

CTeSP

CURSO TÉCNICO SUPERIOR PROFISSIONAL

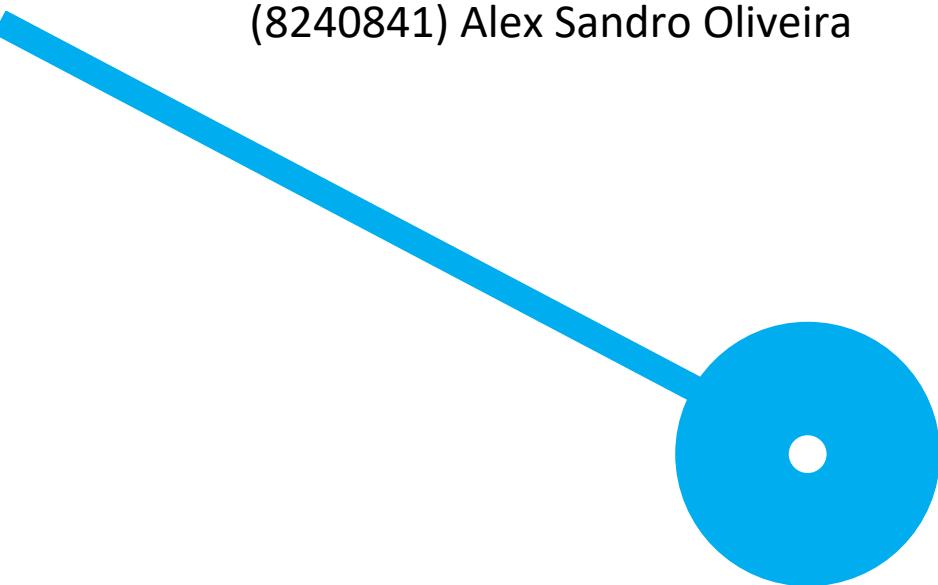
Desenvolvimento para a Web e Dispositivos Móveis

matchPet

(8240579) João Carlos Teixeira Lopes

(8240582) Pedro Miguel Pinto Pereira

(8240841) Alex Sandro Oliveira



1. Introdução.....	3
1.1. Propósito do documento	3
1.2. Visão do Produto e Análise do Problema	3
1.2.1. Contexto e Problemática	3
1.2.2. Proposta de Solução	4
1.2.3. Objetivos do Produto	5
1.3. Definições, acrónimos e abreviaturas	6
1.4. Resenha do resto do documento	6
2. Requisitos do Projeto	8
2.1. Requisitos funcionais	8
2.2. Requisitos não funcionais	9
2.3. Requisitos de Bonificação	9
3. Arquitetura e Design.....	10
3.1. Arquitetura geral	10
3.2. Modelo de dados	10
4. Mockups	11
4.1. Launch Screen.....	11
4.2. Explore (Página Principal)	11
4.3. Following.....	12
4.4. Details	12
4.5. Settings	13
4.6. Filters	13
4.7. Close to you	14

4.8.	Notifications.....	14
5.	Estado das funcionalidades	15
5.1.	Implementadas	15
6.	Trabalho Futuro	16
6.1.	Integração API Petfinder.....	16
6.2.	Funcionalidades de bonificação	16
7.	Conclusão.....	17
8.	Bibliotecas e dependências	19
8.1.	Frameworks Nativos da Apple.....	19
8.2.	Referências Técnicas.....	20

1. Introdução

1.1. Propósito do documento

Este documento apresenta o relatório do projeto desenvolvido no âmbito da Unidade Curricular de Programação para Dispositivos Móveis II (PDM, do curso de Desenvolvimento para a Web e Dispositivos Móveis. O projeto consiste no desenvolvimento de uma aplicação iOS nativa para consulta e adoção de animais, integrando a API Adopt a pet.

1.2. Visão do Produto e Análise do Problema

1.2.1. Contexto e Problemática

O abandono e a falta de adoção de animais representam um desafio social significativo em Portugal e no mundo. Milhares de animais aguardam por um lar em abrigos e organizações de proteção animal, enquanto potenciais adotantes enfrentam dificuldades em encontrar informação sobre animais disponíveis para adoção.

Problemas Identificados

Para potenciais adotantes:

- Informação dispersa em múltiplas plataformas e redes sociais*
- Dificuldade em encontrar animais que correspondam às suas preferências e condições*
- Falta de acesso a informação detalhada e fotografias de qualidade*
- Impossibilidade de acompanhar animais com interesse de forma organizada*

Para organizações de proteção animal:

- Limitações na divulgação dos animais disponíveis*

- *Alcance reduzido do público potencialmente interessado*
- *Dificuldade em manter bases de dados atualizadas e acessíveis*
- *Recursos limitados para marketing e promoção individual de cada animal*

Impacto Social:

- *Taxa de adoção inferior ao desejável*
- *Permanência prolongada de animais em abrigos*
- *Custos elevados de manutenção para organizações*
- *Menor bem-estar animal devido a condições de abrigo prolongadas*

1.2.2. Proposta de Solução

A *matchPet* surge como resposta a estes desafios, apresentando-se como uma aplicação móvel nativa iOS que simplifica todo o processo de descoberta e adoção de animais.

Visão do Produto

A *matchPet* tem como objetivo ser uma plataforma prática e facilitar a adoção de animais através de dispositivos iOS.

Proposta de Valor

Para Adotantes:

- *Pesquisa inteligente: Filtros avançados por espécie, raça, idade, género e localização*
- *Descoberta facilitada: Notificações diárias com sugestões de animais*
- *Informação completa: Galerias de imagens, descrições detalhadas e vídeos*
- *Geolocalização: Descoberta de animais nas proximidades do utilizador*

Para Organizações:

- *Maior visibilidade: Exposição dos animais a um público mais amplo*
- *Aumento de adoções: Facilitação do contacto com potenciais adotantes*

Impacto Social:

- *Aceleração do processo de adoção responsável*
- *Redução do tempo de permanência em abrigos*
- *Maior sensibilização para a causa animal*
- *Contribuição para o bem-estar e proteção animal*

1.2.3. Objetivos do Produto

Objetivos de Curto Prazo (Milestone 1):

- Desenvolver estrutura base da aplicação com dados mock
- Implementar navegação e interfaces principais
- Criar sistema de favoritos com persistência local
- Estabelecer sistema de notificações

Objetivos de Médio Prazo (Milestone 2):

- Integrar API Petfinder para dados reais
- Implementar funcionalidades de filtros avançados
- Otimizar performance e gestão de cache

Objetivos de Longo Prazo:

- Implementar geolocalização para animais.
- Integração com elementos iOS (Contacts, Messages, CallKit)

- Adicionar sistema de recomendações inteligentes

1.3. Definições, acrónimos e abreviaturas

API- Application Programming Interface - Interface de programação de aplicações.

Core Data- Framework da Apple para persistência de dados e gestão de modelo de objetos.

GCD- Grand Central Dispatch - Sistema da Apple para programação concorrente.

HIG- Human Interface Guidelines - Diretrizes de design de interface da Apple.

iOS- Sistema operativo móvel da Apple.

JSON- JavaScript Object Notation - Formato de dados.

MVVMModel-View-ViewModel - Padrão arquitetural.

PDM II-Programação para Dispositivos Móveis II.

REST Representational State Transfer - Estilo arquitetural para APIs.

SwiftUI-Framework declarativo da Apple para construção de interfaces.

TTL Time To Live - Tempo de validade de cache.

UNUserNotificationCenter- Framework da Apple para gestão de notificações.

UUID Universally Unique Identifier - Identificador único universal.

Xcode- IDE oficial da Apple para desenvolvimento iOS.

1.4. Resenha do resto do documento

O documento está estruturado de forma a apresentar progressivamente todos os aspetos do projeto:

- **Secção 2** detalha os requisitos funcionais e não funcionais do sistema.
- **Secção 3** descreve a arquitetura e design da solução.
- **Secção 4** apresenta os mockups das interfaces principais.

- **Secção 5** enumera todas as funcionalidades implementadas.
- **Secção 6** conclui com reflexões.
- **Secção 7** apresentam referências e anexos.

2. Requisitos do Projeto

2.1. Requisitos funcionais

RF01 - Listagem de Animais

- *O sistema deve apresentar lista de animais disponíveis para adoção.*
- *Cada item deve mostrar: imagem, nome, espécie, raça, género, idade e localização.*
- *O sistema deve suportar paginação.*
- *O número de animais por página deve ser configurável.*

RF02 - Filtragem

- *O sistema deve permitir filtrar por espécie.*
- *O sistema deve permitir filtrar por raça.*
- *O sistema deve permitir filtrar por género.*
- *O sistema deve permitir filtrar por idade.*

RF03 - Detalhe do Animal

- *O sistema deve apresentar informação detalhada do animal selecionado.*
- *Deve incluir: galeria de imagens, descrição completa, comentários, vídeos.*
- *Deve permitir adicionar/remover dos favoritos.*

RF04 - Sistema de Following

- *O sistema deve manter lista de animais favoritos.*
- *O sistema deve permitir adicionar animais à lista.*
- *O sistema deve permitir remover animais da lista.*
- *A lista deve ser persistida localmente.*

RF05 - Notificações

- *O sistema deve enviar notificação diária com animal aleatório.*
- *A hora da notificação deve ser configurável.*
- *A notificação deve abrir o detalhe do animal ao ser aberta.*

RF06 - Definições

- *O sistema deve permitir configurar tema (claro/escuro).*
- *Deve permitir configurar hora de notificação diária.*

RF07 - Partilha

- *O sistema deve permitir partilha com elementos do iOS.*

2.2. Requisitos não funcionais

RNF01 - Performance

- *A aplicação deve responder em menos de 2 segundos a interações do utilizador.*
- *O carregamento de imagens deve ser assíncrono.*
- *A lista deve fazer scroll suave mesmo com muitos elementos.*

RNF02 - Usabilidade

- *Interface deve seguir as Human Interface Guidelines da Apple.*
- *Feedback visual para todas as ações do utilizador.*
- *Mensagens de erro claras e acionáveis.*

RNF03 - Compatibilidade

- *Suporte para ecrãs de diferentes dimensões.*

2.3. Requisitos de Bonificação

Descoberta de animais nas proximidades

- *Uso de GPS para localização.*

Interação com elementos iOS

- *Messages (enviar mensagem sobre animal).*

3. Arquitetura e Design

3.1. Arquitetura geral

A aplicação segue uma arquitetura baseada em **MVVM (Model-View-ViewModel)** adaptada ao paradigma do *SwiftUI*.

Componentes principais:

1. **Views (SwiftUI)**: Interface do utilizador reativa.
2. **Managers**: Lógica de negócio e coordenação.
3. **Core Data**: Persistência local.
4. **API Client**: Comunicação com serviços externos.
5. **User Defaults**: Configurações do utilizador.

3.2. Modelo de dados

Animal

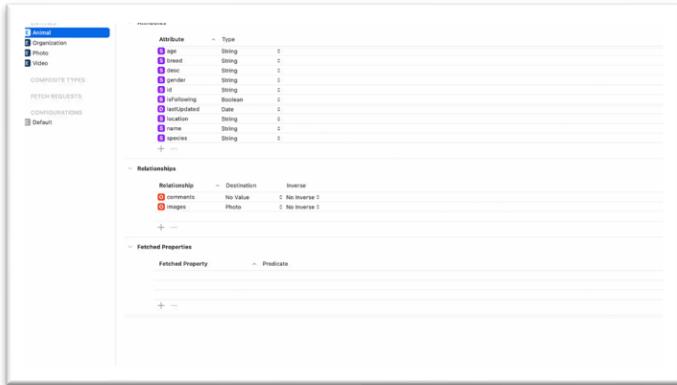


Figure 1 – Entidade Animal e respetivos atributos e relações no modelo de dados.

4. Mockups

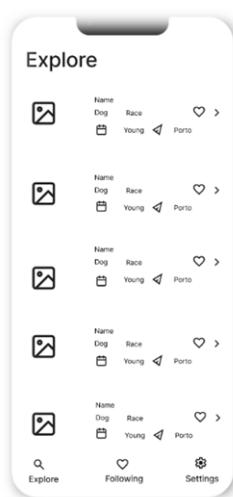
4.1. Launch Screen



Descrição: Ecrã de abertura da aplicação matchPet, apresentando o log da aplicação.

Objetivo: Criar uma introdução visual e identificar a aplicação logo no arranque. Este ecrã é exibido durante o carregamento inicial da mesma.

4.2. Explore (Página Principal)



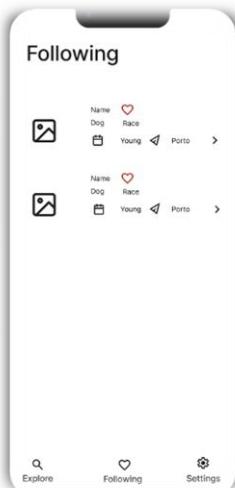
Descrição: Página principal onde o utilizador pode explorar animais disponíveis para adoção.

Elementos principais:

- Lista de animais com foto, nome, espécie, idade e localização;
- Ícones de navegação na parte inferior: explore, following, settings.

Objetivo: Permitir ao utilizador descobrir facilmente animais que estão disponíveis na plataforma.

4.3. Following

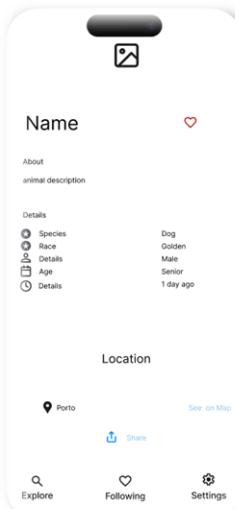


Descrição: Mostra os animais que o utilizador marcou como favoritos (ou que está “a seguir”).

Elementos principais: Lista semelhante ao ecrã Explore, mas filtrada para mostrar apenas os favoritos.

Objetivo: Facilitar o acesso rápido aos animais guardados pelo utilizador para acompanhar o estado de adoção.

4.4. Details



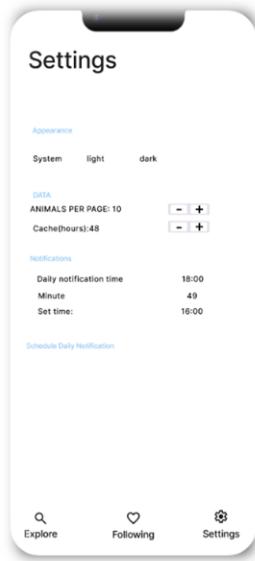
Descrição: Página individual com as informações detalhadas de um animal.

Elementos principais:

- Foto do animal;
- Nome, espécie, idade, sexo e localização;
- Botão de “Following” e opção para partilhar.

Objetivo: Apresentar informações completas sobre o animal e incentivar o contacto ou adoção.

4.5. Settings



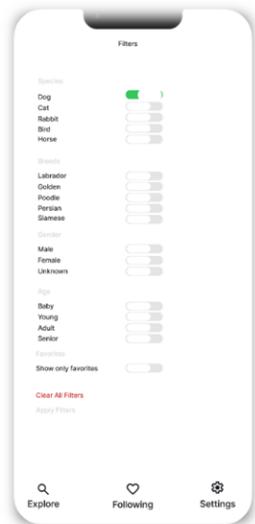
Descrição: Secção de configurações da app.

Elementos principais:

- Alteração de tema;
- Definição das notificações;
- Organização dos animais por página;
- Armazenamento em cache.

Objetivo: Apresentar informações completas sobre o animal e incentivar o contacto ou adoção.

4.6. Filters



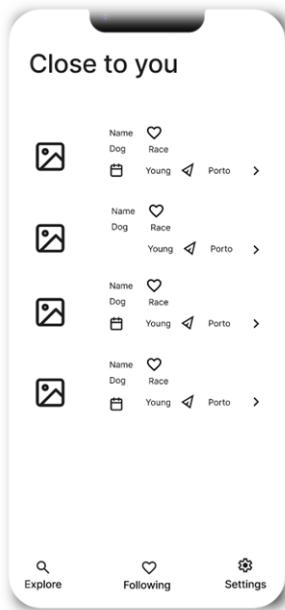
Descrição: Página que permite filtrar a pesquisa de animais com base em critérios específicos.

Elementos principais:

- Campos como espécie, idade, sexo, tamanho, raça e distância;
- Filtro para apresentar apenas animais que “seguimos”.

Objetivo: Ajudar o utilizador a otimizar os resultados e encontrar o animal ideal para adoção.

4.7. Close to you



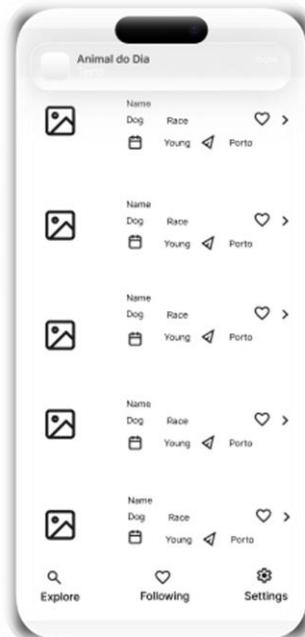
Descrição: Mostrar animais disponíveis para a adoção próximos da localização do utilizador.

Elementos principais:

-Lista de animais semelhante ao ecrã Explore, mas filtrada por proximidade geográfica.

Objetivo: Incentivar a adoção local e facilitar encontros presenciais com os animais.

4.8. Notifications



Descrição: Notificação lançada na hora preferencial selecionada nas Settings para conhecer novo animal aleatório.

Objetivo: Relembrar o utilizador com um animal, para incentivar a adoção o mesmo.

5. Estado das funcionalidades

5.1. Implementadas

- Definição do App icon e do Launch screen.
- Navegação e organização geral da aplicação.
- Definições/settings da aplicação.
- Mock data para desenvolvimento.
- Configuração do CoreData.
- Definição do data model.
- Configuração e lançamento de notificações.
- Suporte para ecrãs de diferentes dimensões .
- Integração da api Petifinder (Nem sempre disponível), o que nos obrigou a usar como fallback a api com dados estáticos, fornecida no moodle da unidade curricular.
- Notificação diária de um animal aleatório da api, agendada pelo utilizador.

5.1.1.1. Bonificação

- Implementação das funcionalidades de bonificação particularmente a descoberta de animais no mapa e a localização atual do utilizador.
- Messages (enviar mensagem sobre animal).

6. Trabalho Futuro

6.1. Integração API Petfinder

-Integração completa da API Petfinder com substituição dos dados mock por informação real e dinâmica. Consequentemente nos settings, implementar opções para: número de animais por página e duração do cache de dados

6.2. Funcionalidades de bonificação

-Implementação das funcionalidades de bonificação particularmente a descoberta de animais nas redondezas baseada na localização em que o utilizador se encontra.

7. Conclusão

O presente relatório documenta o desenvolvimento da aplicação matchPet, uma solução iOS nativa que visa facilitar e promover a adoção responsável de animais através da integração com a API Adopt a pet e a API fornecida no moodle da unidade curricular .

Resultados Alcançados

Durante o desenvolvimento do projeto, foi possível estabelecer as fundações sólidas para uma aplicação funcional. A arquitetura MVVM escolhida, aliada às tecnologias nativas da Apple (SwiftUI e Core Data), garante uma base técnica robusta e alinhada com as melhores práticas de desenvolvimento iOS.

Os mockups desenvolvidos demonstram uma interface intuitiva e visualmente apelativa, respeitando as Human Interface Guidelines da Apple. A estrutura de navegação implementada proporciona uma experiência de utilizador fluida, facilitando a descoberta de animais disponíveis para adoção.

Desafios Identificados

O principal desafio técnico foi a integração da API Adopt a Pet, que exigiu implementação cuidadosa de gestão de estados, tratamento de erros e otimização de performance, particularmente no carregamento assíncrono de imagens e gestão de cache.

A implementação das funcionalidades de bonificação, especialmente a geolocalização para descoberta de animais nas proximidades, representou um desafio adicional que envolve considerações de privacidade e permissões do utilizador.

Reflexão Final

O desenvolvimento da matchPet revelou-se uma experiência enriquecedora que combinou desafios técnicos interessantes com um propósito social significativo. A aplicação das competências adquiridas na Unidade Curricular, aliada à utilização de tecnologias modernas do ecossistema Apple, resulta num projeto com potencial real de impacto positivo para todos os apaixonados por animais.

A concretização bem-sucedida do projeto demonstra não apenas a capacidade técnica de desenvolver aplicações iOS, mas também a importância de considerar o utilizador final.

8. Bibliotecas e dependências

8.1. Frameworks Nativos da Apple

SwiftUI

- **Versão:** iOS 15.0+
- **Propósito:** Framework declarativo para construção de interfaces de utilizador
- **Utilização:** Desenvolvimento de todas as views da aplicação
- **Documentação:** <https://developer.apple.com/documentation/swiftui>

Core Data

- **Versão:** iOS 15.0+
- **Propósito:** Persistência de dados e gestão de modelo de objetos
- **Utilização:** Armazenamento local de animais favoritos e cache
- **Documentação:** <https://developer.apple.com/documentation/coredata>

UserNotifications (UNUserNotificationCenter)

- **Versão:** iOS 15.0+
- **Propósito:** Gestão e agendamento de notificações locais
- **Utilização:** Notificações diárias de animais e alertas personalizados
- **Documentação:** <https://developer.apple.com/documentation/usernotifications>

8.2. Referências Técnicas

Documentação Consultada

- **Apple Developer Documentation** - <https://developer.apple.com/documentation/>
- **Human Interface Guidelines** - <https://developer.apple.com/design/human-interface-guidelines/>
- **Swift Programming Language Guide** - <https://docs.swift.org/swift-book/>

Recursos de Aprendizagem

- *Powerpoints fornecidos na Unidade curricular*
- <https://www.youtube.com/watch?v=v1grovTJ5Y> (vídeo de auxílio à implementação da launch screen)