Instituto Superior de Tecnologias Avançadas

Licenciatura em Informática

Pet Project

Desenvolvimento de uma aplicação móvel em Flutter

Realizado por:

João Maria Neves Pereira Bugalho N°45176

Coordenador: Prof. Dr. Pedro Brandão

Orientadora: Prof. Dra. Dulce Mourato

Lisboa, Ano Letivo 2022/2023

Instituto Superior de Tecnologias Avançadas

Licenciatura em Informática

Pet Project

Desenvolvimento de uma aplicação móvel em Flutter

Realizado por:

João Maria Neves Pereira Bugalho N°45176

Coordenador: Prof. Dr. Pedro Brandão

Orientadora: Prof. Dra. Dulce Mourato

Lisboa, Ano Letivo 2022/2023

Agradecimentos

Quero em primeiro lugar agradecer à minha família, pelo apoio dado durante não só o desenvolvimento deste trabalho, mas como também durante todo meu percurso académico.

Por fim, quero agradecer à minha noiva, que me deu não só apoio e motivação durante o desenvolvimento do trabalho, mas também ideias para o desenvolvimento do trabalho.

Quero agradecer também à orientação disponibilizada pela docente desta unidade curricular, e também à instituição por todo o conhecimento adquirido durante o meu percurso nesta instituição.

Resumo

Este trabalho envolveu o desenvolvimento de uma aplicação móvel utilizando o Framework Flutter e a plataforma Supabase. A aplicação monitoriza ecossistemas aquáticos, destacando os parâmetros de pH, amônia e nitrato. A escolha do Flutter proporcionou uma solução multiplataforma eficiente, enquanto o Supabase permitiu o acesso a uma base de dados remota.

O projeto resultou numa aplicação funcional que regista amostras, trocas de água e gere aquários e peixes. Este trabalho realça a importância do controlo destes parâmetros na saúde dos ecossistemas aquáticos.

Palavras-Chave: Aplicação móvel; Gestão de aquários; pH; Supabase; Flutter.

Abstract

This work involved the development of a mobile application using the Flutter framework and the Supabase platform. The application monitors aquatic ecosystems, highlighting the parameters of pH, ammonia, and nitrate. The choice of Flutter provided an efficient cross-platform solution, while Supabase allowed access to a remote database.

The project resulted in a functional application that registers samples, and water changes, and manages aquariums and fish. This work highlights the importance of controlling these parameters for the health of aquatic ecosystems.

Keywords: Mobile application; Aquarium maintenance; pH; Supabase; Flutter.

Índice

Introdução 1

Revisão da Literatura 2

Materiais e Métodos de Desenvolvimento 6

Resultados e Discussão de Resultados..... 8

Conclusão..... 17

Referências Bibliográficas..... 18

Índice de Figuras

Figura 1 - Ciclo do Nitrogênio 4
Figura 2 - Fluxo Aplicacional 7
Figura 3 - Estrutura de pastas do Flutter 8
Figura 4 - Esquema da base de dados 9
Figura 5 - Classe main 10
Figura 6 - Código do ecrã Login e Registo 10
Figura 7 - Ecrã de Login 11
Figura 8 - E-mail e pop-up de autenticação 11
Figura 9 - E-mail de confirmação 12
Figura 10 - Homepage 13
Figura 11 – Ecrã Lista de Aquários 13
Figura 12 – Upload de Imagens e Widget 14
Figura 13 – Visão de Aquário 14
Figura 14 – Upload de Amostras e trocas de água 15
Figura 15 - Visão do Peixe 15
Figura 16 - Página de Perfil 16

Introdução

Este projeto consiste na implementação de uma aplicação para a gestão de aquários. Um aquário é um micro ecossistema que reproduz em ambiente controlado um rio ou oceano, tendo assim processos químicos e naturais que podem ser complexos e bastante instáveis, dai ser necessário o conhecimento individual de quem gosta de ter um aquário como hobbies, e de ter uma organização à volta do mesmo, dai esta aplicação ter surgido, para ser a base desta organização.

Neste trabalho abordou-se a seleção de tecnologias, para o desenvolvimento de uma aplicação móvel voltada para o controle de ecossistemas aquáticos. Esta aplicação tem como objetivos principais facilitar a utilização e a resolução de problemas de um aquário, dando ao utilizador as ferramentas necessárias para controlar o histórico do mesmo, e os parâmetros necessários de serem controlados para resolução de problemas.

A escolha das tecnologias, como o Flutter, é fundamentada na capacidade de desenvolvimento de aplicativa móvel multiplataforma de alto desempenho. Além disso, a aplicação faz uso da plataforma Supabase para manter uma base de dados remota, fornecendo autenticação e acesso a dados em tempo real. O trabalho também destaca a importância de parâmetros críticos, como o pH, amônia e nitrogênio, para a saúde dos ecossistemas aquáticos, citando suas implicações e impactos.

Revisão da Literatura

O foco inicial do projeto, manteve-se à volta da escolha das tecnologias a utilizar no desenvolvimento deste artefacto digital. Na escolha da tecnologia utilizada para o desenvolvimento da aplicação mobile foram abordadas diversas tecnologias. Segundo Wu (2018, p.6), *"O Flutter é uma estrutura multiplataforma que visa o desenvolvimento de aplicativos móveis de alto desempenho."*, permitindo que conseguisse desenvolver uma plataforma mobile tanto para iOS como para Android utilizando apenas um código-base.

Existem diversas vantagens do Flutter enquanto Framework, sendo elas também referidas por Wu (2018):

> *O Flutter é único. Em vez de utilizar visualizações da web ou confiar nos widgets OEM do dispositivo, o Flutter renderiza todos os componentes da visualização usando seu próprio mecanismo de renderização de alto desempenho. Essa natureza oferece a possibilidade de criar aplicativos que são tão de alto desempenho quanto os aplicativos nativos podem ser (Wu, 2018, p.6).*

Utilizando o Flutter o seu próprio mecanismo de renderização permitindo não depender de "tradução" entre iOS e Android, onde existe a hipótese de causar erros específicos dependendo da plataforma, evitamos retrabalho e trabalhamos de forma estável nas duas plataformas.

Outras vantagens do Flutter também referido por Wu (2018), é a sua linguagem de programação Dart:

> *O Dart foi originalmente desenvolvido como um substituto e sucessor do JavaScript. Assim, ele implementa a maioria das características importantes do próximo padrão do JavaScript (ES7), como as palavras-chave "async" e "await". No entanto, para atrair desenvolvedores que não estão familiarizados com JavaScript, o Dart tem uma sintaxe semelhante à Java (Wu, 2018, p.7).*

Sendo que Dart se assemelha bastante a Java a nível de sintaxe, foi escolhida essa tecnologia para desenvolvimento do projeto, uma vez que tem implementadas bastantes funcionalidades existentes no JavaScript que é uma das tecnologias que é mais utilizada no desenvolvimento de aplicações móveis que por sua vez define um padrão de mercado.

Para além da escolha de tecnologias de programação também foi necessário escolher uma tecnologia onde ter uma base de dados remota e segura que permitisse a aplicação funcionar na normalidade. Para tal foi escolhida a utilização de uma plataforma chamada Supabase que fornece produtos e serviços para ajudar desenvolvedores a programarem as suas aplicações. Entre estes serviços ela fornece autenticação, base de dados PostgreSQL,

armazenamento de ficheiros e APIs auto geradas. A biblioteca de acesso ao Supabase tem suporte para as linguagens de JavaScript, Python e Dart.

Segundo Amanuel (2022) uma das grandes vantagens da Supabase é que uma API gerada automaticamente para a utilização de dados na base de dados: *"A API RESTful lida com operações CRUD e pode ser útil para recuperar dados de várias tabelas usando junções profundamente aninhadas. A API em tempo real escuta as alterações no banco de dados."* facilitando assim o desenvolvimento da aplicação cliente que consome estes dados.

Ao pesquisar sobre os parâmetros e dados essenciais para se manter um ecossistema aquático foram diversos os parâmetros abordados, entre eles o pH, a amônia e o nitrato.

O pH segundo Araújo (2021, p.19):

> *trata-se do potencial hidrogeniónico, é o que representa a concentração de íons de H+ expressa em uma escala anti-logarítmica, trazendo uma indicação do nível de acidez, neutralidade ou alcalinidade da água em uma faixa que vai de 0 a 14, sendo formado por sólidos dissolvidos e gases dissolvidos. Ocorre de forma natural por meio da dissolução de rochas, absorção de gases atmosféricos, oxidação da matéria orgânica e pelo processo de fotossíntese ou antrópica com os despejos domésticos e industriais (Araújo, 2021, p.19).*

Estas alterações de pH que acontecem de forma natural dentro de um ecossistema aquático, podem ter impactos na vida que existe dentro de um aquário como referido por Cunha (2019, p.57), *"O pH é um parâmetro muito importante nos aquários de água salgada. Uma pequena variação do seu valor pode causa reações imediatas no bem-estar da vida animal no ecossistema"*

O pH não só é importante em aquários de água salgada como é referido na citação acima como é em todos os tipos de aquários devido ao impacto que tem num ecossistema e nos organismos que o habitam como referido por Michael (n.d) *"Alguns peixes são sensíveis a mudanças repentinas no pH, mostrando sinais de angústia em quedas abruptas de até 0,2 pontos de pH."* fazendo com que ele seja dos parâmetros mais importantes a serem controlados num aquário.

A par do pH existe outro fator bastante importante dentro de um aquário que é o Ciclo de Nitrogénio. Este ciclo é responsável por eliminar o nitrogênio conforme diz Araújo (2021, p.20):

> *Nitrogênio – Encontra-se alternado em várias formas de oxidação na biosfera, sendo no meio aquático achado como nitrogênio: molecular, orgânico, amônia, nitrito e nitrato onde sua formação se dá como sólidos em suspensão e dissolvidos. É constituído naturalmente em proteínas, clorofila e vários*

outros compostos biológicos e nas atividades antrópicas como nos despejos domésticos e industriais, excrementos de animais e fertilizantes (Araújo, 2021, p.20).

Segundo a citação acima o nitrogénio é necessário ser eliminado do aquário para não se acumularem excrementos animais e comida que não foi consumida. Para isso na natureza existe este ciclo do Nitrogênio que funciona conforme está explicado figura abaixo segundo o site Aqueon.com:

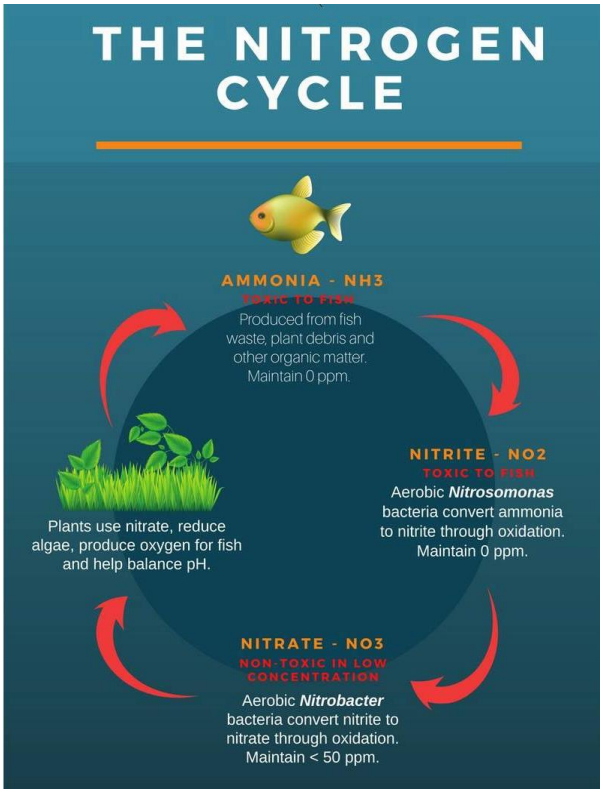

Figura 1 - Ciclo do Nitrogênio (Aqueon, 2021)

Parte deste ciclo gera amônia que segundo o site Aqueon.com:

> *é formada do metabolismo de proteínas e é o principal produto residual dos peixes. A maior parte da amônia dos peixes é excretada através das guelras, com relativamente pouco sendo perdido através da urina e fezes. A amônia também é formada quando o alimento não consumido ou outras matérias orgânicas em um aquário se decompõem. Altas concentrações de amônia na água dificultam a eliminação da amônia pelos peixes de seus corpos. Esse acúmulo de amônia pode causar estresse, danos às guelras e aos órgãos internos e, eventualmente, a morte* (Department, 2023, para.1).

Como é possível observar, os níveis de amónia têm de ser controlados dentro do ambiente de um aquário devido ao facto de ser um químico que surge quando existem

organismos que habitam o ecossistema e nunca pode existir em demasia, se não é impossível ser eliminado naturalmente por este ciclo.

Após a eliminação da amônia surge o nitrito que apesar de ser eliminado naturalmente neste ciclo natural em grandes concentrações pode causar os seguintes efeitos segundo o site Aqueon.com:

> *O nitrito inibe a troca de oxigênio para o sangue dos peixes, causando-lhes asfixia. Os peixes que estão experimentando toxicidade por nitrito geralmente respiram rapidamente e engasgam na superfície (embora existam outros motivos para isso), e seus filamentos branquiais mudam de uma cor vermelha brilhante para marrom opaco ou cinza. O único nível de nitrito seguro é zero (Aqueon, 2021, para.11)*

Controlando estes 3 fatores, conseguimos entender a saúde de um micro ecossistema aquático e resolver possíveis problemas que possam surgir na jornada de manter o mesmo.

Materiais e Métodos de Desenvolvimento

O projeto desenvolvido no decorrer do ano letivo, utilizou metodologia qualitativa com observação participante pois segundo Merriam (2015): "A observação participante é uma abordagem valiosa em pesquisas qualitativas, pois permite aos pesquisadores aprofundar a compreensão dos fenômenos sociais em seu contexto natural.".

Esta metodologia traz vantagens para o desenvolvimento deste tipo de aplicações porque tal como diz Merriam (2015): *"A observação participante permite que os pesquisadores se envolvam ativamente no ambiente estudado, o que, por sua vez, facilita a obtenção de dados ricos e a interpretação dos significados subjacentes aos comportamentos observados.".*

Numa primeira fase, realizou-se uma revisão bibliográfica, utilizando artigos científicos como principal fonte, de modo a definir parâmetros de controle de micro ecossistemas aquáticos. Esta revisão bibliográfica teve como objetivo também, a análise de tecnologias de modo a determinar quais utilizar, de modo a construir uma aplicação cujo uso seja útil ao utilizador. Após a revisão bibliográfica, seguiu-se para a definição das linguagens de programação a utilizar no projeto, assim como as Frameworks utilizadas.

O desenvolvimento deste projeto deu-se da seguinte maneira: em Fevereiro foi efetuada a leitura de artigos científicos e levantamento de requisitos, de Março a Abril foi feita a definição do style guide, de Abril a Junho foi desenvolvida a aplicação ao mesmo tempo que de modo continuo foi desenvolvido a parte teórica do projeto académico.

Na página seguinte irá estar representada em forma de fluxograma uma aproximação de algoritmo de como a aplicação funciona de forma lógica e como a mesma se interliga entre si.

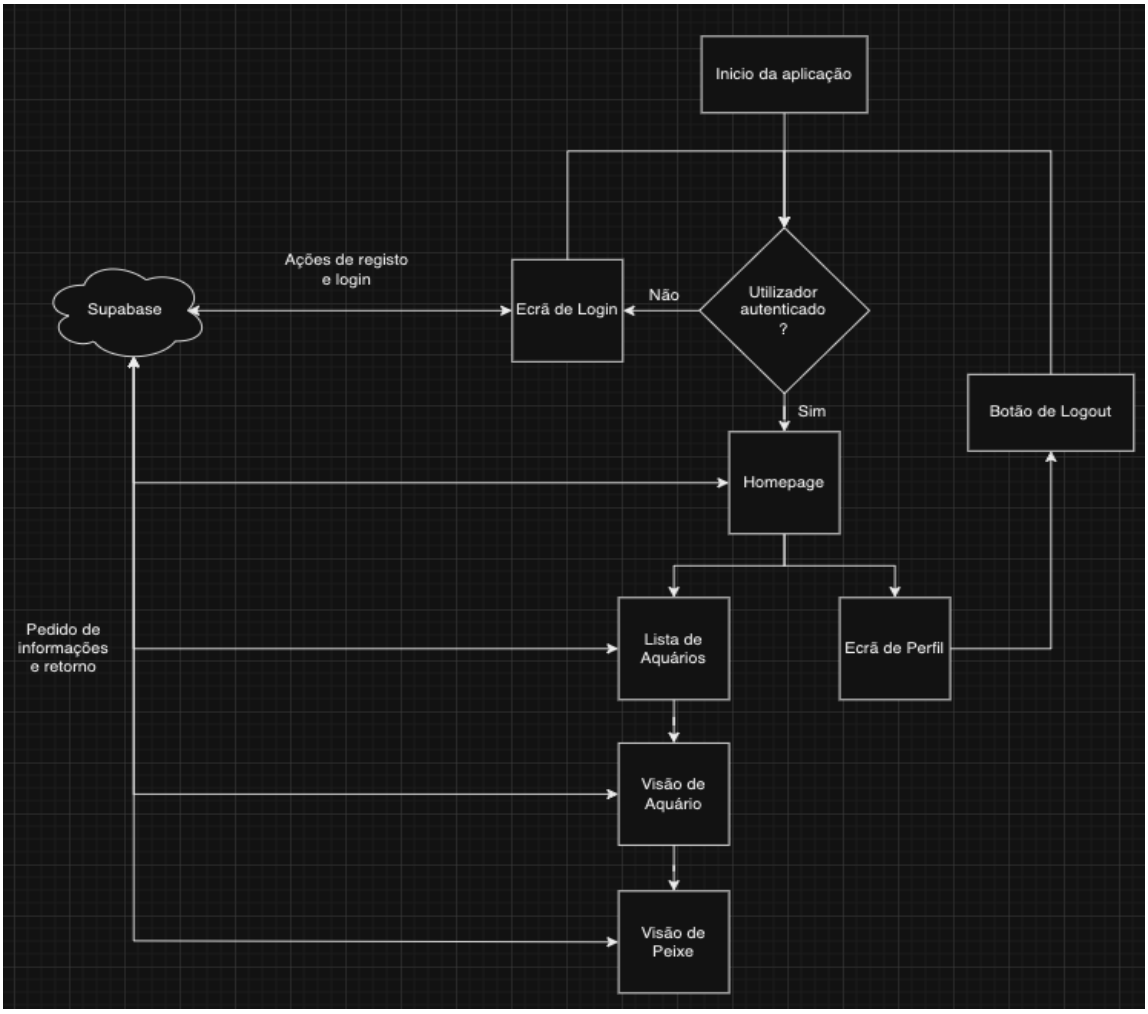

Figura 2 - Fluxo Aplicacional (Bugalho, 2023, p.1).

Resultados e Discussão de Resultados

De modo a se utilizar uma Framework multiplataforma, foi determinada a utilização do Flutter, uma vez que esta Framework contém uma grande comunidade, resultando em várias bibliotecas mantidas pela mesma, e é mantida pela Google. Esta Framework utiliza a linguagem de programação Dart, que assim como a Framework, é também mantida pela Google.

A base de dados utilizada foi PostgreSQL devido a ser uma base de dados de código aberto e gratuita com bastante facilidade de expansão e customização.

Devido à escolha destas duas tecnologias para implementação da aplicação foi utilizada Supabase, uma concorrente de código aberto da Firebase da Google que disponibiliza uma plataforma de desenvolvimento de configuração fácil e gráfica. Utilizando Supabase foi possível ter acesso a uma base de dados relacional em PostgreSQL e utilizar uma biblioteca disponibilizada pela própria plataforma para uso no Flutter.

O começo da implementação começou por gerar o projeto utilizando as ferramentas que o Flutter dispõe. Isto gera um projeto com uma estrutura semelhante à que se encontra na figura abaixo:

Figura 3 - Estrutura de pastas do Flutter (Bugalho, 2023, p.1).

Como podemos observar, ele gera as pastas: android, iOS, Linux, MacOS, web e Windows para compilação dos projetos para as diferentes plataformas suportadas. Depois gera automaticamente uma pasta build com configurações para estas compilações. O código propriamente dito do projeto encontra-se na pasta lib. Dentro dela existe um main.dart que é por onde a aplicação começa a correr.

Foi dividido por pastas as diversas utilidades de código usadas no projeto, nos screens estão todos os ecrãs da aplicação desenhados, no services os acessos às bases de dados e sistemas de autenticação definidos e nos widgets estão definidas as 3 páginas que fazem parte da barra de navegação por baixo de toda a aplicação.

A estrutura de base de dados por trás da aplicação é bastante simples, tendo apenas as relações entre as amostras de água do aquário que guardam um histórico das mesmas ficando associadas a um aquário assim como os peixes:

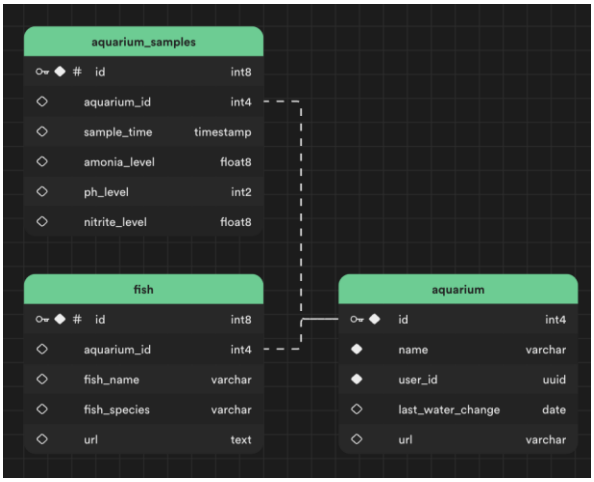

Figura 4 - Esquema da base de dados (Bugalho, 2023, p.1).

A classe main.dart foi implementada com o objetivo de fazer definido o fluxo inicial da aplicação.

Aqui foi configurado o servidor Supabase ao qual a aplicação se deveria conectar, através da função Supabase.initialize e passando os parâmetros de Login no Supabase e depois chamando o runApp(). Esta função vai correr a aplicação chamando o primeiro Widget Stateless da aplicação.

A única função deste widget é definir o layout geral da aplicação, estando definido como uma MaterialApp e definindo se deve começar a aplicação pelo login ou pela homepage, dependendo se o utilizador já está autenticado em cache ou não.

```dart
import 'package:academic_project/screens/homepage.dart';
import 'package:academic_project/services/AuthService.dart';
import 'package:flutter/material.dart';
import 'package:flutter_easyloading/flutter_easyloading.dart';
import 'package:supabase_flutter/supabase_flutter.dart';
import 'package:academic_project/screens/login.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Supabase.initialize(
    url: 'https://mtawjegcdqifqwgaxhzi.supabase.co',
    anonKey:
        'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBh
  );
  runApp(MyApp());
}

final client = Supabase.instance.client;

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Academic-Project',
      home: client.auth.currentSession != null
          ? const HomePage()
          : const LoginPage(),
      builder: EasyLoading.init(),
    );
  }
```

Figura 5 - Classe main (Bugalho, 2023, p.2).

A classe login.dart vai carregar um Widget construído da seguinte maneira como está abaixo:

```dart
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('Sign In')),
      body: ListView(
        padding: const EdgeInsets.symmetric(vertical: 18, horizontal: 12),
        children: [
          TextFormField(
            controller: _emailController,
            decoration: const InputDecoration(labelText: 'Email'),
          ),
          TextFormField(
            controller: _passwordController,
            decoration: const InputDecoration(labelText: 'Password'),
            obscureText: true,
          ),
          const SizedBox(height: 18),
          ElevatedButton(
            onPressed: _isLoading ? null : _signInOtp,
            child:
                Text(_isLoading ? 'Loading' : 'Login with One Time Password'),
          ),
          ElevatedButton(
              onPressed: _isLoading ? null : _signIn,
              child: Text(
                  _isLoading ? 'Loading' : 'Login with Email and Password')),
          ElevatedButton(
              onPressed: _isLoading ? null : _signUp,
              child: Text(
                  _isLoading ? 'Loading' : 'Register with Email and Password')),
        ],
      ),
    );
  }
```

Figura 6 - Código do ecrã Login e Registo (Bugalho, 2023, p.2).

O Widget Scaffold cria um ecrã com a implementação básica de elementos com a estrutura de Material Design. Dentro do body deste Scaffold foi criada uma Lista para carregar um conjunto de elementos filhos com 2 controladores de texto, um para o e-mail e outro para password. Estes controladores já foram inicializados anteriormente antes desta função build. Também existem dois botões que vão carregar as funções de login e de registo. O resultado deste ecrã é o seguinte:

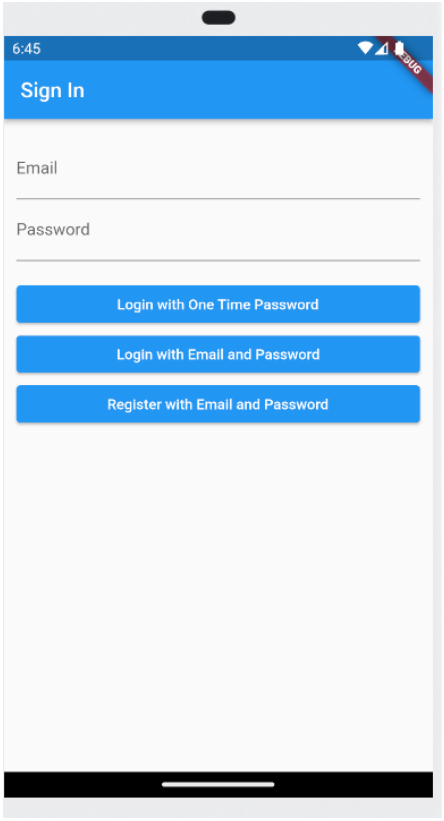

Figura 7 - Ecrã de Login (Bugalho, 2023, p.3).

Após preencher o e-mail o utilizador pode fazer a autenticação de One Time Password que envia um código para o e-mail do mesmo e abre um pop-up à semelhança das imagens abaixo:

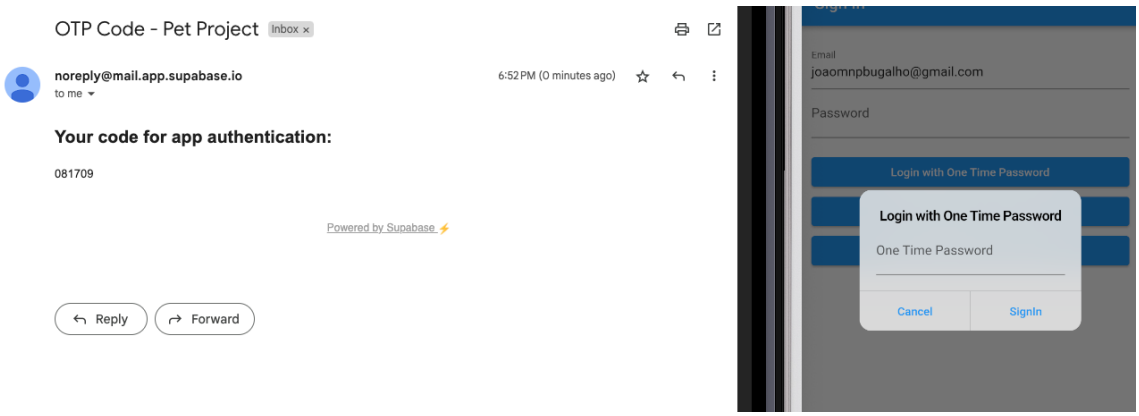

Figura 8 - E-mail e pop-up de autenticação (Bugalho, 2023, p.3).

O código ao ser colocado na caixa de texto dentro do pop-up e clicando SignIn vai autenticar o utilizador e abrir a Homepage.

Já ao efetuar um Login normal, com password, a aplicação irá tratar da autenticação com o Supabase e caso esteja correta passar para a Homepage à semelhança da autenticação anterior.

Ao registar uma conta, preenchendo os campos de e-mail e password e de seguida clicando no botão vai enviar para o e-mail preenchido uma mensagem com um link de validação de conta à semelhança da figura abaixo:

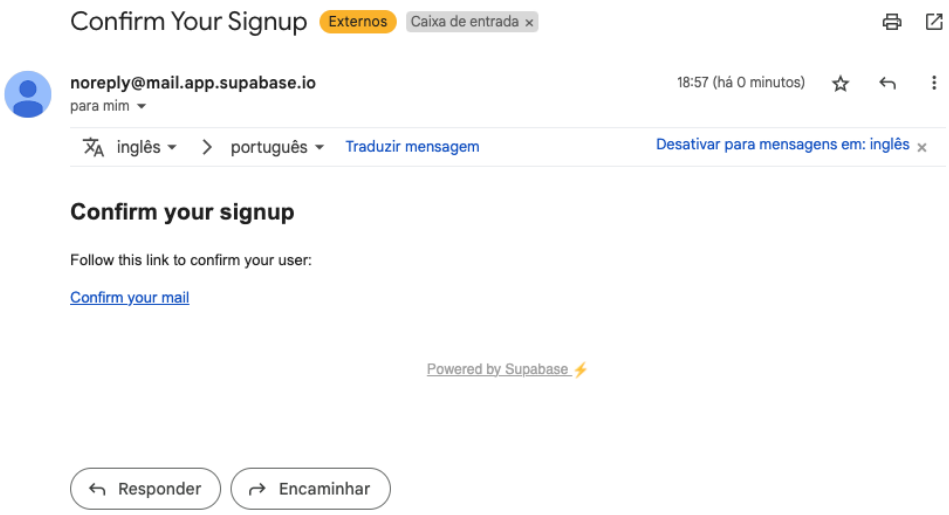

Figura 9 - E-mail de confirmação (Bugalho, 2023, p.3).

Este link presente no e-mail encaminha para um url local inexistente, mas comunica ao Supabase que o utilizador confirmou a conta. Após isto poderá voltar à aplicação e efetuar Login de forma normal.

Na página da aplicação Homepage é mostrado um overview dos diversos aquários, no caso de um utilizador que tem apenas um aquário criado que efetuou a troca de água no dia 17/09/2023 e que a última amostra foi tirada no mesmo dia. Isto permite que existindo diversos aquários o utilizador possa tomar medidas baseadas nos parâmetros apresentados. Caso a troca de água tenha sido efetuada há mais de sete dias vai apresentar a data a vermelho visto que é aconselhável efetuar essas trocas de sete em sete dias.

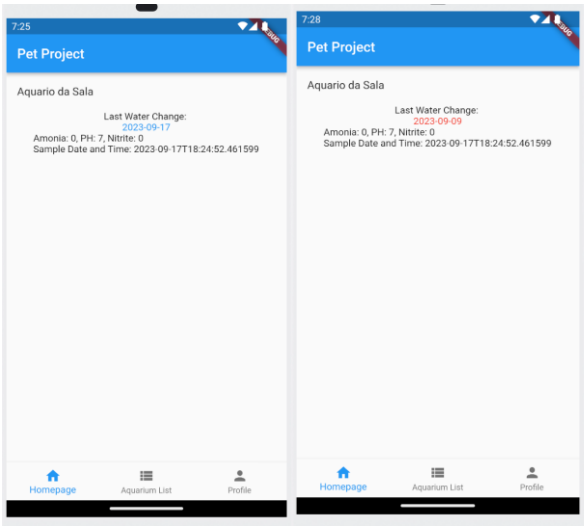

Figura 10 - Homepage (Bugalho, 2023, p.4).

No ecrã de lista de Aquários podemos visualizar e gerir os aquários já existentes ou adicionar novos aquários clicando no botão no fim do ecrã e preenchendo o nome do aquário. Também é possível apagar aquários existentes selecionando o botão de 3 pontos do lado direito de cada nome aquário.

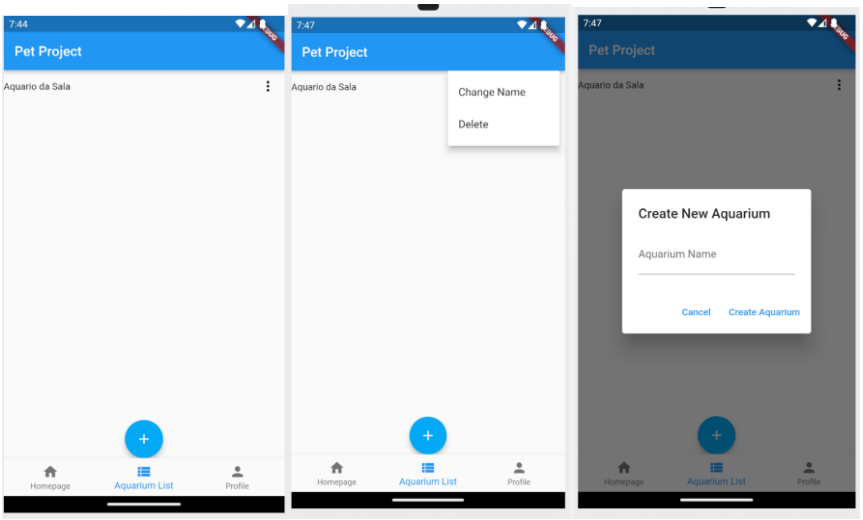

Figura 11 – Ecrã Lista de Aquários (Bugalho, 2023, p.4).

O upload de imagens para a visão de aquário e para a visão de peixe é tratado pelos respetivos widgets que tratam do upload e do load destas imagens no ecrã através do código abaixo:

```
    final picker = ImagePicker();
    final imageFile = await picker.pickImage(
      source: ImageSource.gallery,
      maxWidth: 300,
      maxHeight: 300,
    );
    if (imageFile == null) {
      return;
    }
    setState(() => _isLoading = true);

    try {
      final bytes = await imageFile.readAsBytes();
      final fileExt = imageFile.path.split('.').last;
      final fileName = '${DateTime.now().toIso8601String()}.$fileExt';
      final filePath = fileName;
      await Supabase.instance.client.storage
          .from('aquarium_images')
          .uploadBinary(
            filePath,
            bytes,
            fileOptions: FileOptions(contentType: imageFile.mimeType),
          );
      final imageUrlResponse = await Supabase.instance.client.storage
          .from('aquarium_images')
          .createSignedUrl(filePath, 60 * 60 * 24 * 365 * 10);
      widget.onUpload(imageUrlResponse);
    } on StorageException catch (error) {
      if (mounted) {
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(
            content: Text(error.message),
            backgroundColor: Theme.of(context).colorScheme.error,
          ),
        );
      }
    } catch (error) {
      if (mounted) {
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(
            content: const Text('Unexpected error occurred'),
            backgroundColor: Theme.of(context).colorScheme.error,
          ),
        );
      }
    }
```

```
@override
Widget build(BuildContext context) {
  return Column(
    children: [
      if (widget.imageUrl == null || widget.imageUrl!.isEmpty)
        ClipRRect(
            borderRadius: BorderRadius.circular(8.0),
            child: Container(
              width: 150,
              height: 150,
              color: Colors.grey,
              child: const Center(
                child: Text('No Image'),
              ),
            ))
      else
        ClipRRect(
            borderRadius: BorderRadius.circular(8.0),
            child: Image.network(
              widget.imageUrl!,
              width: 150,
              height: 150,
              fit: BoxFit.cover,
            )),
      ElevatedButton(
        onPressed: _isLoading ? null : _upload,
        child: const Text('Upload'),
      ),
    ],
  );
}
```

Figura 12 – Upload de Imagens e Widget (Bugalho, 2023, p.4).

Na visão do aquário temos diversas opções: dar upload a uma nova imagem associada a este aquário, registar uma troca simples de água, registar novos registos de amostras abrindo uma pop-up e adicionar e gerir peixes existentes.

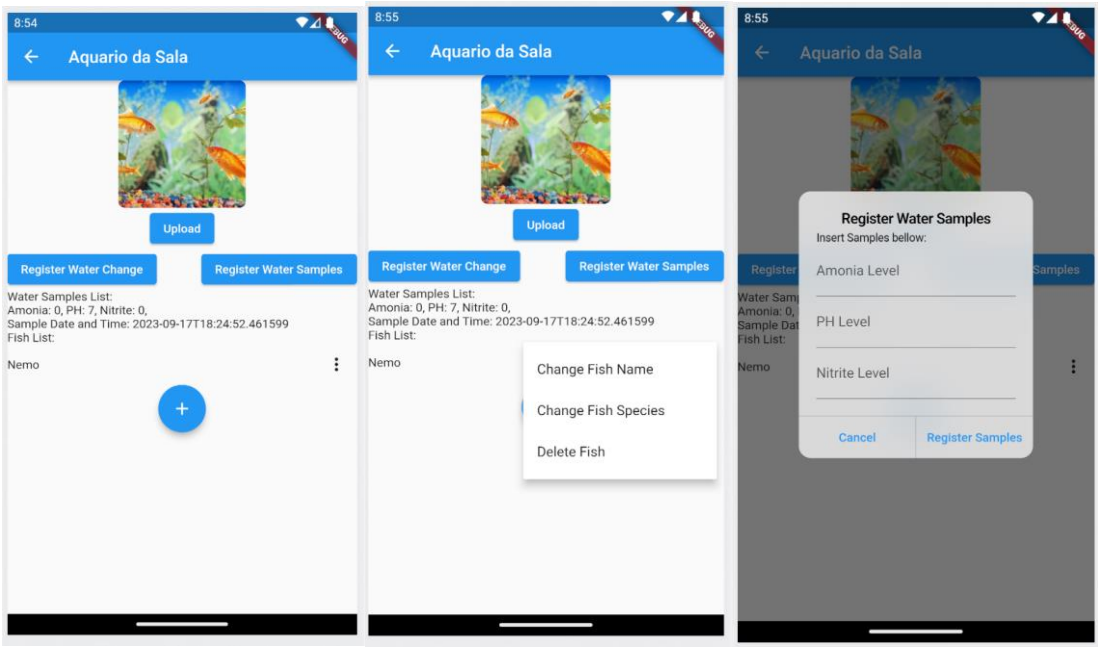

Figura 13 – Visão de Aquário (Bugalho, 2023, p.5).

O código que foi utilizado para o upload de trocas de água e de análises de água encontra-se na figura 20 abaixo:

```
Future<void> _registerWaterSamples(int aquariumId) async {
  EasyLoading.show(status: 'loading...');
  final List<Map<String, dynamic>> data =
      await Supabase.instance.client.from('aquarium_samples').insert({
    'aquarium_id': aquariumId,
    'amonia_level': double.parse(_amoniaController.text.trim()),
    'ph_level': int.parse(_phController.text.trim()),
    'nitrite_level': double.parse(_nitriteController.text.trim())
  }).select();

  setState(() {
    parametersList = data;
  });
  EasyLoading.dismiss();
}

Future<void> _registerWaterChange(int aquariumId) async {
  EasyLoading.show(status: 'loading...');
  DateTime now = DateTime.now();
  DateTime date = DateTime(now.year, now.month, now.day);
  await Supabase.instance.client.from('aquarium').update(
      {'last_water_change': date.toIso8601String()}).eq('id', aquariumId);
  EasyLoading.dismiss();
  context.showSnackBar(message: 'Water Change Registred');
}
```

Figura 14 – Upload de Amostras e trocas de água (Bugalho, 2023, p.5).

Na visão de peixe é possível editar a imagem do peixe que temos criado, em cima da mesma existe o nome do mesmo e por baixo a espécie, no caso da figura 11 abaixo, um Betta de nome Nemo.

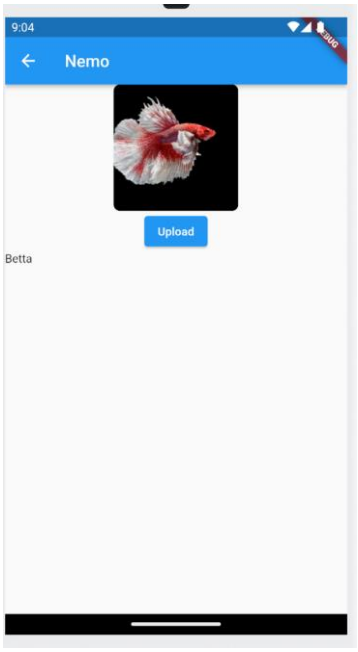

Figura 15 - Visão do Peixe (Bugalho, 2023, p.6).

É uma página bastante simples que apenas possui um botão de Sign out que vai remover a sessão deste utilizador na aplicação e encaminhá-lo para a página de Login:

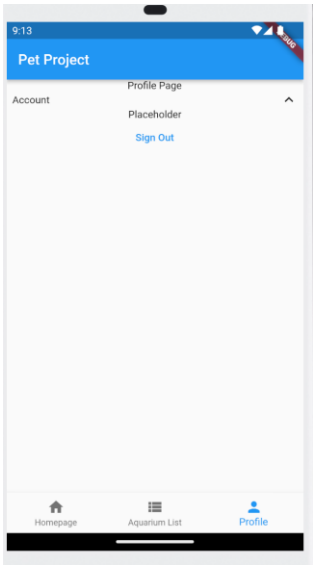

Figura 16 - Página de Perfil (Bugalho, 2023, p.6).

A vantagem desta aplicação é que permite que aos utilizadores centralizar a informação sobre parâmetros da água e servir de suporte para a gestão diária de um aquário.

Conclusão

Este projeto forneceu insights valiosos sobre desenvolvimento de aplicativos móveis e tecnologias de programação, particularmente Flutter, porque me permitiu explorar esta plataforma como um todo e conhecer mais um dos paradigmas da programação. Apesar das restrições de tempo, o projeto alcançou vários recursos essenciais para o gerenciamento de ecossistemas aquáticos, como o controlo de pH, de níveis de amónia e nitrito e também permitiu que eu conhecesse mais dos ciclos que existem dentro de um aquário.

Existem algumas funcionalidades da aplicação que não foram implementadas devido à janela de tempo disponibilizada para o desenvolvimento da mesma. Uma vez que a aplicação surgiu em contexto académico, a ideia seria no futuro aproveitar esta base desenvolvida e implementar as restantes funcionalidades, de modo a não só acabar a aplicação, mas também crescer enquanto programador e explorar esta área de mobile.

O trabalho futuro pode se concentrar em aprimorar as interfaces de usuário, expandir a funcionalidade e explorar áreas como análise de dados e notificações. As referências fornecidas fornecem uma base de conhecimento científico essencial para o desenvolvimento de um aplicativo robusto de gerenciamento de ecossistemas.

Referências Bibliográficas

Amanuel, A. (2022). *Supabase vs Firebase: Evaluation of performance and development of Progressive Web Apps.*

Aqueon (2021). Freshwater Aquarium Water Quality: The Nitrogen Cycle & Optimal Water Chemistry. URL: Aqueon.com. https://www.aqueon.com/articles/freshwater-aquarium-water-quality

Araújo, J. (2021). *Análise de qualidade da água em um aquário: a importância do sistema de filtração da água.*

Bugalho, J. (2023). *Diário de Investigação - Observação de Campo.* Não impresso. Pp. 6.

Cunha, J. (2019). *Controlador de Aquário de Água Salgada (Doctoral dissertation).*

Department, F. (2023). *Florida Department of Agriculture & Consumer Services*. Fdacs.gov. https://www.fdacs.gov/Consumer-Resources/Recreation-and-Leisure/Aquarium-Fish/Aquarium-Water-Quality-Nitrogen-Cycle

Merriam, S. B., & Tisdell, E. J. (2015). *Qualitative research: A guide to design and implementation. John Wiley & Sons.*

Michael, S. (n.d.). *Is low pH a problem in a saltwater aquarium?* https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=a21aed3946254d081fad1b531e8b71d9163001cd

Wu, W. (2018). *React Native vs Flutter, Cross-platforms mobile application frameworks.*