



UNIVERSIDADE
LUSÓFONA

The Captain Planet Initiative - DEISI2209

Trabalho Final de curso

Relatório Intercalar 2º Semestre

Rui Moreira, 21600035, LEI

Orientador: Prof. Bruno Cipriano

Co-orientador: Prof. Rodrigo Correia

Departamento de Engenharia Informática da Universidade Lusófona

Centro Universitário de Lisboa

12 de abril de 2026

www.ulusofona.pt

Direitos de cópia

The Captain Planet Initiative, Copyright de Rui Moreira, ULHT.

A Escola de Comunicação, Arquitectura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Agradecimientos

Resumo

O presente trabalho descreve o desenvolvimento da Captain Planet Initiative, uma aplicação móvel educativa orientada para a sensibilização ambiental através de mini-jogos interativos. A solução foi concebida como um hub agregador de experiências jogáveis, integrando atualmente três mini-jogos com temáticas relacionadas com reciclagem, sustentabilidade e literacia ecológica, promovendo aprendizagem ativa e envolvimento contínuo de crianças e jovens.

A implementação assenta numa arquitetura modular em monorepo, recorrendo a Flutter para a aplicação principal e a Flame para os componentes de jogo 2D. Esta abordagem permite separar responsabilidades entre o hub, os mini-jogos e a biblioteca partilhada, favorecendo reutilização, extensibilidade e manutenção evolutiva. A persistência de dados é atualmente assegurada localmente, constituindo uma base técnica para futura evolução da solução.

A análise comparativa com aplicações existentes evidencia limitações recorrentes ao nível da fragmentação temática, da profundidade pedagógica e da integração de diferentes experiências num único sistema. Neste contexto, a solução proposta distingue-se pela combinação entre estrutura modular, diversidade lúdica e alinhamento com os Objetivos de Desenvolvimento Sustentável.

Os resultados alcançados até ao momento confirmam a viabilidade técnica da abordagem e demonstram que a solução já permite validar o conceito central do projeto. A aplicação apresenta-se, assim, como uma base funcional e evolutiva para futuras extensões, nomeadamente ao nível da persistência alargada, da introdução de novos mini-jogos e da eventual integração de componentes remotas.

Palavras chave:

Gamificação, Educação Ambiental, ODS, Flutter, Serious Games, Literacia Digital

Abstract

This work describes the development of the Captain Planet Initiative, an educational mobile application aimed at promoting environmental awareness through interactive mini-games. The solution was designed as a hub that aggregates playable experiences and currently integrates three mini-games focused on recycling, sustainability, and environmental literacy, fostering active learning and sustained engagement among children and young users.

The implementation is based on a modular monorepo architecture, using Flutter for the main application and Flame for the 2D game components. This approach makes it possible to separate responsibilities between the hub, the mini-games, and the shared library, supporting reuse, extensibility, and evolutionary maintenance. Data persistence is currently ensured locally, providing a technical foundation for the future evolution of the solution.

The comparative analysis of existing applications highlights recurring limitations in terms of thematic fragmentation, pedagogical depth, and the integration of different experiences within a single system. In this context, the proposed solution stands out through its combination of modular structure, gameplay variety, and alignment with the Sustainable Development Goals.

The results achieved so far confirm the technical feasibility of the approach and show that the solution already makes it possible to validate the project's central concept. The application therefore presents itself as a functional and evolvable foundation for future extensions, particularly in terms of expanded persistence, the introduction of new mini-games, and the possible integration of remote components.

Key-words:

Gamification, Environmental Education, SDGs, Flutter, Serious Games, Digital Literacy

Índice

| | |
|---|------|
| Agradecimentos..... | iii |
| Resumo | iv |
| Abstract | v |
| Índice | vi |
| Lista de Figuras | viii |
| Lista de Tabelas | ix |
| Lista de Siglas..... | x |
| 1 Introdução | 1 |
| 1.1 Enquadramento..... | 1 |
| 1.2 Motivação e Identificação do Problema..... | 1 |
| 1.3 Objetivos | 1 |
| 1.4 Estrutura do Documento..... | 2 |
| 2 Pertinência e Viabilidade | 3 |
| 2.1 Pertinência | 3 |
| 2.2 Viabilidade..... | 3 |
| 2.2.1 Alinhamento com os Objetivos de Desenvolvimento Sustentável (ODS) | 3 |
| 2.2.2 Viabilidade Técnica | 4 |
| 2.2.3 Viabilidade Económica | 4 |
| 2.2.4 Viabilidade Social | 4 |
| 2.3 Análise Comparativa com Soluções Existentes | 5 |
| 2.3.1 Soluções existentes..... | 5 |
| 2.3.2 Análise de benchmarking..... | 7 |
| 2.4 Proposta de inovação e mais-valias | 8 |
| 2.5 Identificação de oportunidade de negócio | 8 |
| 3 Especificação e Modelação..... | 10 |
| 3.1 Análise de Requisitos..... | 10 |
| 3.1.1 Enumeração de Requisitos..... | 10 |
| 3.1.2 Descrição detalhada dos requisitos principais | 12 |
| 3.1.3 Casos de Uso/ <i>User Stories</i> | 13 |
| 3.2 Modelação..... | 14 |

| | | |
|-------|---|----|
| 3.2.1 | Modelo de dados do MVP..... | 14 |
| 3.2.2 | Modelo conceptual de evolução futura..... | 16 |
| 3.2.3 | Outras modelações relevantes..... | 17 |
| 3.3 | Protótipos de Interface..... | 18 |
| 4 | Solução Proposta..... | 19 |
| 4.1 | Apresentação..... | 19 |
| 4.2 | Arquitetura..... | 19 |
| 4.3 | Tecnologias e Ferramentas Utilizadas..... | 20 |
| 4.3.1 | Flutter..... | 21 |
| 4.3.2 | Dart..... | 21 |
| 4.3.3 | Flame..... | 21 |
| 4.3.4 | SQLite..... | 21 |
| 4.3.5 | Visual Studio Code, Git, Github e Python..... | 21 |
| 4.4 | Ambientes de Teste e de Produção..... | 22 |
| 4.5 | Abrangência..... | 22 |
| 4.6 | Componentes..... | 23 |
| 4.6.1 | Aplicação principal (Hub App)..... | 23 |
| 4.6.2 | Módulos de mini-jogo..... | 23 |
| 4.6.3 | Persistência local..... | 23 |
| 4.7 | Interfaces..... | 23 |
| 4.7.1 | Interfaces principais do hub..... | 24 |
| 4.7.2 | Ecrã do mini-jogo Eco Sort..... | 25 |
| 4.7.3 | Ecrã do mini-jogo Eco Guess..... | 26 |
| 4.7.4 | Ecrã do mini-jogo Eco Proto..... | 27 |
| 5 | Testes e Validação..... | 28 |
| 6 | Método e Planeamento..... | 29 |
| 6.1 | Planeamento inicial..... | 29 |
| 6.2 | Análise Crítica ao Planeamento..... | 30 |
| | Bibliografia..... | 32 |
| | Anexo 1 – Formulário de declaração de uso de ferramentas de Inteligência Artificial a anexar a relatório..... | 33 |
| | Glossário..... | 37 |

Lista de Figuras

| | |
|---|----|
| Figura 1 - App Recycle Game | 5 |
| Figura 2 - App Ocean Cleanup Game | 6 |
| Figura 3 - App ECOKIDS – Ecologia Infantil | 6 |
| Figura 4 - App Ponto Verde | 7 |
| Figura 5 - Diagrama de Use Cases | 14 |
| Figura 6 - Modelo de dados simplificado do MVP | 15 |
| Figura 7 - Modelo conceptual de evolução futura | 16 |
| Figura 8 - Modelo de classes simplificado da arquitetura modular | 17 |
| Figura 9 - Mapa aplicacional da solução | 18 |
| Figura 10 - Arquitectura lógica da solução proposta | 20 |
| Figura 11 - Perímetros tecnológicos da solução | 21 |
| Figura 12 - Páginas principais do hub: Jogos, Pontuações e Definições | 25 |
| Figura 13 - Ecrã representativo do mini-jogo Eco Sort | 26 |
| Figura 14 - Ecrã representativo do mini-jogo Eco Guess | 26 |
| Figura 13 - Ecrã representativo do mini-jogo Eco Proto | 27 |
| Figura 16 - Organograma em formato Gantt | 29 |
| Figura 17 - Quadro Kanban de organização para o projecto | 30 |

Lista de Tabelas

| | |
|--|----|
| Tabela 1 - Enquadramento do projecto com ODS | 4 |
| Tabela 2 - Benchmarking das soluções existentes | 7 |
| Tabela 3 - Lista de requisitos funcionais | 10 |
| Tabela 4 - Lista de requisitos não funcionais | 12 |
| Tabela 5 - Lista de User Stories | 13 |
| Tabela 6 - Abrangência de conhecimentos adquiridos nas UCs | 22 |

Lista de Siglas

| | |
|-------|--|
| API | Interface de Programação de Aplicações |
| DGE | Direção-Geral da Educação |
| HTML | Linguagem de Marcação de Hipertexto |
| JSON | JavaScript Object Notation |
| JSONB | JavaScript Object Notation Binary |
| LEI | Licenciatura em Engenharia Informática |
| ODS | Objetivos de Desenvolvimento Sustentável |
| ONU | Organização das Nações Unidas |
| PPMD | Projeto-Piloto de Manuais Digitais |
| PRR | Plano de Recuperação e Resiliência |
| RED | Recursos Educativos Digitais |
| REST | Representational State Transfer |
| SQL | Structured Query Language |
| TFC | Trabalho Final de Curso |
| UC | Unidade Curricular |

1 Introdução

A educação ambiental representa, na atualidade, um dos pilares fundamentais para garantir a sustentabilidade do planeta a longo prazo. Com a chegada das tecnologias móveis e a crescente familiaridade das camadas mais jovens com dispositivos digitais, surge a oportunidade de aliar o entretenimento à pedagogia.

Neste capítulo, descreve-se o enquadramento prático e a envolvente do problema em análise, formulando o caso de estudo que serve de base a este Trabalho Final de Curso (TFC). Será demonstrado que a solução a desenvolver responde a uma necessidade real de ferramentas educativas mais envolventes e interativas.

1.1 Enquadramento

O conceito de gamificação refere-se à aplicação de mecânicas e dinâmicas de jogos em contextos que não são de jogo, com o objetivo de motivar comportamentos e facilitar a aprendizagem. No contexto da educação ambiental, as metodologias tradicionais de ensino enfrentam, por vezes, dificuldades em captar a atenção continuada de crianças e jovens.

O projeto "The Captain Planet Initiative" enquadra-se na área do *Mobile Learning* (m-learning) e dos *Serious Games*. O uso de jogos digitais como ferramenta de sensibilização para as alterações climáticas e gestão de resíduos permite simular cenários e consequências de ações humanas, tornando conceitos abstratos em desafios tangíveis. A solução proposta visa criar um ecossistema digital onde o utilizador aprende jogando, recebendo feedback imediato sobre as suas ações virtuais, o que reforça comportamentos positivos a aplicar no mundo real.

1.2 Motivação e Identificação do Problema

A motivação para este projeto advém da urgência climática global e da observação de que a sensibilização precoce é a chave para a mudança de mentalidades [1]. No entanto, existe uma lacuna no mercado de aplicações que consigam ser, simultaneamente, divertidas ao nível de jogos comerciais e pedagogicamente rigorosas. Muitas soluções existentes são excessivamente teóricas ou carecem de mecânicas de jogo (jogabilidade) que retenham o utilizador [2].

O problema que se pretende resolver é a falta de ferramentas digitais integradas e atrativas que ensinam boas práticas ambientais (como a correta separação de lixo ou a preservação da biodiversidade) de uma forma que não seja auferida pelas crianças como "uma aula", mas sim como um desafio lúdico.

1.3 Objetivos

O objetivo geral deste trabalho é desenvolver uma plataforma multimédia educativa focada na sustentabilidade.

Os objetivos específicos são:

1. Desenvolver uma aplicação móvel (Android/iOS) que agregue, no mínimo, três mini-jogos distintos com temáticas ambientais.

2. Implementar um sistema de persistência de dados que permita o registo de pontuações localmente, visando futuramente o registo global de pontuações.
3. Promover a consciencialização sobre a reciclagem, poluição e preservação da natureza, alinhando o conteúdo com os Objetivos de Desenvolvimento Sustentável (ODS) da Organização das Nações Unidas (ONU).
4. Futuramente, criar um portal web para visualização das classificações globais, promovendo a competitividade saudável.

1.4 Estrutura do Documento

O presente relatório encontra-se organizado da seguinte forma:

- No Capítulo 2, é apresentada a análise da pertinência e viabilidade do projeto, incluindo o enquadramento face aos Objetivos de Desenvolvimento Sustentável, a análise comparativa com soluções existentes e a identificação de mais-valias da proposta.
- No Capítulo 3, detalha-se a especificação e modelação da solução, incluindo os requisitos funcionais e não funcionais, casos de uso, modelo de dados e organização da navegação aplicacional.
- No Capítulo 4, descreve-se a solução proposta, com destaque para a arquitetura, tecnologias utilizadas, ambientes de exploração, componentes principais e interfaces implementadas.
- No Capítulo 5, é apresentado o plano de testes e validação da solução, incluindo a abordagem adotada, os critérios de validação, os testes realizados e as principais limitações da fase atual.
- No Capítulo 6, descreve-se o método de trabalho seguido e o planeamento do projeto, incluindo o cronograma adotado e a análise crítica da evolução do trabalho face ao plano inicial.
- Nos anexos, são incluídos elementos complementares de apoio ao relatório, nomeadamente o guião detalhado de testes e o formulário de utilização de inteligência artificial.

2 Pertinência e Viabilidade

Neste capítulo, analisa-se a relevância do projeto face ao panorama atual e a sua capacidade de execução. Será incluído em anexo um inquérito de pertinência (a desenvolver) para validar as premissas junto do público-alvo.

2.1 Pertinência

A pertinência do The Captain Planet Initiative assenta na transformação progressiva do panorama educativo português, marcada por uma adoção crescente de Recursos Educativos Digitais (RED) e pela integração sistemática de tecnologias digitais em contexto escolar [1]. Esta tendência é evidenciada por iniciativas nacionais como o Projeto-Piloto de Manuais Digitais (PPMD)[3], que no ano letivo 2023/2024 envolveu 103 agrupamentos/escolas, 1165 turmas, 3827 professores e cerca de 23600 alunos, demonstrando a aceitação institucional e operacional dos recursos digitais no ensino básico. No ano letivo seguinte, a participação manteve-se expressiva, envolvendo 81 agrupamentos, 674 turmas e 13635 alunos, reforçando a continuidade do investimento e o interesse das comunidades educativas[4].

Paralelamente, o Plano de Recuperação e Resiliência (PRR) inclui, na sua componente dedicada à Transição Digital na Educação, um financiamento significativo orientado para a criação, aquisição e implementação de RED, formação de docentes e modernização pedagógica. Desde o lançamento da iniciativa foram contratualizados 272 milhões de euros para esta componente [5], refletindo o compromisso político com a digitalização do ensino. A este enquadramento somam-se decisões governamentais recentes que autorizam despesa destinada especificamente ao desenvolvimento de recursos digitais para o ensino básico, consolidando a viabilidade e relevância destas soluções no sistema educativo.

Neste contexto, ferramentas interativas baseadas em Gamificação têm ganho protagonismo como estratégia para aumentar o envolvimento dos alunos e facilitar a compreensão de temas complexos, nomeadamente no domínio da educação ambiental. Ao transformar conteúdos como a cadeia de reciclagem, o ciclo do carbono ou a preservação da biodiversidade em desafios lúdicos, a aplicação proposta enquadra-se nas práticas modernas de aprendizagem ativa e contribui para um acesso mais intuitivo, motivador e eficaz ao conhecimento.

2.2 Viabilidade

A viabilidade do projeto é analisada sob diversas perspetivas para garantir que a solução é exequível e sustentável.

2.2.1 Alinhamento com os Objetivos de Desenvolvimento Sustentável (ODS)

O projeto intersecta diretamente com os seguintes objetivos da Agenda 2030 da Organização das Nações Unidas (ONU) [6]apresentados na tabela seguinte.

Tabela 1 - Enquadramento do projecto com ODS

| Objetivos de Desenvolvimento Sustentável | Enquadramento no Projecto |
|---|--|
| 12 - Produção e Consumo Sustentáveis | Abordado através de jogos focados na reciclagem e redução de desperdício. |
| 13 - Ação Climática | Tema transversal a toda a aplicação, no intuito de sensibilizar os utilizadores para as consequências do aquecimento global. |
| 14 - Proteger a Vida Marinha | Abordado em mini-jogos sobre a limpeza dos oceanos e poluição plástica. |
| 15 - Proteger a Vida Terrestre | Abordado em mini-jogos sobre reflorestação e proteção da biodiversidade. |

2.2.2 Viabilidade Técnica

A viabilidade técnica do projeto é robusta. A solução assenta em tecnologias maduras, amplamente documentadas e com comunidades ativas — Flutter [7] e Flame [8] — o que reduz drasticamente o risco de bloqueios técnicos durante o desenvolvimento. A stack escolhida permite criar uma aplicação multiplataforma a partir de uma única base de código, eliminando a necessidade de equipas ou componentes paralelos para Android e iOS. O uso de bases de dados local (SQLite) e de tecnologias Web standard para o backend garantem compatibilidade, estabilidade e facilidade de manutenção. Em suma, não há dependências exóticas nem requisitos tecnológicos que coloquem em causa a execução do protótipo ou a evolução futura da solução.

2.2.3 Viabilidade Económica

Os custos associados ao projeto são reduzidos, uma vez que todo o desenvolvimento é realizado com ferramentas gratuitas ou open-source. O único investimento real prende-se com o tempo de implementação e, numa fase pós-prototipagem, com a manutenção de um servidor para alojar a API e as tabelas de classificações. Estes custos são baixos e escaláveis, permitindo ajustar a infraestrutura consoante o número de utilizadores. A distribuição digital via lojas de aplicações implica despesas residuais, limitadas eventualmente a taxas de publicação. Assim, mesmo num cenário de adoção mais alargada, o modelo económico mantém-se sustentável.

2.2.4 Viabilidade Social

A viabilidade social é elevada, porque o projeto toca num tema altamente relevante — a sustentabilidade ambiental — e apresenta-o através de uma abordagem lúdica facilmente assimilável por crianças e jovens. A familiaridade generalizada com dispositivos móveis e a crescente integração de recursos digitais nas escolas reforçam a probabilidade de aceitação.

Além disso, o formato de mini-jogos, o feedback imediato e a estética amigável alinham-se com práticas contemporâneas de aprendizagem ativa, tornando o produto socialmente pertinente e potencialmente impactante junto do público-alvo.

2.3 Análise Comparativa com Soluções Existentes

2.3.1 Soluções existentes

Atualmente, existem diversas aplicações móveis dedicadas à educação ambiental, embora a maioria apresente limitações ao nível da abrangência temática ou da profundidade pedagógica. Entre as soluções analisadas destacam-se a Recycle Game [9], a Ocean Cleanup Game [10], a ECOKIDS – Ecologia Infantil [11] e a Ponto Verde [12], que serão descritas de seguida.

2.3.1.1 Recycle Game

Aplicação focada exclusivamente na separação de resíduos. A mecânica é simples — geralmente baseada em *drag & drop* — e embora cumpra o papel de introdução aos ecopontos, carece de variedade de desafios e de conteúdos educativos adicionais. A experiência é, na prática, limitada a um único tipo de mini-jogo.



Figura 1 - App Recycle Game

2.3.1.2 Ocean Cleanup Game

Centrada na poluição marítima, esta aplicação coloca o jogador perante a tarefa de eliminar lixo do oceano. Embora possua uma componente visualmente apelativa com cores salientes, a sua utilidade educativa fica restringida ao tema específico dos oceanos sem a aprofundar, não sendo ainda uma solução integrada de literacia ambiental.



Figura 2 - App Ocean Cleanup Game

2.3.1.3 ECOKIDS – Ecologia Infantil

Aplicação mais abrangente e orientada para o público infantil, incluindo vários mini-jogos simples sobre proteção ambiental. No entanto, muitos dos seus conteúdos são estáticos, com pouca profundidade pedagógica e as mecânicas são pouco envolventes ou intuitivas, o que potencialmente limita a retenção dos utilizadores. No entanto, esta é a app que se alinha com o maior número de ODS e que se assemelha aos objectivos da solução proposta nesta comparação (12, 13, 14 e 15).



Figura 3 - App ECOKIDS – Ecologia Infantil

2.3.1.4 Ponto Verde

Aplicação oficial da Sociedade Ponto Verde (SPV) ligada à temática da reciclagem, com conteúdos e *quizzes* educativos relevantes, que recorre ao conceito de gamificação para difusão de conhecimento. Contudo, visto que o público-alvo desta app abrange um grande conjunto de faixas etárias, exige criação de conta para aceder a funcionalidades chave, o que representa uma

barreira significativa para o público infantil, quer por restrições legais associadas à idade mínima, quer pela complexidade adicional no *onboarding*. Para além disso, cada *quizz* só pode ser jogado uma única vez e à medida que o utilizador sobe de nível ou acumula pontos, novos *quizzes* são desbloqueados. As obtenções de novos níveis de estatuto do jogador habilitam-no a ganhar prémios, por isso, é possível afirmar que esta app não pretende prender a atenção do utilizador por longas sessões de jogo, mas sim cativá-lo a frequentar a app com uma periodicidade semanal, ou conforme novos passatempos surjam.

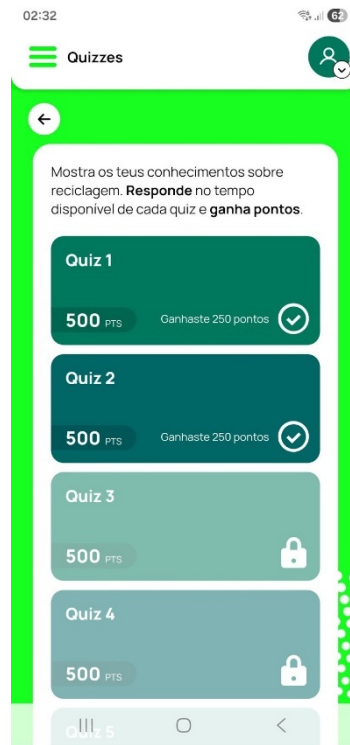


Figura 4 - App Ponto Verde

No conjunto, as soluções existentes abordam tópicos relevantes, mas fazem-no de forma fragmentada: cada aplicação cobre um único domínio ou apresenta barreiras de acesso, falta de variedade ou a carência de integração pedagógica.

2.3.2 Análise de benchmarking

Para comparar de forma estruturada as funcionalidades essenciais, foi realizado um benchmarking entre as soluções existentes e o projeto proposto. A análise focou-se em critérios críticos: variedade de mini-jogos, existência de leaderboard global, presença de conteúdos educativos explícitos, alinhamento com os ODS e acessibilidade para crianças.

A tabela seguinte sintetiza os resultados do benchmarking:

Tabela 2 - Benchmarking das soluções existentes

| Característica | Recycle Game | Ocean Cleanup | ECOKIDS | Ponto Verde | Solução Proposta |
|----------------------|--------------|---------------|---------|-------------|------------------|
| Múltiplos mini-jogos | Não | Não | Sim | Não | Sim |

| | | | | | |
|--|---------|----------|----------|---------------|-----------------------------|
| Leaderboard global | Não | Sim | Não | Não | Sim |
| Conteúdo educativo explícito | Parcial | Moderado | Moderado | Bom | Alto |
| Alinhamento com ODS | Parcial | Parcial | Parcial | Não explícito | Explícito (ODS 12,13,14,15) |
| Barreiras de acesso (ex.: criação conta) | Não | Não | Não | Sim | Não |
| Foco em múltiplos temas ambientais | Não | Não | Parcial | Parcial | Sim |

2.4 Proposta de inovação e mais-valias

O Captain Planet Initiative destaca-se pela integração, numa única aplicação, de múltiplas mecânicas de jogo e várias temáticas ambientais, cada uma alinhada de forma explícita com os ODS 12, 13, 14 e 15. Ao contrário de soluções como o ECOKIDS, que apresentam atividades simples e pouco estruturadas, o projeto propõe mini-jogos autónomos, mas coerentes, cada um concebido com um objetivo pedagógico claro e mensurável.

Esta abordagem garante não só maior diversidade lúdica, mas também uma ligação direta entre a experiência do utilizador e conteúdos educativos relevantes. A aplicação acrescenta ainda uma dimensão de competitividade saudável através de um portal web com classificações globais, reforçando a motivação e a repetição das atividades.

A arquitetura modular (hub + mini-jogos + API + website) facilita a expansão futura e permite adicionar novos conteúdos sem comprometer o sistema. Assim, a Captain Planet Initiative diferencia-se por combinar variedade, rigor pedagógico e dimensão social, oferecendo uma solução mais completa e evolutiva do que as alternativas existentes.

2.5 Identificação de oportunidade de negócio

Apesar de o projeto ter natureza académica e de o seu propósito principal ser a promoção de literacia ambiental alinhada com os Objetivos de Desenvolvimento Sustentável (ODS), a solução apresenta um conjunto de oportunidades de valorização prática que podem ser exploradas por entidades externas.

Do ponto de vista não-comercial, existe um interesse crescente por parte de escolas, autarquias e organizações não-governamentais em dispor de recursos digitais interativos que apoiam campanhas de sensibilização ambiental. A aplicação, ao combinar mecânicas de jogo com conteúdos educativos explicitamente mapeados para os ODS 12, 13, 14 e 15, pode ser integrada em atividades de sala de aula, semanas temáticas de sustentabilidade ou iniciativas municipais dirigidas ao público jovem. O formato modular em mini-jogos e a possibilidade de apresentar

classificações globais tornam a solução adequada para desafios escolares, programas de cidadania ambiental ou eventos de educação informal.

Paralelamente, existe igualmente potencial para exploração comercial, caso se opte por esse caminho. A aplicação pode ser distribuída em lojas digitais (Google Play e App Store), quer como produto gratuito de sensibilização, quer como aplicação *freemium* com funcionalidades adicionais, conteúdos educativos *premium* ou personalização para instituições. A estrutura cliente-servidor e o sistema de *leaderboards* permitem criar competições patrocinadas, parcerias com marcas sustentáveis ou versões licenciadas para escolas e empresas que desejem promover práticas ambientais junto da comunidade escolar ou dos seus colaboradores.

Assim, ainda que o objetivo inicial não seja a comercialização, o projeto revela capacidade de gerar valor social e/ou económico, podendo evoluir para um recurso educativo reutilizável em larga escala ou para um produto digital com mercado próprio.

3 Especificação e Modelação

3.1 Análise de Requisitos

A definição dos requisitos foi realizada com base na necessidade de desenvolver uma aplicação educativa interativa, centrada na aprendizagem através de mini-jogos e suportada por uma arquitetura modular. Importa salientar que o levantamento de requisitos não se limita ao estado atual da implementação, incluindo também funcionalidades previstas para evolução futura, nomeadamente integração com backend e leaderboards globais.

Durante o desenvolvimento, verificaram-se ajustes aos requisitos inicialmente definidos, sobretudo ao nível da priorização, tendo sido privilegiada a construção de um Produto Mínimo Viável (MVP) funcional.

3.1.1 Enumeração de Requisitos

A definição rigorosa dos requisitos é fundamental para o sucesso do desenvolvimento, servindo como guião técnico para a implementação da aplicação em Flutter e dos componentes de jogo em Flame. Nesta secção, os requisitos são categorizados pela sua natureza e hierarquizados quanto à sua importância (Obrigatório, Desejável ou Opcional), assegurando que o foco do desenvolvimento inicial recai sobre as funcionalidades críticas para o Produto Mínimo Viável (MVP).

3.1.1.1 Requisitos Funcionais (RF)

Os Requisitos Funcionais (RF) descrevem os comportamentos específicos do sistema e as interações permitidas aos utilizadores. A lista seguinte detalha as funcionalidades que a aplicação deve disponibilizar, abrangendo desde a navegação na interface (hub) e a mecânica dos mini-jogos, até à gestão de pontuações e persistência de dados. Estes requisitos traduzem diretamente as necessidades pedagógicas e lúdicas do projeto em especificações de software.

Tabela 3 - Lista de requisitos funcionais

| ID | Requisito | Prioridade | Impacto | Estado | Critério de Aceitação |
|-------|---------------------------------|-------------|---------|--------------|---|
| RF-01 | Ecrã inicial com lista de jogos | Obrigatório | Alto | Implementado | O utilizador terá à sua disposição a lista de ícones dos mini-jogos ao abrir a app. |
| RF-02 | Iniciar jogo a partir do hub | Obrigatório | Alto | Implementado | O jogo carrega após o toque no seu respectivo card da lista. Navegação funcional |

| | | | | | |
|-------|-------------------------------------|-------------|-------|---------------------------|---|
| RF-03 | Ecrã de seleção de dificuldade | Obrigatório | Médio | Parcialmente Implementado | Dificuldades "Fácil", "Intermédio" e "Difícil" afetam as regras do jogo. |
| RF-04 | Persistência de pontuações locais | Obrigatório | Médio | Parcialmente Implementado | As pontuações persistem localmente. |
| RF-05 | Leaderboard global | Opcional | Baixo | Não Implementado | Consulta via API dos rankings. |
| RF-06 | Leaderboard por jogo | Opcional | Baixo | Não implementado | Rankings filtrados por jogo |
| RF-07 | Feedback imediato (visual / sonoro) | Obrigatório | Alto | Parcialmente Implementado | Elementos visuais e sonoros mostram se o utilizador acerta ou erra no jogo. |
| RF-08 | Configurar sons dos mini-jogos | Obrigatório | Baixo | Parcialmente Implementado | Activa ou desactiva a reprodução de sons dos mini-jogos após clicar na opção de mute. |
| RF-09 | Elementos educativos (Dicas/Factos) | Obrigatório | Alto | Parcialmente Implementado | Exibição de curiosidade sobre o ODS no final do nível. |

3.1.1.2 Requisitos Não Funcionais (RNF)

Os Requisitos Não Funcionais (RNF) estabelecem as restrições técnicas e os critérios de qualidade sob os quais a solução deve operar. Ao contrário das funcionalidades, estes requisitos definem como o sistema deve ser construído e comportar-se. A lista abaixo foca-se em aspetos críticos como a compatibilidade multiplataforma (Android/iOS), a utilização obrigatória da stack tecnológica Flutter e Flame, o desempenho em dispositivos móveis e a conformidade com as temáticas ambientais propostas.

Tabela 4 - Lista de requisitos não funcionais

| ID | Requisito | Prioridade | Descrição | Estado |
|--------|-----------------------------|-------------|---|---------------------------|
| RNF-01 | Multiplataforma Mobile | Obrigatório | Compatível com Android e iOS. | Implementado |
| RNF-02 | Funcionamento Offline | Obrigatório | Jogável sem acesso à internet. | Implementado |
| RNF-03 | Sincronização online | Opcional | Jogável sem acesso à internet (sincroniza scores posteriormente, assim que a conexão se restabelecer). | Não Implementado |
| RNF-04 | Tecnologias Flutter & Flame | Obrigatório | O código deve ser desenvolvido utilizando a framework Flutter e biblioteca Flame. | Implementado |
| RNF-05 | Promoção de ODS | Obrigatório | Conteúdo alinhado com ODS 12, 13, 14, 15. | Parcialmente Implementado |
| RNF-06 | Autenticação API REST | Opcional | A Autenticação será efectuada através de Token simples, baseado em device_id (para MVP). Em versões futuras, pode evoluir para OAuth2 ou Firebase Auth. | Não Implementado |
| RNF-07 | API REST | Opcional | Esta interface funcionará via Protocolo HTTPS (REST) através do envio de mensagens no formato de dados JavaScript Object Notation (JSON). | Não Implementado |

3.1.2 Descrição detalhada dos requisitos principais

RF-01 — Hub de Jogos

- **Objetivo:** Centralizar acesso aos jogos
- **Descrição:** Interface com lista de jogos disponíveis
- **Critério de aceitação:** Jogos acessíveis via UI

RF-04 — Persistência de Pontuações

- **Objetivo:** Permitir acompanhar evolução do jogador
- **Descrição:** Armazenamento local de pontuações por jogo
- **Dependências:** SQLite, ScoreRepository
- **Critério de aceitação:** Dados persistem após reinício da app

RF-09 — Elementos Educativos

- **Objetivo:** Garantir valor pedagógico
- **Descrição:** Apresentação de factos ecológicos
- **Estado:** Parcial

3.1.3 Casos de Uso/*User Stories*

A representação dos cenários de utilização da solução assenta no ator principal do sistema, o jogador, que interage diretamente com a aplicação móvel através do hub e dos mini-jogos disponíveis. Nesta fase do projeto, a exploração da solução centra-se sobretudo na navegação entre módulos, na execução dos jogos e na futura integração de funcionalidades complementares, como pontuações e definições.

As user stories seguintes refletem os principais cenários de utilização identificados, articulando a experiência real de exploração da aplicação com os requisitos funcionais previamente descritos.

Tabela 5 - Lista de User Stories

| ID | Descrição |
|-------|---|
| US-01 | Como jogador, quero aceder a um hub com vários mini-jogos, para poder escolher a experiência que pretendo jogar. |
| US-02 | Como jogador, quero iniciar rapidamente um mini-jogo a partir do respetivo cartão no hub, para reduzir passos intermédios até à ação principal. |
| US-03 | Como jogador, quero experimentar diferentes tipos de mecânica, para manter a experiência variada e mais envolvente. |
| US-04 | Como jogador, quero selecionar a dificuldade quando essa opção estiver disponível, para ajustar o desafio ao meu nível. |

| | |
|-------|--|
| US-05 | Como jogador, quero que a minha pontuação fique guardada localmente, para poder acompanhar o meu desempenho ao longo do tempo. |
| US-06 | Como jogador, quero receber feedback durante ou no final do jogo, para compreender se o meu desempenho foi correto ou incorreto. |
| US-07 | Como jogador, quero receber informação ecológica associada aos mini-jogos, para aprender enquanto jogo. |
| US-08 | Como jogador, quero consultar futuramente as minhas pontuações e classificações, para comparar resultados e acompanhar a minha evolução. |

A Figura 5 apresenta o diagrama de casos de uso correspondente, evidenciando a interação do jogador com o hub, os mini-jogos, a persistência local de pontuações e as funcionalidades previstas para evolução futura.

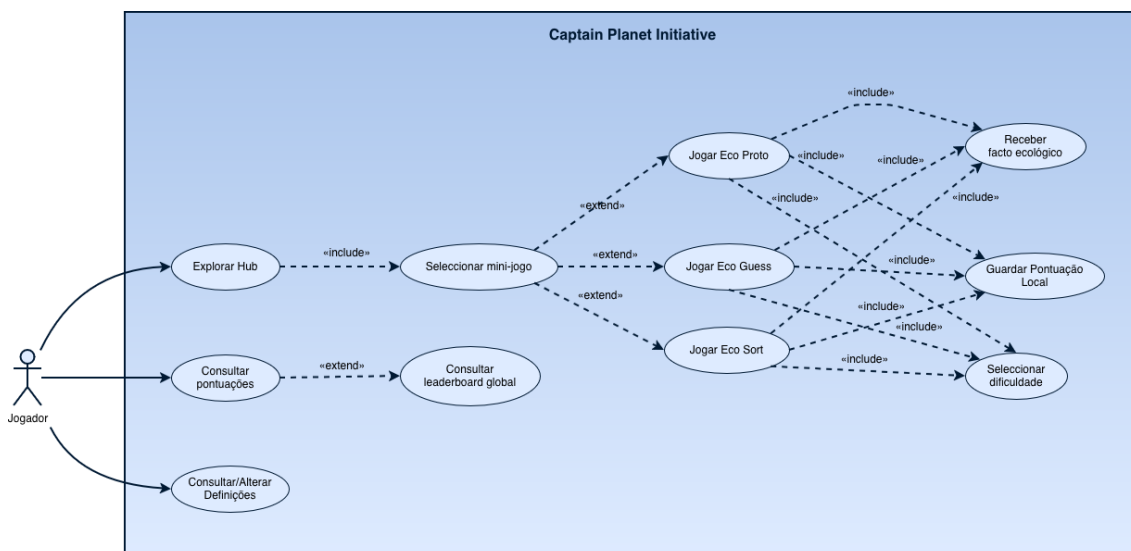


Figura 5 - Diagrama de Use Cases

3.2 Modelação

A modelação da solução deve ser analisada em duas perspetivas complementares: por um lado, o modelo de dados atualmente suportado pela versão atual da solução, centrado na persistência local de pontuações; por outro, o modelo conceptual de evolução futura, pensado para acomodar uma solução mais completa, com conteúdos educativos estruturados, classificação global e eventual integração com backend.

Esta distinção é importante porque o estado atual do projeto privilegia a validação funcional e arquitetural do núcleo da aplicação, não estando ainda totalmente materializada a estrutura de dados mais alargada prevista para fases posteriores. Assim, a modelação apresentada nesta secção procura refletir simultaneamente o que já existe de forma implementável e aquilo que permanece definido como evolução coerente da solução.

3.2.1 Modelo de dados do MVP

Na fase atual do projeto, a persistência encontra-se centrada no registo local de pontuações por jogo. Para esse efeito, foi definido um modelo simplificado, adequado ao estado de

desenvolvimento do MVP e à necessidade de suportar funcionamento offline sem dependência de serviços externos.

A principal tabela considerada nesta fase é a tabela scores, responsável por armazenar os resultados obtidos nas sessões de jogo. A sua estrutura base inclui os seguintes atributos:

- id - identificador único do registo;
- game_id - identificador do mini-jogo a que a pontuação diz respeito;
- score - valor numérico da pontuação obtida;
- metrics_json - conjunto de métricas específicas da sessão, codificadas de forma flexível;
- duration_ms - duração da sessão em milissegundos;
- created_at_ms - instante temporal de criação do registo;
- synced_at - instante temporal da sincronização, quando aplicável.

A Figura 6 apresenta o diagrama entidade-relação simplificado correspondente ao modelo atualmente suportado pela persistência local da implementação atual.

| SCORES | | | |
|---------|---------------|----|--|
| TEXT | id | PK | Identificador único da pontuação (UUID) |
| TEXT | game_id | | Identificador do mini-jogo |
| INTEGER | score | | Pontuação obtida |
| TEXT | metrics_json | | Métricas específicas da sessão |
| INTEGER | duration_ms | | Duração da sessão em ms |
| INTEGER | created_at_ms | | Timestamp de criação em epoch ms |
| INTEGER | synced_at | | Timestamp da sincronização em epoch ms; NULL se não sincronizado |

Figura 6 - Modelo de dados simplificado do MVP

A adoção deste modelo reduzido foi uma decisão deliberada. O objetivo não foi construir, desde já, uma estrutura relacional completa para todos os cenários futuros, mas sim garantir uma base persistente mínima que permita evoluir a funcionalidade de pontuações sem bloquear a implementação do restante sistema.

O uso do campo metrics_json merece particular justificação. Esta solução foi adotada para acomodar diferenças entre mini-jogos com mecânicas distintas, evitando a criação prematura de estruturas específicas por jogo. A principal vantagem desta abordagem é a flexibilidade imediata: novos módulos podem guardar métricas próprias sem obrigar a alterações frequentes ao schema local. Em contrapartida, esta opção sacrifica parte da normalização relacional e dificulta consultas mais estruturadas sobre essas métricas. Trata-se, por isso, de um compromisso típico de MVP, aceitável nesta fase, mas com margem clara para refinamento futuro.

Para além disso, a utilização de synced_at em vez de um simples indicador booleano permite representar explicitamente se a pontuação já foi sincronizada e, em caso afirmativo, em que

momento essa operação ocorreu, mantendo assim maior expressividade sem aumentar significativamente a complexidade do modelo.

3.2.2 Modelo conceptual de evolução futura

Para suportar a evolução futura da solução, foi também definido um modelo conceptual mais completo, orientado para uma estrutura relacional mais rica e mais próxima de uma normalização formal. Este modelo prevê a existência de entidades próprias para jogadores, mini-jogos, factos educativos e pontuações.

As entidades principais consideradas são as seguintes:

- **users** – representa o jogador ou dispositivo, permitindo futuramente distinguir utilizadores e suportar identificação persistente;
- **mini_games** – descreve cada mini-jogo disponível no hub, incluindo metadados como nome, descrição ou área temática;
- **facts** – armazena os factos ecológicos associados aos diferentes jogos, suportando a componente educativa da aplicação;
- **game_scores** – regista as pontuações obtidas por utilizador e por mini-jogo, incluindo métricas específicas da sessão e informação de sincronização.

Neste modelo, um utilizador pode ter várias pontuações, cada pontuação encontra-se associada a um mini-jogo e cada mini-jogo pode possuir vários factos educativos. Esta estrutura é mais adequada a uma solução com backend, classificação global e gestão mais rigorosa de conteúdos.

A Figura 7 apresenta o diagrama entidade-relação correspondente a este modelo conceptual de evolução.

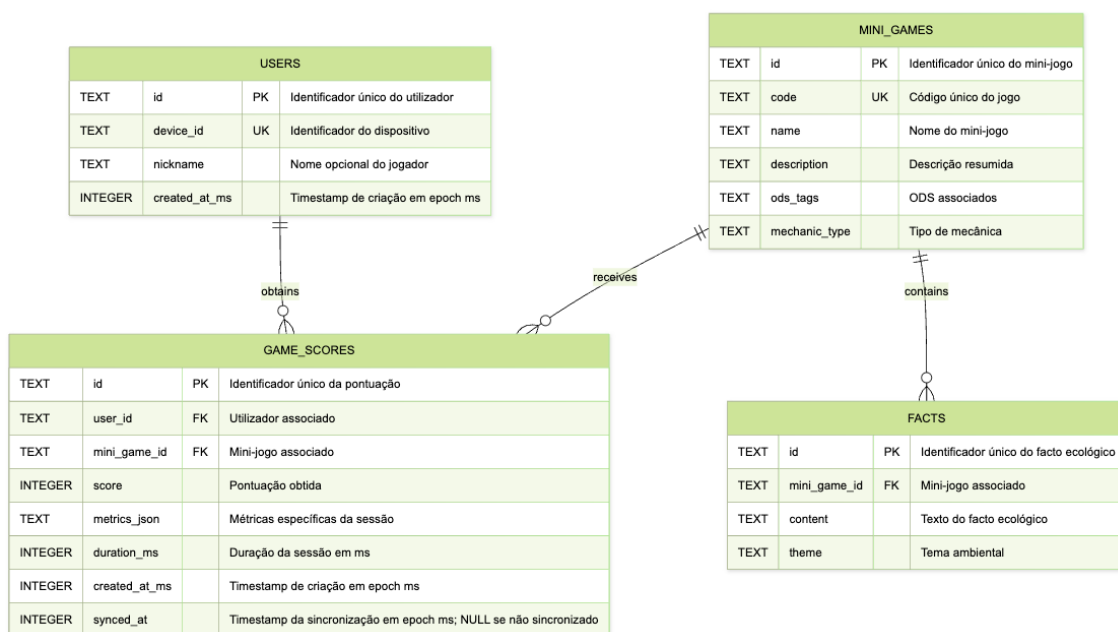


Figura 7 - Modelo conceptual de evolução futura

A manutenção deste modelo no relatório é importante por duas razões. Em primeiro lugar, mostra que o projeto não está limitado ao protótipo atual e possui uma linha de evolução coerente. Em segundo lugar, permite distinguir claramente entre decisões tomadas para

acelerar a implementação do MVP e a estrutura de dados mais robusta necessária para uma futura versão expandida da solução.

3.2.3 Outras modelações relevantes

Para além da modelação de dados, considera-se relevante apresentar uma modelação estrutural dos principais componentes de software associados à persistência e à integração modular dos jogos. Esta necessidade decorre da arquitetura adotada, assente em packages independentes e numa camada partilhada de serviços comuns.

Neste contexto, a Figura 8 apresenta um modelo de classes simplificado envolvendo os elementos centrais da persistência e da composição da aplicação, nomeadamente:

- GameModule
- GameRegistry
- LocalScoreRepositorySqlite
- ScoreEntry
- ScoreRepository

Esta modelação é pertinente porque permite compreender a relação entre o registo de módulos de jogo, a abstração de persistência e a representação das pontuações no sistema. Ao contrário do diagrama entidade-relação, que descreve sobretudo dados, o modelo de classes torna visível a forma como as responsabilidades técnicas foram distribuídas no código.

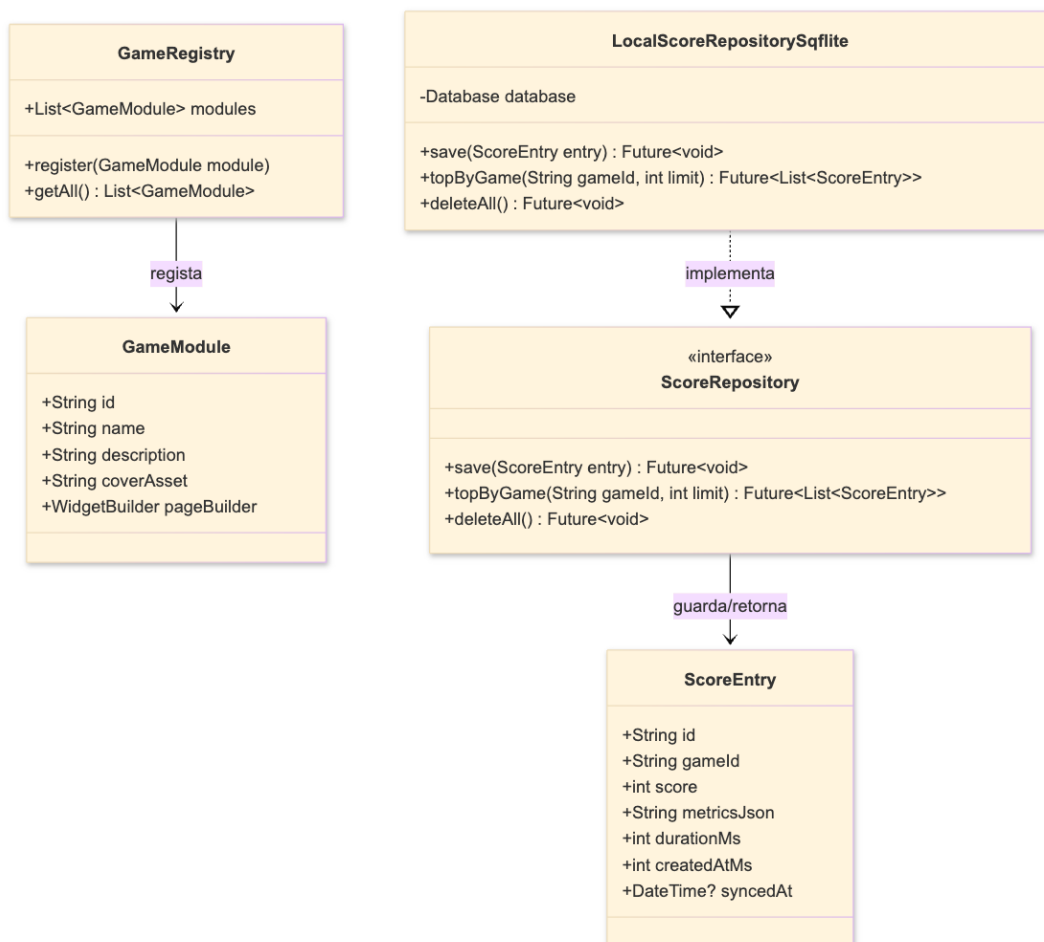


Figura 8 - Modelo de classes simplificado da arquitetura modular

Em conjunto, estas modelações permitem representar tanto a base atual da corrente fase do desenvolvimento, como a evolução prevista da solução, articulando a perspetiva de dados com a perspetiva estrutural do software.

3.3 Protótipos de Interface

Nesta fase do projeto, a definição da interface deixou de se centrar apenas num protótipo isolado e passou a refletir a organização global da aplicação enquanto hub de mini-jogos. Assim, esta secção apresenta sobretudo a estrutura de navegação e o mapa aplicacional da solução, evidenciando os ecrãs principais e a forma como o utilizador circula entre eles.

A aplicação encontra-se organizada em torno de uma navegação principal composta por três áreas base: 1) Jogos, 2) Pontuações e 3) Definições.

A partir da área de jogos, o utilizador pode aceder aos mini-jogos atualmente integrados na solução, sendo eles actualmente o Eco Sort, o Eco Guess e o Eco Proto.

Esta organização traduz a opção de concentrar a experiência principal da aplicação num hub central, a partir do qual os diferentes módulos são descobertos e iniciados. A principal vantagem desta abordagem é a clareza da navegação e a facilidade de expansão futura, uma vez que novos jogos podem ser adicionados mantendo a mesma lógica estrutural.

A Figura 9 apresenta o mapa aplicacional atual da solução, mostrando a relação entre o ecrã principal, os mini-jogos disponíveis e as páginas complementares já previstas na navegação.

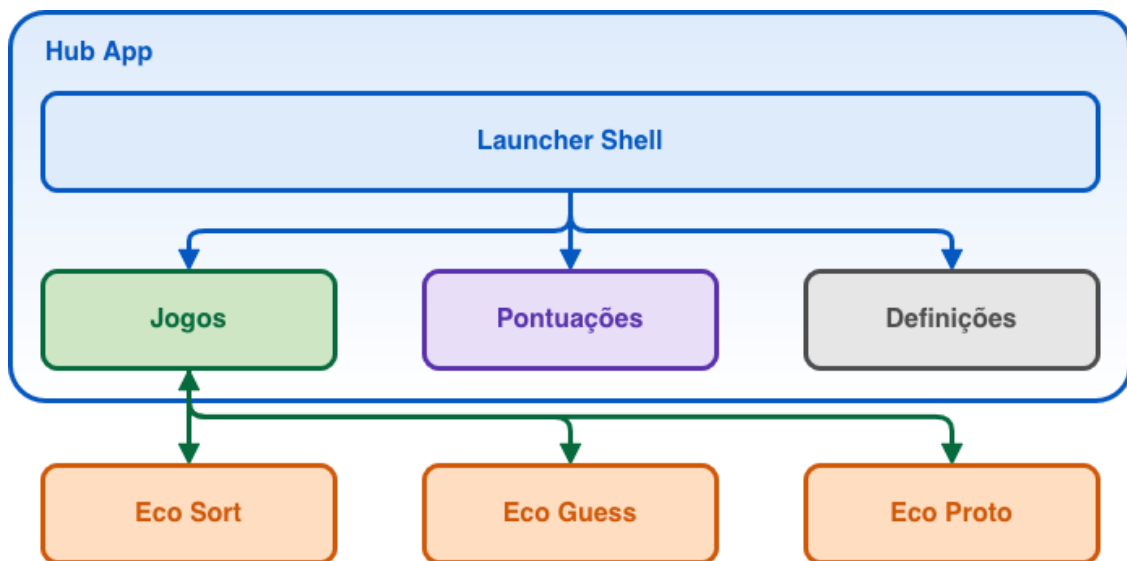


Figura 9 - Mapa aplicacional da solução

Nesta fase, os ecrãs dos mini-jogos já permitem validar a integração entre Flutter e Flame, bem como a navegação entre o hub e os módulos jogáveis. Por outro lado, as páginas de Pontuações e Definições encontram-se ainda em estado inicial, funcionando como placeholders estruturais para funcionalidades em desenvolvimento.

Importa referir que esta secção tem como objetivo apresentar a estrutura e conceção da navegação, não a descrição detalhada das interfaces implementadas. A análise mais concreta dos ecrãs efetivamente desenvolvidos, acompanhada por capturas representativas da aplicação, é apresentada na Secção 4.7.

4 Solução Proposta

4.1 Apresentação

A solução proposta consiste no desenvolvimento de uma aplicação móvel educativa orientada para a sensibilização ambiental através de mini-jogos interativos. A aplicação foi concebida como um ponto central de acesso a diferentes experiências de jogo, cada uma dedicada a uma temática específica relacionada com sustentabilidade, reciclagem e literacia ecológica.

Nesta fase do projeto, a solução encontra-se materializada sob a forma de um MVP funcional, já capaz de validar o conceito central do trabalho: a utilização de mecânicas lúdicas e interação digital como veículo de aprendizagem informal. Em vez de se limitar a conteúdos expositivos ou questionários estáticos, a aplicação procura envolver o utilizador através de desafios curtos, feedback imediato e progressão baseada em desempenho.

A solução distingue-se das alternativas analisadas no benchmarking por apostar numa abordagem modular, onde vários mini-jogos coexistem num mesmo ecossistema aplicacional, mantendo coerência visual e estrutural. Esta opção permite não apenas diversificar a experiência do utilizador, mas também criar uma base técnica preparada para evolução futura.

O presente capítulo encontra-se organizado da seguinte forma: começa-se por descrever a arquitetura global da solução e a lógica subjacente à sua organização; de seguida apresentam-se as tecnologias e ferramentas utilizadas; posteriormente caracterizam-se os ambientes de exploração, a abrangência curricular e os principais componentes do sistema; por fim, são descritas as interfaces mais representativas da aplicação e as decisões de implementação associadas.

4.2 Arquitetura

A arquitetura da solução foi definida com o objetivo de garantir modularidade, separação de responsabilidades e facilidade de evolução futura. Atendendo ao facto de a aplicação integrar vários mini-jogos com mecânicas distintas, optou-se por uma organização em repositório monolítico (monorepo) modular, em que a aplicação principal funciona como agregador dos diferentes módulos de jogo.

Esta abordagem foi adotada para evitar uma estrutura monolítica, na qual toda a lógica da aplicação ficaria concentrada numa única base de código. Essa alternativa aumentaria o acoplamento entre componentes, dificultaria a manutenção e tornaria mais arriscada a introdução de novos jogos ou funcionalidades. Ao separar os diferentes elementos da solução em módulos independentes, a aplicação torna-se mais clara, extensível e tecnicamente sustentável.

A arquitetura atualmente implementada encontra-se organizada em quatro blocos principais: a aplicação principal (Hub App), o mecanismo de registo de módulos (Game Registry), os mini-jogos independentes e a biblioteca partilhada (Common Gamekit), que inclui também a base de persistência local. A Figura 10 apresenta esta organização lógica da solução.

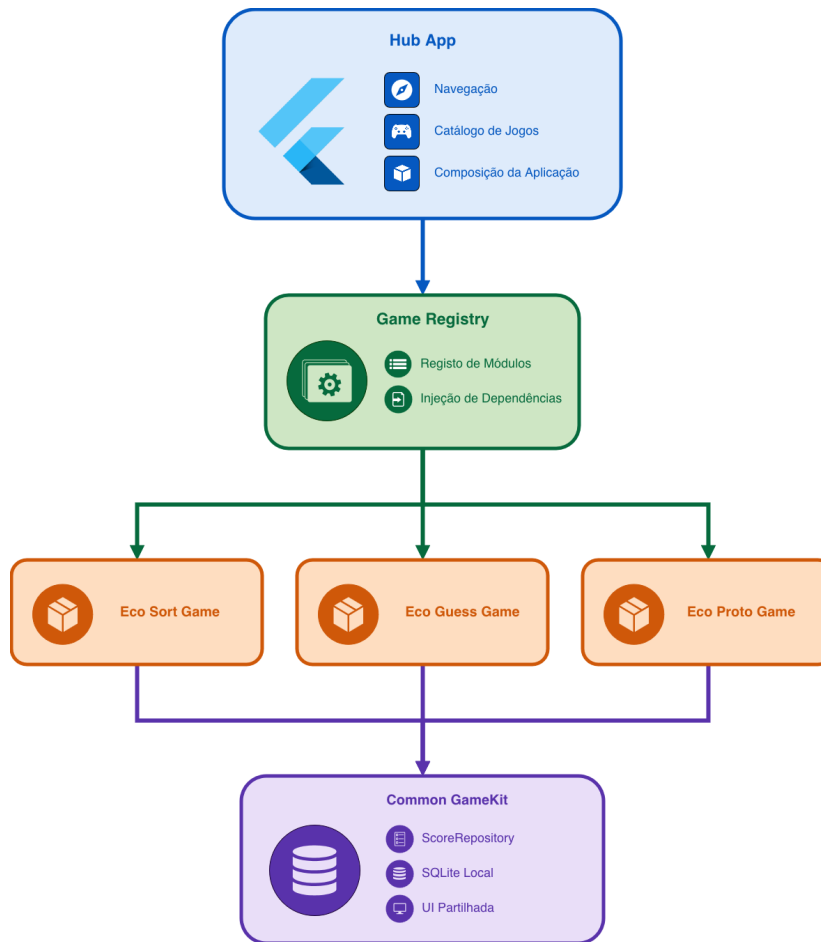


Figura 10 - Arquitectura lógica da solução proposta

A aplicação principal é responsável pela navegação e composição global da experiência. O GameRegistry permite registar e disponibilizar os módulos de jogo através de uma interface comum. Os mini-jogos são implementados como packages autónomos, encapsulando a sua lógica e recursos próprios. Por fim, o common_gamekit concentra funcionalidades transversais, reduzindo duplicação e promovendo consistência técnica entre módulos.

Em termos globais, a arquitetura adotada permite sustentar o estado atual do projeto e, ao mesmo tempo, preparar a aplicação para futuras extensões, como a introdução de novos mini-jogos ou a evolução da camada de persistência.

4.3 Tecnologias e Ferramentas Utilizadas

A implementação da solução recorre a um conjunto de tecnologias selecionadas em função da sua adequação aos requisitos do projeto, da maturidade do respetivo ecossistema e da sua capacidade para suportar uma aplicação móvel modular com componentes interativos.

A Figura 11 apresenta os perímetros tecnológicos da solução, distinguindo a camada mobile, os módulos de jogo, a persistência local e o conjunto de ferramentas de apoio ao desenvolvimento.

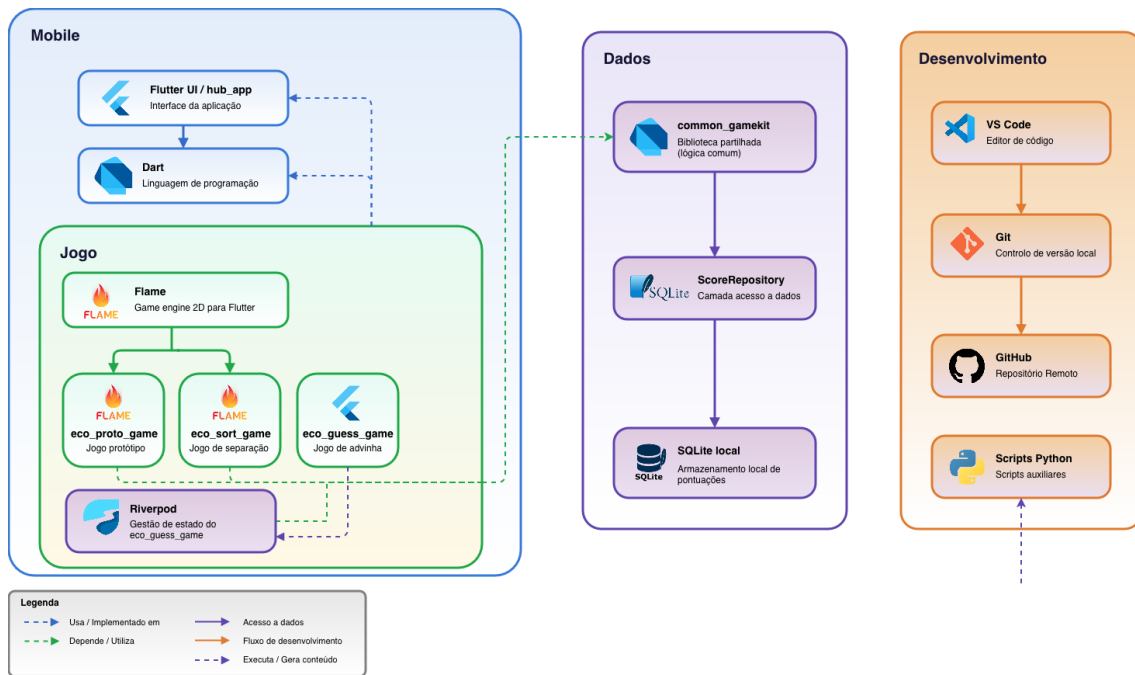


Figura 11 - Perímetros tecnológicos da solução

4.3.1 Flutter

O desenvolvimento da aplicação foi realizado em Flutter, escolhido por permitir a construção de uma solução móvel multiplataforma a partir de uma única base de código. Esta escolha responde diretamente à necessidade de reduzir esforço de desenvolvimento e manutenção, mantendo uma interface consistente entre plataformas.

4.3.2 Dart

A linguagem utilizada em toda a solução é Dart, tanto na aplicação principal como nos módulos de jogo e na biblioteca partilhada. A sua utilização favorece a organização do código em classes, contratos e packages, o que é coerente com a arquitetura definida para o projeto.

4.3.3 Flame

Para a implementação dos mini-jogos foi utilizada a framework Flame, que fornece uma base adequada para rendering 2D, game loop e gestão de componentes interativos. Esta escolha evita a utilização exclusiva de widgets tradicionais do Flutter para comportamentos de jogo, o que seria menos apropriado neste contexto.

4.3.4 SQLite

A persistência local é assegurada através de SQLite, permitindo armazenar pontuações no dispositivo sem dependência de conectividade. No módulo eco_guess_game foi ainda adotado Riverpod para gestão de estado, com o objetivo de separar de forma mais clara a lógica da sessão de jogo da respetiva interface.

4.3.5 Visual Studio Code, Git, Github e Python

Ao nível das ferramentas de desenvolvimento, foram utilizados Visual Studio Code como ambiente de desenvolvimento, Git e Github para controlo de versões e alojamento do

repositório, e ainda pequenos scripts em Python para apoio à manutenção de catálogos e recursos associados ao `eco_sort_game`.

4.4 Ambientes de Teste e de Produção

Nesta fase, a solução é desenvolvida e testada em ambiente local, com foco na execução em dispositivos móveis compatíveis com Flutter. A aplicação funciona sem dependência obrigatória de serviços remotos, o que está alinhado com o estado atual do projeto e com o requisito de funcionamento offline.

Os testes são realizados em ambiente mobile, recorrendo a emuladores e/ou dispositivos físicos Android. Em termos de recursos, a solução exige apenas capacidades compatíveis com aplicações móveis correntes, nomeadamente capacidade de processamento suficiente para interfaces 2D, armazenamento local para instalação da aplicação e suporte à execução de aplicações Flutter.

A persistência é assegurada localmente através de SQLite, não existindo ainda necessidade de infraestrutura de rede ou backend. Numa fase futura, a eventual introdução de um sistema de pontuações online poderá implicar recursos adicionais de servidor, base de dados remota e conectividade, mas esse cenário não faz ainda parte da versão atual da solução.

4.5 Abrangência

O projeto mobiliza conhecimentos de várias unidades curriculares e áreas científicas do curso, refletindo a natureza integradora do TFC.

Ao nível de Engenharia de Software, destacam-se a análise de requisitos, a modularização da solução e a separação de responsabilidades. Em Algoritmia e Estruturas de Dados e Linguagens de Programação II, aplicam-se conhecimentos de estruturação de classes, abstrações e organização modular em Dart. Na área de Computação Móvel, são utilizados conceitos relacionados com desenvolvimento multiplataforma, navegação e interfaces em Flutter.

Em Computação Gráfica, o projeto aplica conhecimentos associados à construção de mini-jogos 2D com Flame. Em Bases de Dados, são mobilizados conceitos de modelação e persistência local através de SQLite. Por fim, a arquitetura foi pensada de forma a permitir futura evolução para cenários de Computação Distribuída, embora essa componente ainda não esteja implementada.

Tabela 6 - Abrangência de conhecimentos adquiridos nas UCs

| Unidades Curriculares | Conhecimentos Aplicados |
|----------------------------------|--|
| Engenharia de Software | Requisitos, modularização, componentes e arquitetura |
| Algoritmia e Estruturas de Dados | Organização lógica dos jogos e fluxos de interação |
| Linguagens de Programação II | Estruturação de classes, packages e abstrações aplicadas em Dart |
| Bases de Dados | Persistência local e modelação de pontuações |

| | |
|-----------------------|---|
| Computação Móvel | Desenvolvimento mobile em Flutter e navegação |
| Computação Gráfica | Rendering e interação 2D com Flame |
| Sistemas Distribuídos | Preparação arquitetural para futura integração remota |

4.6 Componentes

A solução encontra-se dividida em componentes com responsabilidades distintas, o que permite melhorar a organização do sistema e facilitar a sua evolução. Nesta fase, destacam-se a aplicação principal, os módulos de mini-jogo, a biblioteca partilhada e a camada de persistência local.

4.6.1 Aplicação principal (Hub App)

O Hub App (`hub_app`) constitui o ponto de entrada da solução e é responsável pela navegação principal da aplicação, pela apresentação do catálogo de jogos e pela composição da experiência global. É também nesta componente que se encontram estruturadas as páginas base da aplicação, embora as secções de pontuações e definições permaneçam ainda em desenvolvimento.

4.6.2 Módulos de mini-jogo

Os mini-jogos são implementados como packages independentes, cada um contendo a sua lógica, assets e interface próprios. Atualmente, a solução integra os módulos `eco_sort_game`, `eco_guess_game` e `eco_proto_game`. Esta organização permite isolar responsabilidades e facilita a manutenção e extensão futura do sistema.

4.6.3 Biblioteca partilhada

O `common_gamekit` concentra funcionalidades reutilizáveis, incluindo componentes de interface, diálogos comuns e contratos de persistência. A sua utilização permite reduzir duplicação de código e manter maior consistência entre os vários módulos da aplicação.

4.6.3 Persistência local

A persistência local é implementada através de SQLite, recorrendo à abstração `ScoreRepository`. Esta componente suporta o armazenamento de pontuações por jogo e estabelece uma base técnica para futura evolução da gestão de resultados, embora a sua integração funcional ainda não esteja concluída em todos os fluxos da aplicação.

4.7 Interfaces

A presente secção descreve as interfaces mais representativas da aplicação no seu estado atual de desenvolvimento, recorrendo a capturas dos ecrãs implementados e à análise das principais decisões tomadas na sua construção. O foco desta secção incide na concretização visual e funcional da solução, evidenciando o papel de cada ecrã na experiência global do utilizador.

Atendendo ao estado atual do projeto, serão apresentados o ecrã principal do hub, os ecrãs correspondentes aos mini-jogos atualmente integrados e as páginas de pontuações e definições, estas últimas ainda numa fase inicial de implementação. Para cada interface, procura-se

descrever não apenas a sua função, mas também a forma como foi implementada e os critérios que orientaram as principais decisões de organização visual e navegação.

A concepção das interfaces foi orientada por três princípios centrais: acesso rápido ao conteúdo jogável, clareza visual e coerência entre módulos. Esta abordagem pretende reduzir fricção na utilização da aplicação e assegurar uma experiência simples, legível e consistente entre os vários componentes do sistema.

4.7.1 Interfaces principais do hub

O hub constitui o ponto de entrada da aplicação e organiza a navegação principal em três áreas base: Jogos, Pontuações e Definições. Esta estrutura permite concentrar, num único local, tanto o acesso aos mini-jogos como as áreas funcionais complementares da aplicação, contribuindo para uma experiência de utilização mais clara e consistente.

A página Jogos corresponde à área central do hub e apresenta o catálogo de mini-jogos disponíveis. A sua implementação foi realizada em Flutter, recorrendo a cartões visuais (GameCoverCard) gerados a partir do GameRegistry. Esta decisão evita duplicação de configuração na aplicação principal e permite que novos jogos sejam integrados no catálogo de forma simples, desde que devidamente registados. Em termos de interface, optou-se por uma apresentação direta, com imagem de capa, nome e breve descrição, de modo a tornar a seleção do jogo mais imediata e apelativa.

A página Pontuações encontra-se atualmente implementada como placeholder funcional. O seu objetivo é reservar, desde já, o espaço aplicacional onde serão apresentadas as classificações locais e, numa fase posterior, eventuais classificações globais. Apesar de ainda não disponibilizar dados reais ao utilizador, a sua presença é importante para estabilizar a navegação e preparar a evolução funcional da aplicação.

A página Definições encontra-se igualmente numa fase inicial de implementação, funcionando como ponto de entrada para futuras preferências globais da aplicação, nomeadamente controlo de som e aparência. Tal como sucede com a página de pontuações, a sua introdução nesta fase não pretende demonstrar completude funcional, mas antes consolidar a estrutura da navegação principal e evitar crescimento desorganizado em fases posteriores.

Em conjunto, estas três interfaces demonstram que o hub já se encontra estruturado como camada agregadora da aplicação: a área de jogos materializa a funcionalidade principal atualmente disponível, enquanto as áreas de pontuações e definições estabelecem a base necessária para evolução futura sem comprometer a coerência global da solução.

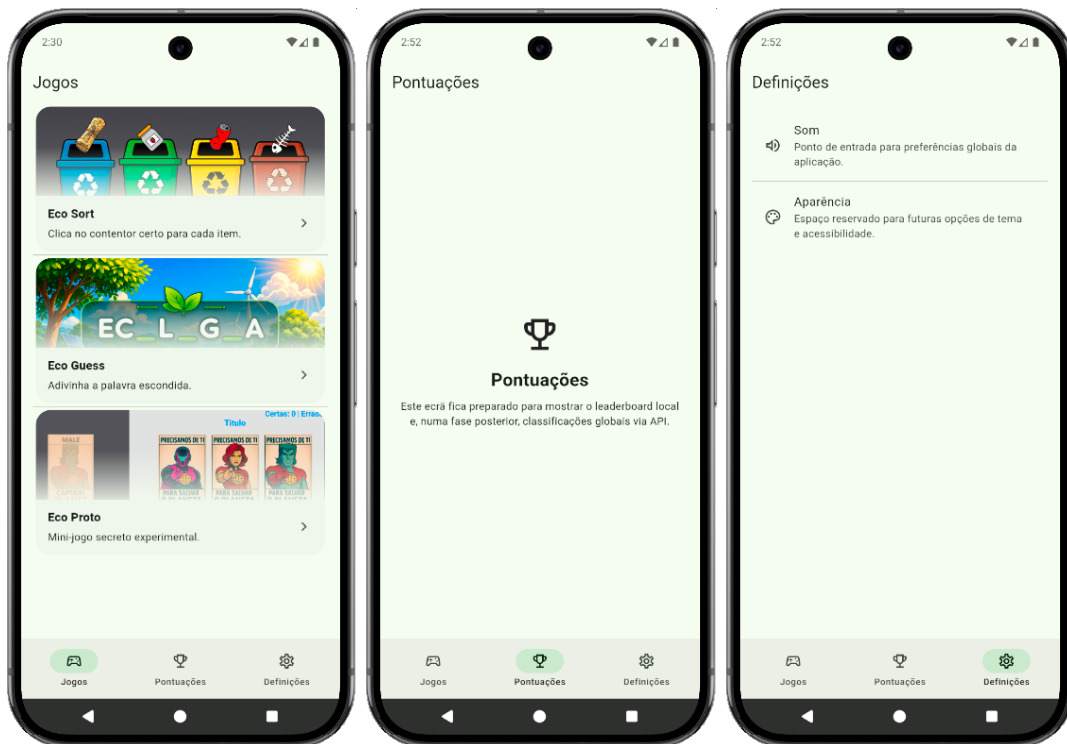


Figura 12 - Páginas principais do hub: Jogos, Pontuações e Definições

4.7.2 Ecrã do mini-jogo Eco Sort

O ecrã do Eco Sort apresenta uma experiência centrada na associação correta entre resíduos e contentores de reciclagem. A interface combina a estrutura da página Flutter com a componente interativa gerida por Flame, permitindo integrar elementos de jogo em tempo real com enquadramento visual consistente no restante sistema.

A implementação foi organizada de forma a separar a mecânica do jogo da navegação da aplicação, evitando que o hub tenha conhecimento direto da lógica interna do mini-jogo. Esta decisão melhora a manutenção e reduz o acoplamento entre módulos. Ao nível visual, a interface privilegia a leitura rápida dos elementos principais, nomeadamente o resíduo a classificar, os contentores e o feedback do jogo.

Nesta fase, o mini-jogo encontra-se parcialmente implementado, faltando ainda concluir aspetos como mecânica de fecho, feedback sonoro imediato e apresentação explícita de erros no final da sessão. Apesar disso, o ecrã já permite validar a lógica base da interação e a viabilidade da arquitetura adotada.



Figura 13 - Ecrã representativo do mini-jogo Eco Sort

4.7.3 Ecrã do mini-jogo Eco Guess

O Eco Guess apresenta uma interface focada na adivinhação de palavras relacionadas com sustentabilidade. O utilizador recebe uma pista textual e interage com a sessão de jogo tentando descobrir a palavra correta, num formato inspirado em mecânicas de dedução progressiva.

Ao contrário dos restantes módulos, este jogo recorre a Riverpod para gestão de estado, permitindo separar a evolução da sessão da camada visual. Esta escolha foi importante porque resolve o problema da dispersão de estado entre vários widgets e facilita a atualização previsível da interface ao longo da ronda. As vantagens concretas são maior clareza estrutural e melhor controlo das transições de estado do jogo.

Do ponto de vista visual, a interface foi desenhada para dar prioridade à pista, ao progresso da palavra e ao estado da sessão, minimizando elementos secundários. Nesta fase, faltam ainda elementos como sons de feedback e apresentação de factos ecológicos no final, mas o ecrã já demonstra a viabilidade da mecânica principal.



Figura 14 - Ecrã representativo do mini-jogo Eco Guess

4.7.4 Ecrã do mini-jogo Eco Proto

O Eco Proto corresponde à migração de um protótipo jogável desenvolvido inicialmente e integrado no novo monorepo como módulo independente. A sua interface foi preservada de forma a manter a experiência originalmente criada, servindo simultaneamente como componente funcional da aplicação e como referência de integração modular. Adicionalmente, este mini-jogo assumirá também futuramente a função de conteúdo oculto (Easter Egg), ficando indisponível na listagem principal por defeito e sendo apenas desbloqueado mediante uma interação específica do utilizador no ecrã de definições, concretamente após dez toques consecutivos numa opção predefinida. Uma vez ativado, o jogo passa então a surgir listado no hub da aplicação, reforçando a componente exploratória da experiência sem comprometer a organização principal da interface.

A principal decisão de implementação neste caso foi manter o jogo encapsulado no seu package próprio, em vez de o reescrever de raiz dentro da aplicação principal. Esta opção resolve o problema de retrabalho desnecessário e permite aproveitar uma base já funcional, reduzindo esforço e validando a estratégia de integração por módulos. Em termos arquiteturais, esta integração demonstra que a estrutura atual consegue acomodar jogos com diferentes históricos de desenvolvimento.



Figura 15 - Ecrã representativo do mini-jogo Eco Proto

5 Testes e Validação

O presente capítulo descreve a abordagem de testes adotada para validar a solução atualmente implementada. Nesta fase do projeto, o objetivo da validação não é ainda demonstrar uma solução final plenamente concluída, mas sim confirmar que a aplicação já funciona de forma coerente com os seus objetivos centrais: disponibilizar um hub de mini-jogos educativos, suportar uma arquitetura modular e validar a viabilidade técnica da proposta.

Atendendo ao estado atual do desenvolvimento, a estratégia de testes incidiu sobretudo sobre três dimensões: validação de componentes críticos através de testes unitários, verificação funcional manual dos principais fluxos da aplicação e confirmação da integração entre a aplicação principal, os módulos de jogo e a biblioteca partilhada.

Os testes unitários atualmente existentes concentram-se sobretudo nos packages `common_gamekit` e `eco_sort_game`. No primeiro, os testes cobrem a criação de módulos de jogo, a estrutura de pontuações e aspetos centrais da persistência local com SQLite. No segundo, os testes incidem sobre parsing de catálogos, controladores de ronda e pontuação, bem como validação de erros e reinicialização de estado. Esta cobertura permite testar elementos de maior risco técnico, nomeadamente a integridade da persistência local, a lógica de scoring e o tratamento de dados estruturados.

Para além dos testes automáticos, foi realizada validação funcional manual dos principais cenários de utilização atualmente disponíveis, incluindo abertura da aplicação, navegação no hub, entrada nos mini-jogos e execução base das respetivas mecânicas. Esta validação permitiu confirmar que a solução já se apresenta como uma aplicação modular jogável, ainda que com diferentes níveis de maturidade entre componentes.

A validação operacional foi realizada em ambiente local de desenvolvimento, recorrendo a execução mobile compatível com Flutter. Dado que a solução funciona atualmente sem backend e sem dependência de serviços externos, não foram ainda realizados testes sobre infraestrutura remota, latência de rede, sincronização online ou disponibilidade de serviços cloud. Do ponto de vista de recursos, a aplicação demonstrou funcionamento compatível com o contexto esperado de uma aplicação móvel com rendering 2D e persistência local ligeira.

Importa reconhecer que a validação atual apresenta limitações relevantes. Não foram ainda realizados testes formais com utilizadores externos, nem existe evidência suficiente para concluir impacto pedagógico real em contexto de utilização. Do mesmo modo, a cobertura de testes permanece desigual entre módulos, existindo ainda validação automática reduzida e por desenvolver no `hub_app`, `eco_guess_game` e `eco_proto_game`.

Ainda assim, os testes realizados até ao momento permitem concluir que a base técnica da solução é viável, que a arquitetura modular funciona de acordo com o esperado e que os componentes mais críticos já possuem validação parcial.

6 Método e Planeamento

6.1 Planeamento inicial

O planeamento do projeto foi definido com base numa abordagem iterativa inspirada em metodologias Agile, de forma a permitir ajustamentos progressivos ao longo do desenvolvimento. Esta opção revelou-se adequada à natureza do trabalho, uma vez que a solução envolve simultaneamente definição de requisitos, arquitetura, integração de módulos independentes, desenvolvimento de mini-jogos e evolução incremental da aplicação principal.

O cronograma foi organizado em torno das principais frentes do projeto: levantamento bibliográfico e benchmarking, modelação e requisitos, conceção da arquitetura, estruturação do monorepo, desenvolvimento da hub_app, integração dos mini-jogos, persistência local, testes técnicos e redação do relatório. Para além destas tarefas nucleares, foram também mantidas no planeamento algumas componentes de evolução futura, como backend, API e leaderboard global, tratadas nesta fase como dependentes do tempo disponível e da maturidade alcançada pela solução local. A Figura 16 apresenta o cronograma revisto do projeto.

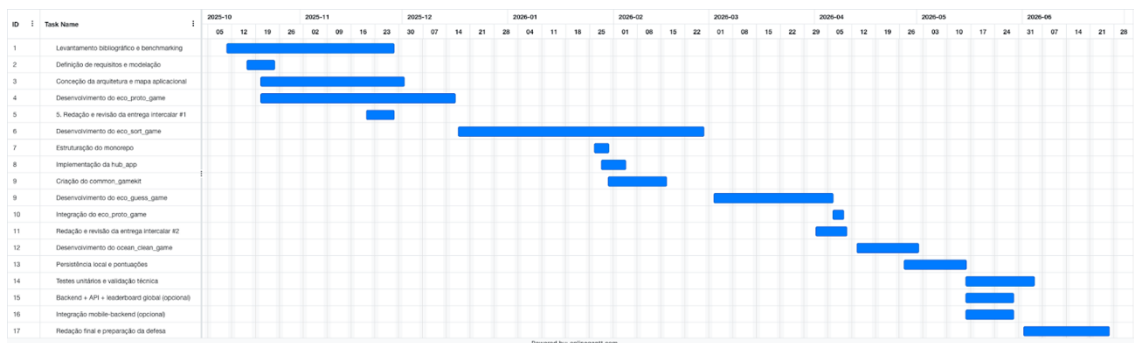


Figura 16 - Organograma em formato Gantt

Para apoiar a organização e acompanhamento do trabalho ao longo do desenvolvimento do projeto, foi utilizado um quadro Kanban na plataforma Trello. Este artefacto permitiu estruturar tarefas, acompanhar o seu estado de execução e manter uma visão contínua da evolução do trabalho, servindo também como instrumento de apoio à comunicação com os professores orientadores. A Figura 17 apresenta o quadro kanban usado durante o desenvolvimento do projeto.

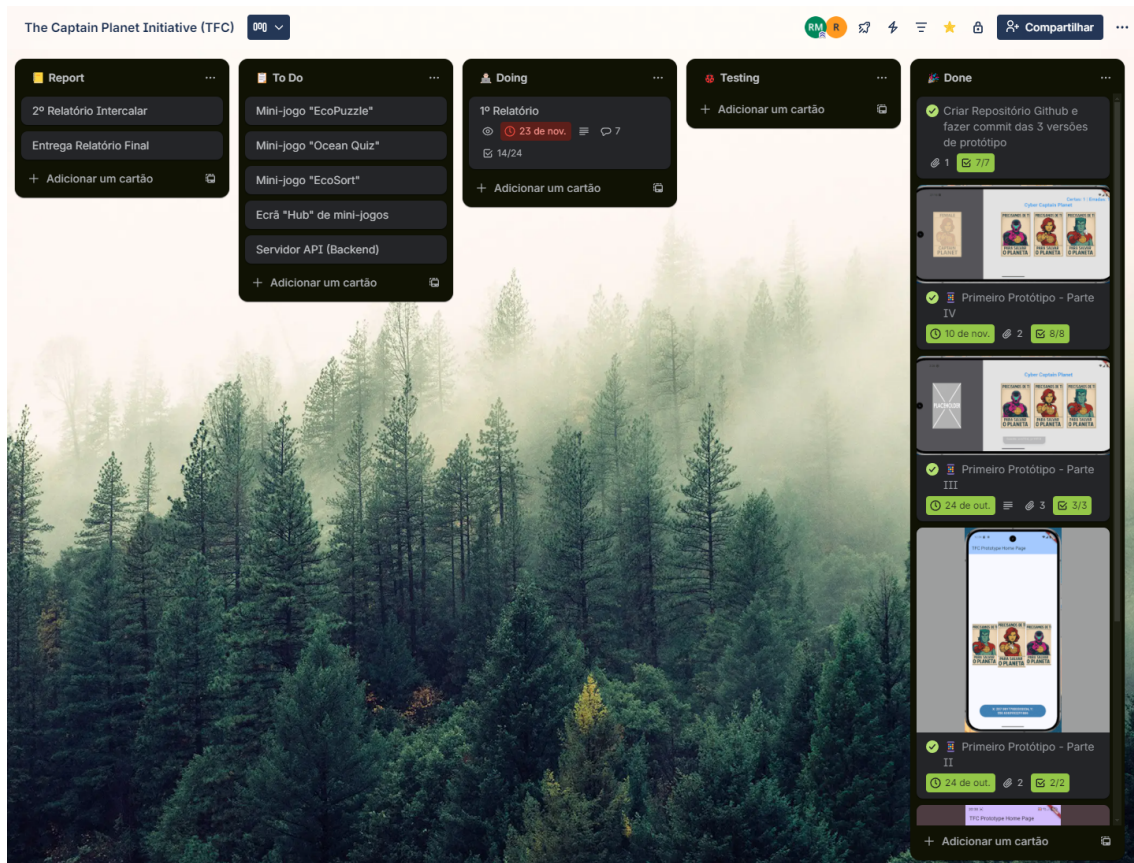


Figura 17 - Quadro Kanban de organização para o projecto

6.2 Análise Crítica ao Planeamento

A análise crítica ao planeamento evidencia que o desenvolvimento do projeto sofreu uma reorientação progressiva face à proposta inicial. Embora o plano original previsse, numa fase relativamente precoce, a implementação de backend, API e leaderboards globais, a evolução real do trabalho demonstrou que a prioridade mais adequada consistia em consolidar primeiro a aplicação móvel, a arquitetura modular e os mini-jogos nucleares.

Em termos práticos, verificou-se que a construção de uma base técnica sólida exigiu mais trabalho estrutural do que inicialmente previsto. A introdução de um monorepo modular, a implementação da `hub_app` enquanto ponto de entrada do sistema, a criação do `common_gamekit` e a integração dos mini-jogos como packages independentes representaram um esforço técnico significativo, mas essencial para garantir coerência, reutilização e escalabilidade da solução.

Esta evolução levou a um ajustamento natural do planeamento: componentes remotas, como backend, API e classificações globais, deixaram de ser encaradas como núcleo obrigatório da fase atual e passaram a ser tratadas como extensões condicionadas ao tempo disponível e ao grau de maturidade alcançado pela solução local. Do mesmo modo, a integração mobile-backend foi adiada, uma vez que a sua concretização dependeria primeiro da existência de um backend funcional e de uma persistência local plenamente estabilizada.

Uma das dificuldades mais relevantes identificadas durante a execução prendeu-se com a modularização dos mini-jogos, em particular devido ao comportamento do Flame no

carregamento de assets a partir de packages. Esta limitação obrigou a adaptações técnicas não previstas inicialmente e teve impacto no ritmo de desenvolvimento, mas contribuiu também para reforçar a robustez da arquitetura atualmente adotada.

Assim, a diferença entre o planejado e o implementado não deve ser interpretada como desvio arbitrário, mas antes como resultado de uma gestão de escopo mais realista. Em vez de dispersar esforço por várias frentes em simultâneo, optou-se por privilegiar a construção de uma base funcional e extensível, capaz de sustentar futuras evoluções com menor risco técnico.

Para a fase seguinte do trabalho, o foco deverá incidir na consolidação dos mini-jogos existentes, na integração efetiva da persistência local, no reforço da validação técnica, na introdução de um novo módulo de jogo e, caso o calendário o permita, na concretização das componentes remotas inicialmente previstas.

Bibliografia

- [1] D. Lyew, J. Moyer, B. Sanderson, E. Cohen, and R. Jereza, “Connecting Early Childhood Development to Climate Change Insights for Communicators,” 2025.
- [2] C. K. W. Tan and H. Nurul-Asna, “Serious games for environmental education,” *Integrative Conservation*, vol. 2, no. 1, pp. 19–42, Mar. 2023, doi: 10.1002/INC3.18.
- [3] Direção-Geral da Educação, “Projeto-Piloto Manuais Digitais Relatório Intermédio-PPMD: Resultados dos Inquéritos por Questionário - Alunos, Professores, Lideranças,” Jun. 2024. Acedido em: Nov. 2025. [Online]. Disponível em: <https://digital.dge.mec.pt/sites/default/files/documents/2024/412-831af993f3cec854e7a9a55b2a8f201b.pdf>
- [4] Direção-Geral da Educação, “Recursos Educativos Digitais | Capacitação Digital das Escolas.” Acedido em: Nov. 2025. [Online]. Disponível em: <https://digital.dge.mec.pt/projeto-piloto-manuais-digitais>
- [5] “272 milhões de euros para a transição digital na educação - Plano de Recuperação e Resiliência - Recuperar Portugal.” Acedido em: Nov. 2025. [Online]. Disponível em: <https://recuperarportugal.gov.pt/2021/10/01/272-milhoes-de-euros-para-a-transicao-digital-na-educacao/>
- [6] “17 Objetivos • ODS - BCSD Portugal.” Acedido em: Nov. 2025. [Online]. Disponível em: <https://ods.pt/ods/>
- [7] “Flutter - Build apps for any screen.” Acedido em: Nov. 2025. [Online]. Disponível em: <https://flutter.dev/>
- [8] “Getting Started — Flame.” Acedido em: Nov. 2025. [Online]. Disponível em: <https://docs.flame-engine.org/latest/>
- [9] “Jogo de Reciclagem – Apps no Google Play.” Acedido em: Nov. 2025. [Online]. Disponível em: <https://play.google.com/store/apps/details?id=game.test1.recycle>
- [10] “Ocean Cleanup Game – Apps no Google Play.” Acedido em: Nov. 2025. [Online]. Disponível em: https://play.google.com/store/apps/details?id=com.droidbg.ocean_cleanup
- [11] “ECOKIDS - Ecologia infantil – Apps no Google Play.” Acedido em: Nov. 2025. [Online]. Disponível em: <https://play.google.com/store/apps/details?id=air.ecokids>
- [12] “Ponto Verde – Apps Android no Google Play.” Acedido em: Nov. 2025. [Online]. Disponível em: <https://play.google.com/store/search?q=ponto+verde&c=apps>

Anexo 1 – Formulário de declaração de uso de ferramentas de Inteligência Artificial a anexar a relatório

Todos os relatórios deverão incluir anexo com cópia, devidamente preenchida, do formulário abaixo.

Assinalar as opções aplicáveis e completar os campos solicitados.

1. Utilização de IA

Não foram utilizadas ferramentas de IA na realização deste trabalho.

Foram utilizadas ferramentas de IA na realização deste trabalho.

2. Ferramentas utilizadas

Assinalar todas as que se aplicam.

Assistência geral à escrita, análise ou ideação

ChatGPT

Microsoft Copilot

Gemini

Claude

Perplexity

Outras. Quais? _____

Assistência à programação / desenvolvimento

GitHub Copilot

Claude

OpenAI Codex

Cursor

Tabnine

Amazon CodeWhisperer / Amazon Q

Outras. Quais? _____

Geração de imagem / design / multimédia

DALL-E

Midjourney

Stable Diffusion

Canva AI / Magic Design

Outras. Quais? Nano Banana _____

Outros usos

Contexto: Ferramentas? ChatGPT _____

3. Fases do trabalho em que foi utilizada IA

- Planeamento do trabalho
- Pesquisa exploratória / levantamento inicial de informação
- Documentação técnica
- Redação do relatório
- Desenho / modelação / arquitetura
- Design / prototipagem / interface
- Geração de código
- Revisão / refatoração / debugging de código
- Criação de testes / casos de teste
- Análise de resultados
- Preparação de apresentação ou materiais auxiliares
- Outros. Quais? _____

4. Tipo de utilização

Descrever sucintamente como a IA foi utilizada.

Exemplos: brainstorming, estruturação de secções, revisão linguística, sugestão de arquitetura, geração de exemplos, explicação de conceitos, geração parcial de código, correção de erros, criação de casos de teste, apoio ao design.

Brainstorming, estruturar e reescrever secções do relatório, sugestão e análise de arquitetura, revisão da modelação de dados e requisitos, geração parcial de código, geração de exemplos, correção de erros, criação de casos de teste e apoio ao design.

5. Partes do trabalho afetadas

Indicar as secções, componentes, módulos, ficheiros, entregáveis ou atividades que foram influenciados pelo uso de IA.

Capítulos 1, 3, 4 e 5 do relatório, modelação de dados, arquitetura da solução, plano de testes, revisão linguística e coerência entre documentação e implementação.

6. Exemplos de *prompt*

Inserir exemplos de *prompt*, diferenciando por âmbito (enquadrado na questão 2) e fase (enquadrado na questão 4)

Assistência geral à escrita, análise ou ideação:

“Ajuda-me a rever e reescrever o capítulo da arquitetura do relatório, alinhando-o com o estado atual do projeto e com uma organização em monorepo modular.”

“Revê os capítulos 3 e 4 e identifica incoerências entre requisitos, modelação, arquitetura e implementação atual.”

Assistência à programação / desenvolvimento:

“Explica o impacto de alterar uma classe abstrata ScoreRepository usada por vários mini-jogos e que consequências isso teria nos packages.”

“Ajuda-me a perceber como modularizar mini-jogos em packages Flutter/Flame mantendo integração com a aplicação principal.”

Geração de imagem / design / multimédia:

“flat design vector icon of a blue recycling bin container, front view, lid slightly open revealing a dark opening slot. The bin is solid flat blue color. On the front, there is a large white universal recycling arrows symbol, and below it, simple white flat icons of a stacked newspaper and a cardboard box. Thick black outlines everywhere, no shadows, no shading, no gradients, 2D mobile game asset, isolated on transparent background”

“Estou a trabalhar numa app que tem um ecrã que irá conter vários minijogos que ao clicar numa das entradas listada, abre esse mesmo minijogo. Preciso que desenes um ecrã de uma hub app. Até ao momento, em anexo este é o ecrã actual. Preciso que me desenvolves uma UI viável para ambos android e ios, com barra de navegação, ecrã de scoreboard e settings”

7. Validação, revisão e intervenção dos autores

Descrever que verificação, revisão, correção, adaptação ou reescrita foi realizada pelos autores.

Nota: se a IA tiver sido usada em código, testes, scripts, modelos, consultas, configurações ou outros artefactos técnicos, deve ser indicado de que forma os autores validaram o funcionamento e confirmaram a sua compreensão.

Todo o conteúdo apoiado por IA foi revisto, validado e adaptado manualmente antes de ser integrado no TFC. No relatório, as sugestões foram sujeitas a reescrita, ajustadas ao estado real

do projeto. e verificadas quanto à coerência entre requisitos, arquitetura, modelação, implementação e planeamento. Sempre que a IA propôs reformulações, estruturas de secções, diagramas, descrições técnicas ou exemplos, procedeu-se à sua revisão crítica, foram corrigidas incorreções, e removeram-se elementos não aplicáveis, adaptando o texto ao contexto académico e à solução efetivamente desenvolvida.

8. Grau de utilização

Residual

Moderado

Extensivo

Utilização homogénea

Grau de uso diferenciado por fase ou componente de trabalho

Descrever sucintamente os diferentes usos.

A utilização foi moderada e homogénea, com maior incidência no apoio à redação e revisão do relatório, análise de arquitetura, modelação, requisitos, testes e planeamento. Houve também uso pontual em apoio à programação, correção de erros, geração de exemplos, diagramas e elementos visuais.

9. Trabalhos em parceria

Protecção de dados confidenciais e recursos proprietários de parceiros

O trabalho foi realizado em parceria com entidade externa ao DEISI

No caso da resposta anterior ser verdadeira, responder às seguintes questões:

O parceiro tem regras para restringir submissão de dados

As submissões validam aplicação de regras de tratamento de dados

Foram implementados mecanismos para restringir a partilha de recursos proprietários

10. Declaração de responsabilidade

Ao assinarem a presente declaração, os autores declaram que:

- a informação acima é verdadeira e reflete o uso efetivo de ferramentas de IA na realização do trabalho;
- compreendem que a IA não substitui autoria nem responsabilidade académica;
- verificaram a validaram e veracidade das referências bibliográficas incluídas no relatório
- assumem integralmente a responsabilidade técnica, científica, ética e académica por todo o conteúdo submetido, incluindo texto, código, modelos, testes, imagens, diagramas e restantes artefactos entregues.

11. Identificação dos autores

Nome(s): Rui Moreira

Número(s): 21600035

Data: 12 / 04 / 2026

Assinatura(s): Rui Pedro de Brito Pires Moreira

Glossário

| | |
|-----------------|---|
| API REST | Interface HTTP que troca dados em JSON entre cliente e servidor. |
| BD Relacional | Estrutura de dados organizada em tabelas ligadas entre si. |
| Facto Ecológico | Mensagem breve com informação ambiental educativa. |
| Flutter | Framework para criar apps móveis multiplataforma. |
| Flame | Motor de jogo 2D para Flutter. |
| Gamificação | Uso de mecânicas de jogo para motivar e ensinar. |
| Leaderboard | Tabela de classificações ordenada por pontuação. |
| Mini-jogo | Jogo curto e autónomo dentro de uma app. |
| Mobile Learning | Aprendizagem através de dispositivos móveis. |
| ODS | Objetivos globais da ONU para o desenvolvimento sustentável. |
| Monorepo | Estratégia de organização em que vários módulos ou packages relacionados são mantidos no mesmo repositório de código. |
| Serious Game | Jogo criado com finalidade educativa. |
| SQLite | Base de dados local e leve usada em dispositivos móveis. |
| TFC | Trabalho Final de Curso |