecer eventuais questões que poderiam ter. A empresa dos Açores foi contactada de forma remota, devido às limitações geográficas.

Cada participante foi solicitado a testar a aplicação AQUA durante uma semana, reportando avistamentos de espécies durante as viagens da sua empresa, assim como explorando as diversas funcionalidades da aplicação. Após a fase de teste em contexto real, solicitou-se aos participantes que preenchessem um questionário semelhante ao da 2ª iteração dos testes de usabilidade, com o objetivo de medir a usabilidade e experiência dos participantes a partir dos questionários SUS e UEQ.

Posteriormente, realizou-se uma breve entrevista com cada participante para aprofundar a sua experiência com a aplicação e identificar eventuais dificuldades encontradas. Além disso, avaliou-se o impacto dos elementos de gamificação presentes na aplicação móvel AQUA na motivação de cada participante.

6.2.2 Participantes

Em relação à distribuição dos participantes, nas empresas A, D e E, um único participante representou cada empresa no estudo, enquanto que na empresa B, dois participantes estiveram envolvidos. Os restantes três participantes faziam parte da empresa C.

Na tabela 15 é apresentada a caracterização de cada empresa em termos de localização, do número e tipo de embarcações, e do número de viagens diárias e de elementos de equipa por embarcação. As empresas B e C utilizam uma aplicação móvel própria para registar os avistamentos.

Qualquer membro disponível da equipa pode reportar os dados dos avistamentos. Assim, os participantes das empresas B e C usaram a sua própria aplicação móvel, e em seguida, repetiram o processo utilizando a aplicação móvel AQUA.

Tabela 15. Caracterização de empresas de observação de cetáceos

Empresa	Localização	Frota	Atividade Diária	Elementos de Equipa por Embarcação
A	Madeira	3 catamarãs	2 viagens por catamarã	5-10
B		1 semi-rígido	3 viagens	3
C		1 semi-rígido	3 viagens	3
D	Açores	1 catamarã, 5-semi-rígidos, 1 barco de fibra	2-3 viagens, variando conforme a embarcação	3
E	Madeira	2 semi-rígidos	3 viagens por embarcação	3

Na tabela 16 são apresentados os dados demográficos dos participantes envolvidos no estudo.

Tabela 16. Dados demográficos dos participantes do estudo

Demográfico	Grupo	Frequência	Percentagem
Género	Feminino	3	37.5%
	Masculino	5	62.5%
	Não indicado	0	0%
Idade	<18	0	0%
	18-24	1	12.5%
	25-34	5	62.5%
	35-44	1	12.5%
	45-54	1	12.5%
	+55	0	0%
Nacionalidade	Portuguesa (PT)	6	75%
	Outra	2	25%

6.2.3 Resultados

– Resultados Quantitativos:

A partir do questionário SUS, calculou-se o *SUS Score* obtido para cada participante, e, de seguida, calculou-se o valor médio do sistema, seguindo as instruções de [56]. Obteve-se um resultado de 82.5, numa escala de 0 a 100, indicando que a aplicação é “excelente” [57] em termos de usabilidade.

Os valores médios obtidos para cada escala avaliada utilizando o questionário UEQ — qualidade pragmática, qualidade hedônica, geral — foram: 0.929, 1.000 e 0.964, respetivamente.

– Resultados Qualitativos:

Para ilustrar os resultados, são citadas no texto algumas declarações dos participantes. As citações seguem o formato U<NúmeroDoParticipante>_<IDdaCitação>, e a lista completa de citações encontra-se no anexo C.3.3.

- **Empresa A:** Na empresa A, o participante n^o1 referiu utilizar com frequência uma aplicação móvel em desenvolvimento por outra entidade para o registo de avistamentos, mencionando que a mesma é confusa e propensa a erros no preenchimento de dados devido à complexidade do processo de reporte de avistamento utilizando a aplicação móvel (U1_1). Desta forma, considerou a aplicação AQUA bastante fluida e fácil de utilizar (U1_2), destacando a sua usabilidade e interface bastante apelativa e satisfatória, em comparação à outra aplicação móvel (U1_3).

Um ponto destacado pelo participante n^o1 foi a sua importância na simplicidade e rapidez da recolha dos dados, sendo a gamificação vista como um elemento interessante para jovens ou entusiastas de espécies marinhas (U1_4). Contudo, para os propósitos da empresa do participante e interesses pessoais, considerou que a gamificação não é prioritária, destacando a prioridade da eficiência na recolha completa dos dados dos avistamentos (U1_5).

Em relação aos elementos de gamificação, o participante n^o1 sugeriu que a raridade das espécies refletisse o grau de ameaça de extinção das mesmas, referindo que a baleia azul, por exemplo, é uma espécie considerada rara em certa época do ano na região da ilha da Madeira, mas comum em outra (U1_6). Considerou que todos os elementos de gamificação incluídos podem ser relevantes do ponto de vista do cliente que realiza as viagens para observação de cetáceos, permitindo que o mesmo possa aprender mais sobre as espécies, assim como ter uma maneira de registar a sua experiência (U1_7).

O participante n^o1 era responsável por comunicar através de um microfone, alternando entre inglês e português, para informar os clientes sobre os avistamentos e fornecer explicações sobre as espécies observada. Além disso, devido à grande dimensão do catamarã e ao elevado número de passageiros, o participante era também responsável por realizar o atendimento aos clientes no bar da embarcação.

Desta forma, o mesmo relatou ter tido dificuldade em reportar os avistamentos no momento exato, não conseguindo capturar imagens das observações, e optando, assim, por realizar o reporte dos avistamentos no final de cada viagem (U1_8).

Além disso, o participante n^o1 demonstrou um interesse da empresa em adotar uma aplicação semelhante à aplicação AQUA para os propósitos internos da empresa, referindo que é uma ferramenta útil para a recolha e análise dos dados dos avistamentos de cetáceos (U1_9).

- **Empresa B e C:** As empresas B e C utilizam a sua própria aplicação móvel para o registo dos avistamentos, estando qualquer membro da equipa responsável por fazer o registo. Desta forma, não houve um controlo uniforme sobre a utilização da aplicação AQUA por todos os participantes de ambas empresas durante o mesmo período de teste.

O membro da equipa que estivesse disponível realizou o reporte dos avistamentos com a aplicação própria, reportando-os novamente utilizando a aplicação AQUA. No entanto,

visto que o utilizador da aplicação não era sempre o mesmo, cada empresa registou-se na aplicação AQUA com uma única conta, comum para todos os participantes.

O participante nº2, da empresa B, achou a aplicação móvel AQUA complexa para uso profissional (U2_1). Ele relatou ter sentido dificuldades ao reportar avistamentos no momento exato, devido a responsabilidades com clientes e outras tarefas na embarcação (U2_2). A adição de gamificação na aplicação não impactou a sua motivação, referindo ter ignorado as recompensas por considerá-las irrelevantes para o objetivo principal de reportar avistamentos (U2_3).

Outro membro da empresa B, o participante nº3, elogiou o design atrativo da aplicação AQUA (U3_1). Comparando ambas aplicações móveis, ele considerou que a aplicação própria era mais complicada, já que não obtinha a localização do avistamento automaticamente. Era necessário utilizar uma aplicação externa para obter as coordenadas geográficas e inseri-las manualmente no formulário (U3_2). O participante considerou que a coleção de cartas de espécies poderia motivar os clientes a completar a sua coleção, mas do ponto de vista profissional, a gamificação acabava por atrapalhar as tarefas (U3_3). O mesmo pensamento aplicou-se nos distintivos, considerados importantes apenas para utilizadores não frequentes em viagens de observação de cetáceos (U3_4).

Em relação ao *ranking*, o participante considera que o mesmo pode gerar competição entre utilizadores, aumentando a pressão para avistar mais espécies, podendo impactar negativamente o ambiente marinho (U3_5).

Os participantes nº4 e nº5, da empresa C, não viram relevância nos elementos de gamificação no contexto profissional (U4_1, U5_1). O participante nº4 mencionou que o uso de duas aplicações diferentes para reportar o mesmo avistamento atrapalhava as suas tarefas na embarcação (U4_2). Ambos opinaram que os elementos de gamificação poderiam motivar entusiastas de observação de cetáceos, mas não eram relevantes para eles, que realizam viagens frequentemente em contexto de trabalho (U4_3, U5_2).

O participante nº6, também da empresa C, considerou a aplicação AQUA mais simples em comparação à aplicação móvel da empresa, referindo que o reporte de avistamento foi mais fácil e rápido (U6_1). Quanto aos elementos de gamificação, não o motivaram muito, embora tenha apreciado a representação visual das cartas de espécies e a diferenciação com base na raridade (U6_2).

- **Empresa D:** Na empresa D, que realiza duas viagens diárias para observação de cetáceos, o participante nº7 descreveu a aplicação móvel AQUA como altamente inovadora, útil e divertida (U7_1). Além disso, descreveu a sua experiência de utilização como muito agradável (U7_2).

O participante não enfrentou dificuldades ao reportar avistamentos devido às responsabilidades na embarcação. Ele ressaltou que após completar as suas tarefas relacionadas com o seu trabalho, reportou imediatamente os avistamentos para garantir a precisão das coordenadas (U7_3).

O mesmo achou particularmente motivador o desbloqueio de cartas de espécies raras e a conquista de distintivos (U7_4). Em relação ao *ranking*, considerou que o mesmo poderia

ser motivador se fosse possível garantir que não há manipulação por parte dos utilizadores (U7_5).

Uma sugestão do participante foi a incorporação de um sistema de níveis para cada espécie. Desta forma, incentivaria os utilizadores a continuar a reportar avistamentos de espécies comuns, que são mais facilmente obtidas. Assim, os utilizadores não deixariam de reportar avistamentos de espécies já desbloqueadas, por já terem obtido a carta da respetiva espécie anteriormente, e aumentaria o envolvimento dos mesmos em tentar alcançar níveis superiores para cada espécie (U7_6).

Além disso, propôs a ideia de fornecer um resumo anual das contribuições de cada utilizador ao final do ano, acreditando que o mesmo poderia aumentar a motivação dos utilizadores (U7_7).

- **Empresa E:** Na empresa E, que realiza três viagens diárias para observação de cetáceos, o participante nº8 expressou uma experiência extremamente positiva com a aplicação móvel AQUA, considerando-a muito fácil de usar (U8_1). No entanto, ele destacou alguns desafios em reportar os avistamentos no momento exato, devido às tarefas na embarcação (U8_2). Isso resultou num certo descontentamento, já que a precisão da localização do avistamento poderia ser afetada quando feita posteriormente.

Em relação aos elementos de gamificação, o participante relatou ter sentido uma enorme satisfação quando desbloqueou cartas de espécies raras e sentiu forte motivação para completar a coleção de cartas (U8_3). Também, ficou entusiasmado por ter alcançado uma posição de destaque (3º lugar) no *ranking* de utilizadores (U8_4).

Apesar de considerar os distintivos como elementos motivadores, o participante os classificou como os menos relevantes em comparação com a coleção de cartas e a o *ranking* de utilizadores (U8_5).

O participante referiu que recomendaria a aplicação AQUA a profissionais envolvidos em reportar avistamentos de cetáceos, devido à utilidade da ferramenta na simplificação da recolha dos dados, que atualmente é feita de forma manual (U8_6).

No entanto, embora tenha encontrado motivação na gamificação, ele acredita que esta teria um impacto mais positivo em pessoas envolvidas na observação de cetáceos por lazer e entretenimento, em vez de num contexto de trabalho (U8_7).

No que diz respeito a sugestões de melhoria, o participante propôs alterar a seleção do número de indivíduos durante o processo de reportar avistamentos. Ele sugeriu adicionar uma barra deslizante para permitir uma maior precisão na especificação do número de indivíduos, em vez de opções limitadas predefinidas (U8_8). Além disso, ele recomendou a adição de uma opção “Unknown” (Desconhecido) como opção de comportamento, uma vez que identificar o comportamento dos animais pode ser difícil em determinadas situações, e a opção “Other” pode ter múltiplos significados (U8_9).

Não se verificou a presença de nenhum participante que tenha escolhido um título para apresentá-lo no *ranking* de utilizadores da aplicação AQUA. Além disso, constatou-se que nenhum dos participantes utilizou a funcionalidade de *tracking* das rotas de viagem disponível na aplicação.

7 Discussão

Durante o processo de desenvolvimento do sistema AQUA, realizou-se diversos testes com o objetivo de avaliar diversos aspetos, incluindo a usabilidade, o envolvimento dos utilizadores e o funcionamento geral da aplicação móvel. Este desenvolvimento centrou-se nos utilizadores, considerando as suas necessidades e sugestões em cada iteração. Ao adotar um desenvolvimento ágil foi possível adaptar continuamente a aplicação AQUA com base no *feedback* dos utilizadores.

Os testes de usabilidade permitiram melhorar gradualmente da aplicação, tornando-a mais fácil de usar, agradável e intuitiva, tendo sido essenciais para o sucesso da avaliação do protótipo no contexto de uso real. Na primeira iteração dos testes (descrita na secção 3.3) observou-se que o redesenho da interface resultou em melhorias significativas na usabilidade, tornando-a mais acessível e intuitiva, principalmente no processo de reportar avistamentos. Esta fase permitiu melhorar os elementos de gamificação já existentes e identificar novas oportunidades, conforme sugerido pelos participantes, a fim de aumentar a motivação dos utilizadores e facilitar a visualização da posição atual do utilizador no ranking. Estas melhorias incluíram a adição de uma coleção de distintivos e a divisão do ranking em local e global. A introdução da coleção de distintivos permitiu uma exploração mais aprofundada dos princípios relacionados ao comportamento individual e à orientação pelo sistema, criando novos objetivos com diversos níveis de dificuldade. Por outro lado, a divisão do ranking permitiu explorar mais os princípios de comportamento social, através de comparações mais específicas entre os utilizadores.

Na segunda iteração dos testes de usabilidade (descrita na secção 6.1), verificou-se melhorias na usabilidade, embora ainda houvessem dificuldades na interação com certos elementos da interface, como na visualização das recompensas após o desbloqueio. Esta iteração confirmou que as melhorias propostas, como o redesenho do processo de reportar avistamentos e a adição e modificações nos elementos de gamificação, durante o processo de desenvolvimento tiveram um impacto positivo na experiência dos utilizadores, em comparação com os resultados da primeira iteração.

A primeira iteração demonstrou que o *redesign* do processo de reportar avistamento foi visto como uma melhoria significativa, tornando-o mais simples e rápido em comparação com a aplicação AQUA 1.0. Na segunda iteração, com base no *feedback* dos participantes que haviam usado a aplicação AQUA 1.0, consideraram que a mesma melhorou significativamente em termos de desempenho e usabilidade. Os participantes que participaram em ambas as iterações conseguiram visualizar a evolução da aplicação, percebendo as melhorias contínuas na sua usabilidade e funcionalidade.

Através dos testes realizados em terra, foi possível preparar o sistema para o seu funcionamento no contexto real. A avaliação do protótipo com profissionais das empresas de observação de cetáceos (descrita na secção 6.2) resultou num *feedback* bastante positivo em relação à aplicação AQUA e à experiência que ela proporciona. Além disso, é importante ressaltar que nenhum dos participantes dessa avaliação final relatou a ocorrência de *bugs* durante o período de teste em alto mar, e todos conseguiram utilizar a aplicação desde o primeiro contacto, sem necessidade de suporte. Desta forma, a avaliação final confirmou que os testes preliminares foram eficazes no sucesso do sistema, demonstrando a sua robustez, mesmo nas condições em alto mar.

- QI1: Quais são as funcionalidades que mais influenciam a satisfação dos utilizadores durante a experiência com a aplicação móvel AQUA?

A melhoria gradual da usabilidade teve um impacto significativo na funcionalidade de reportar avistamentos, que é a principal funcionalidade da aplicação e a mais utilizada pelos utilizadores. Desde o *redesign* da aplicação até à avaliação final, o processo de reportar avistamentos sofreu diversas alterações com o intuito de melhorá-lo. Esta funcionalidade é a mais utilizada pelos utilizadores, já que é responsável por interligar todas as restantes funcionalidades relacionadas com a gamificação. Sem contribuir com o reporte de avistamentos, um utilizador não consegue experimentar o potencial da aplicação AQUA, afetando assim a sua experiência em geral.

A partir da segunda iteração dos testes, foi possível verificar que o processo de reportar avistamentos foi otimizado, eliminando as dificuldades sentidas pelos participantes da primeira iteração. Por exemplo, na primeira iteração, os participantes reportaram um avistamento para cada espécie avistada e tentavam selecionar múltiplos comportamentos para uma mesma espécie. Estes problemas foram resolvidos posteriormente, o que fez com que não se voltassem a verificar durante as restantes fases de avaliação da aplicação.

A avaliação do protótipo no contexto real permitiu verificar que esta funcionalidade foi implementada de forma eficiente e garantindo que havia persistência nos dados recolhidos. Visto que em alto mar não há conexão à Internet, todos os participantes reportavam os seus avistamentos, sendo estes armazenados localmente para submissão posterior. Quando chegavam a terra e a conexão era estabelecida, os mesmos realizavam a submissão dos reportes no histórico de reportes, tendo-se verificado que a mesma era feita com sucesso, sem perda de informação.

Desta forma, o progresso do sistema permitiu garantir que a recolha dos dados é feita da melhor forma, sem necessidade de Internet para utilizar a aplicação AQUA. Além disso, é importante destacar que, em diferentes fases de teste, alguns participantes sugeriram adicionar novas funcionalidades na aplicação que pudessem melhorar a recolha dos dados dos avistamentos. Na segunda iteração, um participante sugeriu permitir que vários utilizadores colaborassem no reporte de um avistamento, garantindo uma recolha mais completa dos dados, já que várias pessoas contribuem para o preenchimento dos dados do mesmo avistamento. Por outro lado, na avaliação final, um participante recomendou modificar a forma como é feita a especificação do número de indivíduos e dos comportamentos de cada espécie avistada, permitindo que os dados sejam mais precisos.

Isto demonstra que, apesar de ter havido uma melhoria significativa na usabilidade do processo de reportar avistamentos, há diversos caminhos que podem ser tomados para explorar ainda mais esta funcionalidade e garantir que a mesma permite uma recolha mais rápida, completa e precisa dos dados.

Dado que a maioria dos testes realizados teve uma duração relativamente curta, os participantes não tiveram oportunidade de explorar a totalidade das funcionalidades da aplicação. Por exemplo, a funcionalidade de *tracking* das rotas de viagem não foi possível explorar devido ao local e tempo dos testes. No entanto, é importante salientar que, apesar de não ter sido usada durante nenhum teste de usabilidade ou avaliação no contexto real, os testes funcionais (descritos na secção 5.4) demonstraram que a mesma foi implementada de forma eficiente e garante uma utilização discreta sem afetar negativamente o desempenho do dispositivo. Ainda, estes testes permitiram detetar alguns problemas que foram resolvidos posteriormente, contribuindo para um melhor funcionamento da aplicação móvel.

Em relação à visualização do mapa de avistamentos, esta funcionalidade foi melhorada de acordo com o *feedback* obtido na segunda iteração dos testes, permitindo visualizar detalhadamente as informações dos avistamentos apresentados. Assim, os utilizadores podem identificar as zonas mais propícias ao avistamento de determinadas espécies.

Na avaliação do protótipo final com profissionais de observação de cetáceos, verificou-se que a empresa A já utilizou uma aplicação semelhante e que as empresas B e C possuem a sua própria aplicação móvel para reportar avistamentos, sugerindo que a aplicação AQUA pode ser uma ferramenta digital valiosa para estas empresas. Como era esperado, a avaliação revelou um interesse e necessidade dos participantes em adquirir uma solução semelhante que facilite a recolha e análise dos dados dos avistamentos, principalmente nos participantes que realizam o reporte dos avistamentos manualmente em papel, devido ao facto do processo ser mais demorado e exigir uma inserção digital dos dados posteriormente.

É importante mencionar que a experiência prévia com aplicações semelhantes pode ter gerado comparações frequentes entre a aplicação AQUA e as aplicações anteriores, potencialmente influenciando as perceções e avaliações dos participantes, visto que a correspondência entre as expectativas dos utilizadores baseadas em experiências passadas e a experiência real desempenham um papel significativo na formação de opiniões sobre a aplicação [64]. No entanto, em geral, todos esses participantes preferiram o *design* e usabilidade da aplicação AQUA, demonstrando que todo o esforço para melhorar a experiência dos utilizadores foi recompensado.

Em todas as avaliações, utilizou-se o questionário SUS para medir a usabilidade. A comparação dos resultados iniciais com a avaliação do protótipo final revelou um aumento no SUS *Score* (+2.5), sugerindo que as alterações feitas a cada iteração contribuíram para melhorar a usabilidade da aplicação móvel.

Além disso, na segunda iteração dos testes de usabilidade e na avaliação do protótipo final, utilizou-se o questionário UEQ para complementar a avaliação da experiência dos participantes. Os resultados reforçaram o *feedback* dos participantes, concluindo que a experiência de utilização é agradável e satisfatória para os utilizadores da aplicação AQUA.

Em suma, a funcionalidade que mais influenciou a satisfação dos utilizadores durante a utilização da aplicação móvel AQUA foi a sua funcionalidade principal, reportar avistamentos, tendo-se verificado que a mesma sofreu melhorias significativas ao longo de todo o processo. É importante destacar que as limitações temporais e de contexto de utilização não permitiram avaliar completamente a influência de cada funcionalidade na satisfação dos utilizadores. Ainda assim, acredita-se que ao melhorar o reporte dos avistamentos, toda a experiência foi beneficiada, uma vez que a maioria das funcionalidades da aplicação depende direta ou indiretamente dela.

- QI2: Quais os elementos de gamificação com potencial de envolver os utilizadores com a aplicação móvel AQUA?

A introdução de gamificação na aplicação AQUA teve como objetivo ultrapassar a desmotivação dos utilizadores, identificado em [28] e melhorar a experiência de utilização. Inicialmente, propôs-se a incorporação de dois elementos de gamificação: a coleção de cartas de espécies e o *ranking* de utilizadores.

Após a realização da primeira iteração dos testes, foram identificadas oportunidades de melhoria no *design* destes elementos. Estas modificações demonstraram ser eficazes ao facilitar a compre-

ensão da raridade das cartas e melhorar a visualização do *ranking*, ao dividi-lo em local e global e aumentando o número de utilizadores apresentados. A alteração da cor das cartas das espécies, com base em [58, 59], para facilitar a distinção da raridade, permitiu que posteriormente todos os utilizadores conseguissem identificar a raridade da espécie pela cor das cartas rapidamente. Além disso, nesta iteração, surgiu a ideia da inclusão da coleção de distintivos, que revelou ser considerado motivador para os participantes.

Na segunda iteração, todos os elementos de gamificação planeados estavam devidamente implementados na aplicação AQUA. Os utilizadores tiveram a oportunidade de explorar cada um desses elementos e fornecer *feedback* valioso sobre os mesmos. Ao avaliar a relevância de cada distintivo, identificou-se que, em geral, todos os distintivos são considerados relevantes pelos participantes, embora alguns tenham sido classificados como menos relevantes, como os relacionados com o reporte de determinado número de espécies (baleias e golfinhos) e o desbloqueio de 5 cartas de espécies comuns. Apesar de serem considerados menos relevantes, os valores médios obtidos são valores intermédios na escala. Assim, os resultados sugerem que os distintivos podem ter um papel importante no envolvimento dos utilizadores e na motivação de participar ativamente no reporte de avistamentos e incidentes.

Um participante considerou o *design* dos distintivos inconsistente com o restante *design* da aplicação. É importante notar que os ícones utilizados eram temporários e alguns foram usados para distintivos diferentes, visto que esta fase de testes tinha como objetivo avaliar a relevância dos distintivos na aplicação AQUA, não tendo sido aplicado um *design* definitivo nos distintivos.

Na avaliação final com os profissionais em observação de cetáceos, já com o *design* definitivo dos distintivos, os mesmos foram o elemento de gamificação considerado menos significativo. Os profissionais desbloquearam distintivos durante o período de utilização da aplicação AQUA, mas não proporcionaram o mesmo nível de satisfação que o desbloqueio das cartas de espécies, tendo apenas um participante destacado os distintivos como motivadores.

Embora tenham sido considerados menos significativos na avaliação final, é importante notar que este elemento foi sugerido pelos participantes da primeira iteração dos testes, e foram considerados relevantes para utilizadores não profissionais na segunda iteração.

No que diz respeito ao *ranking* de utilizadores, na segunda iteração dos testes, os participantes tentaram visualizar o perfil de outros utilizadores para comparar as contribuições. Esta funcionalidade foi adicionada posteriormente na versão final da aplicação, uma vez que os participantes consideraram que poderia ser bastante motivadora, permitindo competir com outros utilizador e ver as recompensas de cada um. A seleção de títulos também foi considerada relevante, tendo os participantes gostado de ter uma forma de se destacar em relação aos restantes no *ranking*, embora se encontrem numa posição inferior.

Embora a literatura [37] sugira que o *ranking* é um dos elementos mais eficazes para motivar os utilizadores, na avaliação final, não foi considerado relevante pelos participantes. Embora um participante ter expressado entusiasmo por ter atingido uma posição de destaque, outros manifestaram desconfiança em relação à validade dos dados apresentados. Além disso, nenhum dos participantes selecionou um título para apresentá-lo no *ranking*, reforçando a pouca relevância do mesmo para os utilizadores.

Para transmitir uma maior confiança aos utilizadores, recomenda-se implementar uma validação rigorosa dos dados reportados. Esta validação torna-se crucial devido à possibilidade de haverem

utilizadores inexperientes que possam ter dificuldade em reportar os avistamentos corretamente, bem como de haver manipulação dos dados para obter recompensas e alcançar posições mais altas no *ranking*.

Uma possível abordagem seria considerar apenas os reportes validados para o desbloqueio de recompensas. Entretanto, é importante reconhecer que a validação de dados pode ser uma tarefa complexa, especialmente ao lidar com uma grande quantidade de avistamentos não validados. Neste contexto, a criação de uma plataforma *web* dedicada à validação dos reportes poderia oferecer uma solução mais prática, em vez da mesma ser feita utilizando a aplicação móvel AQUA. Essa plataforma permitiria que a pessoa responsável pela validação obtivesse uma visão mais abrangente de todos os dados, simplificando a análise detalhada de cada avistamento e utilizador.

Além disso, a validação pode ser apenas necessária para novos utilizadores. Se um utilizador demonstrar consistência e integridade nos seus reportes ao longo do tempo, é possível ponderar validar automaticamente os seus reportes, reduzindo a carga de trabalho associada à validação e aumentando a confiança transmitida aos utilizadores em relação aos dados apresentados na aplicação móvel.

Na segunda iteração dos testes, alguns participantes sugeriram utilizar inteligência artificial para a validação dos dados, embora esta possa ser complexa, tendo em conta que as imagens capturadas geralmente mostram apenas partes do animal.

Quanto à coleção de cartas de espécies, a maioria dos participantes a considerou o elemento de gamificação mais relevante e a que demonstrou ser mais satisfatória para os utilizadores. No entanto, um participante da segunda iteração não viu utilidade nas cartas, sugerindo melhorá-las através da criação de *puzzles* que permitam aos utilizadores desbloquear novas recompensas ao combinar cartas desbloqueadas.

Na avaliação com profissionais de observação de cetáceos, embora apenas dois participantes tenham considerado este elemento motivador, os restantes consideraram que o mesmo pode ser interessante para entusiastas de observação de cetáceos. Além disso, vários participantes ressaltaram o *design* das cartas de espécies, assim como o facto de haver uma distinção por raridade.

No entanto, é importante realçar que a gamificação é um conceito muito amplo. Finckenhagen [65] ressalta que não se deve generalizar no desenho de sistemas gamificados, referindo que cada contexto é único, e a personalização é a chave para o sucesso.

No caso da aplicação AQUA, a mesma foi desenhada com base nas necessidades de indivíduos com pouco ou nenhum conhecimento em relação à biodiversidade marinha. Desta forma, os elementos de gamificação implementados na aplicação podem ter sido considerado pouco relevantes para profissionais, mas os mesmos podem ser considerados motivadores para outros utilizadores não profissionais, tal como se verificou com a coleção de distintivos.

Sugere-se, assim, que a aplicação AQUA seja adaptada de acordo com o tipo de utilizador (profissional ou não profissional). No entanto, é crucial realizar uma análise aprofundada e um estudo dos elementos de gamificação que mais poderiam motivar cada tipo de utilizador no contexto de observação de cetáceos. Posteriormente, recomenda-se incorporar cada elemento na aplicação para atender às necessidades e interesses de cada grupo de utilizadores.

Acredita-se que a utilização da aplicação AQUA num ambiente profissional tenha influenciado a experiência e o impacto da gamificação. Os participantes estavam concentrados nas suas tarefas,

colocando a aplicação móvel como uma tarefa secundária não prioritária. Além disso, o contexto e as características de cada empresa, como o fluxo de trabalho, número de embarcações e o número de viagens realizadas, podem ter influenciado os resultados obtidos no estudo.

Ainda, acredita-se que o mesmo motivo pode ter feito com que os participantes se limitassem a reportar os avistamentos durante as viagens, não chegando a explorar as restantes funcionalidades da aplicação.

De acordo com vários estudos [66,67], a implementação de gamificação em ambientes profissionais pode ter um impacto positivo na motivação dos trabalhadores, desde que seja cuidadosamente implementada, considerando as particularidades do ambiente de trabalho e as necessidades individuais de cada funcionário. No entanto, os benefícios demonstram ser evidentes quando a gamificação está integrada nas atividades inerentes às responsabilidades dos profissionais.

Embora a avaliação final tenha sido conduzida num contexto profissional, a utilização da aplicação AQUA representou ser uma tarefa adicional. Consequentemente, a gamificação não teve o mesmo impacto que poderia ter tido se, por exemplo, o processo de reportar de reportar avistamentos através da aplicação AQUA substituísse a tarefa dos profissionais de registar os avistamentos em papel ou através de aplicações móveis próprias da empresa.

Apesar de num número limitado de participantes terem sido obtidas opiniões semelhantes nalguns aspetos, considera-se necessário realizar um estudo com um número mais abrangente de participantes, visto que embora todos os participantes terem conhecimentos e interesse em relação à biodiversidade marinha, os gostos pessoais de cada participante podem ter influenciado a forma como a gamificação foi percebida.

A disponibilização da aplicação móvel ao público em geral pode permitir que a mesma seja testada por qualquer pessoa, num período indefinido e num contexto não controlado, ou seja, sem participação numa avaliação com um protocolo estabelecido. Assim, é possível obter uma maior representação dos participantes e tirar conclusões mais completas em relação ao impacto da gamificação na motivação dos utilizadores a longo prazo.

Com base nos resultados de todas as avaliações realizadas ao longo do processo, é possível concluir que a coleção de cartas de espécies demonstrou ser o elemento de gamificação com maior potencial de envolver os utilizadores da aplicação AQUA. Embora a coleção de distintivos e o *ranking* de utilizadores não terem demonstrado ser relevante para os profissionais de observação de cetáceos, os mesmos foram considerados relevantes para participantes não profissionais e fora do contexto profissional. Desta forma, acredita-se que é fundamental adaptar a aplicação às diferentes necessidades dos profissionais e entusiastas, além de ser necessário explorar ainda mais o impacto dos elementos de gamificação em utilizadores envolvidos na observação de cetáceos por lazer, dada a diferença entre os resultados obtidos entre profissionais e não profissionais.

- QI3: De que forma podemos aumentar a partilha dos dados recolhidos a partir da aplicação móvel AQUA para a monitorização de espécies de cetáceos

A aplicação móvel AQUA é uma ferramenta para a recolha de dados fundamentais no que diz respeito aos avistamentos de cetáceos. Esses dados têm um enorme potencial para compreender melhor as populações de cetáceos e, posteriormente, tomar medidas de conservação das mesmas no ecossistema marinho. No entanto, é importante destacar que a aplicação AQUA foi especificamente

desenvolvida para a recolha de dados de avistamentos de cetáceos, sendo desta forma desenvolvida uma API para permitir a máxima utilização destes dados.

A API desenvolvida serve como um ponte de ligação entre os dados recolhidos pela aplicação AQUA e outros sistemas externos que queiram utilizar essas informações para a monitorização das espécies marinhas. Este componente foi desenvolvido para permitir o acesso exclusivamente a utilizadores autorizados, garantindo que os dados recolhidos sobre os avistamentos estão protegidos e acessíveis apenas a sistemas cuja autorização seja atribuída. Através do desenvolvimento dos *endpoints* descritos no Apêndice D.3, a API permite simplificar a obtenção e tratamento de informações específicas.

Ao envolver profissionais de observação de cetáceos na avaliação da aplicação, foi possível dar início à construção de uma valiosa fonte de dados de avistamentos de cetáceos acessível para outros sistemas externos. Assim, acredita-se que, através da API desenvolvida, é possível aumentar a partilha dos dados recolhidos, podendo estes dados ser utilizados para realizar a monitorização das espécies, possibilitando a tomada de medidas de conservação e proteção das mesmas.

8 Limitações e Trabalho Futuro

Nesta secção são discutidas as limitações encontradas durante o processo de implementação e testes do sistema AQUA, juntamente com sugestões que poderiam melhorar o sistema no futuro. As limitações abordadas abrangem tanto aspetos técnicos quanto metodológicos da avaliação do protótipo realizado.

As seguintes limitações estão relacionadas com as tecnologias selecionadas para o desenvolvimento do sistema AQUA:

- **Sistema de *Hosting* Render:** Para a hospedagem da API desenvolvida e da base de dados *online*, utilizou-se a plataforma Render. O Render oferece uma variedade de planos de subscrição para a hospedagem de aplicações e serviços. Devido a limitações orçamentárias, optou-se pelo plano de subscrição gratuito. No entanto, é importante notar que o servidor utilizado para hospedar a API e a base de dados no plano gratuito apresentava diversas restrições em termos de desempenho e capacidade de armazenamento.

Durante os testes realizados, observou-se que, apesar das limitações do plano gratuito, o desempenho geral do sistema era aceitável. No entanto, foi identificado um atraso perceptível na obtenção das respostas do servidor.

Uma das limitações significativas do plano gratuito era a restrição de permitir apenas cinco acessos simultâneos. Esta limitação de escalabilidade é uma consideração importante, visto que a aplicação móvel AQUA é uma aplicação de ciência cidadã destinada a ser utilizada por diversos utilizadores simultaneamente.

Além disso, é relevante mencionar que o plano gratuito colocava o servidor em modo de espera caso não houvessem pedidos recebidos durante um período contínuo de 15 minutos. Quando um pedido era recebido enquanto o servidor estava em modo de espera, o servidor precisava de realizar uma reativação que levava cerca de 2 minutos. Desta forma, para manter o servidor ativo e não impactar os testes dos utilizadores, foi necessário realizar pedidos constantes à API;

- **Armazenamento de Imagens e Vídeos:** Como já mencionado anteriormente, durante a fase de implementação do sistema AQUA detetou-se uma limitação no PostgreSQL em relação ao armazenamento de imagens e vídeos. Para superar essa limitação, utilizou-se a plataforma Firebase para gerir o armazenamento *online* desse tipo de dados. Embora esta escolha tenha permitido ultrapassar esta limitação, ela introduziu mais complexidade no sistema AQUA, ao adicionar uma nova dependência que pode impactar a modificação do sistema no futuro.

Recomenda-se que, em trabalhos futuros, haja um aumento do plano de subscrição do sistema de *hosting* utilizado, de forma a aumentar a escalabilidade e disponibilidade do sistema AQUA.

Como já referido anteriormente, a realização da avaliação do protótipo final permitiu recolher dados valiosos em relação aos avistamentos de cetáceos, enriquecendo a base de dados *online*. Desta forma, foi possível dar início à construção de uma base de dados contendo informação relevantes sobre a ocorrência destas espécies, podendo ser acedido por sistemas externos autorizados para a monitorização da biodiversidade marinha.

Contudo, é importante realçar que, até a data, nenhum sistema externo realizou chamadas à API desenvolvida com o objetivo de aceder os dados recolhidos a partir da aplicação móvel

AQUA. Logo, mesmo que tenham sido realizados alguns testes na API, é importante avaliar o comportamento do sistema AQUA com outros sistemas externos.

Apesar do desenvolvimento da API ser uma das principais contribuições deste trabalho, limitações temporais não permitiram a realização de testes mais específicos na mesma, como, por exemplo, testes de desempenho e testes de disponibilidade. Assim, recomenda-se no futuro realizar estes testes de forma a avaliar se as qualidades do sistema são verificadas.

Além de limitações técnicas, identificou-se também algumas limitações relacionadas com os recursos disponíveis:

- **Testes em iOS:** Apesar da flexibilidade do Flutter em suportar o desenvolvimento para iOS e Android em simultâneo, não foi possível testar a aplicação móvel nesse sistema operativo devido à ausência de um dispositivo compatível para realizar os testes.
- **Testes em Alto Mar:** Devido à inexistência de uma embarcação própria e à dificuldade em realizar testes em alto mar, os testes tiveram que ser realizados em terra, resultando num ambiente consideravelmente distinto daquele em que a aplicação seria usada;
- **Imagens das Espécies:** A disponibilização de imagens das espécies na aplicação móvel também demonstrou ser uma limitação. Desta forma, houve necessidade de solicitar imagens das espécies a uma empresa de observação de cetáceos.

Como trabalho futuro, recomenda-se testar a aplicação AQUA no sistema operativo iOS de forma a verificar se a mesma é compatível. Desta forma, é possível identificar possíveis correções necessárias para adaptar a aplicação desenvolvida ao sistema operativo iOS, e permitir que os participantes de futuros testes possam utilizar o seu próprio dispositivo, não afetando assim os resultados dos testes.

Embora os testes tenham sido realizados em terra, a avaliação do protótipo final foi realizada em alto mar, onde não foi verificado nenhum problema no funcionamento da aplicação móvel. Contudo, é importante realizar mais testes em alto mar para identificar possíveis problemas que possam ocorrer num ambiente onde a conexão à Internet é instável.

As seguintes limitações metodológicas enfrentadas ao longo da avaliação do protótipo final são relevantes para a compreensão dos resultados e conclusões obtidas:

- **Limitações Temporais e Amostrais:** O número limitado de participantes contribuiu para uma quantidade muito reduzida de resultados, afetando a abrangência das conclusões da avaliação.

Além disso, a avaliação enfrentou uma limitação devido ao curto período de teste da aplicação móvel AQUA (uma semana). Esse tempo limitado pode ter impedido que os participantes explorassem completamente a aplicação móvel, incluindo todos os aspetos da gamificação e as suas possíveis influências na experiência. Uma avaliação mais prolongada teria permitido uma análise mais abrangente dos impactos da gamificação ao longo do tempo;

- **Leque de Utilizadores:** A avaliação realizada foi feita com participantes pertencentes a equipas de empresas de observação de cetáceos. Desta forma, houve uma limitação no leque de utilizadores, deixando em aberto a possibilidade de explorar a usabilidade da aplicação móvel e o impacto da gamificação entre públicos como crianças, entusiastas de cetáceos, e qualquer cidadão que não pertença a nenhuma dessas equipas;

- **Contexto de Utilização:** As empresas de observação de cetáceos recrutadas operam de maneiras distintas, em relação às tarefas e o número de viagens realizadas por dia, podendo ter criado cenários de teste distintos. Além disso, acredita-se que a utilização da aplicação AQUA em contexto profissional influenciou os resultados obtidos;
- **Recrutamento de Empresas:** O processo de recrutamento de empresas para participação na avaliação revelou-se desafiador, visto que a realização dos testes foi feita durante o verão, período de alta procura de realização de viagens de observação de cetáceos, limitando a disponibilidade das empresas em testar a aplicação móvel durante as suas viagens;
- **Métodos de Recolha de Dados:** Acredita-se que, possivelmente, o facto dos participantes estarem cientes da sua participação num estudo que envolvia responder a questionários e à realização de entrevistas, possa ter influenciado também os resultados;
- **Dispositivos:** Alguns participantes precisaram de utilizar um dispositivo fornecido pela equipa, fazendo com que tivessem de se adaptar a utilizar um novo dispositivo, com um sistema operativo distinto na maioria dos casos. Acredita-se que o processo de adaptação pode ter afetado a experiência de utilização.

Com base nas limitações apresentadas, conclui-se que a avaliação realizada com as empresas de observação de cetáceos foi influenciado por diversos fatores que, por sua vez, limitaram os resultados e conclusões obtidos. Como trabalho futuro, propõem-se realizar uma avaliação durante um período mais longo, contemplando um maior número de participantes e um leque mais completo de utilizadores, incluindo aqueles sem nenhum vínculo direto a empresas de observação de cetáceos. Assim, acredita-se que será possível realizar uma identificação mais precisa da influência de cada elemento de gamificação na motivação de cada grupo de utilizadores.

Propõe-se, também, incluir métricas diretas na aplicação móvel, capazes de medir o envolvimento e a motivação dos participantes, facilitando a recolha dos dados. Entre essas métricas, sugere-se a medição do tempo total de utilização e a frequência de inicialização da aplicação por parte do utilizador, assim como solicitar avaliações durante a interação com a aplicação.

Considera-se importante explorar novos elementos de gamificação, já que a realização dos testes de usabilidade e da avaliação do protótipo final permitiu identificar novos elementos que os participantes consideram que podem ser motivadores e tornar a experiência de utilização da aplicação móvel AQUA mais envolvente. Esta exploração torna-se relevante também visto que alguns elementos de gamificação podem ser mais relevantes para uns utilizadores e pouco relevantes para outro.

Apesar das limitações identificadas, foi possível identificar aspetos para melhorar o sistema AQUA e continuar a explorar mecanismos de gamificação para envolver os utilizadores na recolha de dados sobre as espécies marinhas.

9 Conclusão

A presente dissertação teve como objetivo melhorar a experiência dos utilizadores e o desempenho da aplicação AQUA, através do redesenho da interface de utilizador, a integração de elementos de gamificação, e da reimplementação técnica do sistema.

Ao longo do processo, a realização dos testes de usabilidade e dos testes funcionais desempenharam um papel crucial no sucesso do sistema, evidenciando uma notável evolução na experiência dos utilizadores e na funcionalidade da aplicação.

O desenvolvimento de uma API do sistema AQUA tem um enorme potencial para facilitar a partilha dos dados dos avistamentos com sistemas externos, aumentando a disponibilidade destes dados para monitorização de cetáceos.

Os elementos de gamificação incorporados na aplicação AQUA revelaram ser mais eficazes no aumento da motivação dos utilizadores que não trabalham profissionalmente na observação de cetáceos, destacando a coleção de cartas de espécies.

A realização da avaliação final em colaboração com empresas de observação de cetáceos demonstrou que esta plataforma tem um enorme potencial para a ciência, embora a gamificação não tenha sido considerada relevante durante o contexto profissional. No entanto, embora determinar o impacto a longo prazo destes elementos tenha sido um desafio devido às restrições temporais inerentes a uma dissertação de mestrado, as conclusões obtidas fornecem uma base para futuras investigações nesta área.

As contribuições deste trabalho, que incluíram o desenho e desenvolvimento do sistema AQUA, composto por uma aplicação de ciência cidadã gamificada dedicada à recolha de dados sobre avistamentos de cetáceos, e por uma API para partilha desses dados com sistemas externos, além da avaliação do impacto da gamificação no envolvimento e experiência dos utilizadores, esperam inspirar novas questões e direções para trabalhos futuros.

Acredita-se que a continuação do desenvolvimento e exploração do sistema desenvolvido tem um enorme potencial de unir a ciência cidadã, a tecnologia e a ciência na monitorização da biodiversidade marinha para compreender e proteger as espécies de cetáceos que habitam nos nossos oceanos.

Referências

- [1] Ccm, “Espécies aquáticas ameaçadas em portugal,” acessado a 2022-08-23. [Online]. Available: https://ccm.marinha.pt/pt/aquariovgama_web/aprendermais_web/conservacao_web/Paginas/especiesaquaticasameacadasemportugal.aspx
- [2] “World cetacean alliance.” [Online]. Available: <https://worldcetaceanalliance.org/who-we-are/why-cetaceans/>
- [3] R. C. an assistant director in the IMF’s Institute for Capacity Development., T. C. is professoremeritus at the University of Notre Dame’s Mendoza College of Business., C. F. is professor of the practice of economics, director of undergraduate studies in the Duke University Economics Department., and S. O. is a research analyst in the IMF’s Institute for Capacity Development., “Nature’s solution to climate change – imf,” Dec 2019. [Online]. Available: <https://www.imf.org/en/Publications/fandd/issues/2019/12/natures-solution-to-climate-change-chami>
- [4] M. Santos-Carvalho, F. Barilari, M. J. Pérez-Alvarez, L. Gutiérrez, G. Pavez, H. Araya, C. Anguita, C. Cerda, and M. Sepúlveda, “Impacts of whale-watching on the short-term behavior of fin whales (*balaenoptera physalus*) in a marine protected area in the southeastern pacific,” *Frontiers in Marine Science*, vol. 8, 2021.
- [5] “Milhares têm procurado atividades marítimo-turísticas na madeira (com vídeo),” Jul 2022, acessado a 2022-08-20. [Online]. Available: https://www.jm-madeira.pt/regiao/ver/178510/Milhares_tem_procurado_atividades_maritimo-turisticas_na_Madeira_com_video
- [6] “Whale watching - management and conservation,” acessado a 2022-08-24. [Online]. Available: <https://iwc.int/management-and-conservation/whalewatching>
- [7] “Responsible whale watching,” Sep 2022, acessado a 2022-08-03. [Online]. Available: <https://www.futurismo.pt/responsible-whale-watching-guidelines/>
- [8] “The benefits and impacts of whale watching,” Nov 2022, acessado a 2022-08-24. [Online]. Available: <https://wwhandbook.iwc.int/en/responsible-management/benefits-and-impacts-of-whale-watching>
- [9] “Contribución a la ciencia y la conservación,” Nov 2022, acessado a 2022-08-25. [Online]. Available: <https://wwhandbook.iwc.int/en/preparing-for-a-trip/contributing-to-science-and-conservation>
- [10] “Citizen science,” acessado a 2022-09-08. [Online]. Available: <https://education.nationalgeographic.org/resource/citizen-science>
- [11] “What is citizen science?” acessado a 2022-09-09. [Online]. Available: <https://www.nps.gov/subjects/citizenscience/citizen-science.htm>
- [12] D. Rotman, J. Preece, J. Hammock, K. Procita, D. Hansen, C. Parr, D. Lewis, and D. Jacobs, “Dynamic changes in motivation in collaborative citizen-science projects,” in *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, ser. CSCW ’12. New York, NY, USA: Association for Computing Machinery, 2012, p. 217–226.

- [13] M. AL-Smadi, “Gameducation: Using game mechanics and dynamics to enhance online learning,” 11 2014.
- [14] “Intertagua - proyecto,” acedido a 2022-09-18. [Online]. Available: <https://intertagua.eu/proyecto/>
- [15] M. Mendes and D. Nunes, “Dissertação: A mobile platform for enhancing biodiversity monitoring with citizen engagement,” acedido a 2022-09-12. [Online]. Available: <https://fenix.tecnico.ulisboa.pt/cursos/meic-a/dissertacao/565303595502787>
- [16] M.-C. Yuen, I. King, and K. Leung, “A survey of crowdsourcing systems,” 10 2011, pp. 766–773.
- [17] “10 benefits of citizen science,” acedido a 2022-09-08. [Online]. Available: <https://magazines.rivm.nl/en/2018/10/citizen-science/10-benefits-citizen-science>
- [18] “Report invasive plants - apps on google play,” acedido a 2022-08-14. [Online]. Available: https://play.google.com/store/apps/details?id=com.mobanode.invasivespecies&hl=en_US&gl=US
- [19] “Seek by inaturalist · inaturalist,” acedido a 2022-08-14. [Online]. Available: https://www.inaturalist.org/pages/seek_app
- [20] “Iwdg reporting app,” acedido a 2022-08-15. [Online]. Available: <https://iwdg.ie/iwdg-reporting-app/>
- [21] “Iwdg reporting app - apps on google play,” acedido a 2022-08-15. [Online]. Available: <https://play.google.com/store/apps/details?id=ie.iwdg.reportingapp>
- [22] “Irish sailing news,” acedido a 2022-08-14. [Online]. Available: <https://www.sailing.ie/News/id/3743/iwdg-reporting-app-launched>
- [23] “White shark tracking app - awsc,” acedido a 2022-08-14. [Online]. Available: <https://www.atlanticwhiteshark.org/sharktivity-app>
- [24] Conserve.IO, “Sharktivity,” Jun 2016, acedido a 2022-08-14. [Online]. Available: <https://apps.apple.com/us/app/sharktivity/id1097933510>
- [25] “Maui dolphin; whale tracker map.” [Online]. Available: <https://www.pacificwhale.org/research/hawaii-wdt-sightings-map/>
- [26] P. W. Foundation, “Whale and dolphin tracker,” acedido a 2022-08-14. [Online]. Available: <https://apps.apple.com/pt/app/whale-and-dolphin-tracker/id1336754335>
- [27] S. User, “Seafari - introduction,” acedido a 2022-08-14. [Online]. Available: <http://www.seafariapp.org/>
- [28] M. Dionisio, M. Mendes, M. Fernandez, V. Nisi, and N. Nunes, “Aqua: Leveraging citizen science to enhance whale-watching activities and promote marine-biodiversity awareness,” *Sustainability*, vol. 14, no. 21, 2022.
- [29] J. Chen, “Flow in games (and everything else),” vol. 50, no. 4, p. 31–34, apr 2007.
- [30] B. Morschheuser, J. Hamari, and J. Koivisto, “Gamification in crowdsourcing: A review,” in *Proceedings of the 2016 49th Hawaii International Conference on System Sciences (HICSS)*, ser. HICSS ’16. USA: IEEE Computer Society, 2016, p. 4375–4384.

- [31] B. Morschheuser, J. Hamari, J. Koivisto, and A. Maedche, “Gamified crowdsourcing: Conceptualization, literature review, and future agenda,” *International Journal of Human-Computer Studies*, vol. 106, pp. 26–43, 2017.
- [32] M. Swanson, “Extrinsic vs. intrinsic motivation in gamification marketing.” [Online]. Available: <https://www.gamify.com/gamification-blog/extrinsic-vs.-intrinsic-motivation-in-gamification-marketing>
- [33] Olivieri and Olivieri, “A motivação intrínseca, os gamers e a gamificação,” Mar 2013. [Online]. Available: <https://opusphere.com/a-motivacao-intrinseca-os-gamers-e-a-gamificacao/>
- [34] E. Bontempi, “Intrinsic and extrinsic motivation: Implications in school, work, and psychological well-being,” Jul 2022. [Online]. Available: <https://www.excelsior.edu/article/types-of-motivation/>
- [35] P. Herzig, M. Ameling, B. Wolf, and A. Schill, *Gamification in Education and Business*, 01 2015, pp. 431–450.
- [36] G. Zichermann and C. Cunningham, *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps*, 1st ed. O’Reilly Media, Inc., 2011.
- [37] E. D. Mekler, F. Brühlmann, K. Opwis, and A. N. Tuch, “Do points, levels and leaderboards harm intrinsic motivation? an empirical analysis of common gamification elements,” in *Proceedings of the First International Conference on Gameful Design, Research, and Applications*, ser. Gamification ’13. New York, NY, USA: Association for Computing Machinery, 2013, p. 66–73.
- [38] J. Krath and H. von Kortzfleisch, “Designing gamification and persuasive systems: a systematic literature review,” 04 2021.
- [39] “Duolingo.” [Online]. Available: <https://www.duolingo.com/>
- [40] J. B. J. 2, J. Bilham, B. Bilham, B. Bhutani, and A. Bhutani, “Case study: How duolingo utilises gamification to increase user interest,” Jul 2021. [Online]. Available: <https://raw.studio/blog/how-duolingo-utilises-gamification/>
- [41] N. Turner, “How fitbit have nailed gamification: 10 lessons you can learn from them,” May 2023. [Online]. Available: <https://www.uxforthemasses.com/10-gamification-lessons/>
- [42] S. Singh, “Native vs hybrid vs cross platform - what to choose in 2023?” Dec 2022, acessado a 2022-10-14. [Online]. Available: <https://www.netsolutions.com/insights/native-vs-hybrid-vs-cross-platform/>
- [43] “Native vs cross-platform development: Pros and cons revealed,” acessado a 2022-10-14. [Online]. Available: <https://www.uptech.team/blog/native-vs-cross-platform-app-development>
- [44] S. Kidecha, S. Kidecha, J. 27, and Like, “Native vs hybrid vs cross-platform - dzone,” Jul 2022, acessado a 2022-10-14. [Online]. Available: <https://dzone.com/articles/native-vs-hybrid-vs-cross-platform-how-and-what-to>
- [45] “Native vs hybrid vs cross-platform - what to choose and when?” Oct 2021, acessado a 2022-10-14. [Online]. Available: <https://etraverse.com/blog/native-vs-hybrid-vs-cross-platform-what-to-choose-and-when/>

- [46] “Build apps for any screen,” acedido a 2022-09-18. [Online]. Available: <https://flutter.dev/>
- [47] “Layouts in flutter.” [Online]. Available: <https://docs.flutter.dev/ui/layout>
- [48] “Top 7 frameworks for backend development,” Jun 2022, acedido a 2022-10-15. [Online]. Available: <https://www.geeksforgeeks.org/top-7-frameworks-for-backend-development/>
- [49] Admin-Intelvision, “Node.js vs .net core: What to choose in 2022,” Jul 2022, acedido a 2022-10-15. [Online]. Available: <https://intelvision.pro/blog/node-js-vs-net-core-what-to-choose-in-2022/>
- [50] Javinpaul, “10 best backend frameworks for web development in 2022,” Mar 2022, acedido a 2022-10-15. [Online]. Available: <https://medium.com/javarevisited/10-best-backend-frameworks-for-web-development-8d19e337f774>
- [51] Maitray-Gadhavi, “Best backend frameworks for backend development in 2023,” Oct 2022, acedido a 2022-10-15. [Online]. Available: <https://radixweb.com/blog/best-backend-frameworks>
- [52] C. Gor, “Django vs laravel vs node.js: Which one you should opt for in 2022?” Jul 2022, acedido a 2022-10-15. [Online]. Available: <https://www.esparkinfo.com/blog/laravel-vs-django-vs-nodejs.html>
- [53] “Top 10 backend frameworks: Which is the best option for you?” Dec 2022, acedido a 2022-10-15. [Online]. Available: <https://blog.back4app.com/backend-frameworks/>
- [54] D. Sagita, “Using database in asp.net core,” Feb 2019, acedido a 2022-11-11. [Online]. Available: <https://medium.com/@daniel.sagita/using-database-in-asp-net-core-f69f99048bb>
- [55] S. Bhatia, “Postgresql vs mysql: Everything you need to know in 2022,” acedido a 2022-11-11. [Online]. Available: <https://hackr.io/blog/postgresql-vs-mysql>
- [56] W. T, “Measuring and interpreting system usability scale (sus),” Feb 2021. [Online]. Available: <https://uiuxtrend.com/measuring-system-usability-scale-sus/>
- [57] A. Bangor, P. Kortum, and J. Miller, “Determining what individual sus scores mean: Adding an adjective rating scale,” *J. Usability Studies*, vol. 4, no. 3, p. 114–123, may 2009.
- [58] “Color coded loot.” [Online]. Available: <https://www.giantbomb.com/color-coded-loot/3015-4702/>
- [59] K. Memmott, “Blue? green? gold? enough is enough - i want games to decide on a rarity system,” Jun 2021. [Online]. Available: <https://www.ggrecon.com/articles/i-want-games-to-decide-on-a-rarity-system/>
- [60] K. Wiegiers, “Exploring quality attribute requirements,” Jul 2014, acedido a 2022-08-05. [Online]. Available: <https://www.batimes.com/articles/exploring-quality-attribute-requirements/>
- [61] Jun 2023. [Online]. Available: https://pt.wikipedia.org/wiki/F%C3%B3rmula_de_haversine
- [62] R. T. Fielding and J. Reschke, “Rfc 7231: Hypertext transfer protocol (http/1.1): Semantics and content,” Jun 2014. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc7231#section-6.5.1>

- [63] D. Erhabor, “Acid transactions and implementation in a postgresql database - aviator blog,” Mar 2023. [Online]. Available: <https://www.aviator.co/blog/acid-transactions-postgresql-database/>
- [64] “Expectation confirmation theory,” Mar 2023. [Online]. Available: https://en.wikipedia.org/wiki/Expectation_confirmation_theory
- [65] K. R. Finckenhagen, “Contextual factors and successful gamification,” 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:211115422>
- [66] A. Aziz, A. Mushtaq, and M. Anwar, “Usage of gamification in enterprise: A review,” 03 2017, pp. 249–252.
- [67] C. Perryer, N. Celestine, B. Scott-Ladd, and C. Leighton, “Enhancing workplace motivation through gamification: Transferrable lessons from pedagogy,” *The International Journal of Management Education*, vol. 14, p. 327–335, 07 2016.

A Apêndice - Tabelas

Cada caso de uso principal é descrito da seguinte forma:

- **UCx:** nomenclatura utilizada para distinguir e referenciar cada caso de uso, onde x representa o identificar único atribuído ao caso de uso.
- **Ator Principal:** o utilizador ou sistema externo que interage com o sistema no caso de uso.
- **Objetivo:** descreve a finalidade que o ator principal deseja alcançar ao realizar o caso de uso.
- **Pré-Condições:** as condições ou situações que se devem verificar para que o caso de uso possa ser iniciado.
- **Pós-Condições:** as condições ou situações que se devem verificar após o sucesso da conclusão do caso de uso.
- **Cenário Principal:** a sequência de passos que descrevem o fluxo da interação entre o ator principal e o sistema no caso de uso.
- **Cenário Alternativo (Opcional):** a sequência de passos que descrevem o fluxo da interação alternativo quando algo fora do fluxo normal, ou seja, do cenário principal, se verifica.
- **Extensões (Opcional):** as condições ou situações que podem alterar o fluxo no caso de uso.

Tabela 17. Descrição Detalhada do Caso de Uso UC1

UC1: Registrar Utilizador

Ator Principal: Utilizador

Objetivo: Permitir que o utilizador crie uma conta na aplicação.

Pré-Condições:

- O utilizador deve ter acesso à aplicação.
- O utilizador deve ter acesso à internet.

Pós-Condições:

- O utilizador terá uma conta na aplicação e poderá autenticar-se para aceder às funcionalidades disponíveis.

Cenário Principal:

1. O utilizador seleciona a opção de “Sign Up”.
2. O sistema apresenta ao utilizador um formulário de registo para o utilizador preencher.
3. O utilizador preenche as informações necessárias para criar a conta, incluindo o seu email, password e confirmação de password.
4. O utilizador confirma as informações inseridas e seleciona a opção “Continue”.
5. O sistema verifica se as informações são válidas.
6. O sistema apresenta ao utilizador um formulário para o utilizador preencher o seu nome, país e cidade.
7. O utilizador preenche a informação solicitada e seleciona a opção “Continue”.
8. O sistema apresenta ao utilizador a opção de adicionar uma foto e uma descrição de perfil.
9. O utilizador carrega uma foto de perfil, escreve uma descrição para o seu perfil e seleciona a opção “Continue”.
10. O sistema cria a conta para o utilizador e redireciona-o para o ecrã de login.

Cenários Alternativos:

9. O utilizador seleciona a opção “Skip”.
10. O sistema cria a conta para o utilizador e redireciona-o para o ecrã de login.

Extensões:

- 5a. Em caso de informações em falta: O sistema apresenta uma mensagem de erro ao utilizador e solicita que o mesmo preencha todos os campos em falta.
- 5b. Em caso de email inválido: O sistema apresenta uma mensagem de erro ao utilizador indicando que o email inserido é inválido.
- 5c. Em caso de passwords não correspondentes: O sistema apresenta uma mensagem de erro ao utilizador indicando que a password e a confirmação de password não são iguais.
- 10a. Em caso de não haver conexão à Internet: O sistema apresenta uma mensagem de erro ao utilizador indicando que não foi possível criar uma conta devido à falta de conexão à Internet.

Tabela 18. Descrição Detalhada do Caso de Uso UC2**UC2:** Autenticar Utilizador**Ator Principal:** Utilizador**Objetivo:** Permitir que o utilizador se autentique na aplicação e aceda às funcionalidades disponíveis.**Pré-Condições:**

- O utilizador deve ter acesso à aplicação.
- O utilizador deve ter acesso à Internet.
- O utilizador deve ter uma conta na aplicação.

Pós-Condições:

- O utilizador terá acesso às funcionalidades disponíveis na aplicação.

Cenário Principal:

1. O utilizador seleciona a opção de “Login”.
2. O sistema apresenta ao utilizador um formulário de login para o utilizador preencher.
3. O utilizador preenche as informações necessárias para efetuar login, incluindo o seu email e password.
4. O utilizador confirma as informações inseridas e seleciona a opção “Login”.
5. O sistema verifica se as informações são válidas.
6. O sistema apresenta uma mensagem de sucesso e redireciona o utilizador para o ecrã principal.

Extensões:

- 5a.** Em caso de informações em falta: O sistema apresenta uma mensagem de erro ao utilizador e solicita que o mesmo preencha todos os campos em falta.
- 5b.** Em caso de email inválido: O sistema apresenta uma mensagem de erro ao utilizador indicando que o email inserido não está registado na aplicação.
- 5c.** Em caso de dados inválidos: O sistema apresenta uma mensagem de erro ao utilizador indicando que os dados inseridos são inválidos.
- 6a.** Em caso de não haver conexão à Internet: O sistema apresenta uma mensagem de erro ao utilizador indicando que não foi possível efetuar login devido à falta de conexão à Internet.

Tabela 19. Descrição Detalhada do Caso de Uso UC3

UC3: Excluir Conta

Ator Principal: Utilizador

Objetivo: Permitir que o utilizador exclua a sua conta.

Pré-Condições:

- O utilizador deve ter acesso à aplicação.
- O utilizador deve ter acesso à Internet.
- O utilizador deve ter login efetuado na aplicação.

Pós-Condições:

- O utilizador deixará de ter acesso à aplicação a partir das credenciais da conta com login efetuado.

Cenário Principal:

1. O utilizador seleciona a opção “Delete account”.
2. O sistema apresenta ao utilizador uma mensagem de confirmação de exclusão da conta, indicando que a ação é irreversível.
3. O utilizador seleciona a opção “Confirm”.
4. O sistema apaga a conta do utilizador, termina a sua sessão, apresenta uma mensagem de sucesso e redireciona-o para o ecrã de criação de contas.

Cenários Alternativos:

3. O utilizador seleciona a opção “Cancel”.
4. O sistema volta ao ecrã anterior e permite que o utilizador continue a utilizar a aplicação normalmente.

Extensões:

- 4a. Em caso de não haver conexão à Internet: O sistema apresenta uma mensagem de erro ao utilizador indicando que não foi possível excluir a sua conta devido à falta de conexão à Internet.

Tabela 20. Descrição Detalhada do Caso de Uso UC4**UC4:** Reportar Avistamento**Ator Principal:** Utilizador**Objetivo:** Permitir que o utilizador reporte um avistamento.**Pré-Condições:**

- O utilizador deve ter acesso à aplicação.
- O utilizador deve ter login efetuado na aplicação.
- O utilizador deve ter permitido o acesso ao GPS do dispositivo.
- O utilizador deve ter permitido o acesso à câmara do dispositivo.

Pós-Condições:

- O utilizador reporta um avistamento.
- O avistamento é armazenado localmente em caso de falha de envio para o servidor.

Cenário Principal:

1. O utilizador seleciona a opção “Sighting Report”.
2. O sistema apresenta ao utilizador as opções de capturar fotografias ou vídeos das espécies avistadas.
3. O utilizador seleciona uma opção e captura as imagens.
4. O sistema apresenta ao utilizador as fotografias e vídeos capturados.
5. O utilizador seleciona a opção “Continue”.
6. O sistema solicita ao utilizador a identificação das espécies avistadas.
7. O sistema apresenta ao utilizador uma lista com as cinco espécies mais avistadas pelos utilizadores.
8. O utilizador seleciona as espécies avistadas e clica em “Continue”.
9. O sistema solicita ao utilizador a indicação do número de indivíduos avistados para cada espécie.
10. O utilizador seleciona uma opção para cada espécie e clica em “Continue”.
11. O sistema solicita ao utilizador a indicação dos comportamentos dos indivíduos avistados para espécie.
12. O utilizador seleciona todos os comportamentos para cada espécie e clica em “Continue”.
13. O sistema apresenta ao utilizador um formulário para adicionar um comentário geral do avistamento.
14. O utilizador escreve um comentário e clica em “Continue”.
15. O sistema apresenta um sumário do avistamento com toda a informação preenchida pelo utilizador, incluindo um mapa com as coordenadas, e a data e hora do avistamento.
16. O utilizador confirma que a informação do avistamento está correta e clica em “Submit Sighting”.
17. O sistema envia o avistamento para o servidor.
18. O sistema apresenta uma mensagem de sucesso ao utilizador e redireciona-o para o ecrã inicial.

Extensões:

- 17a. Em caso de não haver conexão à Internet: O sistema apresenta uma mensagem indicando que o avistamento foi armazenado localmente.

Tabela 21. Descrição Detalhada do Caso de Uso UC5

UC5: Reportar Incidente do Tipo “Dead Animal” ou “Injured Animal”

Ator Principal: Utilizador

Objetivo: Permitir que o utilizador reporte um incidente do tipo “Dead Animal” ou “Injured Animal”.

Pré-Condições:

- O utilizador deve ter acesso à aplicação.
- O utilizador deve ter login efetuado na aplicação.
- O utilizador deve ter permitido o acesso ao GPS do dispositivo.
- O utilizador deve ter permitido o acesso à câmara do dispositivo.

Pós-Condições:

- O utilizador reporta um incidente do tipo “Dead Animal” ou “Injured Animal”.
- O incidente é armazenado localmente em caso de falha de envio para o servidor.

Cenário Principal:

1. O utilizador seleciona a opção “Seen an Incident?”.
2. O sistema solicita ao utilizador a seleção da categoria do incidente.
3. O utilizador seleciona a opção “Dead Animal” ou “Injured Animal” e clica em “Continue”.
4. O sistema apresenta ao utilizador as opções de capturar fotografias ou vídeos das espécies avistadas.
5. O utilizador seleciona uma opção e captura as imagens.
6. O sistema apresenta ao utilizador as fotografias e vídeos capturados.
7. O utilizador seleciona a opção “Continue”.
8. O sistema solicita ao utilizador a identificação das espécies envolvidas no incidente.
9. O sistema apresenta ao utilizador uma lista com as cinco espécies mais avistadas pelos utilizadores.
10. O utilizador seleciona as espécies envolvidas e clica em “Continue”.
11. O sistema solicita ao utilizador a indicação do número de indivíduos envolvidos para cada espécie.
12. O utilizador seleciona uma opção para cada espécie e clica em “Continue”.
13. O sistema solicita ao utilizador a indicação dos comportamentos dos indivíduos envolvidos para espécie.
14. O utilizador seleciona todos os comportamentos para cada espécie e clica em “Continue”.
15. O sistema apresenta ao utilizador um formulário para adicionar um comentário geral do incidente.
16. O utilizador escreve um comentário e clica em “Continue”.
17. O sistema apresenta um sumário do incidente com toda a informação preenchida pelo utilizador, incluindo um mapa com as coordenadas, e a data e hora do incidente.
18. O utilizador confirma que a informação do avistamento está correta e clica em “Submit Incident”.
19. O sistema envia o incidente para o servidor.
20. O sistema apresenta uma mensagem de sucesso ao utilizador e redireciona-o para o ecrã inicial.

Extensões:

- 19a.** Em caso de não haver conexão à Internet: O sistema apresenta uma mensagem indicando que o incidente foi armazenado localmente.

Tabela 22. Descrição Detalhada do Caso de Uso UC6

UC6: Reportar Incidente do Tipo “Other”

Ator Principal: Utilizador

Objetivo: Permitir que o utilizador reporte um incidente do tipo “Other”.

Pré-Condições:

- O utilizador deve ter acesso à aplicação.
- O utilizador deve ter login efetuado na aplicação.
- O utilizador deve ter permitido o acesso ao GPS do dispositivo.
- O utilizador deve ter permitido o acesso à câmara do dispositivo.

Pós-Condições:

- O utilizador reporta um incidente do tipo "Other".
- O incidente é armazenado localmente em caso de falha de envio para o servidor.

Cenário Principal:

1. O utilizador seleciona a opção “Seen an Incident?”.
2. O sistema solicita ao utilizador a seleção da categoria do incidente.
3. O utilizador seleciona a opção “Other” e clica em “Continue”.
4. O sistema apresenta ao utilizador as opções de capturar fotografias ou vídeos das espécies avistadas.
5. O utilizador seleciona uma opção e captura as imagens.
6. O sistema apresenta ao utilizador as fotografias e vídeos capturados.
7. O utilizador seleciona a opção “Continue”.
8. O sistema apresenta ao utilizador um formulário para descrever o incidente.
9. O utilizador escreve a descrição do incidente e clica em “Continue”.
10. O sistema apresenta um sumário do incidente com toda a informação preenchida pelo utilizador, incluindo um mapa com as coordenadas, e a data e hora do incidente.
11. O utilizador confirma que a informação do avistamento está correta e clica em “Submit Incident”.
12. O sistema envia o incidente para o servidor.
13. O sistema apresenta uma mensagem de sucesso ao utilizador e redireciona-o para o ecrã inicial.

Extensões:

- 12a.** Em caso de não haver conexão à Internet: O sistema apresenta uma mensagem indicando que o incidente foi armazenado localmente.

Tabela 23. Descrição Detalhada do Caso de Uso UC7**UC7:** Iniciar Tracking da Rota da Viagem**Ator Principal:** Utilizador**Objetivo:** Permitir que o utilizador inicie o tracking da rota de uma viagem.**Pré-Condições:**

- O utilizador deve ter acesso à aplicação.
- O utilizador deve ter login efetuado na aplicação.
- O dispositivo deve ter a funcionalidade GPS habilitada.

Pós-Condições:

- O utilizador faz tracking da rota de uma viagem.

Cenário Principal:

1. O utilizador seleciona a opção “Start your trip!”.
2. O sistema apresenta ao utilizador um mapa com a opção de iniciar o tracking da rota da viagem.
3. O utilizador seleciona a opção “Start Trip”.
4. O sistema apresenta no mapa as coordenadas iniciais da rota e a cada 5 minutos coloca as coordenadas atuais do utilizador no mapa, traçando a rota.
5. O utilizador clica em “Stop”.
6. O sistema envia a rota da viagem para o servidor.
7. O sistema apresenta ao utilizador uma mensagem de sucesso indicando que a sua rota da viagem foi submetida com sucesso, e apresenta no mapa a rota da viagem completa.

Extensões:

- 6a.** Em caso de não haver conexão à Internet: O sistema apresenta uma mensagem indicando que a rota da viagem foi armazenada localmente.

Tabela 24. Descrição Detalhada do Caso de Uso UC8**UC8:** Alterar Título**Ator Principal:** Utilizador**Objetivo:** Permitir que o utilizador inicie o tracking da rota de uma viagem.**Pré-Condições:**

- O utilizador deve ter acesso à aplicação.
- O utilizador deve ter login efetuado na aplicação.
- O dispositivo deve ter conexão à Internet.
- O utilizador deve ter pelo menos um distintivo desbloqueado.

Pós-Condições:

- O utilizador altera o título do seu perfil.
- O novo título é apresentado no perfil do utilizador.

Cenário Principal:

1. O utilizador acede ao seu perfil de utilizador.
2. O sistema apresenta ao utilizador o seu perfil, incluindo toda a sua informação pessoal preenchida no momento de registo, assim como um botão para alterar o seu título.
3. O utilizador seleciona a opção “Change Title”.
4. O sistema apresenta ao utilizador uma lista com todos os títulos disponíveis.
5. O utilizador seleciona um título diferente do atual e clica em “Select”.
6. O sistema verifica se o novo título escolhido pelo usuário é diferente do título atual.
7. O sistema atualiza o título do utilizador com a nova escolha do utilizador.

Tabela 25. Descrição Detalhada do Caso de Uso UC9**UC9:** Desbloquear Carta de Espécie**Ator Principal:** Utilizador**Objetivo:** Permitir que o utilizador desbloqueie uma carta de uma espécie.**Pré-Condições:**

- O utilizador deve ter acesso à aplicação.
- O utilizador deve ter login efetuado na aplicação.
- O dispositivo deve ter conexão à Internet.
- O utilizador deve ter a carta da espécie bloqueada.

Pós-Condições:

- O utilizador desbloqueia uma carta de uma espécie.
- A carta da espécie é adicionado à coleção de cartas de espécies do utilizador.

Cenário Principal:

1. O utilizador submete um avistamento.
2. O sistema verifica as espécies avistadas e apresenta ao utilizador o número e as cartas das espécies avistadas pela primeira vez após submeter o avistamento.
3. O sistema atualiza a coleção de cartas de espécies do utilizador com as novas cartas desbloqueadas.

Tabela 26. Descrição Detalhada do Caso de Uso UC10**UC10:** Desbloquear Distintivo**Ator Principal:** Utilizador**Objetivo:** Permitir que o utilizador desbloqueie um distintivo.**Pré-Condições:**

- O utilizador deve ter acesso à aplicação.
- O utilizador deve ter login efetuado na aplicação.
- O dispositivo deve ter conexão à Internet.
- O utilizador deve cumprir os requisitos para o desbloqueio do distintivo.
- O utilizador deve ter o distintivo bloqueado.

Pós-Condições:

- O utilizador desbloqueia um distintivo.
- O distintivo é adicionado à coleção de distintivos do utilizador.
- O título do distintivo é adicionado como opção na lista de títulos disponíveis do utilizador.

Cenário Principal:

1. O utilizador submete um avistamento.
2. O sistema verifica as espécies avistadas e apresenta ao utilizador o número e as cartas das espécies avistadas pela primeira vez após submeter o avistamento.
3. O sistema atualiza a coleção de cartas de espécies do utilizador com as novas cartas desbloqueadas.

Tabela 27. Descrição Detalhada do Caso de Uso UC11**UC11:** Visualizar Ranking**Ator Principal:** Utilizador**Objetivo:** Permitir que o utilizador visualize o ranking.**Pré-Condições:**

- O utilizador deve ter acesso à aplicação.
- O utilizador deve ter login efetuado na aplicação.

Pós-Condições:

- O utilizador visualiza o ranking.

Cenário Principal:

1. O utilizador clica em “Ranking”.
2. O sistema apresenta ao utilizador o ranking global, a opção de alternar entre o ranking “Global” ou “Local” e de adicionar filtros ao ranking.
3. O utilizador seleciona a opção “Local”.
4. O sistema atualiza o ranking com os utilizadores com a mesma localização do utilizador.
5. O utilizador seleciona a opção de adicionar um filtro.
6. O sistema apresenta ao utilizador a opção de filtrar o ranking por intervalo de tempo, número de avistamentos reportados, número de espécies desbloqueadas e número de distintivos desbloqueados.
7. O utilizador seleciona a opção de filtrar o ranking por número de espécies desbloqueadas.
8. O sistema atualiza o ranking ordenado por ordem decrescente do número de espécies desbloqueadas por cada utilizador.
9. O utilizador pode alternar entre o ranking "Global" e "Local" a qualquer momento.
10. O utilizador pode alterar o filtro a qualquer momento.

Tabela 28. Descrição Detalhada do Caso de Uso UC12**UC12:** Visualizar Mapa de Avistamentos**Ator Principal:** Utilizador**Objetivo:** Permitir que o utilizador visualize um mapa com todos os avistamentos reportados a partir da aplicação, filtrando por data.**Pré-Condições:**

- O utilizador deve ter acesso à aplicação.
- O utilizador deve ter login efetuado na aplicação.

Pós-Condições:

- O utilizador visualiza o mapa com todos os avistamentos.

Cenário Principal:

1. O utilizador acede ao mapa de avistamentos.
2. O sistema apresenta ao utilizador um mapa com clusters e áreas de todos os avistamentos reportados a partir da aplicação.
3. O utilizador seleciona a opção de adicionar um filtro.
4. O sistema apresenta ao utilizador a opção de filtrar os avistamentos apresentados no mapa por “Last year”, “Last month”, “Last week”, “Last day”.
5. O utilizador seleciona a opção do período que deseja visualizar.
6. O sistema atualiza o mapa e apresenta ao utilizador apenas os avistamentos reportados dentro do período selecionado.
7. O utilizador pode interagir com os avistamentos no mapa, selecionando um cluster para expandi-lo ou fazendo zoom in ou zoom out.

Tabela 29. Descrição Detalhada do Caso de Uso UC13

UC13: Visualizar Informação sobre Espécie

Ator Principal: Utilizador

Objetivo: Permitir que o utilizador visualize a informação sobre uma espécie.

Pré-Condições:

- O utilizador deve ter acesso à aplicação.
- O utilizador deve ter login efetuado na aplicação.

Pós-Condições:

- O utilizador visualiza a informação sobre uma espécie.

Cenário Principal:

1. O utilizador acede ao ecrã de seleção de espécies, onde são apresentadas opções para selecionar o tipo de animal, tamanho e outras informações relevantes.
2. O sistema apresenta ao utilizador três opções para selecionar o tipo de animal, incluindo “Dolphins”, “Whales” ou “Others”.
3. O utilizador seleciona a opção “Dolphins”.
4. O sistema apresenta ao utilizador três opções para selecionar o tamanho das espécies do animal selecionado, incluindo “Small”, “Medium” ou “Large” (apenas uma opção pode ser selecionada).
5. O utilizador seleciona a opção “Large”.
6. O sistema lista todas as espécies de golfinhos com um tamanho “Large”.
7. O utilizador seleciona uma espécie.
8. O sistema redireciona o utilizador para um ecrã com informações detalhadas sobre a espécie selecionada, incluindo uma imagem, nome comum, nome científico, descrição da espécie, de como se pode identificar a espécie e algumas curiosidades.
9. O utilizador pode ler a informação sobre a espécie ou selecionar a opção de ouvir a descrição da espécie.
10. Se o utilizador selecionar a opção de ouvir a informação, o sistema deve reproduzir um áudio com as informações correspondentes. O utilizador pode parar a reprodução a qualquer momento.

Tabela 30. Descrição Detalhada do Caso de Uso UC14**UC14:** Visualizar Histórico de Reportes**Ator Principal:** Utilizador**Objetivo:** Permitir que o utilizador visualize o seu histórico de reportes submetidos e não submetidos.**Pré-Condições:**

- O utilizador deve ter acesso à aplicação.
- O utilizador deve ter login efetuado na aplicação.

Pós-Condições:

- O utilizador visualiza o seu histórico de reportes.

Cenário Principal:

1. O utilizador acede ao seu histórico de reportes (avistamentos e incidentes).
2. O sistema apresenta ao utilizador uma lista com os reportes não submetidos e uma lista com os reportes submetidos, ordenados por data.
3. O utilizador seleciona um reporte.
4. O sistema apresenta ao utilizador detalhes sobre o reporte selecionado, incluindo um mapa com as suas coordenadas, a data e hora do reporte, e espécies envolvidas (caso haja) e as informações preenchidas para cada uma.
5. O utilizador visualiza a informação do reporte e clica em “Close”.
6. O sistema apresenta novamente a lista de reportes.
7. O utilizador clica em “Submit All”.
8. O sistema submete todos os reportes não submetidos e envia-os para o servidor.
9. O sistema atualiza a lista de reportes, colocando os reportes não submetidos na lista de reportes submetidos.

Extensões:**3a.** Se o utilizador selecionar um reporte não submetido:

1. O sistema apresenta ao utilizador a opção “Delete”.
2. Se o utilizador clicar em “Delete”, o sistema apresenta uma mensagem de confirmação.
3. Se o utilizador confirmar, o sistema remove o reporte da lista de reportes não submetidos e atualiza a lista apresentada ao utilizador.

8a. Em caso de não haver conexão à Internet: O sistema apresenta uma mensagem indicando que não foi possível submeter os avistamentos devido à falta de conexão à Internet.

Tabela 31. Descrição Detalhada do Caso de Uso UC15**UC15:** Visualizar Histórico de Rotas de Viagem**Ator Principal:** Utilizador**Objetivo:** Permitir que o utilizador visualize o seu histórico de rotas de viagem submetidas e não submetidas.**Pré-Condições:**

- O utilizador deve ter acesso à aplicação.
- O utilizador deve ter login efetuado na aplicação.

Pós-Condições:

- O utilizador visualiza o seu histórico de rotas de viagem.

Cenário Principal:

1. O utilizador acede ao seu histórico de rotas de viagem (avistamentos e incidentes).
2. O sistema apresenta ao utilizador uma lista com as rotas de viagem não submetidas e uma lista com as rotas de viagem submetidas, ordenados por data.
3. O utilizador seleciona uma rota de viagem.
4. O sistema apresenta ao utilizador detalhes sobre a viagem selecionada, incluindo um mapa com a rota da viagem, a data, a duração, a hora de início e a hora do fim da viagem.
5. O utilizador visualiza a informação da viagem e clica em “Close”.
6. O sistema apresenta novamente a lista de viagens.
7. O utilizador clica em “Submit All”.
8. O sistema submete todas as rotas de viagem não submetidas e envia-as para o servidor.
9. O sistema atualiza a lista de rotas de viagem, colocando as rotas de viagem não submetidas na lista de rotas de viagem submetidas.

Extensões:**3a.** Se o utilizador selecionar uma rota de viagem não submetida:

1. O sistema apresenta ao utilizador a opção “Delete”.
2. Se o utilizador clicar em “Delete”, o sistema apresenta uma mensagem de confirmação.
3. Se o utilizador confirmar, o sistema remove a rota de viagem da lista de rotas de viagem não submetidas e atualiza a lista apresentada ao utilizador.

8a. Em caso de não haver conexão à Internet: O sistema apresenta uma mensagem indicando que não foi possível submeter as rotas de viagem devido à falta de conexão à Internet.

B Apêndice - Imagens

B.1 Aplicação AQUA - Design Final

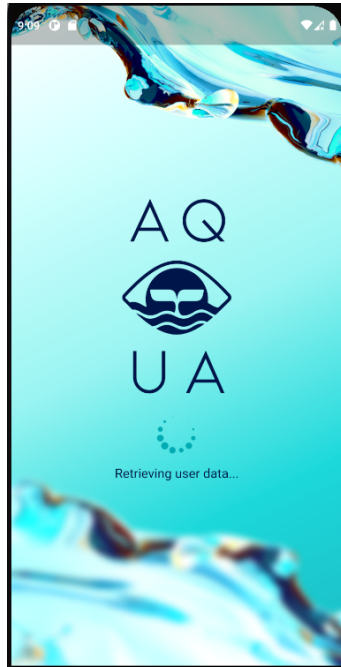


Figura 62. Screenshots da aplicação AQUA: Splash screen

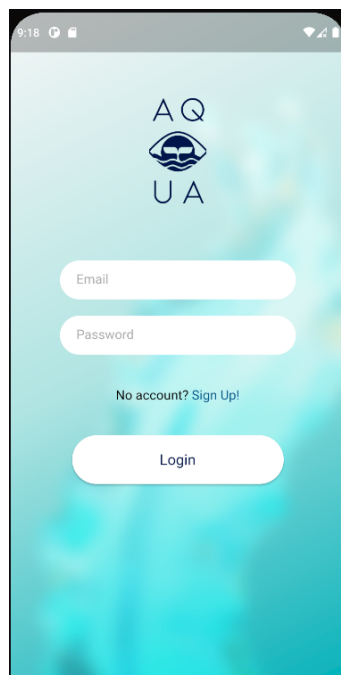


Figura 63. Screenshots da aplicação AQUA: Ecrã de login

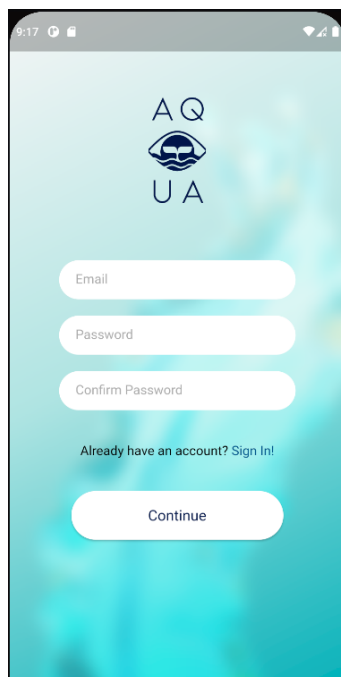


Figura 64. Screenshots da aplicação AQUA: Ecrã de criação de conta

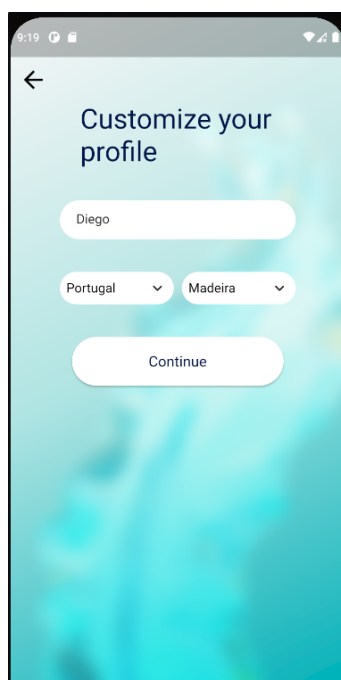


Figura 65. Screenshots da aplicação AQUA: Ecrã de adição de informação ao perfil de utilizador - Parte 1

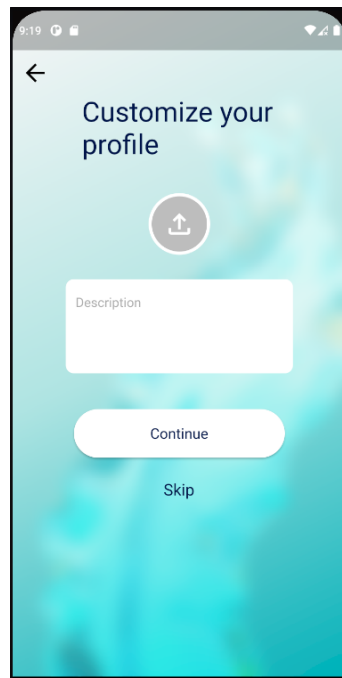


Figura 66. Screenshots da aplicação AQUA: Ecrã de adição de informação ao perfil de utilizador - Parte 2

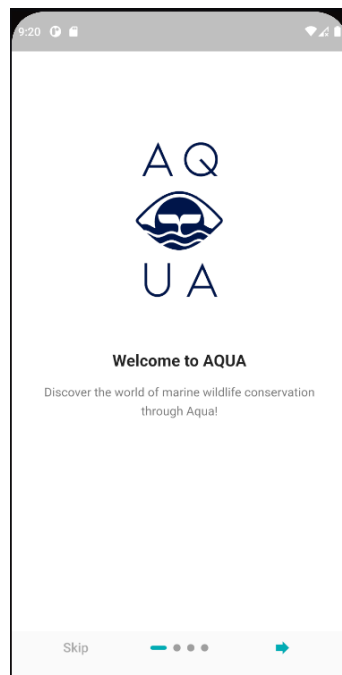


Figura 67. Screenshots da aplicação AQUA: Introdução ao funcionamento da aplicação - Parte 1

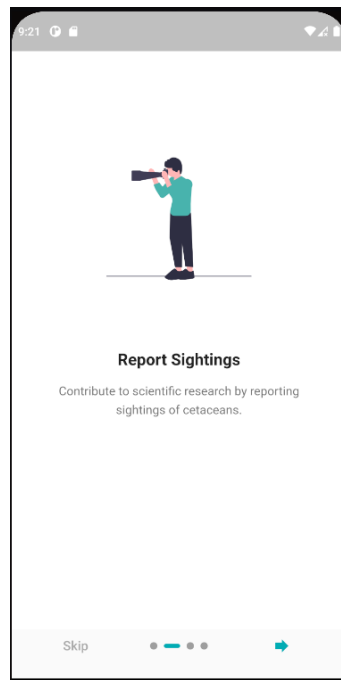


Figura 68. Screenshots da aplicação AQUA: Introdução ao funcionamento da aplicação - Parte 2

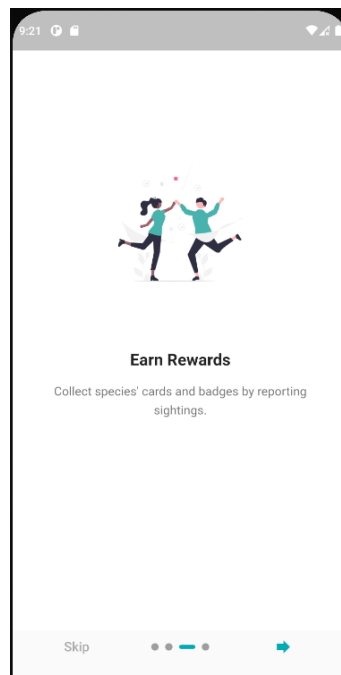


Figura 69. Screenshots da aplicação AQUA: Introdução ao funcionamento da aplicação - Parte 3

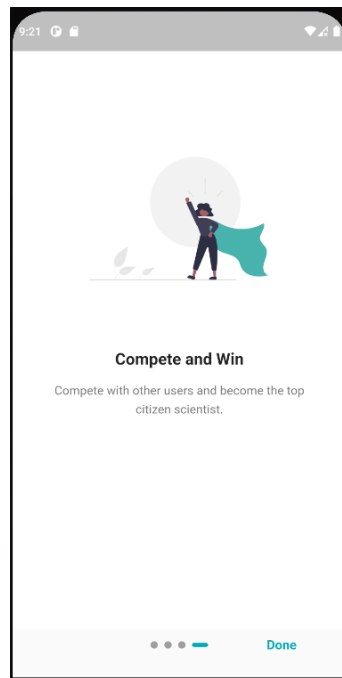


Figura 70. Screenshots da aplicação AQUA: Introdução ao funcionamento da aplicação - Parte 4

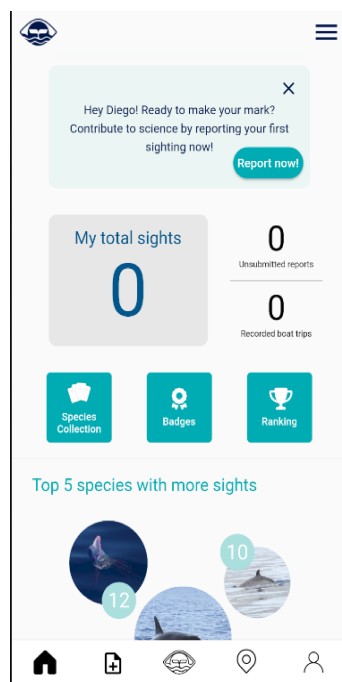


Figura 71. Screenshots da aplicação AQUA: Ecrã inicial incluindo contadores, botões e cinco espécies mais avistadas

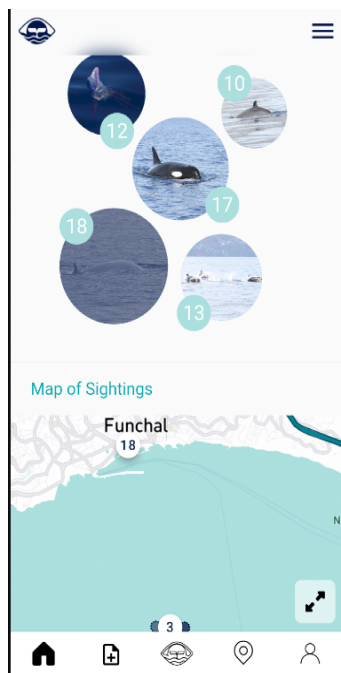


Figura 72. Screenshots da aplicação AQUA: Ecrã inicial incluindo cinco espécies mais avistadas e mapa de avistamentos

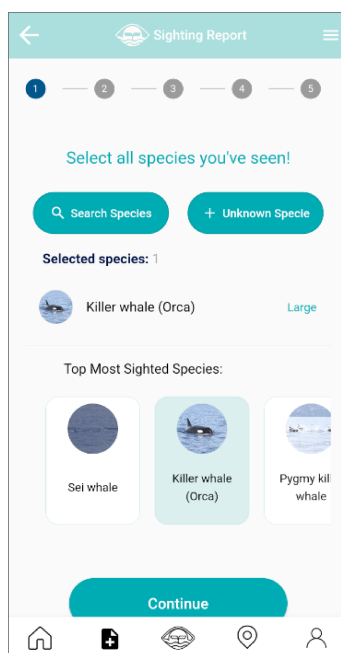


Figura 73. Screenshots da aplicação AQUA: Ecrã de seleção de espécies

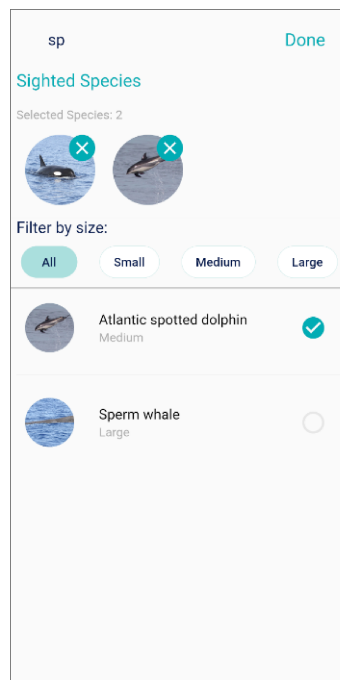


Figura 74. Screenshots da aplicação AQUA: Ecrã de pesquisa de espécies

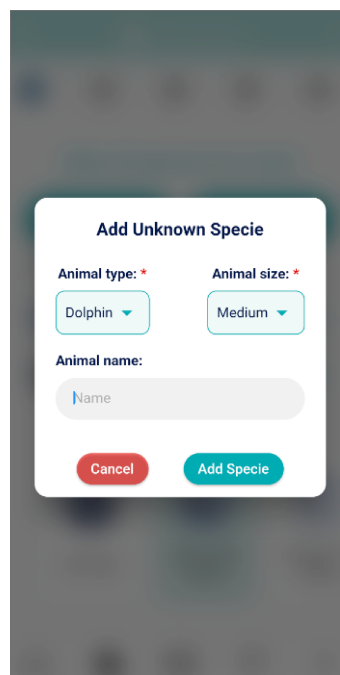


Figura 75. Screenshots da aplicação AQUA: Formulário para adição de espécies desconhecida

The screenshot shows the 'Sighting Report' screen in the AQUA app. At the top, there is a back arrow, the app logo, and the title 'Sighting Report'. Below this is a progress bar with five steps: 1 (checked), 2 (selected), 3, 4, and 5. The main content area features a circular image of a dolphin. Below the image, the text asks 'How many Atlantic spotted dolphins did you see?'. There are five buttons for selection: '1', '2' (highlighted in dark blue), '3', '4-10', '11-25', and '+25'. A teal 'Continue' button is at the bottom. The bottom navigation bar contains icons for home, add, sighting report, location, and profile.

Figura 76. Screenshots da aplicação AQUA: Ecrã de indicação do número de indivíduos das espécies

The screenshot shows the 'Sighting Report' screen in the AQUA app, specifically the behavior selection step. The progress bar at the top shows steps 1 and 2 as completed, step 3 as selected, and steps 4 and 5 as pending. The main content area features a circular image of a dolphin. Below the image, the text asks 'How were the Atlantic spotted dolphins behaving?'. There are four buttons for selection: 'Feeding' (highlighted in dark blue), 'Sleeping', 'Playing', and 'Other' (highlighted in dark blue). A teal 'Continue' button is at the bottom. The bottom navigation bar contains icons for home, add, sighting report, location, and profile.

Figura 77. Screenshots da aplicação AQUA: Ecrã de indicação dos comportamentos das espécies

Sighting Report

Unknown Specie #1

name Unknown Specie

type Dolphin size Medium

Number of Dolphins: 1

Behavior: I don't know

Continue

Figura 78. Screenshots da aplicação AQUA: Ecrã de preenchimento de informação sobre espécie desconhecida

Sighting Report

Any comment?

AQUA

Continue

Skip

Figura 79. Screenshots da aplicação AQUA: Ecrã para adição de comentário

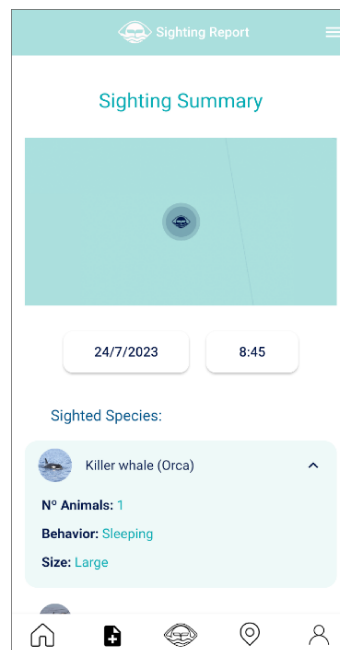


Figura 80. Screenshots da aplicação AQUA: Ecrã do sumário dos dados do avistamento - Parte 1

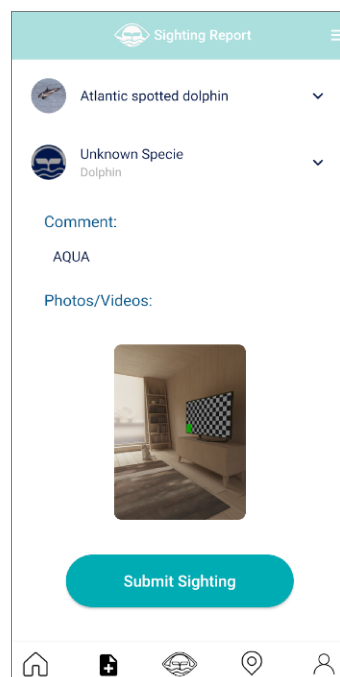


Figura 81. Screenshots da aplicação AQUA: Ecrã do sumário dos dados do avistamento - Parte 2

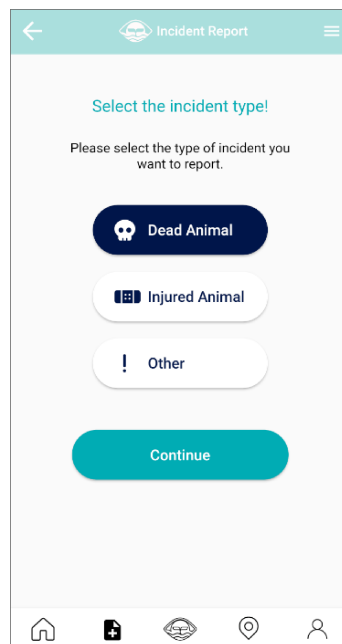


Figura 82. Screenshots da aplicação AQUA: Ecrã de seleção do tipo de incidente

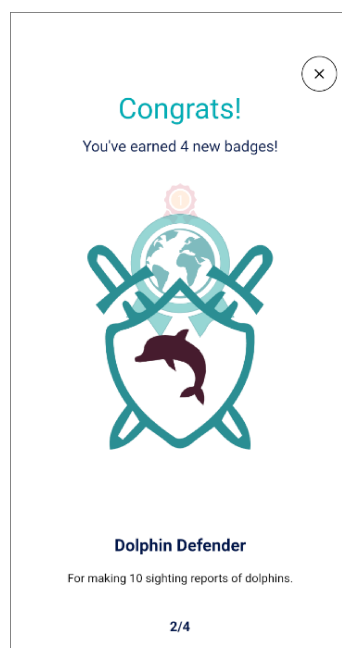


Figura 83. Screenshots da aplicação AQUA: Ecrã de desbloqueio de novos distintivos

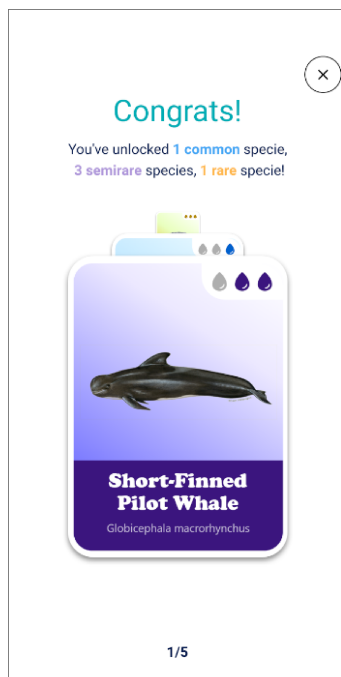


Figura 84. Screenshots da aplicação AQUA: Ecrã de desbloqueio de novas cartas de espécies

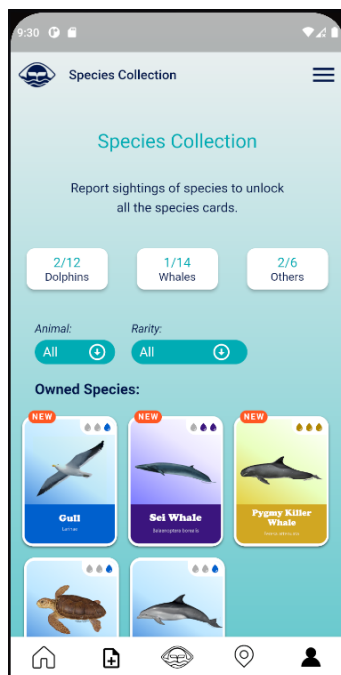


Figura 85. Screenshots da aplicação AQUA: Ecrã da coleção de espécies sem filtros aplicados

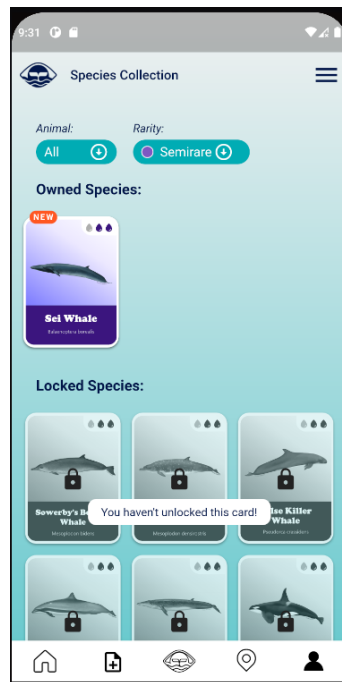


Figura 86. Screenshots da aplicação AQUA: Ecrã da coleção de espécies com filtro aplicado



Figura 87. Screenshots da aplicação AQUA: Visualização de uma carta desbloqueada - Parte 1

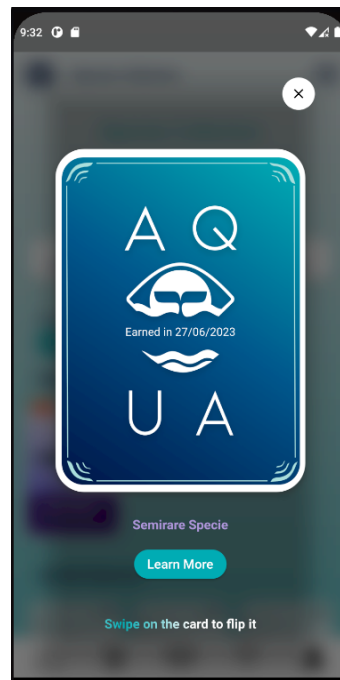


Figura 88. Screenshots da aplicação AQUA: Visualização de uma carta desbloqueada - Parte 2

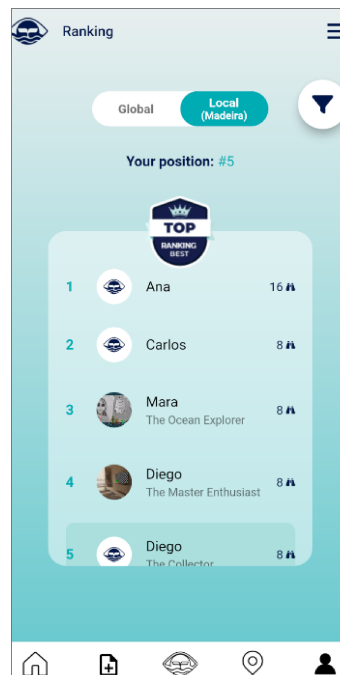


Figura 89. Screenshots da aplicação AQUA: Ecrã do ranking de utilizadores

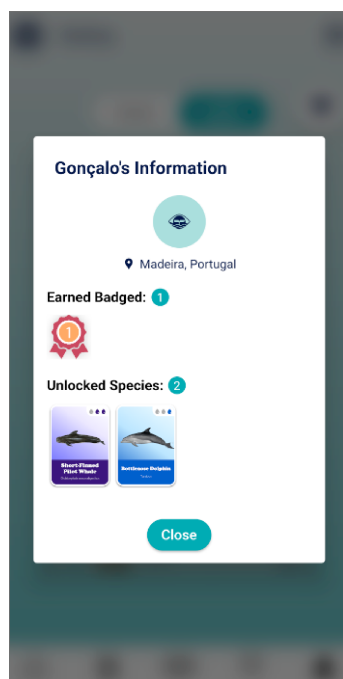


Figura 90. Screenshots da aplicação AQUA: Visualização de perfil de um utilizador do ranking

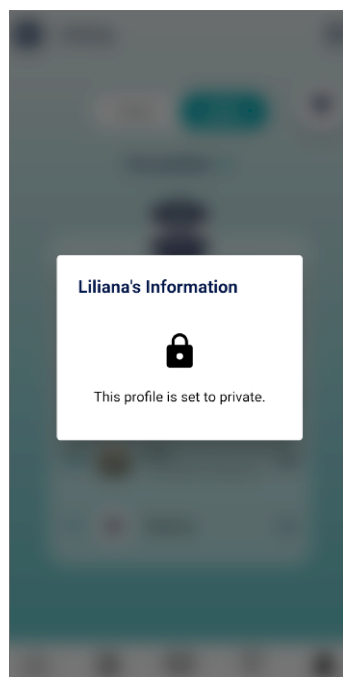


Figura 91. Screenshots da aplicação AQUA: Mensagem ao tentar visualizar um perfil privado

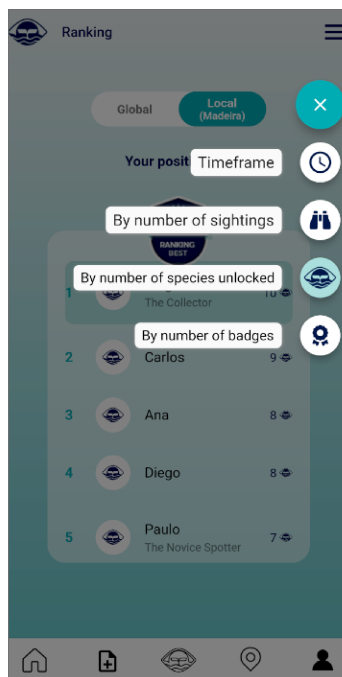


Figura 92. Screenshots da aplicação AQUA: Filtros disponíveis para o ranking

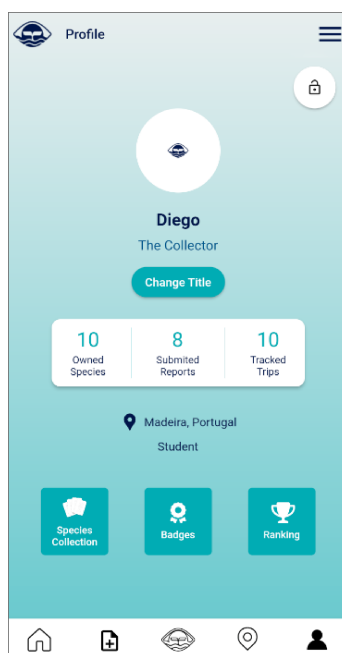


Figura 93. Screenshots da aplicação AQUA: Ecrã de perfil do utilizador

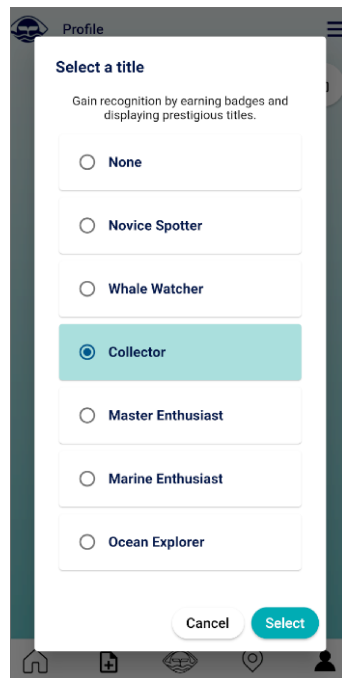


Figura 94. Screenshots da aplicação AQUA: Lista de títulos desbloqueados pelo utilizador

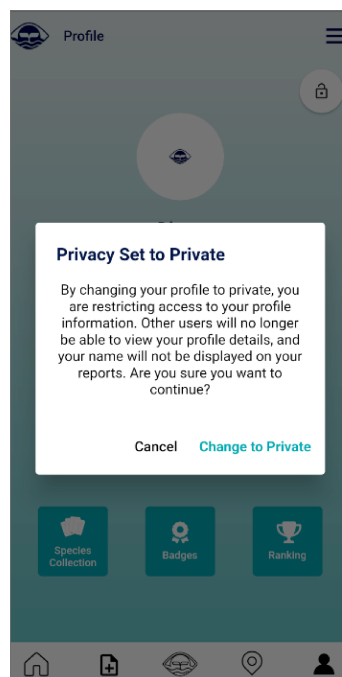


Figura 95. Screenshots da aplicação AQUA: Alteração de privacidade do perfil (de público para privado)



Figura 96. Screenshots da aplicação AQUA: Ecrã de mapa com avistamentos do último mês

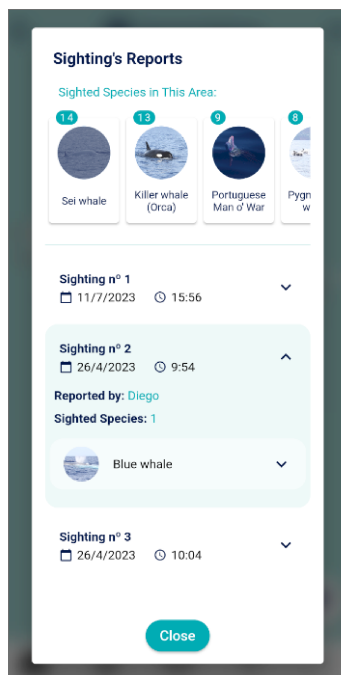


Figura 97. Screenshots da aplicação AQUA: Detalhes do cluster de avistamentos

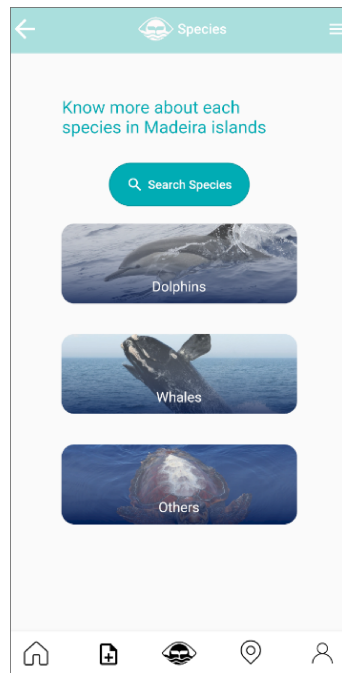


Figura 98. Screenshots da aplicação AQUA: Ecrã de seleção do tipo de animal

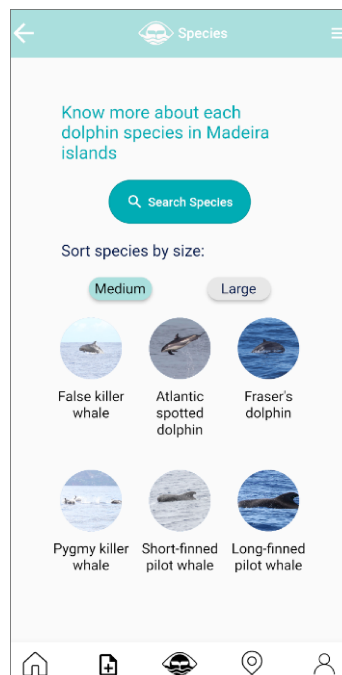


Figura 99. Screenshots da aplicação AQUA: Ecrã de seleção da espécie

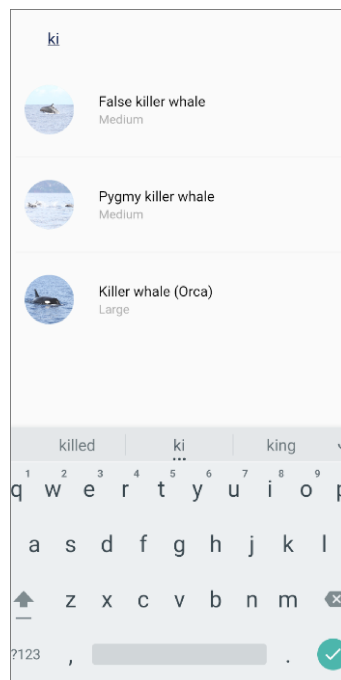


Figura 100. Screenshots da aplicação AQUA: Pesquisa de uma espécie pelo nome

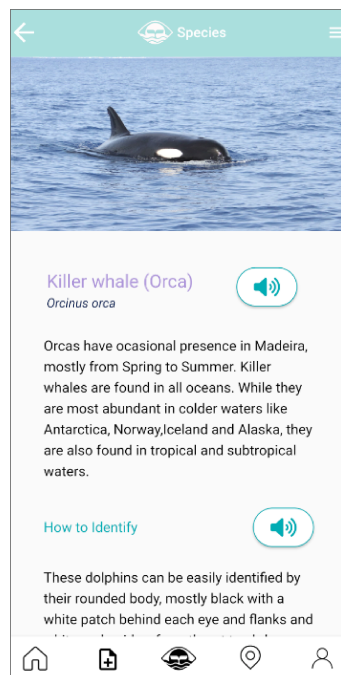


Figura 101. Screenshots da aplicação AQUA: Ecrã com informação sobre a espécie - Parte 1

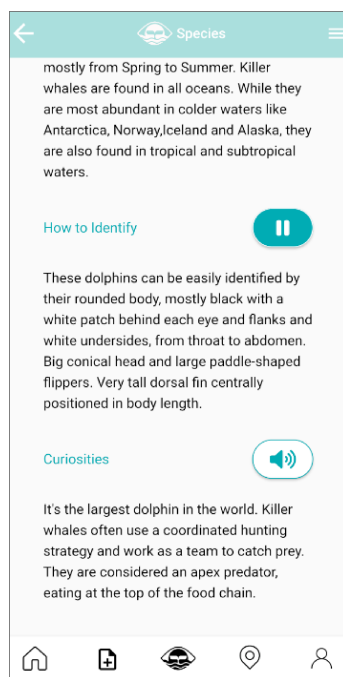


Figura 102. Screenshots da aplicação AQUA: Ecrã com informação sobre a espécie - Parte 2

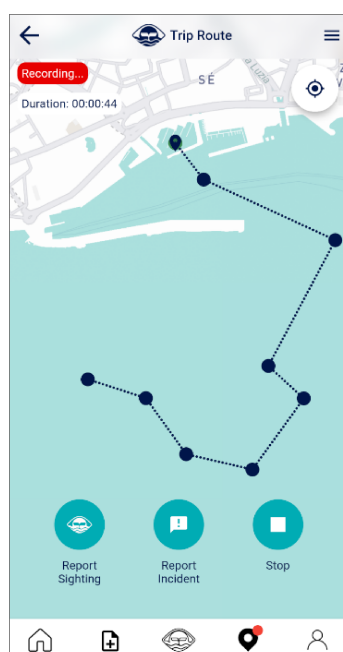


Figura 103. Screenshots da aplicação AQUA: Ecrã durante o tracking de uma rota de viagem

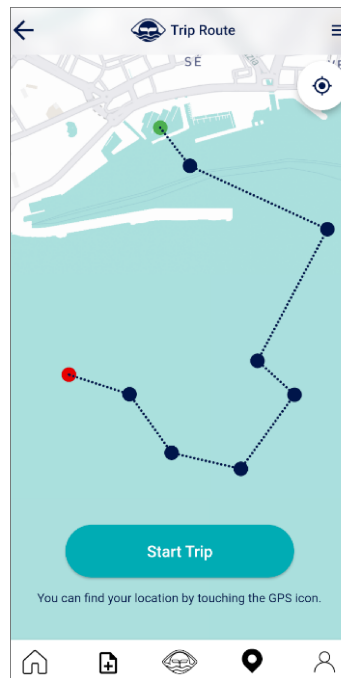


Figura 104. Screenshots da aplicação AQUA: Ecrã após terminar o tracking de uma rota de viagem



Figura 105. Screenshots da aplicação AQUA: Ecrã do histórico de reportes

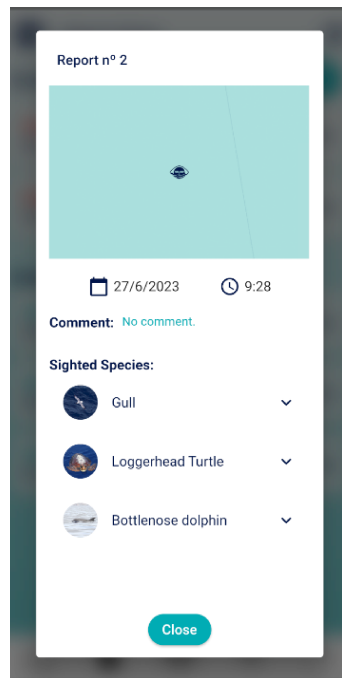


Figura 106. Screenshots da aplicação AQUA: Detalhes de um avistamento submetido

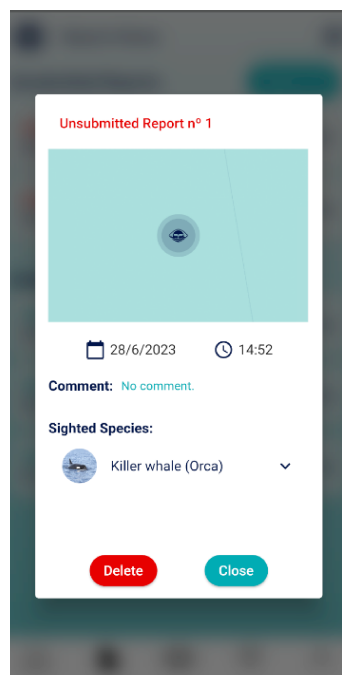


Figura 107. Screenshots da aplicação AQUA: Detalhes de um avistamento não submetido



Figura 108. Screenshots da aplicação AQUA: Ecrã do histórico de reportes, após submissão dos reportes

B.2 Arquitetura do Sistema: Modelo C4

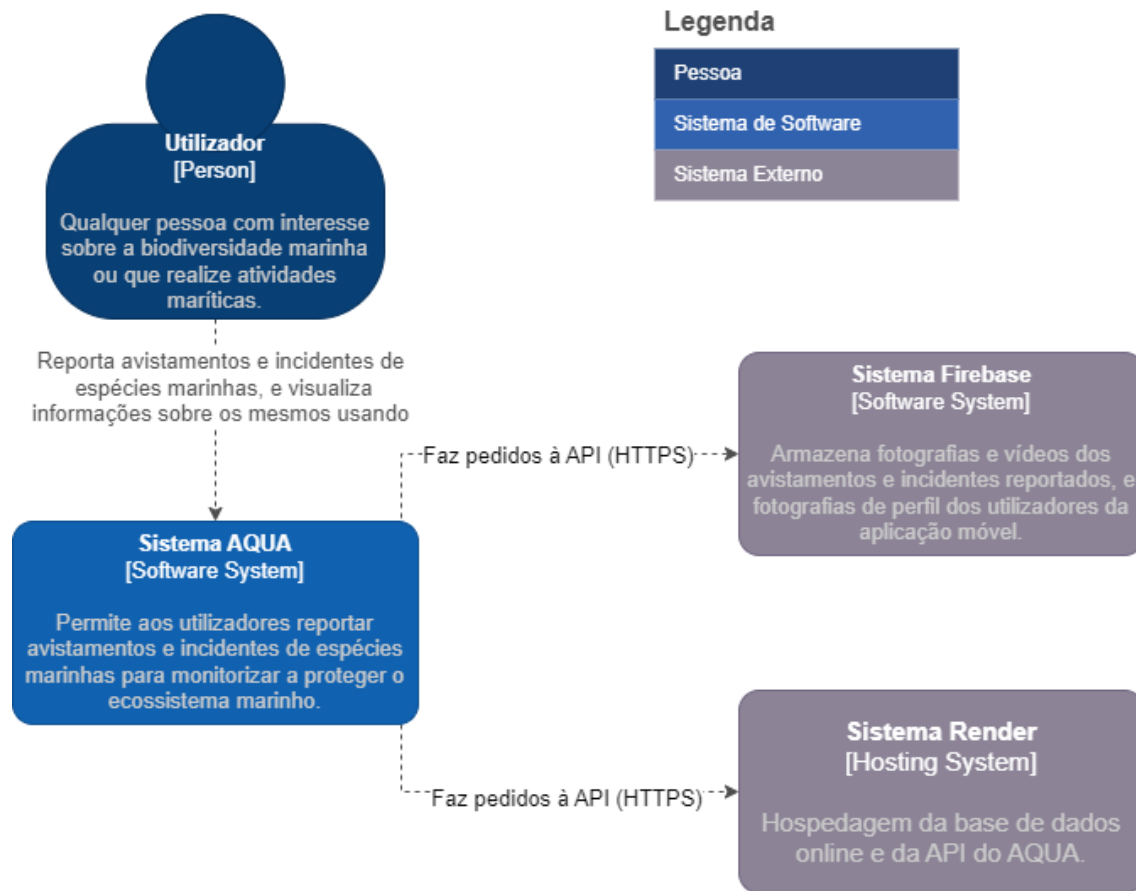


Figura 109. Modelo C4: Diagrama de Contexto

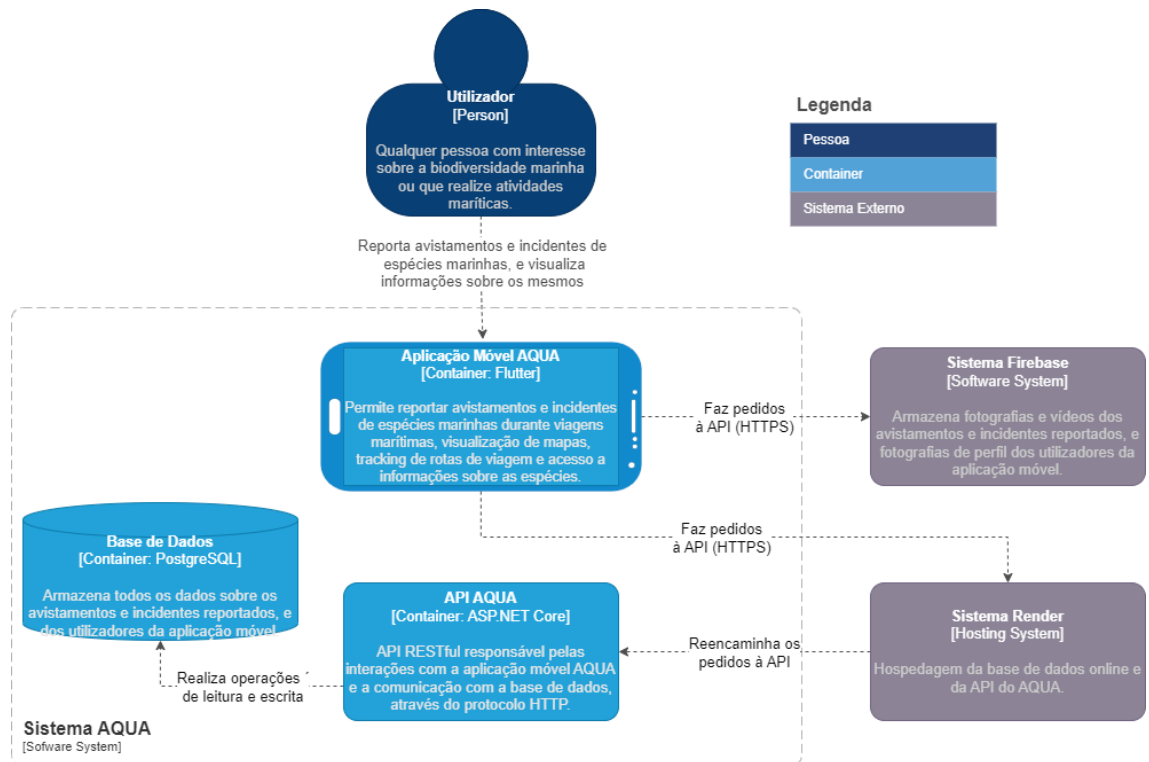


Figura 110. Modelo C4: Diagrama de Containers

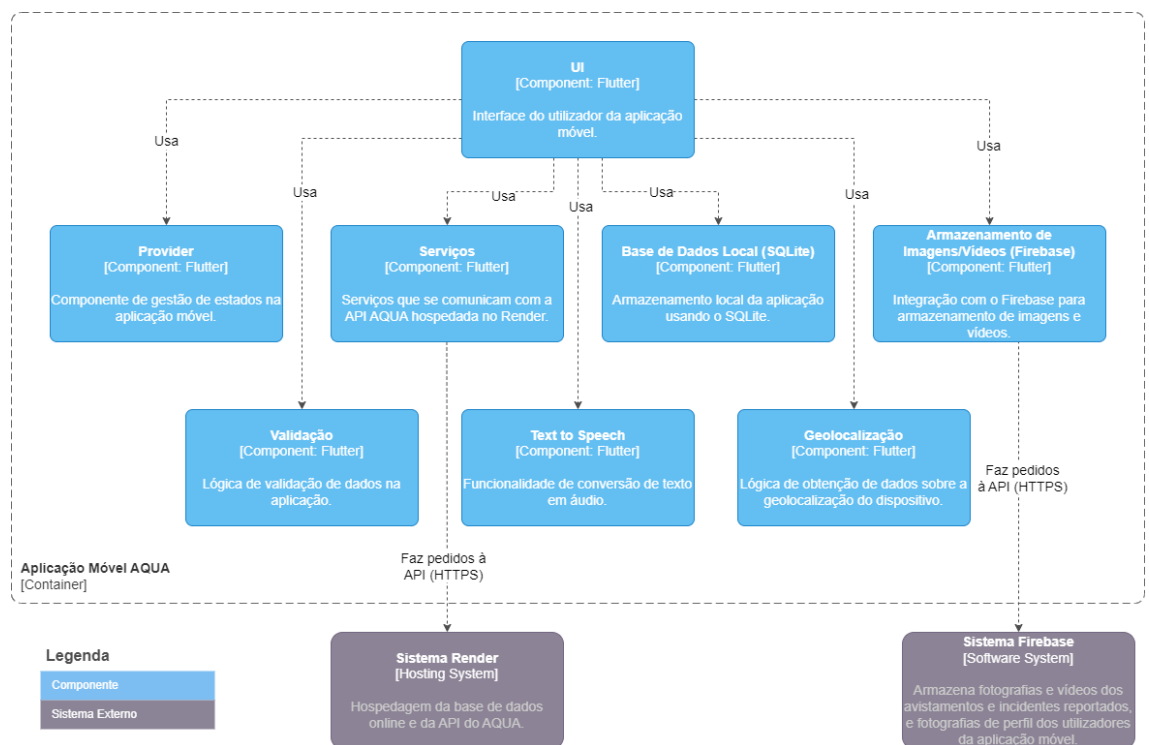
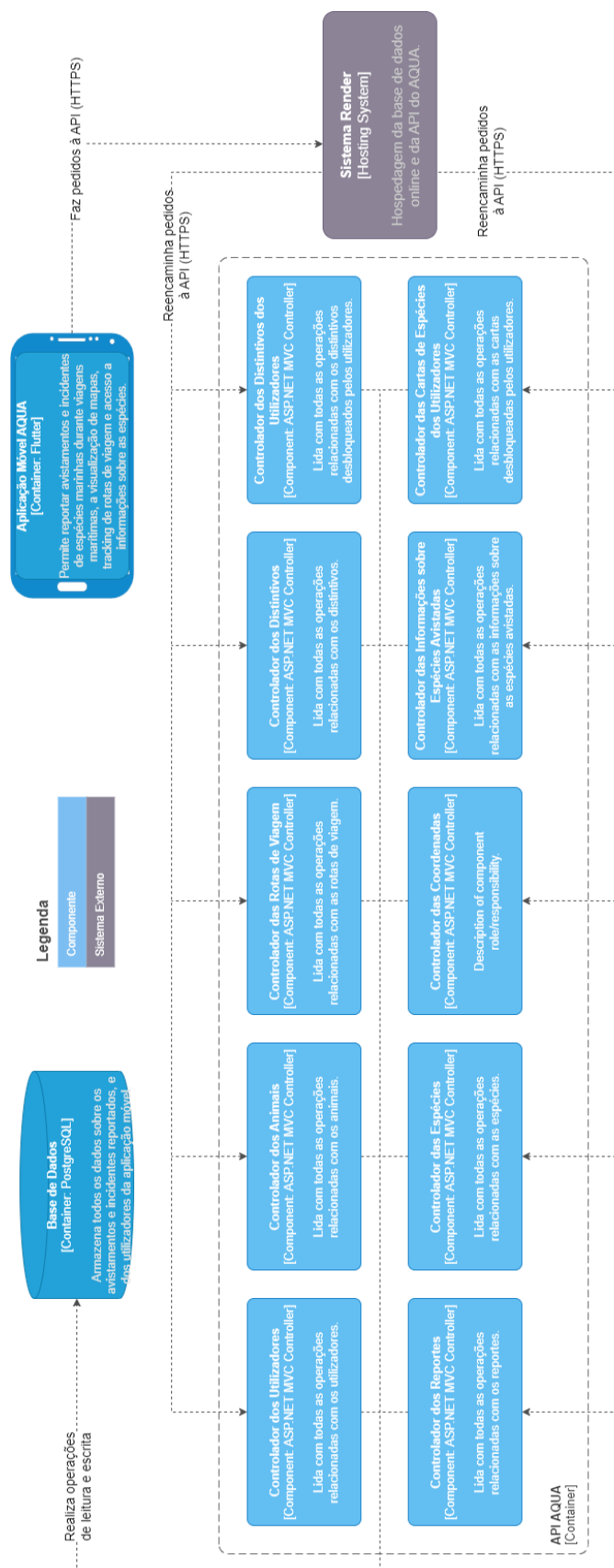


Figura 111. Modelo C4: Diagrama de Componentes (Aplicação Móvel AQUA)



B.3 AQUA 1.0

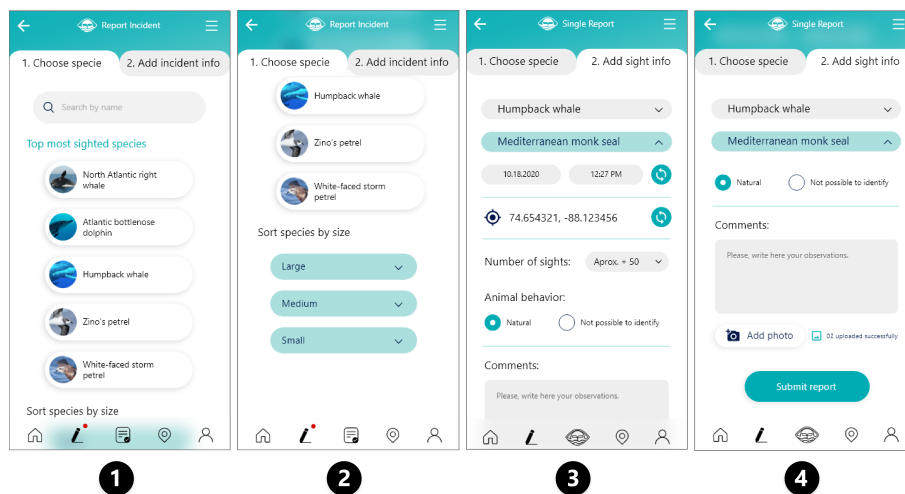


Figura 113. Screenshots da aplicação AQUA (Versão Anterior): (1-2) Ecrãs de seleção de espécies avistadas; (3-4) Ecrãs para preenchimento de dados sobre o avistamento e as espécies avistadas;

B.4 Testes de Usabilidade

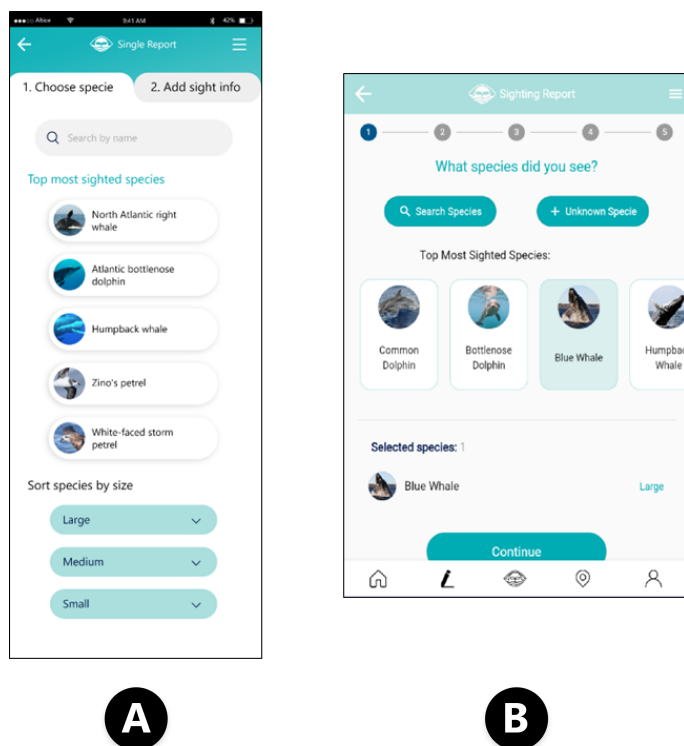


Figura 114. Jogo “This or That” - Seleção de versões da identificação de espécies avistadas: (A) Antes do *redesign*; (B) Após o *redesign* da aplicação.

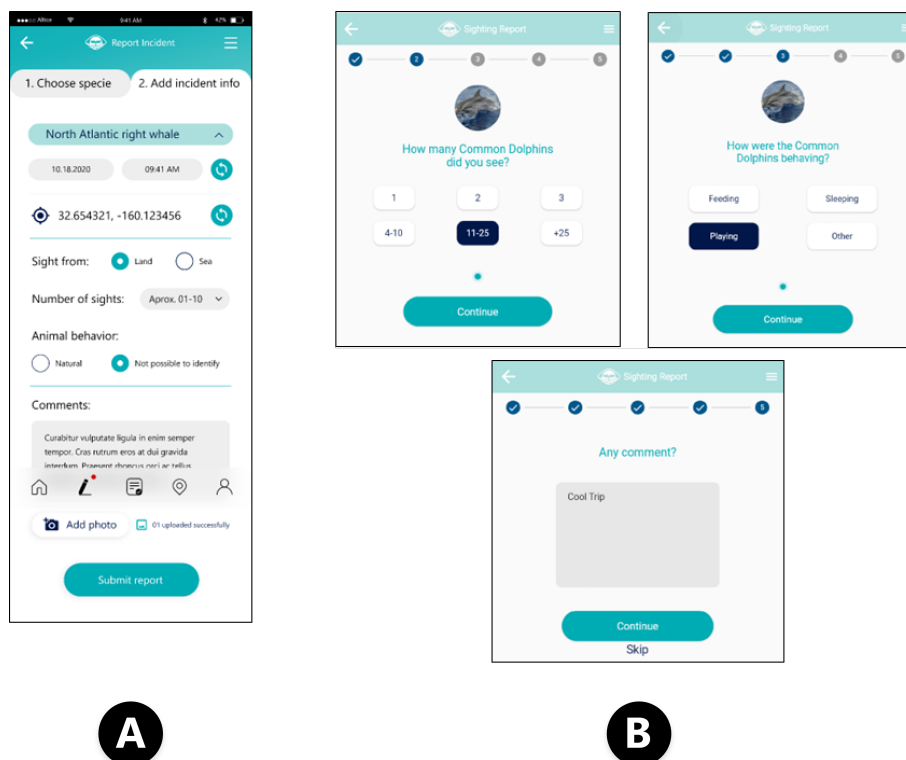


Figura 115. Jogo “This or That” - Seleção de versões da especificação dos detalhes das espécies avistadas e do avistamento: (A) Antes do *redesign*; (B) Após o *redesign* da aplicação.



Figura 116. Jogo “This or That” - Seleção de versões das cartas de espécies: (A) Cartas com raridades diferentes com a mesma cor; (B) Cartas com raridades diferentes com cores distintas.

B.5 Diagramas de Casos de Uso

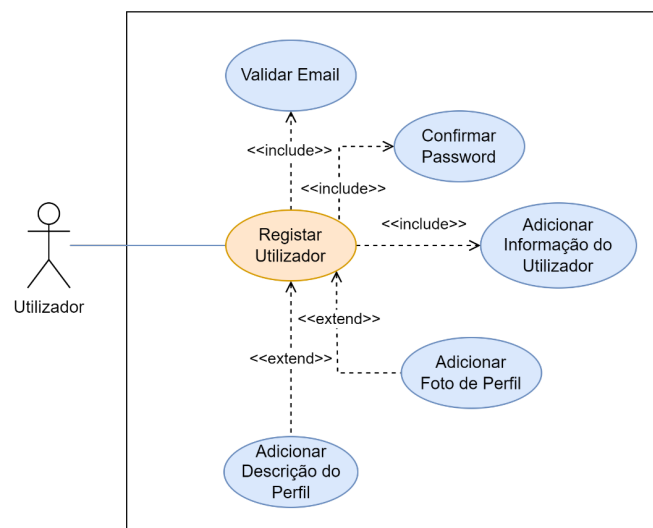


Figura 117. Diagrama de Caso de Uso: Registrar Utilizador.

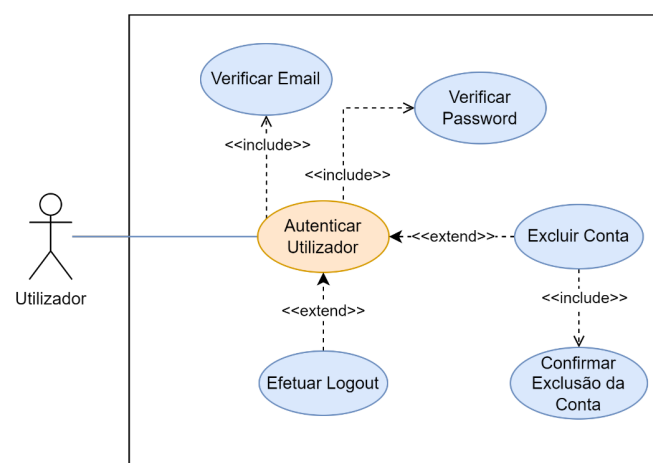


Figura 118. Diagrama de Caso de Uso: Autenticar Utilizador.

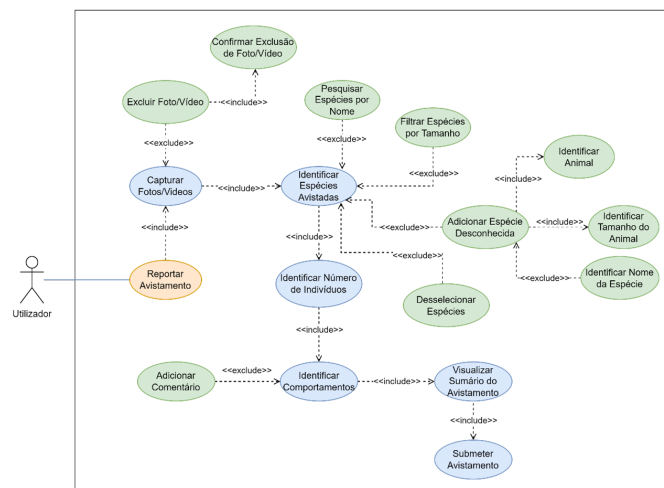


Figura 119. Diagrama de Caso de Uso: Reportar Avistamento.

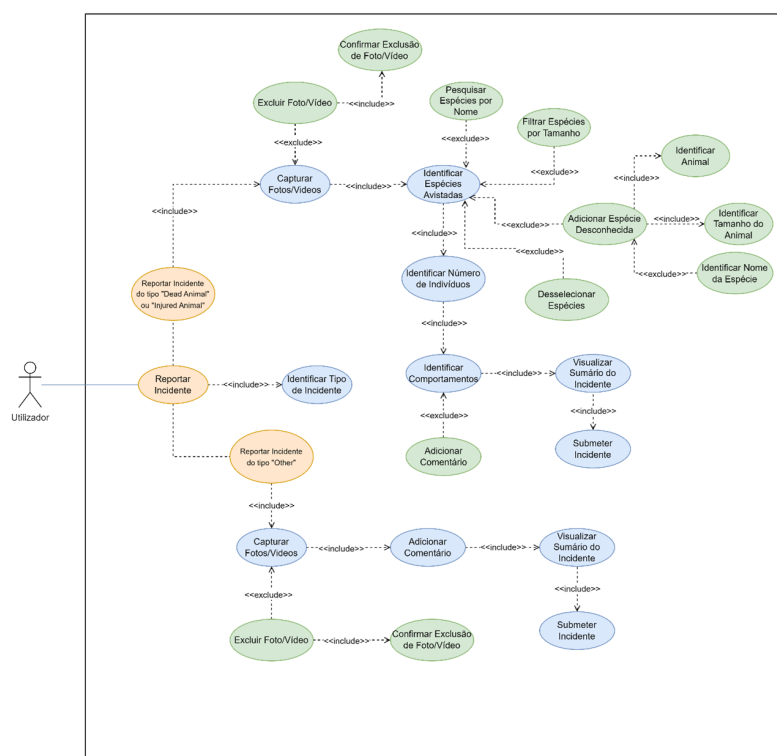


Figura 120. Diagrama de Caso de Uso: Reportar Incidente.

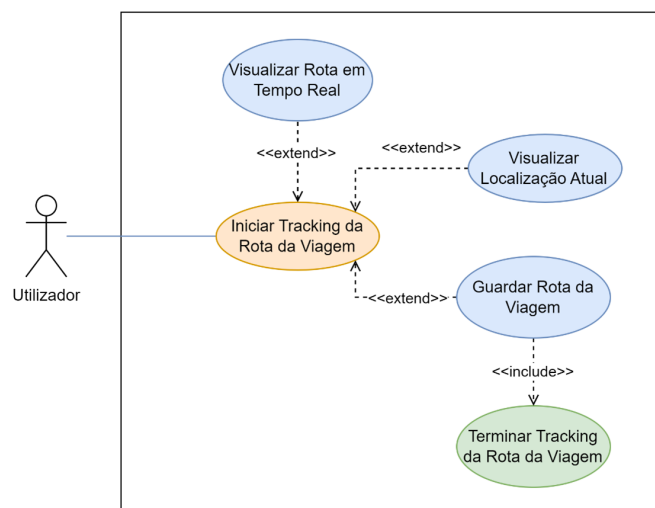


Figura 121. Diagrama de Caso de Uso: Iniciar Tracking da Rota de Viagem.

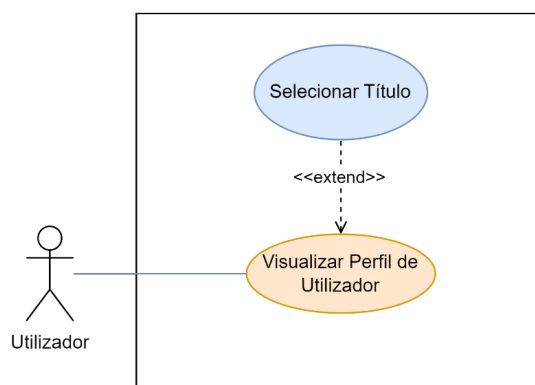


Figura 122. Diagrama de Caso de Uso: Visualizar Perfil de Utilizador.

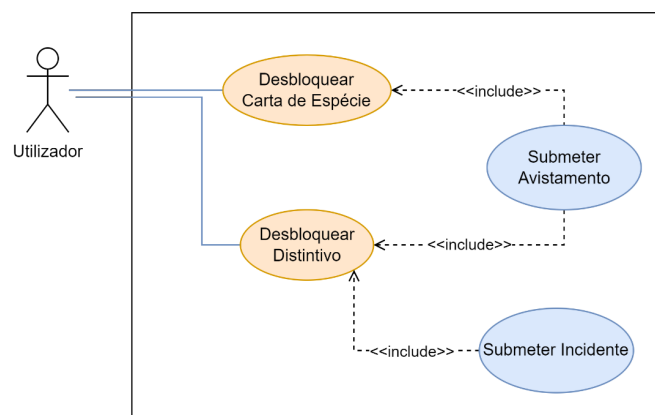


Figura 123. Diagrama de Casos de Uso: Desbloquear Carta de Espécies; Desbloquear Distintivo.

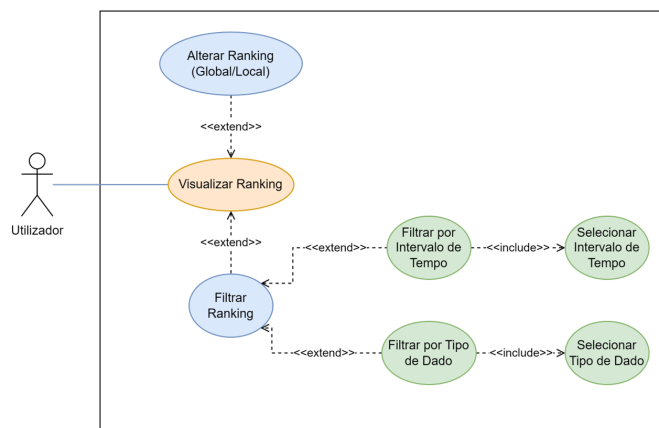


Figura 124. Diagrama de Caso de Uso: Visualizar Ranking.

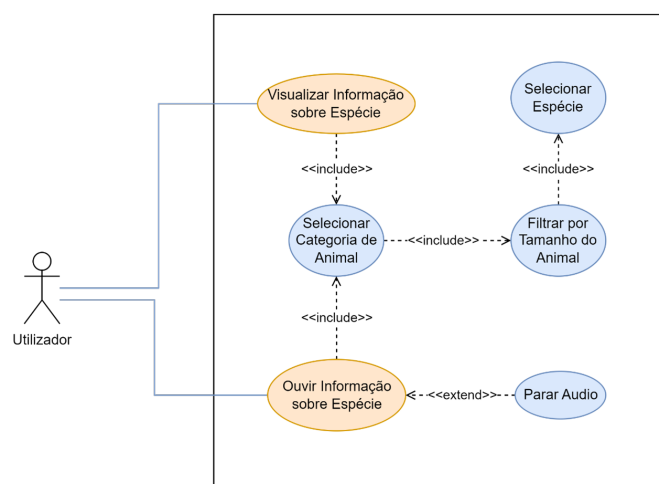


Figura 125. Diagrama de Casos de Uso: Visualizar Informação sobre Espécie; Ouvir Informação sobre Espécies.

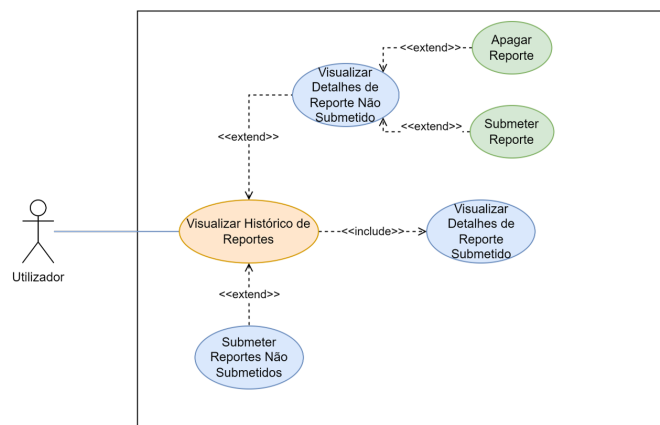


Figura 126. Diagrama de Caso de Uso: Visualizar Histórico de Reportes.

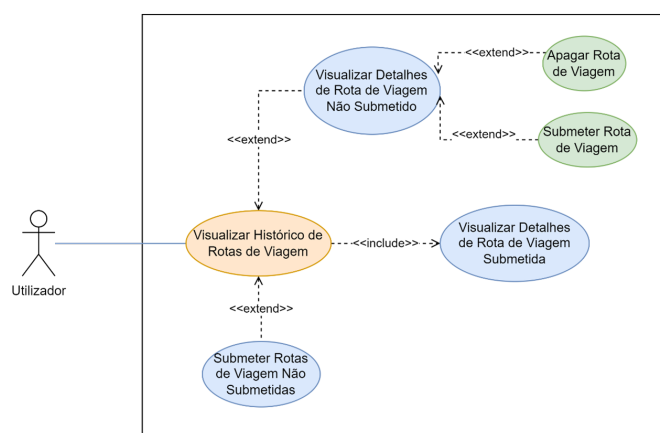


Figura 127. Diagrama de Caso de Uso: Visualizar Histórico de Rotas de Viagem.

C Apêndice - Listas

C.1 Requisitos Funcionais

Cada requisito é identificado pelo prefixo “RF” (Requisito Funcional), seguido das iniciais que representam a respetiva funcionalidade. Por exemplo, um requisito relacionado com o registo será identificado como “RF_R1”, representando o requisito funcional nº1 do registo.

Registo (R):

RF_R1: O sistema deverá permitir que um novo utilizador se registe na aplicação através de um email, uma password e uma confirmação de password.

RF_R2: O sistema deverá validar o email inserido pelo utilizador e verificar se não foi registado anteriormente por outro utilizador.

RF_R3: O sistema deve verificar se a password inserida pelo utilizador corresponde à confirmação de password.

RF_R4: O sistema deverá permitir ao utilizador adicionar o seu nome, país e cidade de residência após o registo.

RF_R5: O sistema deverá permitir ao utilizador carregar uma fotografia de perfil caso pretenda, não sendo obrigatório.

RF_R6: O sistema deverá permitir ao utilizador adicionar uma descrição do seu perfil caso pretenda, não sendo obrigatório.

RF_R7: O sistema deverá apresentar uma mensagem de erro quando o utilizador tentar registar-se sem preencher todos os campos obrigatórios.

RF_R8: O sistema deverá apresentar uma mensagem de erro quando não houver conexão à Internet durante o processo de registo.

Login (L):

RF_L1: O sistema deverá permitir ao utilizador autenticar-se através do seu endereço de e-mail e password registados.

RF_L2: O sistema deverá verificar se o email inserido pelo utilizador está registado na aplicação.

RF_L3: O sistema deverá verificar se a password inserida pelo utilizador corresponde ao email registado.

RF_L4: O sistema deverá permitir o acesso à aplicação caso o email e password inseridos estiverem corretos.

RF_L5: O sistema deverá apresentar uma mensagem de erro se o email ou a password inserida estiverem incorretos.

RF_L6: O sistema deverá apresentar uma mensagem de erro quando não houver conexão à Internet durante o processo de login.

RF_L7: O sistema deverá permitir ao utilizador efetuar logout da aplicação.

RF_L8: O sistema deverá permitir ao utilizador autenticar-se através do seu endereço de e-mail e password registados.

RF_L9: O sistema deverá verificar se o email inserido pelo utilizador está registado na aplicação.

RF_L10: O sistema deverá verificar se a password inserida pelo utilizador corresponde ao email registado.

RF_L11: O sistema deverá permitir o acesso à aplicação caso o email e password inseridos estiverem corretos.

RF_L12: O sistema deverá apresentar uma mensagem de erro se o email ou a password inserida estiverem incorretos.

RF_L13: O sistema deverá apresentar uma mensagem de erro quando não houver conexão à Internet durante o processo de login.

RF_L14: O sistema deverá permitir ao utilizador efetuar logout da aplicação.

Excluir Conta (EC):

RF_EC1: O sistema deverá permitir ao utilizador apagar a sua conta permanentemente.

RF_EC2: O sistema deverá informar ao utilizador que, ao apagar a sua conta, perderá acesso à aplicação, mas que toda a informação reportada utilizando a aplicação será mantida.

RF_EC3: O sistema deverá apresentar uma mensagem de confirmação depois da conta ter sido apagada com sucesso.

RF_EC4: O sistema deverá apresentar uma mensagem de erro se não houver conexão à Internet durante o processo de exclusão da conta.

Reportar Avistamento (RA):

RF_RA1: O sistema deverá permitir ao utilizador capturar imagens ou vídeos utilizando a câmara do dispositivo.

RF_RA2: O sistema deverá permitir ao utilizador apagar imagens ou vídeos carregados.

RF_RA3: O sistema deverá apresentar uma mensagem de confirmação antes da exclusão da imagem ou vídeo.

RF_RA4: O sistema deverá solicitar ao utilizador para identificar as espécies avistadas.

RF_RA5: O sistema deverá apresentar uma lista com as cinco espécies mais reportadas na aplicação.

RF_RA6: O sistema deverá permitir ao utilizador pesquisar espécies específicas por nome.

RF_RA7: O sistema deverá permitir ao utilizador filtrar as espécies por tamanho (small, medium, large).

RF_RA8: O sistema deverá permitir ao utilizador desselecionar espécies previamente selecionadas.

RF_RA9: O sistema deverá permitir ao utilizador identificar o número de indivíduos avistados para cada espécie (1, 2, 3, 4-10, 11-25, ou +25).

RF_RA10: O sistema deverá permitir que o utilizador selecione apenas um número de indivíduos para cada espécie.

RF_RA11: O sistema deverá permitir ao utilizador identificar o comportamento dos indivíduos avistados para cada espécie através das seguintes opções: feeding, playing, sleeping, other.

RF_RA12: O sistema deverá permitir ao utilizador seleccionar vários comportamentos para a mesma espécie.

RF_RA13: O sistema deverá permitir ao utilizador adicionar uma nova espécie (desconhecida) não listada na aplicação, através da indicação do tipo de animal (dolphin, whale, turtle, fish, bird, other) e do tamanho do animal (small, medium, large).

RF_RA14: O sistema deverá permitir ao utilizador especificar o nome da espécie desconhecida, caso o saiba.

RF_RA15: O sistema deverá permitir ao utilizador especificar o número de indivíduos avistados e o comportamento deles, opcionalmente, para a espécie desconhecida.

RF_RA16: O sistema apenas deverá permitir que o utilizador continue o reporte do avistamento se preencher toda a informação para cada espécie.

RF_RA17: O sistema deverá permitir ao utilizador adicionar um comentário geral para o avistamento opcionalmente.

RF_RA18: Após concluir a especificação da informação sobre o avistamento, o sistema deverá apresentar um sumário do mesmo, incluindo a localização num mapa, a data e hora do avistamento, uma lista de espécies avistadas com a informação preenchida para cada uma, o comentário geral, e as imagens e vídeos carregados.

RF_RA19: O sistema deverá permitir ao utilizador submeter o reporte do avistamento. Se houver conexão à Internet, a informação do avistamento é armazenada no servidor e é apresentada uma mensagem de confirmação ao utilizador.

RF_RA20: O sistema deverá armazenar o reporte do avistamento localmente, caso não haja conexão à Internet. O sistema deverá informar o utilizador sobre o armazenamento local.

Reportar Incidente (RI):

RF_RI1: O sistema deverá permitir ao utilizador seleccionar o tipo de incidente a reportar, entre “dead animal”, “injured animal” e “other incident”.

RF_RI2: No caso de seleccionar “dead animal” ou “injured animal”, o processo de reporte do incidente é idêntico ao de reportar um avistamento.

RF_RI3: No caso de seleccionar “other incident”, o utilizador deve capturar as imagens e vídeos do incidente e adicionar uma descrição sobre o mesmo, sendo a descrição obrigatória.

Tracking da Viagem (TV):

RF_TV1: O sistema deverá permitir ao utilizador iniciar o *tracking* da rota da viagem.

RF_TV2: Ao iniciar o tracking, o sistema deverá guardar localmente a localização do utilizador a cada 5 minutos, em *background*.

RF_TV3: O sistema deverá apresentar um mapa indicando cada coordenada guardada, permitindo ao utilizador visualizar a rota da viagem em tempo real.

RF_TV4: O sistema deverá permitir ao utilizador visualizar a sua localização atual no mapa a qualquer momento ao clicar num botão específico.

RF_TV5: O sistema deverá permitir ao utilizador terminar o tracking a qualquer momento, através de um botão.

RF_TV6: Caso o utilizador regresse ao ponto inicial, o tracking é automaticamente terminado.

RF_TV7: Quando o tracking é terminado, a rota da viagem é submetida na base de dados caso haja conexão à Internet. Caso não haja, a mesma é guardada localmente.

Perfil do Utilizador (PU):

RF_PU1: O sistema deverá permitir ao utilizador visualizar o seu perfil.

RF_PU2: No perfil do utilizador deve ser apresentada a foto de perfil, o nome, a descrição pessoal, a localização, o número de espécies desbloqueadas, o número de avistamentos reportados, e o número de viagens reportadas pelo utilizador.

RF_PU3: O sistema deverá permitir ao utilizador selecionar um “título” visível para outros utilizadores. As opções para o “título” devem ser todos os nomes dos distintivos desbloqueados pelo utilizador.

Coleção de Espécies (CE):

RF_CE1: O sistema deverá permitir ao utilizador desbloquear a carta de uma espécie quando reporta o avistamento de uma espécie pela primeira vez e o avistamento for submetido com sucesso.

RF_CE2: O sistema deverá permitir ao utilizador visualizar a sua coleção de cartas de espécies desbloqueadas, indicando o número total de golfinhos, baleias e outras espécies desbloqueadas.

RF_CE3: O sistema deverá separar as espécies desbloqueadas das bloqueadas.

RF_CE4: O sistema deverá permitir ao utilizador filtrar a coleção por tipo de animal (all, dolphin, whale, other) e por raridade da espécie (all, common, semirare, rare).

RF_CE5: O sistema deverá permitir ao utilizador tocar numa carta desbloqueada para visualizá-la com mais detalhe e interagir com ela para visualizar o outro lado da carta, onde é possível visualizar a data de desbloqueio da mesma.

RF_CE6: O sistema deverá apresentar uma mensagem indicando que uma carta está bloqueada, caso o utilizador toque numa carta bloqueada.

Distintivos (D):

RF_D1: O sistema deverá permitir ao utilizador desbloquear um distintivo quando cumpre os requisitos específicos para cada distintivo.

RF_D2: O sistema deverá permitir ao utilizador visualizar os distintivos desbloqueados, incluindo o número total de distintivos desbloqueados.

RF_D3: O sistema deverá permitir ao utilizador tocar num distintivo desbloqueado para visualizar o distintivo com mais detalhe, incluindo o seu título, descrição e a data em que foi desbloqueado.

RF_D4: O sistema deverá apresentar um tooltip indicando que o distintivo está bloqueado, caso o utilizador toque num distintivo bloqueado.

Ranking (R):

RF_R1: O sistema deverá permitir ao utilizador visualizar o ranking global com todos os utilizadores do sistema.

RF_R2: O sistema deverá permitir ao utilizador visualizar um ranking local com os utilizadores com a mesma localização que ele.

RF_R3: O sistema deverá apresentar a posição do utilizador no respetivo ranking.

RF_R4: O sistema deverá permitir ao utilizador filtrar o ranking por número total de avistamentos reportados, de espécies desbloqueadas ou de distintivos desbloqueados pelos utilizadores.

RF_R5: O sistema deverá permitir ao utilizador filtrar o ranking por intervalo de tempo (all time, last year, last month, last week).

RF_R6: Ao selecionar uma opção de filtro, o ranking deve ser atualizado automaticamente, ordenando os utilizadores de acordo com os filtros selecionados.

RF_R7: Para cada utilizador no ranking, o sistema deverá apresentar a sua posição, a sua foto de perfil, o seu nome, o seu título (caso o tenha), e o número total de avistamento reportados, de espécies desbloqueadas ou de distintivos desbloqueados, dependendo do filtro selecionado pelo utilizador.

Mapa de Avistamentos (MA):

RF_MA1: O sistema deverá apresentar um mapa com todos os avistamentos reportados pelos utilizadores na aplicação.

RF_MA2: O sistema deverá apresentar os avistamentos em *clusters* com o número de avistamentos naquela zona.

RF_MA3: o fazer zoom no mapa, o sistema deverá apresentar a área dos avistamentos com maior detalhe.

RF_MA4: O sistema deverá permitir ao utilizador filtrar os avistamentos apresentados no mapa por intervalo de tempo, incluindo o último ano, último mês, última semana ou últimas 24 horas.

Informação sobre Espécies (IE):

RF_IE1: O sistema deverá permitir ao utilizador selecionar o tipo de animal (dolphins, whales, others).

RF_IE2: O sistema deverá permitir ao utilizador filtrar as espécies do animal selecionado por tamanho (small, medium, large).

RF_IE3: Ao selecionar uma espécie, o sistema deverá apresentar uma imagem da espécie, o seu nome comum, nome científico, descrição da espécie, uma descrição de como é possível identificar a espécie, e algumas curiosidades sobre a mesma.

RF_IE4: O sistema deverá permitir ao utilizador ouvir a descrição da espécie, de como é possível identificá-la, e as curiosidades, em vez de ler o texto.

RF_IE5: O sistema deverá permitir que o utilizador pare o áudio da descrição da espécie, de como é possível identificá-la, e das curiosidades, a qualquer momento, ao clicar num botão de pausa.

Histórico de Reportes (HR):

RF_HR1: O sistema deverá permitir ao utilizador visualizar o seu histórico de reportes (avistamentos e incidentes) na aplicação, ordenados por data.

RF_HR2: O sistema deverá permitir ao utilizador apagar um reporte, caso ainda não tenha sido submetido.

RF_HR3: O sistema deverá permitir ao utilizador submeter todos os reportes não submetidos de uma vez, caso tenha conexão à Internet.

RF_HR4: O sistema deverá apresentar uma mensagem de erro caso o utilizador tente submeter os reportes e não tenha conexão à Internet.

RF_HR5: O sistema deverá permitir ao utilizador visualizar os detalhes sobre cada reporte.

RF_HR6: O sistema deverá apresentar um mapa com as coordenadas do avistamento, a data e hora em que foi feito, o comentário geral e informações sobre cada espécie avistada.

RF_HR7: O sistema deverá apresentar um mapa com as coordenadas do incidente, o tipo de incidente, informações sobre as espécies envolvidas (se houver) e o comentário geral.

Histórico de Rotas de Viagem (HRV):

RF_HRV1: O sistema deverá permitir ao utilizador visualizar o seu histórico de rotas de viagem na aplicação, ordenadas por data.

RF_HRV2: O sistema deverá permitir ao utilizador apagar uma rota de viagem, caso não tenha sido submetida.

RF_HRV3: O sistema deverá permitir ao utilizador visualizar a indicação de “Recording...” quando uma rota estiver a ser gravada em tempo real. Ao clicar, será redirecionado para o mapa do tracking da rota da viagem.

RF_HRV4: O sistema deverá permitir ao utilizador visualizar os detalhes de cada rota de viagem, sendo apresentado o mapa com o trajeto da viagem, a data, a duração, a hora de início e a hora de fim da viagem.

C.2 Distintivos

Distintivo “Novice Spotter”: Desbloqueado por reportar um primeiro avistamento.

Distintivo “Sighting Pro”: Desbloqueado por reportar 10 avistamentos.

Distintivo “Citizen Scientist”: Desbloqueado por reportar mais do que 25 avistamentos.

Distintivo “Geographer”: Desbloqueado por reportar avistamentos em diferentes regiões.

Distintivo “Cetacean Expert”: Desbloqueado por identificar 10 espécies de cetáceos distintas.

Distintivo “Dolphin Defender”: Desbloqueado por reportar 10 avistamentos de golfinhos.

Distintivo “Whale Watcher”: Desbloqueado por reportar 5 avistamentos de baleias.

Distintivo “Collector”: Desbloqueado por desbloquear pelo menos 10 cartas de espécies distintas.

Distintivo “Master Enthusiast”: Desbloqueado por desbloquear uma espécie rara.

Distintivo “Marine Enthusiast”: Desbloqueado por desbloquear pelo menos 3 cartas de espécies semiraras.

Distintivo “Ocean Explorer”: Desbloqueado por desbloquear pelo menos 5 cartas de espécies comuns.

Distintivo “Ocean Guardian”: Desbloqueado por reportar um incidente.

C.3 Citações dos Participantes

As citações seguem o formato U<NúmeroDoParticipante>_<IDdaCitação>.

C.3.1 Testes de Usabilidade - 1ª Iteração

U1_1: “só me deixa selecionar um, qual coloco?”

U1_2: “acho que seria mais fácil terem cores distintas”

U1_3: “sim, acho que são motivadores”

U1_4: “acho que era fixe ver o ranking da Madeira”

U1_5: “esta versão é mais simples”

U2_1: “como tem vários, coloco a opção other?”

U2_2: “a espécie rara podia ser dourada (...) dá destaque”

U2_3: “acredito que possam ajudar”

U2_4: “prefiro esta versão”

U3_1: “o ícone repete-se várias vezes”

U3_2: “com cores diferentes dá para identificar logo qual é rara”

U3_3: “se houverem muitas pessoas só vai dar para ver o top 5”

U3_4: “parece super confusa (...) prefiro esta versão, é mais simples”

U4_1: “custou-me identificar as espécies por causa do tamanho das imagens”

U4_2: “ao ganhar recompensas acho que pode motivar”

U4_3: “esta dá para ver quanto falta para terminar”

U5_1: só pelas gotas de água acho que é difícil perceber (...) mudar a cor poderia ajudar

U5_2: “as imagens estão muito pequenas”

U5_3: “em vez de mostrar o top 5, poderia apresentar o top 100”

U6_1: “podia ser adicionado um ranking global e um ranking local, para ver quem reporta mais na Madeira”

U6_2: “a outra está uma confusão (...) prefiro mil vezes esta”

C.3.2 Testes de Usabilidade - 2ª Iteração

U1_1: “como dizia "tell us about yourself"pensei que era apenas para escrever algo sobre mim (...) nem pensei em carregar a foto”

U1_2: “pensava que era para deslizar para ao lado”

U1_3: “esta fonte de texto é difícil de ler”

U1_4: “acho que as cores dão para perceber bem a distinção”

U1_5: “pelo título nunca mais chegava lá”

U1_6: “seria bom ver os dados dos avistamentos ao clicar neles”

U1_7: “não dá para ver os dados do utilizador”

U1_8: “gostei, acho que é bastante intuitiva”

U1_9: “não achei muito motivador (...) não tem muita utilidade (...) talvez cada carta poderia ter uma parte de um puzzle atrás, e o utilizador poderia unir várias cartas para desbloquear novas recompensas”

U1_10: “eu não usaria porque não costumo fazer estas viagens”

U1_11: “a tua versão está muito melhor”

U2_1: “não está a dar (...) vou mudar os dados todos”

U2_2: “acho que se o campo de pesquisa fosse limpo era mais rápido (...) não seria necessário apagar o texto”

U2_3: “costuma ser na horizontal”

U2_4: “talvez se tivesse uma animação daria para perceber”

U2_5: “nos jogos que jogo costuma ser assim”

U2_6: “pensava que dava para ver os detalhes dos avistamentos”

U2_7: “seria bom se desse para ver as recompensas de cada utilizador no ranking, para comparar”

U2_8: “é bonita a aplicação”

U2_9: “para uma pessoa usar isto teria de pagar viagens ou ter um barco (...) não penso que fossem usar com frequência”

U2_10: “não tem nada a ver (...) a outra app estava sempre a crashar”

U3_1: “este ícone é quase sempre usado para editar algo (...) pensei que era ali que se editava”

U3_2: “à primeira vista não percebi (...) quase sempre é na horizontal”

U3_3: “o mapa só mostra pins dos avistamentos, devia dar para clicar e ver melhor os avistamentos”

U3_4: “acho que o título pode ajudar a aumentar a competição”

U3_5: “gostei das cores (...) é clean”

U3_6: “eu não usaria mas talvez um biólogo ache piada”

U4_1: “acho que seria interessante se os avistamentos pudessem ser colaborativos entre membros da mesma equipa (...) assim poderiam competir com outras empresas”

U4_2: “podia ser adicionada alguma indicação de que o scroll é feito na vertical”

U4_3: “dá para perceber bem com as cores”

U4_4: “acho que seria interessante ver os detalhes sobre os avistamentos (...) que espécies foram vistas ali”

U4_5: “diz que tenho dois avistamentos mas quatro espécies desbloqueadas (...) não está errado?”

U4_6: “um título fixe faz com que os outros fiquem com inveja e também o queiram obter”

U4_7: “ficava giro ter um aquário com as espécies que vamos desbloqueando (...) e podemos interagir com elas, dar comida e brincar”

U4_8: “talvez seja interessante para biólogos marinhos”

U4_9: “sinceramente, eu acho que eu iria reportar todas as espécies só para desbloquear tudo (...) penso que as outras pessoas fariam o mesmo”

U4_10: “devia haver uma validação de se os dados estão corretos ou não (...) poderia ser utilizada inteligência artificial para verificar se as imagens se correspondem às espécies que o utilizador selecionou”

U4_11: “melhorou bastante (...) está mais completa”

U5_1: “acho que poderias adicionar uma animação que destaque a carta rara”

U5_2: “não tinha percebido que era isto”

U5_3: “no ranking podias alterar a cor da posição do utilizador”

U5_4: “acho gira a aplicação (...) é bastante simples de usar”

U6_5: “eu acho que uma pessoa normal não utilizaria (...) tem de pagar as viagens”

U6_6: “com inteligência artificial daria para validar se os dados estão corretos ou não”

U6_1: “por exemplo uma carta rara poderia brilhar”

U6_2: “acho que ao clicar num distintivo poderia aparecer o título também (...) assim sabes qual o título que vais obter”

U6_3: “clicar no pin podia mostrar os detalhes daquele avistamento”

U6_4: “podia dar para alternar entre os dados gerais e os dados do utilizador”

U6_5: “talvez destacar a posição atual alterando a cor”

U6_6: “acho que os distintivos não ficam bem (...) não combinam com o resto da app”

U6_7: “poderia ser adicionado um sistema de níveis (...) quando reporta avistamentos, ganha pontos e pode passar de nível (...) assim vai ganhando recompensas quando sobe de nível”

U6_8: “ao desbloquear uma espécie comum poderia dar 1 ponto, e se fosse rara poderia dar 3 pontos”

U6_9: “os distintivos poderiam ter nível também”

U6_10: “as empresas poderiam ver no mapa onde as espécies costumam estar (...) poderiam ir diretamente até lá”

U6_11: “um biólogo não iria reportar dados errados”

U7_1: “acho que devias alterar a fonte, não dá para perceber o que está escrito”

U7_2: “as cores ajudam a perceber qual é comum e qual é rara”

U7_3: “pensava que esse botão era para ouvir o som da espécie (...) talvez devias alterar a posição do botão”

U7_4: “no ecrã inicial, em vez de mostrar dados gerais, acho que devia mostrar dados pessoais também (...) tem dados gerais e dados pessoais misturados”

U7_5: “colocar uma cor mais escura ou clara para ser mais fácil de ver onde estamos”

U7_6: “tens distintivos que são parecidos (...) são progressivos, não consegues desbloquear um sem desbloquear o outro (...) poderiam ser considerados o mesmo distintivo e simplesmente evolui”

U7_7: “penso que seria útil para as empresas (...) poderiam competir entre elas e destacar-se das outras”

U7_8: “as pessoas penso que iriam manipular os dados (...) colocar dados aleatórios para obter as recompensas”

U7_9: “mas se fossem profissionais penso que não iriam colocar dados errados (...) eles querem reportar para contribuir”

C.3.3 Estudo

U1_1: “uma vez pediram-nos para usarmos uma app para reportar os avistamentos, mas era muito confusa (...) facilmente me enganava a preencher o formulário e era necessário voltar atrás várias vezes”

U1_2: “esta app é muito mais fácil de usar (...) é só seguir os passos e continuar”

U1_3: “gostei bastante do design da aplicação e das cores”

U1_4: eu acho que isso é mais interessante talvez para crianças ou pessoas que gostem mesmo deste tipo de animais

U1_5: “eu faço o reporte dos avistamentos para os dados estatísticos da nossa empresa (...) não é muito relevante para nós”

U1_6: “esta parte da raridade acho que devia ser mais dinâmica (...) a baleia azul costuma estar um ou dois meses do ano por aqui perto mas depois ela migra para os Açores (...) acaba por ser rara avista-la o resto do ano”

U1_7: “as recompensas são talvez mais interessantes para os clientes (...) podem aprender mais sobre as espécies e ter um registo do que viram”

U1_8: “eu tive de reportar os avistamentos no final das viagens (...) era muito difícil para mim reportar no momento exato pelas minhas tarefas”

U1_9: “a nossa empresa por acaso quer uma app para fazer o registo (...) nós fazemos o registo em papel e depois temos de passar os dados para o Excel, dá muito trabalho”

U2_1: “acho que a aplicação é demasiado complexa para uso profissional”

U2_2: “foi um pouco complicado fazer o registo no momento exato (...) temos muitas tarefas no barco”

U2_3: “sinceramente eu simplesmente ignorava, só me foquei em reportar os avistamentos”

U3_1: “está gira a aplicação”

U3_2: “é isto que nos falta na nossa aplicação (...) nós temos de ir ao Google Maps e ver as coordenadas e depois voltar à nossa app”

U3_3: “mais interessante para os clientes que são novos no mar e gostariam de adicionar novas espécies à sua coleção, mas não acho que seja interessante para quem está trabalhando, pois é apenas algo que atrapalha a conclusão das tarefas”

U3_4: “mais interessante para quem não faz isso todos os dias”

U3_5: “a competição aumenta a pressão para encontrar os animais e isso por sua vez pressiona os animais (...) não acho que a competição seja justa”

U4_1: “estou a trabalhar, não é relevante para mim”

U4_2: “tinha de reportar na nossa app e depois reportar na outra (...) acabava por ser chato e demorava imenso ”

U4_3: “talvez seja interessante para entusiastas (...) para nós não é”

U5_1: “em contexto de trabalho estamos é focados nas nossas tarefas”

U5_2: “pessoas que gostem de cetáceos talvez achem piada ”

U6_1: “acho que é muito mais simples e rápido fazer o reporte na tua app”

U6_2: “não muito (...) mas gostei das cartas e de haverem umas raras e outras não”

U7_1: “considero inovadora, útil e divertida”

U7_2: “a minha experiência foi muito boa”

U7_3: “sempre tentei reportar os avistamentos logo após fazer as minhas tarefas relacionadas com os avistamentos, principalmente para garantir que as coordenadas registadas eram o mais precisas possível”

U7_4: “achei bastante motivador (...) gostei de ver que desbloqueava espécies raras (...) os distintivos também”

U7_5: “acho motivador mas depois surge aqui a questão de haverem pessoas a reportar avistamentos só para ganhar (...) sei que deve ser difícil mas devia haver um controlo para evitar isso”

U7_6: “o que eu sugeriria é incorporar um sistema de aumento de nível para cada espécie, a fim de também motivar os utilizadores a registarem avistamentos de espécies comuns (...) os utilizadores continuariam a registar o maior número possível de avistamentos, mesmo das espécies mais avistadas com o objetivo de subir de nível”

U7_7: “outra sugestão seria incluir um sumário dos avistamentos de cada utilizador até o final de cada ano (...) algo semelhante ao spotify wrapped (...) acredito que isso também incentivaria as pessoas a usarem mais a aplicação”

U8_1: “gostei imenso (...) é muito fácil de usar”

U8_2: “custou-me reportar os avistamentos no momento por causa das minhas tarefas (...) então tive de reportar mais tarde mas as coordenadas já não eram as mesmas”

U8_3: “adorei as cartas (...) fiquei super contente quando desbloqueei uma carta rara (...) queria desbloqueá-las todas”

U8_4: “fiquei contente quando vi que fiquei em terceiro lugar (...) até fui ver quem eram os que estavam à minha frente”

U8_5: “gostei dos distintivos mas gostei mais das outras coisas”

U8_6: “eu recomendaria a app a qualquer pessoa que reporte avistamentos (...) é muito mais fácil assim, do que ser feito em papel”

U8_7: “eu achei motivador mas talvez seja ainda mais para pessoas que fazem as viagens de vez em quando (...) eu estou a trabalhar”

U8_8: “às vezes via imensos animais e tinha de colocar a opção +25 (...) mas isso pode significar 30 ou 1000 (...) acho que devia ser mais preciso o valor que inserimos (...) talvez com uma barrinha que deslize”

U8_9: “também acho que nos comportamentos devia haver uma opção unknown (...) às vezes não dá para identificar o comportamento então tive de colocar a opção other (...) essa opção pode significar muita coisa”

D Apêndice - Detalhes do Desenvolvimento da API

D.1 Arquitetura MVC: Componentes

– Modelos:

Para o desenvolvimento da API foram implementados vários modelos. Estes modelos têm como objetivo separar as diferentes entidades, melhorando a organização do código e facilitando a manutenção da API. Desta forma, cada modelo representa uma entidade específica do sistema, podendo ter ou não relação com outros modelos distintos. Os modelos implementados na API desenvolvida são:

- **User:** Este modelo é responsável por representar um utilizador do sistema. Na tabela 32 são apresentadas todas as propriedades do modelo 'User', incluindo uma descrição de cada propriedade.

Tabela 32. Propriedades do Modelo 'User'

Propriedade	Descrição
ID	Identificador único do utilizador
Name	Nome do utilizador
Email	<i>Email</i> do utilizador
PasswordHash	<i>Hash</i> da <i>password</i> do utilizador
PasswordSalt	<i>Salt</i> da <i>password</i> do utilizador
Active	Indica se o utilizador está ativo ou não
Role	Papel do utilizador
Location	Localização do utilizador
Description	Descrição do utilizador
ImageUrl	URL da foto de perfil do utilizador
TripRoutes	Lista de IDs de rotas de viagem do utilizador
UnlockedSpecies	Lista de IDs de espécies desbloqueadas pelo utilizador
UnlockedBadges	Lista de IDs de distintivos desbloqueados pelo utilizador
BadgeTitleId	ID do título do distintivo escolhido pelo utilizador

- **UserDto:** Este modelo também é responsável por representar um utilizador do sistema. Contudo, este modelo é um DTO, utilizado para enviar e receber informação dos utilizadores sem expor informações sensíveis, como a *password*. Desta forma, este modelo inclui todas as propriedades do modelo 'User' com exceção das propriedades '**PasswordHash**' e '**PasswordSalt**'.
- **Animal:** Este modelo é responsável por representar um animal do sistema. Ele inclui apenas duas propriedades: a propriedade '**ID**', que é o identificado único do animal, e a propriedade '**Name**', que representa o nome do animal.
- **Specie:** Este modelo é responsável por representar uma espécie de um animal do sistema. Na tabela 33 são apresentadas todas as propriedades do modelo 'Specie', incluindo uma descrição de cada propriedade.

Tabela 33. Propriedades do Modelo 'Specie'

Propriedade	Descrição
ID	Identificador único da espécie
AnimalId	Identificador do animal associado à espécie
Name	Nome da espécie
Size	Tamanho da espécie (<i>Small, Medium, Large</i>)
Rarity	Raridade de encontrar a espécie (<i>Common, Semirare, Rare</i>)
CientificName	Nome científico da espécie
Description	Descrição da espécie
HowToIdentify	Informação sobre como identificar a espécie
Curiosities	Curiosidades sobre a espécie
CardImageUrl	URL da imagem da carta da espécie
ImagesUrl	Lista de URLs de imagens da espécie

- **LogInUserData:** Este modelo é responsável por representar os dados do utilizador que são enviados para validar a autenticação no sistema AQUA. Ele inclui como propriedades, a propriedade '**Email**', que identifica o *email* inserido pelo utilizador, e a propriedade '**Password**' que identifica a *password* inserida no momento do *login*.
- **Badge:** Este modelo é responsável por representar um distintivo da aplicação AQUA. Este modelo incluir várias propriedades: a propriedade '**ID**', que representa o identificador único do distintivo, a propriedade '**Name**' que representa o título do distintivo, a propriedade '**Description**' que representa a descrição de como um utilizador pode desbloquear o distintivo, e a propriedade '**IconUrl**' que representa o URL da imagem do distintivo.
- **UserEarnedBadge:** Este modelo representa uma relação entre um utilizador (User) e um distintivo (Badge) desbloqueado pelo utilizador. Na tabela 34 são apresentadas todas as propriedades do modelo 'UserEarnedBadge', incluindo uma descrição de cada propriedade.

Tabela 34. Propriedades do Modelo 'UserEarnedBadge'

Propriedade	Descrição
ID	Identificador único do distintivo
UserId	Identificador do utilizador associado
User	Referência ao objeto do tipo User associado ao utilizador
BadgeId	Identificador do distintivo associado
Badge	Referência ao objetivo do tipo Badge associado ao distintivo
EarnedDate	Data em que o utilizador desbloqueou o distintivo

- **UserEarnedSpecie:** De igual forma ao modelo anterior, este modelo representa uma relação entre um utilizador (User) e uma espécie (Specie) desbloqueada pelo utilizador. Na tabela 35 são apresentadas todas as propriedades do modelo 'UserEarnedSpecie', incluindo uma descrição de cada propriedade.

Tabela 35. Propriedades do Modelo 'UserEarnedSpecie'

Propriedade	Descrição
ID	Identificador único do distintivo
UserId	Identificador do utilizador associado
User	Referência ao objeto do tipo User associado ao utilizador
SpecieId	Identificador da espécie associada
Specie	Referência ao objetivo do tipo Specie associado à espécie
EarnedDate	Data em que o utilizador desbloqueou a espécie

- **TripRoute:** Este modelo representa uma rota de viagem registada na aplicação. Na tabela 36 são apresentadas todas as propriedades do modelo 'TripRoute', incluindo uma descrição de cada propriedade.

Tabela 36. Propriedades do Modelo 'TripRoute'

Propriedade	Descrição
ID	Identificador único da rota da viagem
UserId	Identificador do utilizador associado à rota
Coordinates	Lista de IDs das coordenadas que compõem a rota
StartDate	Data do início da viagem
EndDate	Data do fim da viagem

- **Coordinate:** Este modelo representa uma coordenada de uma rota de viagem registada. Ele inclui várias propriedades: a propriedade '**ID**', que representa o identificador único da coordenada, a propriedade '**Timestamp**' que indica quando a coordenada foi registada, e a propriedade '**Latitude**' e a propriedade '**Longitude**' que, tal como o nome indica, representam a latitude e longitude da coordenada, respetivamente.
- **TripInformation:** Este modelo representa as informações completas relacionadas com uma viagem. Ele possui a propriedade '**TripRoute**' que representa o objeto do tipo '**TripRoute**', e a propriedade '**Coordinates**' que representa uma lista de objetos do tipo '**Coordinate**'. Este modelo permite agrupar a informação relacionada com uma rota de viagem. Desta forma, é possível submeter na base de dados a informação completa de uma rota de viagem, sem necessidade de chamar dois *endpoints* distintos para armazenar a informação da rota, e posteriormente a informação das coordenadas.
- **TripInformation:** Este modelo representa as informações completas relacionadas com uma viagem. Ele possui a propriedade '**TripRoute**' que representa o objeto do tipo '**TripRoute**', e a propriedade '**Coordinates**' que representa uma lista de objetos do tipo '**Coordinate**'. Este modelo permite agrupar a informação relacionada com uma rota de viagem. Desta forma, é possível submeter na base de dados a informação completa de uma rota de viagem, sem necessidade de chamar dois *endpoints* distintos para armazenar a informação da rota, e posteriormente a informação das coordenadas.
- **Report:** Este modelo representa um reporte submetido na aplicação, podendo este ser um avistamento ou um incidente. Na tabela 37 são apresentadas todas as propriedades do modelo 'Report', incluindo uma descrição de cada propriedade.

Tabela 37. Propriedades do Modelo 'Report'

Propriedade	Descrição
ID	Identificador único da rota da viagem
UserId	Identificador do utilizador associado ao reporte
IsIncident	Indica se é um incidente ou um avistamento
IncidentType	Tipo de incidente (opcional)
DateAndTime	Data e hora do reporte
Latitude	Latitude da localização do reporte
Longitude	Longitude da localização do reporte
SightedSpeciesInformation	Lista de IDs das informações sobre as espécies envolvidas
Comment	Comentário sobre o reporte (opcional)
ImagesUrl	Lista de URLs das imagens do reporte
VideosUrl	Lista de URLs dos vídeos do reporte

- **SightedSpecieInformation:** Este modelo representa um as informações sobre uma espécie envolvida num reporte. Na tabela 38 são apresentadas todas as propriedades do modelo 'SightedSpecieInformation', incluindo uma descrição de cada propriedade.

Tabela 38. Propriedades do Modelo 'SightedSpecieInformation'

Propriedade	Descrição
ID	Identificador único da informação da espécie
IsUnknownSpecie	Indica se a espécie é desconhecida
SpecieName	Nome da espécie (opcional)
SpecieId	Identificador da espécie (opcional)
NumberIndividuals	Número de indivíduos avistados
Behaviors	Lista de comportamentos observados para a espécie

- **ReportAndSightedSpecies:** Este modelo foi implementado com o mesmo objetivo do modelo 'TripInformation'. Ele permite representar a informação completa sobre um reporte através das propriedades 'Report' e 'SightedSpeciesInformation'. A propriedade 'Report' representa um objeto do tipo 'Report' que contém a informação básica sobre o reporte, enquanto que a propriedade 'SightedSpeciesInformation' representa uma lista de objetos do tipo 'SightedSpecieInformation' que contém todas as informações sobre as espécies envolvidas no reporte. Assim, tal como na submissão das rotas de viagem, a submissão dos reportes é simplificada, apenas necessitando chamar um único *endpoint* para armazenar a informação completa de um reporte na base de dados.

– Controladores:

A API desenvolvida é composta por vários controladores responsáveis por receber e tratar todos os pedidos da API. Em cada controlador existem ações que correspondem a diferentes *endpoints* da API e a operações específicas nos dados de cada entidade. Assim sendo temos os seguintes controladores:

- **UsersController:** Lida com todas as operações relacionadas com os utilizadores na API;
- **SpeciesController:** Lida com todas as operações relacionadas com as espécies na API;

- **AnimalsController:** Lida com todas as operações relacionadas com os animais na API;
- **BadgesController:** Lida com todas as operações relacionadas com os distintivos na API;
- **TripRoutesController:** Lida com todas as operações relacionadas com as rotas de viagem na API;
- **CoordinatesController:** Lida com todas as operações relacionadas com as coordenadas das rotas de viagem na API;
- **ReportsController:** Lida com todas as operações relacionadas com os reportes (avistamentos e incidentes) na API;
- **SightedSpecieInformationsController:** Lida com todas as operações relacionadas com as informações sobre as espécies avistadas nos reportes na API;
- **UserEarnedBadgesController:** Lida com todas as operações relacionadas com os distintivos desbloqueados pelos utilizadores na API;
- **UserEarnedSpeciesController:** Lida com todas as operações relacionadas com as cartas desbloqueadas pelos utilizadores na API.

D.2 Boas Práticas

Nesta secção são apresentadas algumas boas práticas adotadas durante o desenvolvimento da API. Estas práticas tiveram como objetivo garantir o desenvolvimento da API com um código eficiente e de fácil compreensão. Entre essas boas práticas podemos destacar:

1. **Separação de responsabilidades:** Tal como já foi referido, ao desenvolver uma arquitetura MVC é garantida a separação de responsabilidades, facilitando a compreensão e manutenção do código.
2. **Utilização do formato JSON:** Este formato é o mais utilizado para o envio e receção de dados através de uma API pela facilidade de manipulação legibilidade e por ser um formato de dados *lightweight*.
3. **Uso de nomes no plural nos *endpoints*:** Para a nomenclatura dos *endpoints* não foram utilizados verbos para descrever o objetivo do *endpoint*. Cada *endpoint* foi definido de forma a permitir compreender a que coleção de dados estamos a aceder. Desta forma, através do método HTTP é possível perceber o tipo de operação que se esta a solicitar na coleção de dados específica. Ao utilizar esta nomenclatura, é evitada a criação de um URL distinto para cada operação de um método HTTP. Por exemplo, em vez de ter um *endpoint* “api/getUsers”, desenvolveu-se um *endpoint* “api/Users”. Assim, para aceder aos dados de todos os utilizadores, seria apenas necessário utilizar o método GET neste *endpoint*. Caso quisermos adicionar um novo utilizador, basta utilizar o método POST do mesmo *endpoint*, em vez de criar um novo *endpoint* com o URL “api/addUser”.
4. **Utilização de métodos de conveniência:** Para retornar as respostas HTTP comuns, utilizou-se os métodos de conveniência da classe ‘**ControllerBase**’, que é a classe base para os controladores de uma API em ASP.NET Core. Estes métodos permitem simplificar o código e tornar o processo de respostas mais eficiente, para além de facilitar a compreensão do mesmo. Na tabela 39 são apresentados alguns métodos de conveniência do ASP.NET Core, assim como a respetiva resposta HTTP.

Tabela 39. Métodos de Conveniência do ASP.NET Core e a respetiva resposta HTTP

Métodos de Conveniência	Resposta HTTP
Ok	200 (OK)
CreatedAtAction	201 (Created)
NoContent	202 (Accepted)
BadRequest	400 (Bad Request)
Unauthorized	401 (Unauthorized)
NotFound	404 (Not Found)
InternalServerError	500 (Internal Server Error)

5. **Validação das entradas:** Utilizou-se a validação das entradas fornecida pelo ASP.NET Core para garantir que os dados recebidos nos *endpoints* são corretos. Para isso, utilizaram-se atributos de validação como o '[Required]', '[StringLength(n)]' e '[EmailAddress]'.
6. **Utilização do *Object-Relational Mapping* (ORM):** Utilizou-se o *Entity Framework* (EF) como ORM, permitindo criar uma camada de abstração sobre a base de dados, e facilitando o desenvolvimento e aumentando a flexibilidade do sistema.
7. **Documentação da API:** Forneceu-se uma documentação clara da API utilizando a ferramenta *Swagger*. Esta ferramenta permite gerar automaticamente a documentação da API.
8. **Testes dos *endpoints*:** Realizou-se testes para garantir a qualidade e funcionalidade correta da API, utilizando o *Postman*.

Algumas práticas mencionadas serão especificadas com mais detalhe ao longo deste Capítulo.

D.3 Endpoints

A seguir, é apresentada uma descrição detalhada de todos os *endpoints* desenvolvidos na API do sistema AQUA:

- **Endpoint “api/Users”:** Este *endpoint* é responsável pela gestão dos dados de todos os utilizadores registados no sistema AQUA. Na tabela 40 são apresentados todos os métodos HTTP desenvolvidos para este *endpoint*, seguido de uma pequena descrição de cada método.

Tabela 40. Métodos HTTP desenvolvidos para o *endpoint* “api/Users”

Endpoint 'api/Users'	
GET	Obter todos os utilizadores do sistema.
POST	Criar um novo utilizador no sistema.

- **GET:** Este método permite obter todos os utilizadores armazenados na base de dados. A resposta da API será uma lista de objetos '**UserDto**', que contém toda a informação não considerada sensível e relevante de cada utilizador. Ao retornar a resposta, a API utiliza o método o método '**Ok**' para indicar que a solicitação foi bem-sucedida, e retorna a lista de utilizadores.
- **POST:** Este método corresponde ao registo de um novo utilizador no sistema. Este método recebe no corpo do pedido um objeto do tipo '**User**', que contém as informações do novo

utilizador, como o nome, *email* e *password*. Inicialmente é feita a verificação se já existe um utilizador com o mesmo *email* na base de dados. Caso exista, é retornada uma resposta '**BadRequest**'. Caso contrário, é gerado o *hash* e o *salt* da *password* do utilizador. Com a password encriptada, o objeto User é adicionado à base de dados, retornando como resposta o código 201, indicando que o utilizador foi registado com sucesso.

- **Endpoint “api/Users/Login”:** Este *endpoint* permite autenticar um utilizador no sistema. Para este *endpoint* apenas foi desenvolvido um método HTTP, o POST, pelo objetivo específico desta funcionalidade. O método POST é o mais adequado para esta funcionalidade pois envolve o envio de informações sensíveis de forma segura no corpo do pedido.
 - **POST:** Este método, tal como referido anteriormente, permite autenticar um utilizador no sistema. Para isso, é obtida a informação do utilizador com o email fornecido na base de dados. Se o utilizador não existir, é retornada uma resposta '**NotFound**'. Caso exista, é verificado se o utilizador se encontra ativo, ou seja, se a sua propriedade '**Active**' é '**true**'. Caso não esteja, é retornada uma resposta '**Unauthorized**'. Se o utilizador estiver ativo, a sua senha fornecida é validada, comparando o *hash* da senha armazenada na base de dados com o da senha fornecida pelo utilizador. Como resposta, a API retorna um '**Ok**'.
- **Endpoint “api/Users/{id}”:** Este *endpoint* é responsável pela gestão dos dados de um utilizador específico, identificado pelo parâmetro “id”. Na tabela 41 são apresentados todos os métodos HTTP desenvolvidos para este *endpoint*. Este *endpoint* recebe como parâmetro o ID do utilizador pretendido. Por exemplo, para aceder aos dados do utilizador com ID “3”, seria necessário enviar um pedido ao *endpoint* “api/Users/{3}” com o método GET.

Tabela 41. Métodos HTTP desenvolvidos para o *endpoint* “api/Users/{id}”

Endpoint 'api/Users/{id}'	
GET	Obter um utilizador específico do sistema.
PUT	Atualizar os dados de um utilizador específico do sistema.
DELETE	Desativa um utilizador específico do sistema.

- **GET:** Este método permite aceder aos dados do utilizador correspondente ao ID fornecido. A API tenta procurar o utilizador com o ID desejado no conjunto de utilizadores registados no sistema. Se não for encontrado, a API retorna uma resposta '**NotFound**' com uma mensagem informando que o utilizador com o ID especificado não foi encontrado. Caso seja encontrado, a API retorna o método '**Ok**' indicando que o pedido foi bem-sucedido. A resposta será um objeto '**UserDto**' contendo a informação do utilizador com o ID fornecido.
- **PUT:** Este método é responsável por atualizar as informações de um utilizador existente no sistema com base no seu ID. Ele recebe o ID do utilizador como parâmetro e um objeto '**UserDto**' com os dados atualizados do utilizador no corpo do pedido. Inicialmente, é feita a procura do utilizador correspondente ao ID fornecido na base de dados. Se ele não for encontrado, é retornada uma resposta '**NotFound**'. De seguida, é verificado se o novo email do utilizador pertence a outro utilizador. Caso exista, é retornada uma resposta '**BadRequest**' informando que o email já está em uso. Depois, são identificados os

novos IDs dos distintivos (**'Badges'**) e espécies (**'Species'**) desbloqueados pelo utilizador, comparado aos que ele já tinha. Se houverem novos distintivos, são criadas instâncias de **'UserEarnedBadge'** e adicionadas à base de dados. O mesmo é feito para as novas espécies, utilizando a classe **'UserEarnedSpecie'**. Posteriormente, as restantes propriedades do utilizados são atualizadas com base nos dados fornecidos no objeto **UserDto** e é retornada uma resposta **'Ok'** com uma mensagem indicando que a informação do utilizador foi atualizada com sucesso.

- **DELETE:** Embora o nome do método seja DELETE, este método é responsável por desativar um utilizador existente no sistema com base no ID recebido. Inicialmente, é feita a procura do utilizador correspondente ao ID fornecido na base de dados. Se ele não for encontrado, é retornada uma resposta **'NotFound'**. De seguida, a propriedade **'Active'** do utilizador é definida como **'false'**, indicando que o utilizador está desativado. Por fim, é retornada uma resposta **'Ok'** com uma mensagem indicando que o utilizador foi desativado com sucesso. Ao desativar um utilizador, o mesmo deixa de ter acesso ao sistema, contudo, desta forma é possível bloquear o acesso do utilizador ao sistema sem excluir completamente os dados relacionados com ele.
- **Endpoint “api/Animals”:** Este *endpoint* é responsável pela gestão dos dados de todos os animais do sistema AQUA. Na tabela 42 são apresentados todos os métodos HTTP desenvolvidos para este *endpoint*, seguido de uma pequena descrição de cada método.

Tabela 42. Métodos HTTP desenvolvidos para o *endpoint* “api/Animals”

Endpoint 'api/Animals'	
GET	Obter todos os animais do sistema.
POST	Criar um novo animal no sistema.

- **GET:** Este método retorna o conjunto de animais existentes no sistema e retorna uma resposta **'Ok'** indicando o sucesso da operação.
 - **POST:** Este método recebe no corpo do pedido um objeto do tipo **'Animal'**, contendo toda a informação do novo animal a ser adicionado à base de dados. Se o animal já existir na base de dados, é retornada uma resposta **'BadRequest'**, indicando que o animal já existe na base de dados. Caso contrário, o mesmo é adicionado e é retornado um *status* 201.
- **Endpoint “api/Animals/{id}”:** Este *endpoint* é responsável pela gestão dos dados de um animal específico, identificado pelo parâmetro “id”. Na tabela 43 são apresentados todos os métodos HTTP desenvolvidos para este *endpoint*. Este endpoint recebe como parâmetro o ID do animal pretendido.

Tabela 43. Métodos HTTP desenvolvidos para o *endpoint* “api/Animals/{id}”

Endpoint 'api/Animals/{id}'	
GET	Obter um animal específico do sistema.
PUT	Atualizar os dados de um animal específico do sistema.
DELETE	Excluir um animal específico do sistema.

- **GET:** Este método permite aceder aos dados de um animal específico com o ID fornecido. Se o animal não for encontrado, é retornado um '**NotFound**', indicando que o animal não foi encontrado. Caso contrário, é retornado um '**Ok**' e toda a informação do animal.
 - **PUT:** Este método permite alterar a informação de um animal específico do sistema. Ele recebe no corpo do pedido um objeto do tipo '**Animal**' com os novos dados do animal. Inicialmente, é verificado se o animal existe na base de dados. Caso não exista, é retornado um '**NotFound**'. Caso contrário, a informação do animal é atualizada na base de dados e é retornado um '**Ok**' com uma mensagem indicando que o animal foi atualizado.
 - **DELETE:** Este método permite excluir um animal específico do sistema. Se o animal existir na base de dados, o mesmo é excluído e é retornado um '**Ok**' com uma mensagem indicando o sucesso da operação.
- **Endpoint “api/Species”:** Este *endpoint* é responsável pela gestão dos dados de todas as espécies do sistema AQUA. Na tabela 44 são apresentados todos os métodos HTTP desenvolvidos para este *endpoint*, seguido de uma pequena descrição de cada método. Os métodos foram desenvolvidos de forma semelhante aos métodos desenvolvidos para o *endpoint* “api/Animals”.

Tabela 44. Métodos HTTP desenvolvidos para o *endpoint* “api/Species”

Endpoint 'api/Species'	
GET	Obter todas as espécies do sistema.
POST	Criar uma nova espécie no sistema.

- **Endpoint “api/Species/{id}”:** Este *endpoint* é responsável pela gestão dos dados de uma espécie específica, identificado pelo parâmetro “id”. Na tabela 45 são apresentados todos os métodos HTTP desenvolvidos para este *endpoint*. Os métodos foram desenvolvidos de forma semelhante aos métodos desenvolvidos para o *endpoint* “api/Animals/{id}”.

Tabela 45. Métodos HTTP desenvolvidos para o *endpoint* “api/Species/{id}”

Endpoint 'api/Species/{id}'	
GET	Obter uma espécie específica do sistema.
PUT	Atualizar os dados de uma espécie específica do sistema.
DELETE	Excluir uma espécie específica do sistema.

- **Endpoint “api/Badges”:** Este *endpoint* é responsável pela gestão dos dados de todos os distintivos do sistema AQUA. Na tabela 46 são apresentados todos os métodos HTTP desenvolvidos para este *endpoint*. Os métodos foram desenvolvidos de forma semelhante aos métodos desenvolvidos para o *endpoint* “api/Animals”.

Tabela 46. Métodos HTTP desenvolvidos para o *endpoint* “api/Badges”

Endpoint ‘api/Badges’	
GET	Obter todos os distintivos do sistema.
POST	Criar um novo distintivo no sistema.

- **Endpoint “api/Badges/{id}”:** Este *endpoint* é responsável pela gestão dos dados de um distintivo específico, identificado pelo parâmetro “id”. Na tabela 47 são apresentados todos os métodos HTTP desenvolvidos para este *endpoint*. Os métodos foram desenvolvidos de forma semelhante aos métodos desenvolvidos para o *endpoint* “api/Animals/{id}”.

Tabela 47. Métodos HTTP desenvolvidos para o *endpoint* “api/Badges/{id}”

Endpoint ‘api/Badges/{id}’	
GET	Obter um distintivo específico do sistema.
PUT	Atualizar os dados de um distintivo específico do sistema.
DELETE	Excluir um distintivo específico do sistema.

- **Endpoint “api/Reports”:** Este *endpoint* é responsável pela gestão dos dados de todos os reportes (avistamentos e incidentes) do sistema AQUA. Na tabela 48 são apresentados todos os métodos HTTP desenvolvidos para este *endpoint*.

Tabela 48. Métodos HTTP desenvolvidos para o *endpoint* “api/Reports”

Endpoint ‘api/Reports’	
GET	Obter todos os reportes do sistema.
POST	Criar um novo reporte no sistema.

- **GET:** Este método é responsável por obter toda a informação sobre os reportes do sistema. Ele retorna o conjunto de reportes existentes e uma resposta ‘Ok’ indicando o sucesso da operação.
- **POST:** Este método é responsável por adicionar um novo reporte ao sistema. Ele recebe no corpo do pedido um objeto ‘ReportAndSightedSpecies’, que contém as informações do reporte e de todas as espécies associadas a esse reporte. Inicialmente, é criada uma instância de ‘SightedSpecieInformation’ com a informação de cada espécie reportada. À medida que cada instância é criada, a mesma é adicionada à base de dados e o ID do objeto é adicionado a uma lista. Ao adicionar todas as instâncias de ‘SightedSpecieInformation’, é criada uma instância de ‘Report’ com toda a informação do reporte, sendo atribuída à propriedade ‘SightedSpeciesInformation’ de ‘Report’ a lista com todos os IDs adicionados anteriormente. Por fim, o reporte é adicionado à base de dados e é retornado um ‘Ok’ indicando o sucesso da operação. Caso ocorra alguma falha, é retornado um ‘BadRequest’ com uma mensagem indicando a exceção identificada.

- **Endpoint “api/Reports/{id}”:** Este *endpoint* é responsável pela gestão dos dados de um reporte específico, identificado pelo parâmetro “id”. Na tabela 49 são apresentados todos os métodos HTTP desenvolvidos para este *endpoint*.

Tabela 49. Métodos HTTP desenvolvidos para o *endpoint* “api/Reports/{id}”

Endpoint ‘api/Reports/{id}’	
GET	Obter um reporte específico do sistema.
PUT	Atualizar os dados de um reporte específico do sistema.
DELETE	Excluir um reporte específico do sistema.

- **GET:** Este método permite aceder aos dados de um reporte específico com o ID fornecido. Se o reporte não for encontrado, é retornado um ‘**NotFound**’, indicando que o reporte não foi encontrado. Caso contrário, é retornado um ‘**Ok**’ e toda a informação do reporte.
 - **PUT:** Este método permite alterar a informação de um reporte específico do sistema. Ele recebe no corpo do pedido um objeto do tipo ‘**Report**’ com os novos dados do reporte. Inicialmente, é verificado se o reporte existe na base de dados. Caso não exista, é retornado um ‘**NotFound**’. Caso contrário, a informação do reporte é atualizada na base de dados e é retornado um ‘**Ok**’ com uma mensagem indicando o sucesso da operação.
 - **DELETE:** Este método permite excluir um reporte específico do sistema. Se o reporte existir na base de dados, o mesmo é excluído e é retornado um ‘**Ok**’ com uma mensagem indicando o sucesso da operação.
- **Endpoint “api/Reports/User/id”:** Este *endpoint* é responsável pela gestão dos dados de todos os reportes (avistamentos e incidentes) de um utilizador específico do sistema. Este *endpoint* recebe como parâmetro o ID do utilizador pretendido. Para este método apenas foi desenvolvido o método GET.
 - **GET:** Este método é responsável por obter todos os reportes de um utilizador específico. Ele procura todos os reportes do sistema que tenham como propriedade ‘**UserId**’ o ID fornecido. Caso não seja encontrado nenhum reporte, é retornado um ‘**NotFound**’ com uma mensagem indicando que o utilizador com o ID fornecido não tem reportes. Caso contrário, é retornado um ‘**Ok**’ com toda a informação dos reportes do utilizador.
 - **Endpoint “api/Reports/ByDate”:** Este *endpoint* é responsável pela gestão dos dados de todos os reportes (avistamentos e incidentes) reportados entre duas datas. Este *endpoint* recebe como parâmetro dois ‘**DateTime**’, correspondentes à data inicial (‘**startDate**’) e à data final (‘**endDate**’). Para este método apenas foi desenvolvido o método GET.
 - **GET:** Este método é responsável por obter todos os reportes entre as duas datas recebidas no corpo do pedido. Ele procura todos os reportes do sistema cuja data de reporte esteja compreendida entre as duas datas recebidas. Caso não seja encontrado nenhum reporte, é retornado um ‘**NotFound**’ com uma mensagem indicando que o utilizador com o ID fornecido não tem reportes. Caso contrário, é retornado um ‘**Ok**’ com toda a informação

dos reportes reportados entre o período recebido.

- **Endpoint “api/SightedSpeciesInformation”:** Este *endpoint* é responsável pela gestão dos dados de todas as espécies associadas a algum reporte do sistema AQUA. Na tabela 50 são apresentados todos os métodos HTTP desenvolvidos para este *endpoint*.

Tabela 50. Métodos HTTP desenvolvidos para o *endpoint* “api/SightedSpeciesInformation”

Endpoint ‘api/SightedSpeciesInformation’	
GET	Obter as espécies associadas a algum reporte do sistema.
POST	Criar uma nova espécie associada a algum reporte no sistema.

- **GET:** Este método é responsável por obter toda a informação sobre as espécies associadas a algum reporte no sistema. Ele retorna um ‘Ok’ contendo toda a informação das espécies.
 - **POST:** Este método recebe no corpo do pedido um objeto ‘SightedSpecieInformation’ e é responsável por adicionar uma nova espécie associada a algum reporte na base de dados. Como resultado é retornado um *status* 201.
- **Endpoint “api/SightedSpeciesInformation/{id}”:** Este *endpoint* é responsável pela gestão dos dados de uma espécie específica associada a algum reporte, identificado pelo parâmetro “id”. Na tabela 51 são apresentados todos os métodos HTTP desenvolvidos para este *endpoint*.

Tabela 51. Métodos HTTP desenvolvidos para o *endpoint* “api/SightedSpeciesInformation/{id}”

Endpoint ‘api/SightedSpeciesInformation/{id}’	
GET	Obter uma espécie específica associada a algum reporte do sistema.
PUT	Atualizar os dados de uma espécie específica associada a algum reporte do sistema.
DELETE	Excluir uma espécie específica associada a algum reporte do sistema.

- **GET:** Este método permite obter os dados de uma espécie específica associada a algum reporte com o ID fornecido. Se a espécie não for encontrada, é retornado um ‘NotFound’, indicando que a espécie não foi encontrada. Caso contrário, é retornado um ‘Ok’ e toda a informação da espécie específica.
- **PUT:** Este método recebe no corpo do pedido um objeto do tipo ‘SightedSpecieInformation’ com os novos dados da espécie. Inicialmente, é verificado se a espécie específica existe na base de dados. Caso não exista, é retornado um ‘NotFound’. Caso contrário, a informação da espécie é atualizada na base de dados e é retornado um ‘Ok’ com uma mensagem indicando o sucesso da operação.
- **DELETE:** Este método permite excluir uma espécie específica associada a algum reporte do sistema. Se a espécie existir na base de dados, a mesma é excluída e é retornado um ‘Ok’ com uma mensagem indicando o sucesso da operação.

- **Endpoint “api/Coordinates”:** Este *endpoint* é responsável pela gestão dos dados de todas as coordenadas das rotas de viagem do sistema AQUA. Na tabela 52 são apresentados todos os métodos HTTP desenvolvidos para este *endpoint*.

Tabela 52. Métodos HTTP desenvolvidos para o *endpoint* “api/Coordinates”

Endpoint ‘api/Coordinates’	
GET	Obter todas as coordenadas do sistema.
POST	Criar uma nova coordenada no sistema.

- **GET:** Este método é responsável por obter todas as coordenadas das rotas de viagem no sistema. Ele retorna um ‘Ok’ contendo toda a informação das coordenadas.
 - **POST:** Este método recebe no corpo do pedido um objeto ‘Coordinate’. Ele adiciona-o à base de dados e retorna um *status* 201.
- **Endpoint “api/Coordinates/{id}”:** Este *endpoint* é responsável pela gestão dos dados de uma coordenada específica, identificado pelo parâmetro “id”. Na tabela 53 são apresentados todos os métodos HTTP desenvolvidos para este *endpoint*.

Tabela 53. Métodos HTTP desenvolvidos para o *endpoint* “api/Coordinates/{id}”

Endpoint ‘api/Coordinates/{id}’	
GET	Obter uma coordenada específica do sistema.
PUT	Atualizar os dados de uma coordenada específica do sistema.
DELETE	Excluir uma coordenada específica do sistema.

- **GET:** Este método permite obter os dados de uma coordenada específica com o ID fornecido. Se a coordenada não for encontrada, é retornado um ‘NotFound’, indicando que a coordenada não foi encontrado. Caso contrário, é retornado um ‘Ok’ e toda a informação da coordenada específica.
 - **PUT:** Este método recebe no corpo do pedido um objeto do tipo ‘Coordinate’ com os novos dados da coordenada. Inicialmente, é verificado se a coordenada específica existe na base de dados. Caso não exista, é retornado um ‘NotFound’. Caso contrário, a informação da coordenada é atualizada na base de dados e é retornado um ‘Ok’ com uma mensagem indicando o sucesso da operação.
 - **DELETE:** Este método permite excluir uma coordenada específica do sistema. Se a coordenada existir na base de dados, a mesma é excluída e é retornado um ‘Ok’ com uma mensagem indicando o sucesso da operação.
- **Endpoint “api/UsersEarnedBadges”:** Este *endpoint* é responsável pela gestão dos dados dos distintivos desbloqueados pelos utilizadores do sistema. Na tabela 54 são apresentados todos os métodos HTTP desenvolvidos para este *endpoint*.

Tabela 54. Métodos HTTP desenvolvidos para o *endpoint* “api/UsersEarnedBadges”

Endpoint ‘api/UsersEarnedBadges’	
GET	Obter todos os distintivos desbloqueados pelos utilizadores do sistema.
POST	Criar um novo distintivo desbloqueado por um utilizador do sistema.

- **GET:** Este método é responsável por retornar todos os distintivos desbloqueados pelos utilizadores do sistema. Ele retorna um ‘Ok’ contendo toda a informação dos distintivos desbloqueados.
 - **POST:** Este método recebe no corpo do pedido um objeto do tipo ‘UserEarnedBadge’ com os dados de desbloqueio do distintivo por um utilizador. O objeto é armazenado na base de dados e é retornado um *status* 201.
- **Endpoint “api/UsersEarnedBadges/{id}”:** Este *endpoint* é responsável pela gestão dos dados de um distintivo específico desbloqueado por um utilizador, identificado pelo parâmetro “id”. Na tabela 55 são apresentados todos os métodos HTTP desenvolvidos para este *endpoint*.

Tabela 55. Métodos HTTP desenvolvidos para o *endpoint* “api/UsersEarnedBadges/{id}”

Endpoint ‘api/UsersEarnedBadges/{id}’	
GET	Obter um distintivo específico desbloqueado por um utilizador do sistema.
PUT	Atualizar os dados de um distintivo específico desbloqueado por um utilizador do sistema.
DELETE	Excluir um distintivo específico desbloqueado por um utilizador do sistema.

- **GET:** Este método permite obter os dados de um distintivo específico desbloqueado por um utilizador com o ID fornecido. Se o distintivo não for encontrado, é retornado um ‘NotFound’, indicando que o distintivo não foi encontrado. Caso contrário, é retornado um ‘Ok’ e toda a informação do distintivo específico.
 - **PUT:** Este método recebe no corpo do pedido um objeto do tipo ‘UserEarnedBadge’ com os novos dados do distintivo desbloqueado por um utilizador. Inicialmente, é verificado se o distintivo específico existe na base de dados. Caso não exista, é retornado um ‘NotFound’. Caso contrário, a informação do distintivo é atualizada na base de dados e é retornado um ‘Ok’ com uma mensagem indicando o sucesso da operação.
 - **DELETE:** Este método permite excluir distintivo desbloqueado por um utilizador do sistema. Se o distintivo existir na base de dados, o mesmo é excluído e é retornado um ‘Ok’ com uma mensagem indicando o sucesso da operação.
- **Endpoint “api/UsersEarnedSpecies”:** Este *endpoint* é responsável pela gestão dos dados das cartas das espécies desbloqueadas pelos utilizadores do sistema. Na tabela 56 são apresentados todos os métodos HTTP desenvolvidos para este *endpoint*. Os métodos foram desenvolvidos de forma semelhante aos métodos desenvolvidos para o *endpoint* “api/UsersEarnedBadges”.

Tabela 56. Métodos HTTP desenvolvidos para o *endpoint* “api/UsersEarnedSpecies”

Endpoint ‘api/UsersEarnedSpecies’	
GET	Obter todos as cartas das espécies desbloqueadas pelos utilizadores do sistema.
POST	Criar uma nova carta de uma espécie desbloqueada por um utilizador do sistema.

- **Endpoint “api/UsersEarnedSpecies/{id}”:** Este *endpoint* é responsável pela gestão dos dados de um distintivo específico desbloqueado por um utilizador, identificado pelo parâmetro “id”. Na tabela 57 são apresentados todos os métodos HTTP desenvolvidos para este *endpoint*. Os métodos foram desenvolvidos de forma semelhante aos métodos desenvolvidos para o *endpoint* “api/UsersEarnedBadges/{id}”.

Tabela 57. Métodos HTTP desenvolvidos para o *endpoint* “api/UsersEarnedSpecies/{id}”

Endpoint ‘api/UsersEarnedSpecies/{id}’	
GET	Obter uma carta de uma espécie desbloqueada por um utilizador do sistema.
PUT	Atualizar os dados de uma carta de uma espécie desbloqueada por um utilizador do sistema.
DELETE	Excluir uma carta de uma espécie desbloqueada por um utilizador do sistema.

D.4 Entity Framework

O *Entity Framework* é um *Object-Relational Mapping* disponibilizado pelo ASP.NET Core que apresenta diversas vantagens para o desenvolvimento da API. Com ele, é possível criar uma camada de abstração sobre a base de dados, permitindo mapear entidades e relações dos modelos implementados para tabelas e colunas na base de dados relacional. Desta forma, não é necessário lidar diretamente com SQL para a realização de operações nos dados. Além disso, o EF oferece métodos que permitem realizar as operações CRUD de forma rápida e simples.

Ainda, uma das maiores vantagens do *Entity Framework* é a sua capacidade de criar modelos automaticamente a partir de uma base de dados existente, ou gerar migrações para atualizar a estrutura da base de dados. Isto significa que é possível realizar alterações nos modelos implementados e refletir essas alterações na base de dados utilizando migrações, o que facilita a manutenção e evolução do sistema AQUA.

Nesta secção será abordado como o EF foi utilizado no desenvolvimento da API do sistema AQUA, destacando aspetos importantes, como a configuração do contexto da base de dados, e a simplificação das operações e migrações de dados.

- **Configuração do contexto da base de dados:**

A configuração do contexto da base de dados foi feita na classe ‘**ApiDbContext**’ que apresenta uma relação de herança com a classe ‘**DbContext**’ disponibilizada pelo *Entity Framework*. Nesta classe foram definidas as propriedades ‘**DbSet**’, que correspondem aos modelos de dados que representam as tabelas da base de dados.

Esta configuração é feita no construtor, onde é recebido um objeto que contém as opções de configuração do contexto, como a ‘*connection string*’ para realizar a conexão à base de dados.

Desta forma, é possível utilizar as propriedades ‘**DbSet**’ para aceder e manipular os dados das tabelas correspondentes. Na figura 128 é possível observar as diferentes propriedades definidas.

```

public class ApiDbContext : DbContext
{
    0 referências
    public ApiDbContext(DbContextOptions<ApiDbContext> options) : base(options) { }

    11 referências
    public DbSet<User> Users { get; set; }
    6 referências
    public DbSet<Animal> Animals { get; set; }
    7 referências
    public DbSet<Specie> Species { get; set; }
    11 referências
    public DbSet<Report> Reports { get; set; }
    7 referências
    public DbSet<SightedSpecieInformation> SightedSpeciesInformation { get; set; }
    7 referências
    public DbSet<TripRoute> TripRoutes { get; set; }
    7 referências
    public DbSet<Coordinate> Coordinates { get; set; }
    6 referências
    public DbSet<Badge> Badges { get; set; }
    6 referências
    public DbSet<UserEarnedBadge> UsersEarnedBadges { get; set; }
    6 referências
    public DbSet<UserEarnedSpecie> UsersEarnedSpecies { get; set; }
}

```

Figura 128. Propriedades do tipo 'DbSet' da classe 'ApiDbContext'

Para obter todos os utilizadores do sistema, pode utilizar-se '`apiContext.Users.ToList()`', sendo '`apiContext`' uma instância do '`ApiDbContext`'. Ao seguir esta abordagem, o acesso aos dados é simplificado, visto que as operações nos dados podem ser facilmente realizadas diretamente nas propriedades '`DbSet`'.

– Implementação dos modelos:

Na implementação dos modelos apresentados anteriormente foram utilizadas anotações que são disponibilizadas pelo *Entity Framework* para configurar o mapeamento dos modelos criados com as tabelas da base de dados. Entre essas anotações podemos destacar as seguintes:

- **[Table("NomeDaTabela")]:** Utilizou-se esta anotação para especificar o nome da tabela da base de dados associada ao modelo criado. Desta forma, o EF consegue fazer o mapeamento correto entre o modelo e a tabela correspondente na base de dados. Por exemplo, na figura 129 é possível observar a utilização da anotação '`[Table("user")]`' na classe '`User`'. Assim, é possível especificar que o modelo '`User`' refere-se à tabela '`user`' da base de dados.

```

[Table("user")]
4 referências
public class User
{

```

Figura 129. Utilização da anotação '[Table("user")]' na classe 'User'

- **[Key]:** Esta anotação foi utilizada para indicar a propriedade do modelo que representa a chave primária da tabela, permitindo que o EF identifique corretamente a chave primária na realização de operações nos dados.
- **[Required]:** Utilizou-se esta anotação para indicar que uma propriedade é obrigatória, ou seja, que não pode ser nula na base de dados.
- **[StringLength(n)]:** Utilizou-se esta anotação para definir o tamanho máximo de uma propriedade do tipo *String*, onde '`n`' representa o comprimento máximo dos dados.

- **[ForeignKey("Propriedade")]:** Utilizou-se esta anotação para especificar a chave estrangeira numa relação entre entidades. Isto permite que o EF estabeleça corretamente as associações entre as entidades em questão. Por exemplo, na figura 130 é possível observar a utilização desta anotação, onde é especificado que a propriedade 'UserId' e a propriedade 'BadgeId' correspondem a duas chaves estrangeiras, na classe 'UserEarnedBadge'.

```

[Table("userEarnedBadge")]
5 referências
public class UserEarnedBadge
{
    [Key]
    0 referências
    public int Id { get; set; }

    3 referências
    public int UserId { get; set; }

    [ForeignKey("UserId")]
    0 referências
    public User User { get; set; }

    3 referências
    public int BadgeId { get; set; }

    [ForeignKey("BadgeId")]
    0 referências
    public Badge Badge { get; set; }

    3 referências
    public DateTime EarnedDate { get; set; }
}

```

Figura 130. Utilização da anotação '[ForeignKey("UserId")] e '[ForeignKey("BadgeId")]' na classe 'UserEarnedBadge'

- **[EmailAddress]:** Esta anotação foi utilizada para validar se uma determinada propriedade contém um endereço de *email* válido. Isto permite que o EF realize a validação do email fornecido automaticamente.

A utilização destas anotações disponibilizadas pelo *Entity Framework* permitiram simplificar o mapeamento entre os modelos e as tabelas, evitando problemas e inconsistências nos dados. Além disso, estas anotações facilitaram a validação dos dados.

– Operações CRUD:

O *Entity Framework* disponibiliza um conjunto de recursos que facilitam a implementação das operações CRUD (*Create, Read, Update, Delete*) na API. No sistema AQUA, as operações CRUD permitiram o desenvolvimento de todos os *endpoints* necessários para a gestão dos dados da aplicação.

Por exemplo, para a operação 'Create', responsável pela criação de um novo elemento na base de dados, o EF fornece o método 'Add' que permite adicionar o respetivo elemento à coleção de dados em questão. Por exemplo, na figura 131 como o método é utilizado para adicionar um novo distintivo (Badge) à base de dados.


```

[HttpPost]
0 referências
public IActionResult CreateBadge([FromBody] Badge badge)
{
    _apiDbContext.Badges.Add(badge);
    _apiDbContext.SaveChanges();
}

```

Figura 131. Utilização do método 'Add' para adicionar um novo distintivo à base de dados

No caso da operação 'Read', responsável pela obtenção de dados da base de dados, o EF disponibiliza o método 'Find' que recebe como parâmetro a chave primária, e permite encontrar o elemento correspondente. Também, o método 'FirstOrDefault' permite encontrar o primeiro elemento cuja condição seja verificada. Na figura 132 é possível observar a utilização do método 'FirstOrDefault' para obter os dados de uma coordenada específica a partir do seu ID.

```

[HttpGet("{id}")]
0 referências
public IActionResult GetCoordinateById(int id)
{
    var coordinate = _apiDbContext.Coordinates.FirstOrDefault(x => x.Id == id);

    if (coordinate == null)
    {
        return NotFound($"Coordinate with id '{id}' not found.");
    }

    return Ok(coordinate);
}

```

Figura 132. Utilização do método 'FirstOrDefault' para obter os dados de uma coordenada específica da base de dados

Além disso, o *Entity Framework* permite simplificar a operação 'Update', de atualização dos dados. Ao modificar diretamente um modelo correspondente, basta utilizar o método 'SaveChanges' para refletir essas alterações na base de dados.

Por fim, no caso da operação 'Delete', que é responsável por eliminar elementos, o EF disponibiliza o método 'Remove'. De igual forma, o método 'SaveChanges' é utilizado para confirmar e refletir a alteração na base de dados.

A utilização deste métodos facilitam o desenvolvimento da API e evita a necessidade de escrever *queries* em SQL diretamente, simplificando o código e melhorando a sua legibilidade.

Nas figuras 133 e 134 é possível observar como as operações 'Update' e 'Delete', respetivamente, foram realizadas utilizando os recursos disponibilizados pelo EF.

```

user.Name = userObj.Name;
user.Email = userObj.Email;
user.Description = userObj.Description;
user.Location = userObj.Location;
user.ImageUrl = userObj.ImageUrl;
user.UnlockedBadges = userObj.UnlockedBadges;
user.UnlockedSpecies = userObj.UnlockedSpecies;
user.TripRoutes = userObj.TripRoutes;
user.BadgeTitleId = userObj.BadgeTitleId;
_apiDbContext.SaveChanges();

_apiDbContext.Animals.Remove(animal);
_apiDbContext.SaveChanges();

```

Figura 134. Exemplo da operação 'Delete' de um animal

Figura 133. Exemplo da operação 'Update' de um utilizador

– **Migração da base de dados:**

Uma outra vantagem do Entity Framework é a facilidade com que é possível realizar migrações da base de dados. Ele permite que possamos utilizar diferentes sistemas gestores de base de dados, como o MySQL e PostgreSQL, sem necessidade de modificar significativamente o código.

Além disso, as migrações permitem refletir as alterações realizadas nos modelos diretamente na estrutura das tabelas da base de dados. Isto significa que, ao realizar modificações nos modelos, pode-se criar e aplicar migrações que atualizam a estrutura da base de dados de forma controlada, o que facilita a manutenção e evolução do sistema.

Graças a esta capacidade do EF, foi possível desenvolver a API e a base de dados de forma progressiva e simultânea, permitindo uma evolução flexível e controlada do sistema ao longo do tempo.