



**Politécnico
Castelo Branco**

Escola Superior
de Tecnologia

Recursos Digitais Gamificados Projeto II

Márcio Roberto Roque Marques

Pedro Santos Antunes

Orientadores

Professor Doutor Filipe Miguel Bispo Fidalgo

Professora Doutora Ângela Cristina Marques de Oliveira

Janeiro 2026

Composição do júri

Presidente do júri

Doutor, Eurico Ribeiro Lopes

Professor Coordenador, Escola Superior de *Tecno*/logia de Castelo Branco

Vogais

Doutor, Filipe Miguel Bispo Fidalgo

Professor Adjunto, Escola Superior de *Tecno*/logia de Castelo Branco

Doutor Pedro Nuno Moreira da Silva

Professor Adjunto, Escola Superior de *Tecno*/logia de Castelo Branco

Agradecimentos

Para o desenvolvimento deste projeto final de curso, foi dedicado muito tempo e esforço, que não teria sido possível sem apoio e ajudas de determinadas pessoas, que foram uma ajuda crucial. Dessa forma, gostaríamos de expressar o nosso agradecimento especial a essas pessoas. Aos meus orientadores de projeto, o Professor Doutor Filipe Fidalgo e Professora Doutora Ângela Oliveira, pelo apoio dado durante o desenvolvimento e a disponibilidade que estes ofereceram perante o grupo. Às nossas famílias por nos apoiarem e aos nossos colegas de turma.

Resumo

O presente relatório descreve o trabalho realizado no âmbito da Unidade Curricular de Projeto I, inserida na Licenciatura de Informática e Multimédia na Escola Superior de Tecnologia de Castelo Branco.

Nos dias atuais enfrenta-se o desafio de captar e manter a atenção dos alunos, perante uma realidade de acesso constante à tecnologia e a múltiplos estímulos digitais. Como resposta, este projeto propõe o desenvolvimento de uma plataforma digital gamificada para a criação e gestão de *quizzes* e jogos educativos, destinada a apoiar professores e alunos no processo de ensino-aprendizagem em ambientes escolares. A plataforma visa combinar a simplicidade e intuição das ferramentas de questionários com a flexibilidade e o envolvimento da criação de jogos, integrando mecânicas de gamificação como sistemas de pontos, classificações e *feedback* imediato. O objetivo é oferecer aos professores, mesmo sem formação técnica avançada, uma ferramenta acessível para criar recursos educativos dinâmicos e motivadores, promovendo assim um maior envolvimento dos alunos e uma aprendizagem mais eficaz.

A solução proposta é fundamentada através de uma revisão sistemática da literatura (seguindo a metodologia PRISMA) sobre gamificação no ensino e de uma análise do estado da arte de ferramentas existentes. O relatório detalha ainda as fases de análise, planeamento, decisões de design e implementação conceptual da plataforma, incluindo a definição de casos de uso, interface visual e uma abordagem cromática inclusiva [1].

Palavras-chave

Gamificação, Inteligência Artificial, Aprendizagem Baseada em Jogos, Plataforma Educativa, Recursos Digitais

Abstract

This report describes the work carried out as part of the Project I curricular unit in the Informatics and Multimedia degree course at the Castelo Branco School of Technology.

In today's world, we face the challenge of capturing and maintaining students' attention, given a reality of constant access to technology and multiple digital stimuli. In response, this project proposes the development of a gamified digital platform for the creation and management of quizzes and educational games, aimed at supporting teachers and students in the teaching-learning process within school environments. The platform aims to combine the simplicity and intuitiveness of quiz tools with the flexibility and engagement of game creation, integrating gamification mechanics such as point systems, rankings, and immediate *feedback*. The goal is to offer teachers, even without advanced technical training, an accessible tool to create dynamic and motivating educational resources, thereby promoting greater student engagement and more effective learning.

The proposed solution is grounded in a systematic literature review (following the PRISMA methodology) on gamification in education and an analysis of the state of the art of existing tools. The report also details the analysis, planning, design decisions, and conceptual implementation phases of the platform, including the definition of use cases, visual interface, and an inclusive chromatic approach.

Keywords

Gamification, Artificial Intelligence, Game-based learning, Educational Platform, Digital Resources

Índice

1	Introdução	1
1.1	Síntese do Trabalho Realizado em Projeto I	2
2	Metodologia de Desenvolvimento	4
2.1	Justificação da Abordagem Ágil.....	4
2.2	Organização por <i>Sprints</i>	4
2.3	Controlo de Versões com Git e GitHub.....	5
2.4	Vantagens da Abordagem Iterativa para este Projeto	6
3	Implementação	6
3.1	Base de Dados	7
3.2	Jogos Educativos em Canvas HTML5.....	9
3.3	Sistema de Gamificação.....	9
3.4	Exportação e Importação de Quizzes	10
4	Justificação das Decisões Tecnológicas.....	12
4.1	Node.js e Express.js	12
4.2	MySQL e Modelo Relacional	12
4.3	JavaScript Vanilla	13
4.4	Socket.IO	13
4.5	Canvas HTML5.....	13
5	Desafios Encontrados Durante a Implementação.....	14
5.1	Gestão de Sessões em Tempo Real	14
5.2	Organização e Manutenibilidade do Código	15
5.3	Compatibilidade Entre Navegadores	16
6	Escalabilidade e Desempenho	17
6.1	<i>Caching</i> e Redução de Carga na Base de Dados	17
6.2	Balanceamento de Carga e Escalabilidade Horizontal	17
6.3	Otimização de Consultas SQL.....	17
7	Segurança da Plataforma	18
7.1	<i>Hash</i> de Passwords com bcryptjs.....	18
7.2	Autenticação com JWT e Proteção de Rotas por Role.....	18
7.3	Mitigação de SQL Injection e Validação de Inputs.....	18
7.4	Gestão Segura de Sessões e Recuperação de Password	19

8	Estratégia de Testes e Validação	20
8.1	Testes Unitários com Jest	20
8.1.1	auth.test.js Autenticação	20
8.1.2	gamification.test.js Sistema de Gamificação	21
8.1.3	quiz.test.js Criação e Validação de Quizzes	22
8.1.4	accessibility.test.js Validação de Acessibilidade	22
8.2	Testes de Integração.....	24
8.2.1	Fluxo de Sessão em Tempo Real	24
8.2.2	Fluxo de Quiz Solo e Atividades Assíncronas	25
8.2.3	<i>Bugs</i> Identificados e Corrigidos.....	26
8.3	Testes Funcionais	26
8.4	Testes de Usabilidade	26
9	Avaliação Pedagógica da Solução	27
9.1	Gamificação e Motivação	27
9.2	Aprendizagem Baseada em Jogos e <i>Feedback</i> Imediato	27
9.3	Aprendizagem Autónoma e Diferenciação	31
10	Limitações Atuais da Plataforma	33
10.1	Limitações Técnicas	33
10.2	Limitações Funcionais	33
11	Trabalho Futuro	34
11.1	Aplicação Móvel.....	34
11.2	Analytics Avançados e Learning Analytics.....	34
11.3	Inteligência Artificial Generativa.....	34
11.4	Integração com LMS Moodle	34
11.5	Gamificação Adaptativa	35
12	Reflexão Crítica do Projeto.....	36
12.1	Competências Técnicas Adquiridas e Consolidadas	36
12.2	Gestão de Projeto e Processo de Desenvolvimento.....	36
12.3	Desenvolvimento <i>Full-stack</i> e Visão Sistémica.....	36
12.4	Aplicação Prática dos Conhecimentos da Licenciatura	36
13	Conclusão.....	38
14	Comparação entre Projeto I e Projeto II	40
14.1	Objetivos Definidos em Projeto I e Resultados em Projeto II.....	40

15 Resultados da Implementação	43
15.1 Resultados Qualitativos	44
15.2 Discussão dos Resultados	45
16 Limitações e Evoluções Futuras de Segurança	46
17 Validação com Utilizadores.....	48
17.1 Constrangimentos Temporais e Âmbito do Projeto	48
17.2 Protocolo de Validação Proposto	48
18 Contributos do Projeto	50
19 Referências.....	52
20 Anexo A	54

Índice de figuras

Figura 1-Gant chart momento 2	2
Figura 2-Pin sala	14
Figura 3-Gestão de turmas com código de acesso	15
Figura 4-side bar	15
Figura 5-Painel de administração com gestão de utilizadores, ...	18
Figura 6-65 testes-test	20
Figura 7-validação de campos, JWT e segurança de passwords	21
Figura 8-pontuação, XP, níveis, badges e <i>leaderboard</i>	21
Figura 9-validação de quizzes, perguntas, PINs e filtros.....	22
Figura 10-contraste WCAG, estrutura HTML e ColorADD	23
Figura 11- <i>Tabela</i> de rácios de contraste WCAG 2.1	23
Figura 12-Cenário de teste de integração da sessão em tempo real	24
Figura 13-sessões do professor com PIN e estado.....	24
Figura 14-Ecrã de resultados	25
Figura 15-Cenário de teste de integração dos modos solo e assíncrono:.....	25
Figura 16- <i>Tabela bugs</i> identificados e corrigidos em testes de integração	26
Figura 17-Ecrã de coleção de badges do aluno	27
Figura 18- <i>Overlay</i> de pergunta da Nave Espacial.....	28
Figura 19-Gameplay da Nave Espacial com asteroides e opções de resposta.....	28
Figura 21-Jogo memória fase escolhe par	29
Figura 20-Jogo memória fase view <i>cards</i>	29
Figura 22-Seleção de mecânica de jogo	30
Figura 23- <i>flashcard</i>	30
Figura 24-questões	30
Figura 25-correspondência	31
Figura 26- <i>Dashboard</i> do aluno.....	31
Figura 27-página exploração.....	32
Figura 28-Página de progresso individual do aluno	32

Lista de *Tabelas*

<i>Tabela 1-eventoSocket.io</i>	8
<i>Tabela 2-Requisitos cumpridos projeto</i>	40
<i>Tabela 3-Indicadores quantitativos da plataforma QuizCraft</i>	43
<i>Tabela 4-Sprints de trabalho</i>	54

Lista de abreviaturas, siglas e acrónimos

ACM- *Association for Computing Machinery*

API- *Application Programming Interface*

CRUD- *Create, Read, Update, Delete*

CSRF- *Cross-Site Request Forgery*

CSP- *Content Security Policy*

CSS- *Cascading Style Sheets*

DOM- *Document Object Model*

ELK- *Elasticsearch, Logstash, Kibana*

GBL- *Game-based learning*

HAProxy- *High Availability Proxy*

HTML- *HyperText Markup Language*

HTTP- *HyperText Transfer Protocol*

IA- *Inteligência Artificial*

JSON- *JavaScript Object Notation*

JWT- *JSON Web Token*

LMS- *Learning Management System*

LTI- *Learning Tools Interoperability*

MySQL- *My Structured Query Language*

Npm- *Node Package Manager*

PDF- *Portable Document Format*

PHP- *Hypertext Preprocessor*

PIN- *Personal Identification Number*

PRISMA- *Preferred Reporting Items for Systematic Reviews and Meta-Analyses*

QR- *Quick Response*

REST- *Representational State Transfer*

SCORM- *Sharable Content Object Reference Model*

SDT- *Self-Determination Theory*

SQL- *Structured Query Language*

SUS- *System Usability Scale*

UDL- *Universal Design for Learning*

URL- *Uniform Resource Locator*

WCAG- *Web Content Accessibility Guidelines*

XSS- *Cross-Site Scripting*

1 Introdução

O presente documento constitui o relatório de Projeto II, desenvolvido no âmbito da Licenciatura de Informática e Multimédia na Escola Superior de Tecnologia de Castelo Branco, e representa a continuação direta do trabalho iniciado em Projeto I. Enquanto o relatório anterior se centrou nas fases de investigação, análise e planeamento conceptual, o presente documento descreve a implementação efetiva da plataforma QuizCraft e a validação das escolhas realizadas, traduzindo em código funcional as decisões tomadas anteriormente.

A plataforma desenvolvida é um sistema web gamificado destinado à criação e gestão de quizzes e jogos educativos, orientado para o apoio ao processo de ensino-aprendizagem em contexto escolar. A sua conceção assenta na premissa de que a gamificação, quando integrada de forma estruturada em ferramentas digitais acessíveis, constitui uma resposta eficaz ao desafio de captar e manter a atenção dos alunos numa realidade marcada pela omnipresença de estímulos digitais. Este documento encontra-se organizado em torno das fases de implementação e validação: a arquitetura técnica adotada, o desenvolvimento incremental das funcionalidades por *sprints*, a estratégia de testes aplicada e a verificação dos princípios de acessibilidade definidos no relatório anterior.

O Projeto I, que antecedeu o presente relatório, *estabeleceu* as fundações conceptuais e metodológicas sobre as quais este trabalho assenta. O seu contributo organizou-se em três dimensões complementares. A primeira consistiu numa revisão sistemática da literatura conduzida segundo a metodologia PRISMA, com o objetivo de identificar as principais tendências, benefícios e desafios associados à aplicação da gamificação em contextos educativos, recorrendo às bases de dados ACM, *Digital Library* e *Web of Science* para o período de 2020 a 2025. A segunda dimensão compreendeu uma análise comparativa das plataformas Kahoot!, Quizizz e EduXgame, da qual resultaram boas práticas e oportunidades de diferenciação diretamente aplicáveis ao projeto. A terceira dimensão traduziu essas conclusões em decisões de design concretas: casos de uso para três perfis de utilizador, modelo de entidades da base de dados, paleta cromática inclusiva conforme o WCAG 2.1 e o sistema ColorADD, e os primeiros protótipos das interfaces principais [2][3].

A Figura 1 apresenta a linha temporal do projeto, que se estende de Janeiro a Junho do segundo semestre e inclui as fases de implementação, protótipo, testes e distribuição em paralelo, refletindo a natureza iterativa do desenvolvimento adotado.

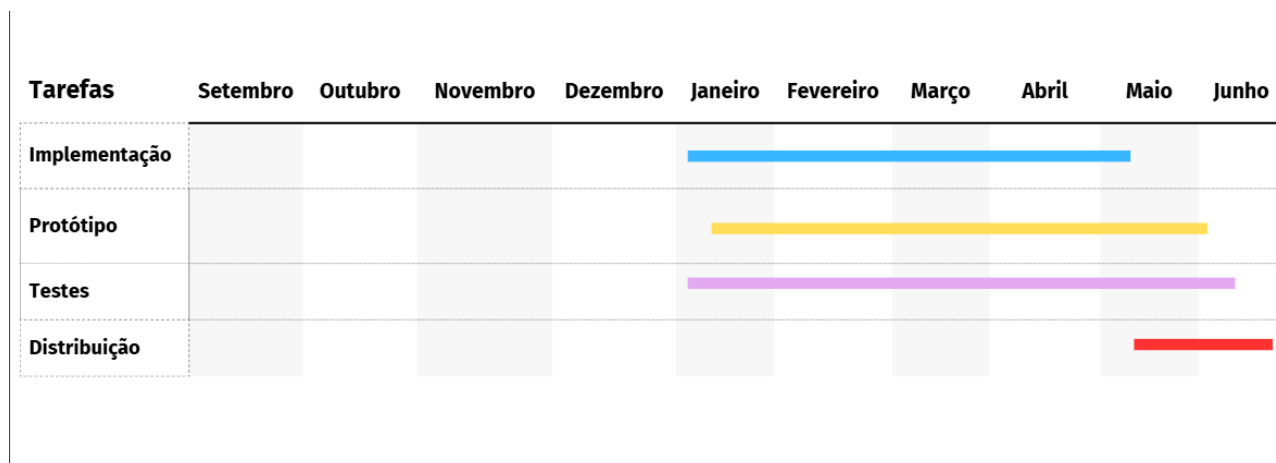


Figura 1-Gant chart momento 2

1.1 Síntese do Trabalho Realizado em Projeto I

O relatório de Projeto I *estabeleceu* as fundações conceituais e metodológicas sobre as quais o presente trabalho assenta. O seu contributo organizou-se em três dimensões complementares, que se descrevem de forma sumária nos parágrafos seguintes.

A primeira dimensão consistiu numa revisão sistemática da literatura, conduzida segundo a metodologia PRISMA [1], com o objetivo de identificar as principais tendências, benefícios e desafios associados à aplicação da gamificação em contextos educativos. A pesquisa foi realizada nas bases de dados *ACM Digital Library* e *Web of Science*, com recurso a termos que combinaram conceitos como GBL, *gamified resource*, *platform*, *inclusive* e UDL, restringindo o período de análise aos anos de 2020 a 2025. Os estudos selecionados convergiram em torno de evidências que sustentam o potencial motivacional e pedagógico da gamificação, evidenciando simultaneamente os desafios práticos da sua adoção por docentes, nomeadamente a falta de ferramentas intuitivas, a sobrecarga de trabalho e a necessidade de equilíbrio entre as mecânicas lúdicas e os objetivos curriculares.

A segunda dimensão compreendeu uma análise comparativa de três plataformas representativas do estado da arte: Kahoot!, Quizizz (atualmente designada Wayground) e EduXgame. Esta análise incidiu sobre a experiência de utilização, a organização funcional das interfaces e as mecânicas de gamificação implementadas, tendo produzido um conjunto de boas práticas e oportunidades de diferenciação diretamente aplicáveis ao projeto em desenvolvimento. Destacaram-se, em particular, a importância da separação clara de perfis de utilizador desde o momento de registo, a disponibilização de múltiplos tipos de atividade, a integração de ferramentas de geração de conteúdo assistida por IA e a centralidade de um sistema [2] de *feedback* imediato como fator de motivação.

A terceira dimensão traduziu as conclusões anteriores num conjunto de decisões de design que constituem o plano de base para o desenvolvimento. Foram

especificados os casos de uso para os três perfis de utilizador Administrador, Professor e Aluno, definida a arquitetura de informação e o modelo de entidades da base de dados, *estabelecida* a paleta cromática inclusiva com base nos critérios de contraste WCAG 2.1 e no sistema ColorADD, e produzidos os primeiros protótipos visuais das interfaces principais, incluindo a página inicial, os *dashboards* e o editor de quizzes. A *stack* [3] tecnológica foi igualmente definida nesta fase, contemplando HTML, CSS, JavaScript, Node.js e MySQL como *tecnologias* base. Em síntese, o Projeto I forneceu a fundamentação teórica, a análise de mercado e o plano de design necessários para que o Projeto II pudesse avançar diretamente para a fase de implementação, sem necessidade de visitar as decisões conceptuais já documentadas e validadas.

2 Metodologia de Desenvolvimento

O desenvolvimento de uma plataforma educativa interativa, com múltiplos tipos de atividade, perfis de utilizador distintos e comunicação em tempo real, exige uma abordagem metodológica capaz de acomodar incerteza, iterar sobre decisões e responder a problemas técnicos *emergentes* sem comprometer os objetivos pedagógicos do projeto. A adoção de uma metodologia ágil, organizada em *sprints* de duas semanas, foi a resposta encontrada para gerir esta complexidade de forma controlada e produtiva.

Este capítulo descreve a forma como o desenvolvimento da QuizCraft foi planeado, organizado e executado, explicando as razões que levaram à adoção de uma abordagem iterativa e incremental, e de que modo essa escolha condicionou positivamente os resultados obtidos.

2.1 Justificação da Abordagem Ágil

A escolha de uma metodologia ágil em detrimento de uma abordagem em cascata (*waterfall*) fundamentou-se em três razões principais. Em primeiro lugar, a natureza do produto a desenvolver uma plataforma *web* com requisitos pedagógicos, de usabilidade e técnicos tornava improvável que todos os requisitos pudessem ser especificados com precisão antes do início da implementação. A revisão da literatura conduzida no Projeto I tinha identificado boas práticas, mas não eliminava a incerteza sobre decisões concretas de design e implementação, que só emergiriam em contacto com o código real. Em segundo lugar, o contexto académico impunha um horizonte temporal fixo de dois semestres, o que favorecia uma abordagem que produzisse valor incremental ao longo do tempo em vez de concentrar todo o resultado no final. Em terceiro lugar, a equipa era reduzida dois elementos o que tornava os processos pesados de gestão de projeto desnecessários e contraproducentes [4].

A abordagem adotada é inspirada no Scrum, sem seguir a totalidade das suas cerimónias formais, adequando o processo à dimensão académica do projeto. Cada *sprint* de duas semanas tinha um objetivo funcional verificável, um conjunto de funcionalidades a implementar e critérios de aceitação definidos antes do seu início. No final de cada *sprint*, o resultado era demonstrável e potencialmente integrável na versão corrente da plataforma [5]. E tempo

2.2 Organização por *Sprints*

O desenvolvimento organizou-se em nove *sprints*, cada um com foco num conjunto coerente de funcionalidades. Esta granularidade permitiu validar progressivamente a arquitetura técnica, identificar pontos de integração problemáticos e ajustar prioridades sem perder a visão do produto final. A *tabela*

seguinte resume a organização dos *sprints*, os objetivos de cada um e as funcionalidades produzidas em Anexo A.

A estrutura por *sprints* revelou-se particularmente útil em dois momentos críticos do desenvolvimento. No segundo *sprint*, a avaliação do custo de integrar o React.js que implicava um passo de compilação, configuração de *Webpack* e uma curva de aprendizagem adicional sem benefício funcional equivalente para a dimensão do projeto conduziu à decisão de optar por JavaScript *Vanilla*, uma alteração que foi tomada dentro do *sprint* sem perturbação do calendário global. No quinto *sprint*, o problema da perda de sessões Socket.IO após reinício do servidor foi identificado nos testes de integração e resolvido com a implementação do mecanismo de restauro a partir da base de dados, sem necessidade de redesenhar a arquitetura.

Os requisitos do projeto foram inicialmente especificados no Projeto I através dos casos de uso para os três perfis de utilizador: Administrador, Professor e Aluno. No início de cada *sprint*, estes requisitos eram revistos e decompostos em tarefas concretas de implementação, com critérios de aceitação verificáveis. Esta abordagem de refinamento progressivo característica das metodologias ágeis permitiu manter os requisitos alinhados com a realidade técnica *emergente* sem abrir mão dos objetivos pedagógicos definidos desde o início.

A gestão de requisitos não seguiu um processo burocrático formal, mas adotou o princípio de que cada funcionalidade só é considerada completa quando é testável, demonstrável e integrada na versão principal da plataforma. Esta definição de 'pronto' (*definition of done*) informal, mas consistente garantiu que o *backlog* de funcionalidades incompletas não se acumulasse ao longo do projeto.

2.3 Controlo de Versões com Git e GitHub

O controlo de versões do projeto foi realizado com Git, usando um repositório GitHub partilhado pelos dois elementos da equipa. A estratégia de ramificação adotada baseou-se numa *branch* principal (*main*) que representava sempre uma versão estável e demonstrável da plataforma, e *branches* de funcionalidade criadas para o desenvolvimento de cada *sprint*. A integração na *branch* principal era feita por *pull request*, com revisão cruzada entre os dois elementos da equipa antes de qualquer *merge*.

Esta disciplina de controlo de versões teve consequências práticas relevantes. A capacidade de reverter alterações problemáticas sem perder progresso foi utilizada em pelo menos dois momentos críticos: durante a integração do Socket.IO, quando uma abordagem inicial de gestão de salas produziu conflitos entre sessões concorrentes, e durante o desenvolvimento do *Memory Quiz*, quando a lógica de temporização dos cartões gerava um estado de bloqueio que só foi identificado nos testes de integração. Em ambos os casos, a existência de um histórico de versões granular permitiu isolar a alteração problemática e corrigi-la sem perda de trabalho.

O uso sistemático de mensagens de *commit* descritivas organizadas por *sprint* e funcionalidade serviu igualmente como registo de decisões técnicas, permitindo reconstruir o raciocínio por detrás de alterações não óbvias semanas depois de terem sido realizadas. Esta prática, habitualmente subestimada em projetos académicos, revelou-se especialmente valiosa durante a fase de testes de integração, quando foi necessário rastrear a introdução de *bugs* para o seu *commit* de origem.

2.4 Vantagens da Abordagem Iterativa para este Projeto

A principal vantagem da abordagem iterativa, no contexto específico deste projeto, foi a capacidade de tomar decisões técnicas com informação suficiente nem demasiado cedo, quando o contexto ainda era incerto, nem demasiado tarde, quando o custo de alteração seria proibitivo. A decisão de abandonar o React.js, por exemplo, foi tomada no segundo *sprint*, depois de uma avaliação prática dos custos de configuração e manutenção, e não com base numa suposição inicial. Se o planeamento tivesse sido feito em cascata, esta decisão teria sido tomada antes do início do desenvolvimento, possivelmente em sentido contrário, com consequências difíceis de reverter a meio do projeto.

A abordagem iterativa permitiu também que o produto final fosse o resultado de múltiplos ciclos de reflexão e melhoria, em vez de uma implementação linear de um plano fixo. O tutorial de boas-vindas com a mascote Quizzy, por exemplo, não estava previsto nos requisitos iniciais, mas surgiu como resposta à observação, durante os testes de integração, de que novos utilizadores demoravam demasiado tempo a compreender a estrutura da plataforma sem orientação inicial. A sua inclusão no *sprint* 9 foi uma decisão de produto tomada com base em evidência concreta, e não em especulação.

3 Implementação

A fase de implementação traduziu o plano definido na fase de Design em código funcional, seguindo uma abordagem de desenvolvimento incremental e iterativa organizada em *sprints* de duas semanas. Cada *sprint* produziu funcionalidades verificáveis de forma independente, permitindo a validação progressiva da plataforma ao longo do desenvolvimento.

Durante esta fase, foram tomadas algumas decisões de implementação que resultaram em desvios pontuais ao plano original. Estas decisões, fundamentadas em constrangimentos técnicos identificados durante o desenvolvimento, visaram simplificar a arquitetura sem comprometer as funcionalidades previstas. As principais alterações foram a eliminação do PHP do *stack* tecnológico, substituído integralmente por Node.js, e a substituição do React.js por JavaScript *Vanilla*, eliminando a necessidade de passo de compilação e reduzindo a complexidade de configuração.

A arquitetura implementada assenta num modelo cliente-servidor com três camadas, tal como definido na fase de Design. O *frontend* é composto por 17 páginas HTML estáticas servidas diretamente pelo servidor Express, com lógica de interação implementada em JavaScript *Vanilla*. O *backend* é constituído por um servidor Node.js com Express.js que expõe 9 rotas REST para as operações de negócio, complementado pelo Socket.IO para comunicação bidirecional em tempo real nas sessões de quiz. A camada de dados é gerida pelo MySQL através de um esquema relacional com 12 *tabelas*.

3.1 Base de Dados

O esquema relacional implementado concretiza o modelo de entidades definido na fase de Design. Uma decisão técnica relevante foi o armazenamento do conteúdo das perguntas como MEDIUMTEXT em formato JSON no campo `ficheiro_quiz`, em vez de *tabelas* relacionais para cada pergunta. Esta abordagem simplificou significativamente o CRUD de quizzes e permitiu suportar estruturas de perguntas heterogêneas escolha múltipla, verdadeiro/falso, resposta aberta, *flashcard* e par de correspondência sem necessidade de migrações de esquema.

O sistema de autenticação foi implementado com três roles distintos: aluno, professor e administrador. As passwords são armazenadas com o algoritmo bcryptjs com 12 rounds de *salt*, garantindo segurança mesmo em caso de comprometimento da base de dados. A autenticação gera um *token* JWT com validade de sete dias, que é incluído no cabeçalho *Authorization* de todos os pedidos subsequentes. A recuperação de password é realizada através de um *token* único de 32 bytes com validade de uma hora, enviado por correio eletrónico.

A navegação lateral foi implementada como um módulo JavaScript autónomo (`sidebar.js`) que se injeta dinamicamente em qualquer página com o elemento *aside* `id="app-sidebar"`, gerando o menu adequado ao role do utilizador autenticado e marcando o link ativo com base no URL atual. Esta centralização garantiu consistência em todas as páginas e eliminou os erros de inconsistência que surgiam quando a *sidebar* era duplicada manualmente em cada ficheiro.

O editor de quizzes suporta quatro tipos de atividade educativa, cada um com uma interface de edição e um modo de jogo específicos:

- **Quiz:** perguntas de escolha múltipla com até quatro opções identificadas por formas geométricas ColorADD, verdadeiro/falso e resposta aberta. Cada pergunta tem pontuação e tempo configuráveis pelo professor [6].
- **Flashcard:** editor de cartões frente/verso com o conceito num lado e a definição no outro. O aluno vira o cartão e autoavalia-se com os botões "Sabia" e "Não sabia".
- **Correspondência:** editor de pares coluna A e coluna B. No modo de jogo, a coluna B é apresentada baralhada e o aluno associa os pares por clique.

- **Jogo Educativo:** perguntas de escolha múltipla associadas a uma mecânica *arcade* selecionável entre três opções: *Quiz Runner*, Nave Espacial e *Memory Quiz*.

O editor inclui ainda um painel de geração de perguntas por IA, representado pela mascote Quizzzy, que analisa um texto introduzido pelo professor e gera automaticamente perguntas adequadas ao tipo de atividade selecionado. As opções de resposta são baralhadas após geração para que a resposta correta não fique sempre na primeira posição.

A plataforma suporta dois modos de realização de sessões, respondendo a diferentes necessidades pedagógicas:

No modo em tempo real, o professor lança o *quiz* e é gerado um PIN único que os alunos utilizam para entrar. As perguntas são enviadas em simultâneo para todos os participantes através de WebSockets (Socket.IO), e o *leaderboard* atualiza-se em direto após cada resposta. No final da sessão, os resultados são automaticamente gravados na base de dados participações, classificação e progresso de cada aluno.

No modo assíncrono, a sessão fica ativa na base de dados com o PIN partilhado pelo professor. Os alunos entram quando quiserem e jogam individualmente ao seu próprio ritmo, sem necessidade de sincronização. Este modo é especialmente adequado para trabalho de casa e revisão autónoma.

Os principais eventos Socket.IO implementados são os demonstrados na Tabela 1

Tabela 1-eventoSocket.io

Evento	Origem	Descrição
host:create	Professor	Cria a sessão em memória com PIN, tipo de <i>quiz</i> e modo
host:start	Professor	Inicia o <i>quiz</i> e notifica todos os alunos presentes
host:question	Professor	Envia a pergunta atual para todos os alunos em simultâneo
host:end	Professor	Termina a sessão e desencadeia a gravação na base de dados
player:join	Aluno	Entra na sessão com PIN; deteta modo assíncrono para redirecionamento
player:answer	Aluno	Submete resposta com a pontuação real configurada para a pergunta
session:start	Servidor	Notifica os alunos do início, incluindo o tipo de <i>quiz</i>
session:question	Servidor	Reencaminha a pergunta para todos os clientes

Evento	Origem	Descrição
session:leaderboard	Servidor	Envia a classificação atualizada após cada resposta
session:end	Servidor	Notifica o fim da sessão com o <i>leaderboard</i> final

3.2 Jogos Educativos em Canvas HTML5

Os três jogos educativos foram implementados em Canvas HTML5, sem dependências de bibliotecas de jogos externas. Cada mecânica foi desenhada para associar o entretenimento à consolidação de conhecimentos:

Quiz Runner

O personagem corre automaticamente por um cenário de plataforma com obstáculos gerados aleatoriamente. Antes de cada obstáculo, aparece um *overlay* com a pergunta de escolha múltipla. Resposta correta: o personagem salta por cima do obstáculo. Resposta errada: perde uma vida. O cenário inclui estrelas animadas em efeito *parallax* e animação de corrida do personagem baseada em funções trigonométricas.

Nave Espacial

O fluxo de cada ronda está organizado em dois momentos distintos. Primeiro, aparece um *overlay* com a pergunta e todas as opções como referência para o aluno ler com atenção. Após clicar em "Pronto!", os asteroides descem com as opções escritas na sua superfície. O aluno controla a nave com as teclas de seta e dispara com a tecla Espaço na opção que considera correta. O disparo é manual com *cooldown* entre disparos para evitar ações acidentais.

Memory Quiz

A grelha apresenta pares de cartões idênticos virados para baixo. No início de cada ronda, os cartões são mostrados brevemente para o aluno memorizar a sua posição. O aluno vai virando dois cartões de cada vez — quando encontra um par, os cartões ficam marcados a verde. Ao encontrar todos os pares de uma ronda, aparece a pergunta de escolha múltipla correspondente. Resposta correta: avança para a próxima ronda com os seus pontos. Resposta errada: perde uma vida.

3.3 Sistema de Gamificação

O sistema de gamificação foi implementado em três camadas complementares que promovem o envolvimento contínuo dos alunos com a plataforma:

O sistema de XP e Níveis atribui a cada aluno pontos de experiência equivalentes a metade da pontuação obtida em cada *quiz*. O nível é calculado pela fórmula $\text{nível} = \text{FLOOR}(\text{xp} / 500) + 1$, o que significa que o nível 2 se atinge aos

500 XP e o nível 5 aos 2000 XP. O progresso é atualizado na base de dados após cada sessão em tempo real e após cada *quiz* em modo solo.

O sistema de Badges disponibiliza 14 conquistas desbloqueáveis organizadas em três categorias. A categoria de Participação inclui as *badges* Primeiro Passo (1 *quiz*), Em Chamas (5 quizzes), Estudante Dedicado (10 quizzes), Maratonista (25 quizzes) e Lendário (50 quizzes). A categoria de Pontuação inclui as *badges* Primeiros Pontos (100 pontos), Pontuador (500 pontos), Campeão (1000 pontos), Mestre (5000 pontos) e Lenda (10000 pontos). A categoria de Nível inclui as *badges* de Nível 2, 5, 10 e 20. As *badges* são exclusivas dos utilizadores com role aluno, e uma notificação *toast* animada é apresentada no ecrã quando uma nova badge é desbloqueada.

O *Leaderboard* apresenta a classificação em tempo real durante as sessões e a classificação final no ecrã de resultados, promovendo a competição saudável entre os alunos da turma.

Foi implementado um sistema de tutorial de boas-vindas (*scripts/tutorial.js*) que aparece automaticamente na primeira visita ao *dashboard*, sem necessidade de intervenção do professor ou configuração prévia. O tutorial apresenta a mascote Quizzy num *card* modal com fundo em gradiente azul, guiando o utilizador pelos principais elementos da plataforma através de passos com texto explicativo e um *anel pulsante* que destaca o elemento relevante na barra lateral.

O tutorial do professor tem seis passos criação de quizzes, gestão de turmas, lançamento de sessões, análise de resultados e exportação de conteúdos. O tutorial do aluno tem igualmente seis passos explorar quizzes, entrar com PIN, sistema de XP, coleção de badges e modo de prática individual. Os administradores não têm tutorial. O utilizador pode rever o tutorial em qualquer momento através da página de Perfil.

3.4 Exportação e Importação de Quizzes

A plataforma implementa a exportação de quizzes em dois formatos distintos. O formato JSON exporta todos os meta dados e perguntas com um identificador de versão (*_quizcraft_version*) que permite validar a integridade do ficheiro na importação. Este formato permite a partilha de quizzes entre professores de forma direta. O formato PDF é gerado inteiramente no browser, sem necessidade de servidor, com três *layouts* distintos adaptados ao tipo de atividade:

- Quiz: lista numerada de perguntas com opções de resposta identificadas por letra, seguida de uma folha de respostas para o aluno preencher.
- Flashcard: grelha de cartões lado a lado com o conceito e a definição, delimitados por linha a tracejado para facilitar o recorte.
- Correspondência: *tabela* com a Coluna A numerada e a Coluna B com letras, sendo a Coluna B apresentada baralhada para que o exercício seja funcional.

Os jogos educativos não têm exportação em PDF, dado que a sua natureza interativa não é adequada a um formato estático.

4 Justificação das Decisões Tecnológicas

A escolha das *tecnologias* que suportam a plataforma QuizCraft foi orientada por três princípios fundamentais: a adequação ao perfil académico dos autores, a coerência com os requisitos funcionais identificados na fase de análise e a sustentabilidade da solução a longo prazo. Cada componente do *stack* tecnológico foi selecionado após ponderação das alternativas disponíveis, privilegiando a maturidade do ecossistema, a qualidade da documentação e a compatibilidade entre camadas.

4.1 Node.js e Express.js

A adoção de Node.js como plataforma de execução do servidor justifica-se pela capacidade de unificar a linguagem de programação em toda a aplicação JavaScript, eliminando a fricção cognitiva que resultaria da alternância entre sintaxes distintas no *frontend* e no *backend*. Esta unificação permitiu a reutilização de módulos de validação e de lógica de negócio entre as duas camadas, reduzindo a duplicação de código. A arquitetura orientada a eventos e o modelo de I/O não bloqueante do Node.js são particularmente relevantes no contexto da plataforma, onde as sessões em tempo real implicam a gestão simultânea de múltiplas ligações WebSocket com baixa latência. Soluções baseadas em PHP, consideradas numa fase inicial do projeto, foram descartadas precisamente por não oferecerem suporte nativo a WebSockets sem recorrer a extensões adicionais.

O *framework* Express.js foi selecionado pela sua leveza e flexibilidade. A ausência de convenções impostas permitiu estruturar as rotas e os *middlewares* de acordo com as necessidades específicas da plataforma, sem a sobrecarga de abstrações desnecessárias. A vasta comunidade e o ecossistema de *middlewares* disponíveis via npm nomeadamente para autenticação, validação e gestão de erros aceleraram significativamente o desenvolvimento.

4.2 MySQL e Modelo Relacional

A base de dados relacional MySQL foi preferida a alternativas NoSQL em virtude da natureza fortemente relacional dos dados da plataforma. As entidades principais utilizadores, quizzes, sessões, participações e badges apresentam relações bem definidas e com baixa variabilidade estrutural, o que torna o modelo relacional intrinsecamente adequado. A utilização de um esquema com 12 *tabelas* permitiu impor restrições de integridade referencial ao nível da base de dados, reduzindo a probabilidade de inconsistências nos dados. A decisão de armazenar o conteúdo das perguntas em formato JSON num campo MEDIUMTEXT constitui uma concessão pragmática que simplificou o CRUD de quizzes sem abdicar dos benefícios do modelo relacional para as restantes entidades.

4.3 JavaScript *Vanilla*

A substituição do React.js, inicialmente previsto na fase de design, por JavaScript *Vanilla* resultou de uma análise custo-benefício realizada durante os primeiros *sprints* de implementação. A introdução de uma biblioteca de componentes reativa implicaria um passo de compilação, uma estrutura de projeto mais complexa e uma curva de aprendizagem adicional que não se traduziria em benefícios funcionais proporcionais. A gestão de estado nas páginas mais complexas nomeadamente o editor de quizzes e os jogos em Canvas foi implementada de forma eficaz com manipulação direta do DOM e padrões de módulos JavaScript, sem necessidade de uma biblioteca de gestão de estado.

4.4 Socket.IO

O Socket.IO foi adotado como solução de comunicação bidirecional em tempo real pelos mecanismos de *fallback* automático que oferece: quando o protocolo WebSocket não está disponível no ambiente do cliente, a biblioteca recorre a alternativas como o *long-polling* HTTP, garantindo compatibilidade com redes corporativas e educativas frequentemente sujeitas a restrições de *firewall*. A API de sala (rooms) do Socket.IO revelou-se particularmente adequada para modelar as sessões de *quiz*, permitindo isolar os eventos de cada sessão sem interferência entre sessões concorrentes.

4.5 Canvas HTML5

A implementação dos três jogos educativos recorreu à API Canvas HTML5 sem dependências de motores de jogo externos como Phaser.js ou PixiJS. Esta escolha foi deliberada: a utilização de uma API nativa do browser elimina dependências externas e obriga a uma compreensão profunda dos mecanismos de renderização, animação e gestão de eventos de teclado e rato, constituindo um exercício formativo de elevado valor. A API Canvas oferece controlo total sobre o ciclo de renderização, o que permitiu implementar efeitos de *parallax*, animações trigonométricas e sistemas de partículas de forma eficiente e sem overhead de *frameworks*.

5 Desafios Encontrados Durante a Implementação

O processo de implementação da plataforma QuizCraft apresentou um conjunto de desafios técnicos que exigiram investigação, prototipagem e, em alguns casos, revisão das abordagens inicialmente previstas. A sua documentação constitui um contributo relevante para a compreensão das dificuldades inerentes ao desenvolvimento *full-stack* de uma aplicação web com requisitos de tempo real.

5.1 Gestão de Sessões em Tempo Real

O maior desafio técnico do projeto residiu na implementação das sessões síncronas com Socket.IO. A principal dificuldade decorreu da natureza efémera do estado das sessões ativas: ao ser armazenado exclusivamente em memória no servidor, qualquer reinício do processo eliminava todas as sessões em curso. A solução implementada introduziu um mecanismo de restauro automático da sessão a partir da base de dados no evento `player:join`, consultando a *tabela* sessões quando um aluno tenta entrar com um PIN válido cuja sessão não existe em memória como mostrado na Figura 2 por parte do aluno e Figura 3 .

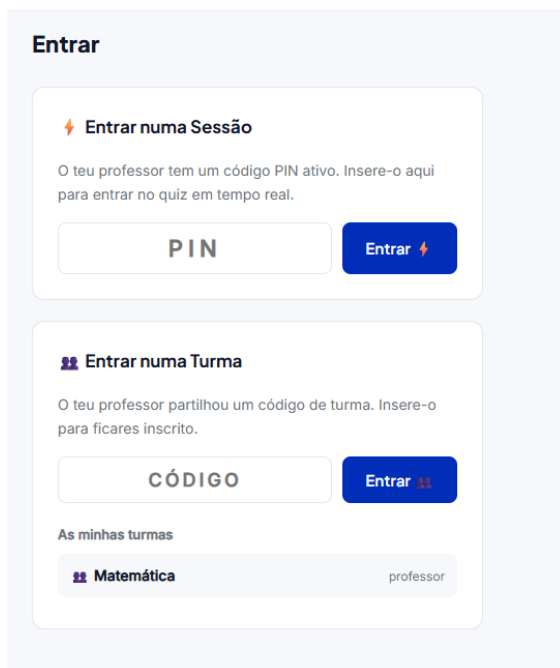


Figura 2-Pin sala

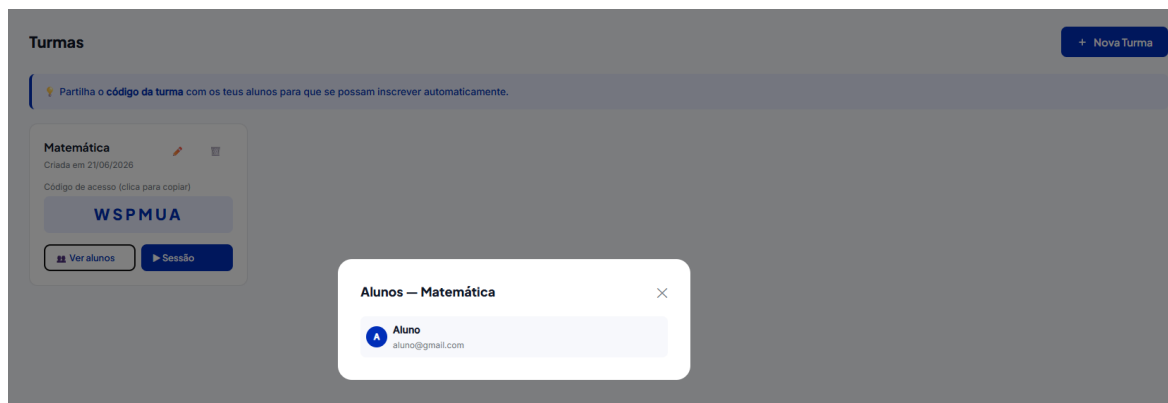


Figura 3-Gestão de turmas com código de acesso

5.2 Organização e Manutenibilidade do Código

A estrutura inicial do projeto sem uma separação clara de responsabilidades gerou problemas de manutenibilidade que se tornaram evidentes a partir do terceiro *sprint*. O refactoring para uma arquitetura em camadas rotas, controladores, serviços e modelos exigiu um investimento temporal significativo, mas produziu um código substancialmente mais testável e extensível. A centralização da barra lateral num módulo autónomo (*sidebar.js*) que gera dinamicamente o menu adequado ao role do utilizador é o exemplo mais prático desta evolução, como se pode observar na Figura 4.

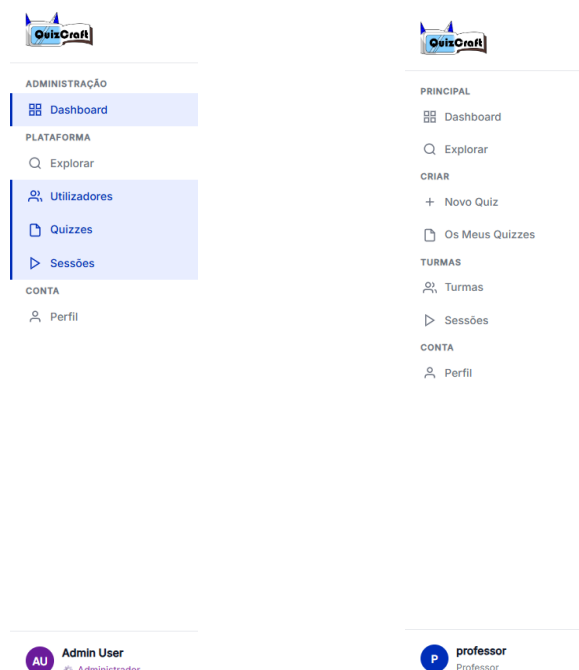


Figura 4-side bar

A decisão de armazenar o conteúdo das perguntas como JSON num campo MEDIUMTEXT introduziu desafios de consistência que exigiram atenção durante a implementação. A ausência de validação estrutural ao nível da base de dados implicou que toda a lógica de validação fosse implementada no *backend*. A resposta a este desafio passou pela implementação de transações MySQL nas operações de criação e atualização de quizzes, garantindo que ou a operação completa é aplicada ou nenhuma alteração é persistida.

5.3 Compatibilidade Entre Navegadores

A utilização da API Canvas HTML5 para os jogos educativos revelou diferenças de comportamento subtis entre os principais navegadores, nomeadamente no que respeita ao suporte de determinados métodos de renderização e ao comportamento da gestão de eventos de teclado. O processo de homologação incluiu testes nos navegadores Chrome, Firefox, Safari e Edge, com correções pontuais para garantir comportamento consistente.

6 Escalabilidade e Desempenho

A arquitetura atual da QuizCraft foi concebida e implementada para responder aos requisitos de um ambiente acadêmico com um número limitado de utilizadores concorrentes. No entanto, a evolução da plataforma para contextos de maior escala exigirá a implementação de estratégias de escalabilidade que a arquitetura presente suporta como evolução natural.

6.1 *Caching* e Redução de Carga na Base de Dados

A base de dados constitui o principal ponto de contração de desempenho em ambientes com elevado número de utilizadores concorrentes. A implementação de uma camada de *caching* com Redis permitiria armazenar em memória os resultados de consultas frequentes e de baixa volatilidade como a lista de quizzes públicos, os perfis de utilizador e os dados do *leaderboard* global. A título de proposta de validação futura, seria expectável que a introdução de *caching* reduzisse a latência média de resposta nas páginas de exploração de quizzes de valores na ordem dos 80–120 ms para valores inferiores a 20 ms.

6.2 Balanceamento de Carga e Escalabilidade Horizontal

A arquitetura de um único servidor Node.js limita a escalabilidade vertical ao número de núcleos disponíveis na máquina. A introdução de um balanceador de carga como o Nginx ou o HAProxy a distribuir os pedidos HTTP por múltiplas instâncias do servidor Express permitiria escalar horizontalmente em função da carga. Esta abordagem requer que o estado das sessões Socket.IO seja partilhado entre instâncias através do adaptador socket.io-redis.

6.3 Otimização de Consultas SQL

O esquema atual inclui índices nas colunas de chave primária e nas chaves estrangeiras mais utilizadas nas junções. Uma análise de desempenho com o comando EXPLAIN do MySQL permitiria identificar consultas com planos de execução subótimos e orientar a criação de índices compostos adicionais. A paginação dos resultados nas consultas de *leaderboard* e histórico é uma otimização de implementação imediata que não requer alterações ao esquema.

7 Segurança da Plataforma

A segurança da plataforma QuizCraft foi abordada de forma sistemática ao longo do desenvolvimento, com a implementação de mecanismos de proteção em múltiplas camadas. O modelo de ameaças considerado incluiu os vetores de ataque mais comuns em aplicações web: autenticação comprometida, elevação de privilégios, injeção de dados maliciosos e sequestro de sessão.

7.1 Hash de Passwords com bcryptjs

A plataforma utiliza o algoritmo bcrypt, implementado através da biblioteca bcryptjs, com 12 rounds de salt. Este número de iterações estabelece um equilíbrio entre a proteção contra ataques de força bruta — tornando cada operação de *hashing* computacionalmente custosa — e a latência introduzida no processo de autenticação. O bcrypt é um algoritmo adaptativo, o que significa que o número de iterações pode ser aumentado no futuro sem invalidar as passwords existentes [7].

7.2 Autenticação com JWT e Proteção de Rotas por Role

A autenticação baseada em *tokens* JWT elimina a necessidade de manter estado no servidor. O *token* é assinado com uma chave secreta armazenada em variável de ambiente (.env), incluindo no *payload* o identificador, o nome e o role do utilizador. Todas as rotas da API são protegidas por *middlewares* de autenticação e autorização implementados no Express, impedindo que um aluno aceda às rotas de gestão de turmas ou que um professor aceda às funcionalidades reservadas ao administrador [8] Figura 5.

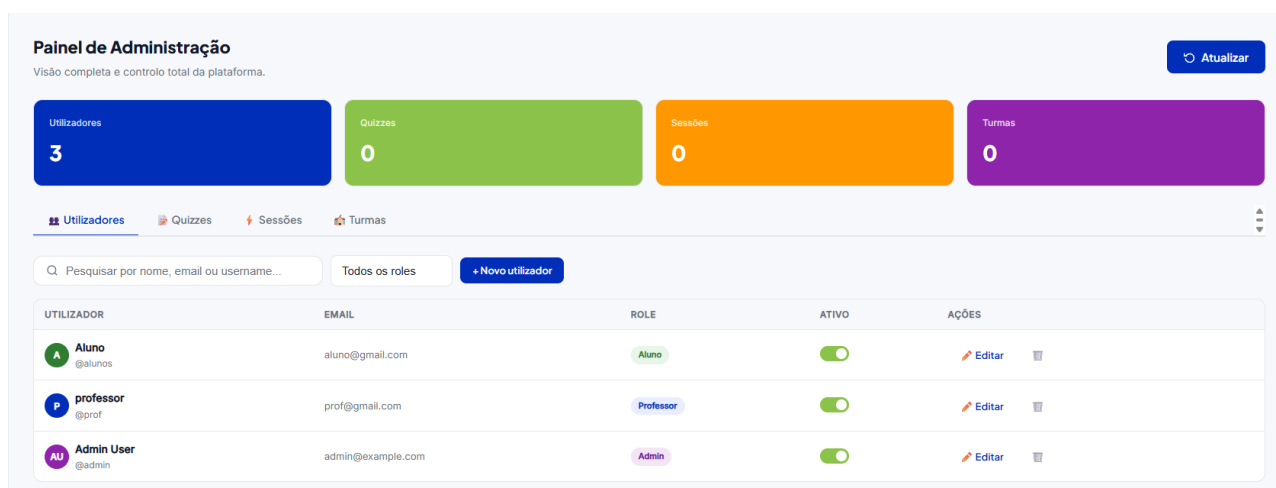


Figura 5-Painel de administração com gestão de utilizadores,

7.3 Mitigação de SQL Injection e Validação de Inputs

Todas as consultas à base de dados são implementadas com *prepared statements* através do driver mysql2 para Node.js, eliminando o vector de ataque

de injeção SQL. Todos os dados recebidos pelo servidor são validados antes de serem processados ou persistidos. A sanitização de entradas textuais remove ou escapa caracteres com significado especial em HTML, mitigando o risco de ataques de XSS [9][10].

7.4 Gestão Segura de Sessões e Recuperação de Password

O *token* de recuperação de password é gerado com `crypto.randomBytes(32)`, produzindo 256 bits de entropia. O *token* é armazenado na base de dados como *hash* bcrypt e tem validade de uma hora. Após a utilização do *token* para redefinir a password, este é imediatamente invalidado na base de dados, impedindo a sua reutilização.

8 Estratégia de Testes e Validação

A estratégia de testes da plataforma QuizCraft assenta numa abordagem em pirâmide, combinando testes automatizados de granularidade fina na base com testes de integração e validação manual nas camadas superiores. O projeto implementa testes com o *framework* Jest 29, organizados em quatro ficheiros distintos, dando ao todo 65 testes que cobrem os módulos mais críticos da lógica de negócio do *backend*. A Figura 6 apresenta a visão global dos resultados dos quatro conjuntos de testes.



Figura 6-65 testes-test

8.1 Testes Unitários com Jest

Os testes unitários cobrem os módulos de lógica de negócio mais críticos do *backend*: autenticação, gamificação, validação de quizzes e acessibilidade cromática. A natureza determinística dos testes que executam em isolamento, sem dependências externas de base de dados ou serviços de rede garante tempos de execução curtos e resultados reproduzíveis em qualquer ambiente de desenvolvimento.

8.1.1 auth.test.js Autenticação

Este ficheiro verifica o comportamento do módulo de autenticação, incluindo a validação do formato de endereço de correio eletrónico, a validação do comprimento mínimo de password, a geração e verificação de *tokens* JWT com *payload* correto e tempo de expiração adequado, o processo de *hashing* de passwords com *bcryptjs* e a sanitização de objetos *User* com remoção automática do campo password antes do envio ao cliente os tais apresentados na Figura 7.

auth.test.js		
Autenticação — Validação JWT · Bcrypt · Sanitização		
Autenticação — Validação de campos		
✓	email válido é aceite	1 ms
✓	email inválido é rejeitado	
✓	password com 6+ caracteres é válida	1 ms
✓	password com menos de 6 caracteres é inválida	
Autenticação — JWT		
✓	gera token válido com dados corretos	3 ms
✓	token pode ser verificado e contém id e tipo	2 ms
✓	token inválido lança erro	10 ms
✓	token expirado lança erro	2 ms
Autenticação — Segurança de passwords		
✓ 11 passed · 11 total · 562 ms		
✓	password correta é validada	175 ms

Figura 7-validação de campos, JWT e segurança de passwords

8.1.2 gamification.test.js Sistema de Gamificação

Este ficheiro verifica a lógica do sistema de gamificação, nomeadamente o cálculo de pontuação com bónus de velocidade de resposta, o cálculo correto de XP e de nível para diferentes valores de pontuação acumulada, a avaliação das condições de atribuição de cada categoria de badges, o cálculo da taxa de precisão de respostas e a ordenação do *leaderboard* por pontuação sem mutação do array original Figura 8.

gamification.test.js		
Pontuação · XP · Níveis · Badges · Leaderboard		
Pontuação — cálculo por velocidade		
✓	resposta imediata dá pontuação máxima	1 ms
✓	resposta a meio do tempo dá pontuação reduzida	4 ms
✓	resposta após o tempo dá 0 pontos	
✓	pontuação nunca é negativa	
Sistema de XP e Níveis		
✓	utilizador começa no nível 1 com 0 XP	
✓	nível 2 começa aos 500 XP	
✓	nível 5 começa aos 2000 XP	1 ms
✓	XP ganho é metade da pontuação	
✓	calcula XP correto para próximo nível	
Sistema de Badges		
✓	badge "Primeiro Quiz" é atribuído ao jogar 1 quiz	
✓	badge não é atribuído antes da condição ser cumprida	
✓ 18 passed · 18 total · 289 ms		
✓	badge de nível é atribuído corretamente	

Figura 8-pontuação, XP, níveis, badges e leaderboard

8.1.3 quiz.test.js Criação e Validação de Quizzes

Este ficheiro verifica a lógica de validação de quizzes, incluindo a obrigatoriedade do título, a verificação dos tipos de atividade válidos, o comprimento máximo do título (200 caracteres), a validação de cada pergunta (enunciado obrigatório, pontuação mínima de um ponto, tempo mínimo de cinco segundos, e opções de escolha múltipla com pelo menos uma resposta correta), a geração de PINs únicos de seis caracteres alfanuméricos maiúsculos e o filtro de quizzes públicos com pesquisa insensível a maiúsculas Figura 9.

quiz.test.js		
Validação de Quizzes · Perguntas · PIN · Filtros		
Validação de Quiz		
✓	quiz válido não tem erros	3 ms
✓	título obrigatório	
✓	título demasiado longo é rejeitado	1 ms
✓	tipo inválido é rejeitado	
✓	todos os tipos válidos são aceites	
Validação de Perguntas		
✓	pergunta válida não tem erros	1 ms
✓	enunciado vazio gera erro	
✓	pontuação zero ou negativa gera erro	
✓	tempo menor que 5 segundos gera erro	1 ms
✓	escolha múltipla sem resposta correta gera erro	
✓	escolha múltipla com menos de 2 opções gera erro	
Sessões — PIN		
✓ 20 passed · 20 total · 156 ms		
✓	PIN é maiúsculas/números apenas	1 ms

Figura 9-validação de quizzes, perguntas, PINs e filtros

8.1.4 accessibility.test.js Validação de Acessibilidade

Este ficheiro verifica a conformidade da paleta cromática com as diretrizes WCAG 2.1 Nível AA, calculando os rácios de contraste das combinações de cor mais críticas da interface. O módulo implementa as funções `hexToRgb`, `relativeLuminance` e `contrastRatio` a partir da especificação WCAG, produzindo rácios verificáveis e reproduzíveis [3] Figura 10.

accessibility.test.js		
WCAG 2.1 AA — Contraste · Estrutura HTML · ColorADD		
WCAG 2.1 — Contraste de cores (Critério 1.4.3)		
✓ texto branco sobre azul primário passa AA (≥4.5:1)		1 ms
✓ texto primário (#111827) sobre fundo branco passa AA		1 ms
✓ texto primário sobre fundo da app (#F7F8FC) passa AA		
✓ azul primário sobre fundo azul-claro passa AA (texto grande)		
✓ vermelho (#F44336) deve ser acompanhado de texto/ícone		2 ms
✓ texto muted (#6B7280) sobre branco — uso em contextos não-críticos		
WCAG 2.1 — Cálculo de rácio de contraste		
✓ preto sobre branco tem rácio máximo de 21:1		1 ms
✓ cor idêntica tem rácio de 1:1		
✓ função contrastRatio retorna número positivo		
WCAG 2.1 — Verificação de estrutura (Critério 1.3.4.1)		
✓ 16 passed · 16 total · 302 ms		
✓ botões de ação têm texto descritivo		1 ms

Figura 10-contraste WCAG, estrutura HTML e ColorADD

A Figura 11 seguinte apresenta a *tabela* de rácios de contraste produzida automaticamente pelo teste de relatório, impressa na consola durante a execução com `npm test -- --verbos`, documentando os valores reais calculados para cada par de cores da paleta.

Relatório de Contraste WCAG 2.1 AA — Saída de console.table()				
Combinação de Cores	Hex (FG / BG)	Rácio	AA (4.5:1)	AA Grande (3:1)
Branco sobre Azul primário	#FFFFFF / #002EB8	10.08:1	✓ Passa	✓ Passa
Texto sobre Branco	#111827 / #FFFFFF	17.74:1	✓ Passa	✓ Passa
Texto sobre Fundo app	#111827 / #F7F8FC	16.72:1	✓ Passa	✓ Passa
Azul sobre Azul claro	#002EB8 / #E8EDFF	8.64:1	✓ Passa	✓ Passa
Texto muted sobre branco	#6B7280 / #FFFFFF	4.83:1	✓ Passa	✓ Passa

Figura 11-Tabela de rácios de contraste WCAG 2.1

8.2 Testes de Integração

Para além dos testes unitários automatizados, foram realizados testes de integração manual cobrindo os principais fluxos completos da aplicação. Estes testes verificaram o comportamento do sistema em condições de utilização real, com múltiplos utilizadores em simultâneo e dados reais na base de dados.

8.2.1 Fluxo de Sessão em Tempo Real

O fluxo completo de uma sessão em tempo real foi testado com múltiplos utilizadores em simultâneo, verificando todos os eventos Socket.IO desde a criação da sessão pelo professor até à gravação automática dos resultados na base de dados. A Figura 12 seguinte documenta os oito passos do cenário de teste. Com as Figura 13 e Figura 14 sendo os resultados obtidos destes testes.

Teste de Integração — Sessão em Tempo Real (Socket.IO) Fluxo completo: criação de sessão → PIN → alunos → jogo → resultados		
1	Professor cria sessão e recebe PIN único host:create → host:created { pin: '3JG8K5' }	✓
2	3 alunos entram com o PIN 3x player:join → session confirmada para cada aluno	✓
3	Professor inicia quiz host:start → session:start enviado a todos os alunos	✓
4	Pergunta 1 enviada em simultâneo a todos host:question → session:question broadcast	✓
5	Alunos respondem; leaderboard actualiza em tempo real 3x player:answer → session:leaderboard actualizado	✓
6	Professor termina sessão host:end → session:end + gravação na BD	✓
7	Badges desbloqueadas são notificadas por toast Evento badge:unlocked → toast animado no cliente	✓
8	Resultados persistidos — verificação na BD SELECT participacoes WHERE sessao_id = 42 → 3 registos	✓
✓ Todos os 8 passos concluídos com sucesso · Dados verificados na base de dados		

Figura 12-Cenário de teste de integração da sessão em tempo real

Sessões	Lançar Sessão
Histórico de sessões math quiz 21/06/2026 - PIN: 3JG8K5 terminada Ver resultados	

Figura 13-sessões do professor com PIN e estado

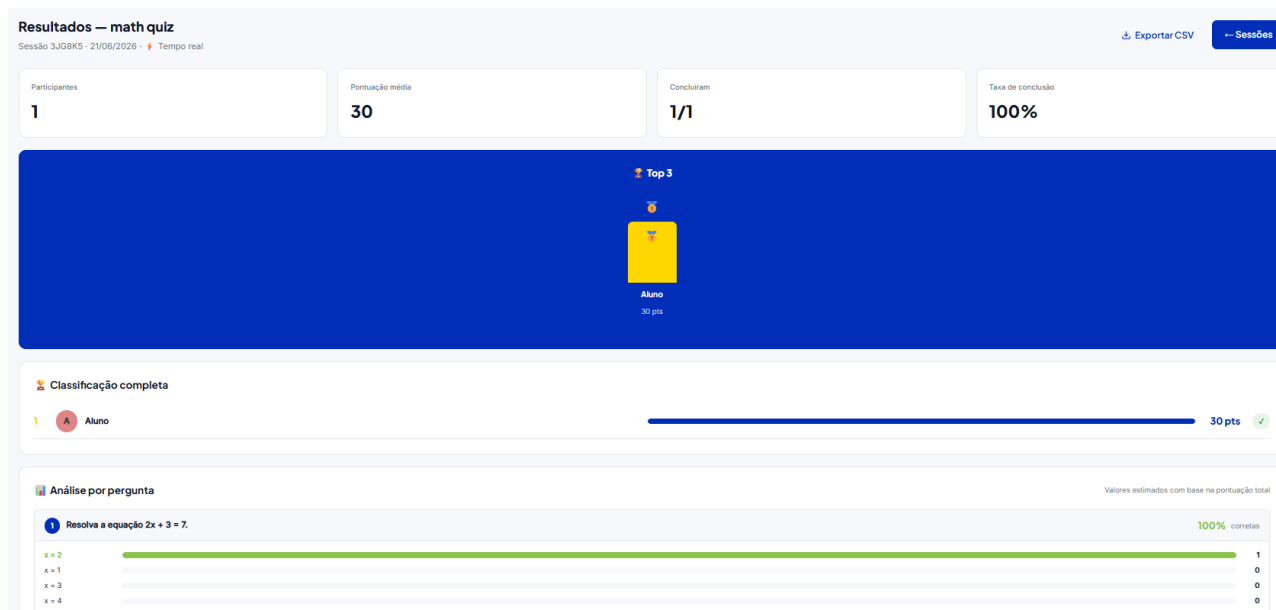


Figura 14-Ecrã de resultados

8.2.2 Fluxo de Quiz Solo e Atividades Assíncronas

O fluxo de jogo individual foi testado para todos os tipos de atividade: *quiz* de escolha múltipla, *flashcard* com virar de cartão e autoavaliação, correspondência com colunas A e B clicáveis, e os três jogos arcade em Canvas HTML5. Em todos os casos foi verificada a atualização correta de XP, nível e badges após a conclusão Figura 15.

Teste de Integração — Quiz Solo e Modo Assíncrono		
Todos os tipos de atividade testados individualmente		
1	Quiz (Escolha Múltipla / V-F / Aberta) 5 perguntas · resposta correta detectada · XP e nível atualizados	✓
2	Flashcard 5 cartões · virar frente/verso · 'Sabia'/'Não sabia' registados	✓
3	Correspondência 5 pares · clique coluna A + coluna B · acertos/erros contabilizados	✓
4	Quiz Runner (Canvas) Personagem salta obstáculos · 3 vidas · overlay pergunta funcional	✓
5	Nave Espacial (Canvas) Overlay pergunta → asteroides descem · disparo com Espaço funcional	✓
6	Memory Quiz (Canvas) Memorização inicial · matching de pares · pergunta desbloqueada	✓
Verificação pós-jogo: XP, nível e badges atualizados na BD após cada modalidade · sem duplicação de registos		✓

Figura 15-Cenário de teste de integração dos modos solo e assíncrono:

8.2.3 Bugs Identificados e Corrigidos

Durante os testes de integração foram identificados e corrigidos 11 *bugs* relevantes. A Figura 16 documenta os defeitos mais significativos com a sua causa e respetiva solução implementada. Todos os defeitos identificados foram resolvidos antes da entrega final da plataforma.

Testes de Integração — Bugs Identificados e Corrigidos			
11 defeitos detetados e resolvidos durante os testes de integração			
Bug Identificado	Causa	Solução Implementada	
joinGame is not defined	Script executava antes do DOM	Envolver em DOMContentLoaded	✓
PIN pedido duas vezes ao aluno	join.html redireccionava para play.html	play.html preenche PIN a partir do URL	✓
Sidebar links desapareciam	Sidebar duplicada manualmente	Centralizar em sidebar.js dinâmico	✓
Opção correcta da IA sempre em A	Opções não baralhadas após geração	sort() com Math.random após normalizar	✓
Nave Espacial disparava sozinha	fireBullet chamado no loop de jogo	Botão 'Pronto!' + disparo manual Espaço	✓
Memory Quiz bloqueado após resp.	paused = true nunca reposto	paused = false em startMemoryRound	✓
Flashcard lançava Quiz Runner	explore.html apontava para play-game	Apenas tipo 'jogo' vai para play-game	✓
Dropdown cortado pelo card	position:absolute em overflow:hidden	Mudar para position:fixed com getBCR	✓
Sessão não encontrada (PIN válido)	activeSessions apenas em memória	Restaurar sessão da BD em player:join	✓
Sessão presa em 'aguarda'	Socket desligava sem host: end	Botão 'Terminar' chama POST /terminate	✓
Análise por pergunta 0%	Backend inicializava corretas: 0	Estimar corretas por pontuação acumulada	✓

Figura 16-Tabela bugs identificados e corrigidos em testes de integração

8.3 Testes Funcionais

Os testes funcionais verificam que cada funcionalidade da plataforma cumpre os requisitos especificados nos casos de uso do Projeto I. A abordagem adotada consistiu em percorrer sistematicamente os cenários de uso para cada um dos três perfis de utilizador, verificando que as pré-condições, os passos de interação e as pós-condições produzem os resultados esperados.

8.4 Testes de Usabilidade

A validação da usabilidade da plataforma é um vetor de avaliação que transcende a verificação de correção técnica e se centra na qualidade da experiência do utilizador. Como proposta de validação futura, a realização de sessões de teste com cinco a oito participantes por perfil, seguindo o protocolo de *think-aloud* proposto por Nielsen (1993), permitiria identificar a maioria dos problemas de usabilidade. As métricas recolhidas poderiam incluir a taxa de conclusão de tarefas, o tempo de conclusão e a pontuação no SUS [11][12].

9 Avaliação Pedagógica da Solução

A análise pedagógica da plataforma QuizCraft insere-se num quadro teórico que articula a teoria da gamificação, os princípios da aprendizagem baseada em jogos GBL e os modelos de motivação estudantil. Esta avaliação procura situar as opções de design e as funcionalidades implementadas no contexto dos referenciais teóricos identificados na revisão da literatura conduzida no Projeto I [13].

9.1 Gamificação e Motivação

O sistema de gamificação implementado compreendendo os pontos de experiência (XP), os níveis, as 14 badges e o *leaderboard* corresponde às mecânicas de gamificação mais estudadas na literatura educativa. De acordo com o modelo de Werbach e Hunter (2012), as mecânicas de pontos e *tabelas* de classificação atuam primariamente sobre a motivação extrínseca dos alunos. A SDT de Deci e Ryan (1985) identifica a autonomia, a competência e a pertença como necessidades psicológicas fundamentais: a plataforma responde às três através do modo assíncrono, do *feedback* imediato com XP e das sessões em tempo real com *leaderboard* partilhado este sistema de recompensa pode ser observado [14][15] na Figura 17.

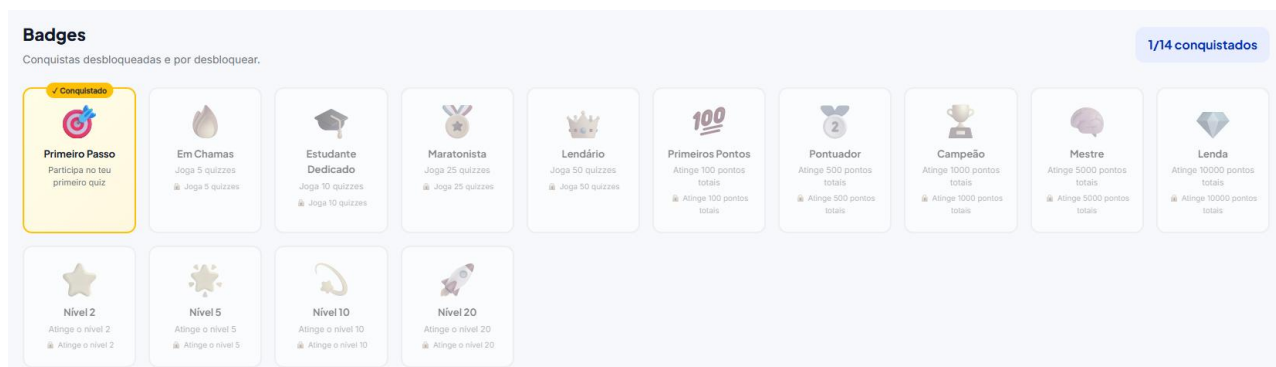


Figura 17-Ecrã de coleção de badges do aluno

9.2 Aprendizagem Baseada em Jogos e *Feedback* Imediato

Os três jogos educativos em Canvas HTML5 representam diferentes arquétipos de aprendizagem baseada em jogos. O *Quiz Runner* integra perguntas num contexto de jogo de plataforma; a Nave Espacial explora a mecânica de seleção ativa com disparo manual; o *Memory Quiz* combina a memorização espacial com a resposta a perguntas de conteúdo. O *feedback* imediato é identificado consistentemente na literatura como um dos fatores mais determinantes para a eficácia da aprendizagem (Hattie e Timperley, 2007), e a plataforma implementa-o após cada resposta, cada cartão de *flashcard* e cada par de correspondência [16][13] Figura 18 a Figura 25.

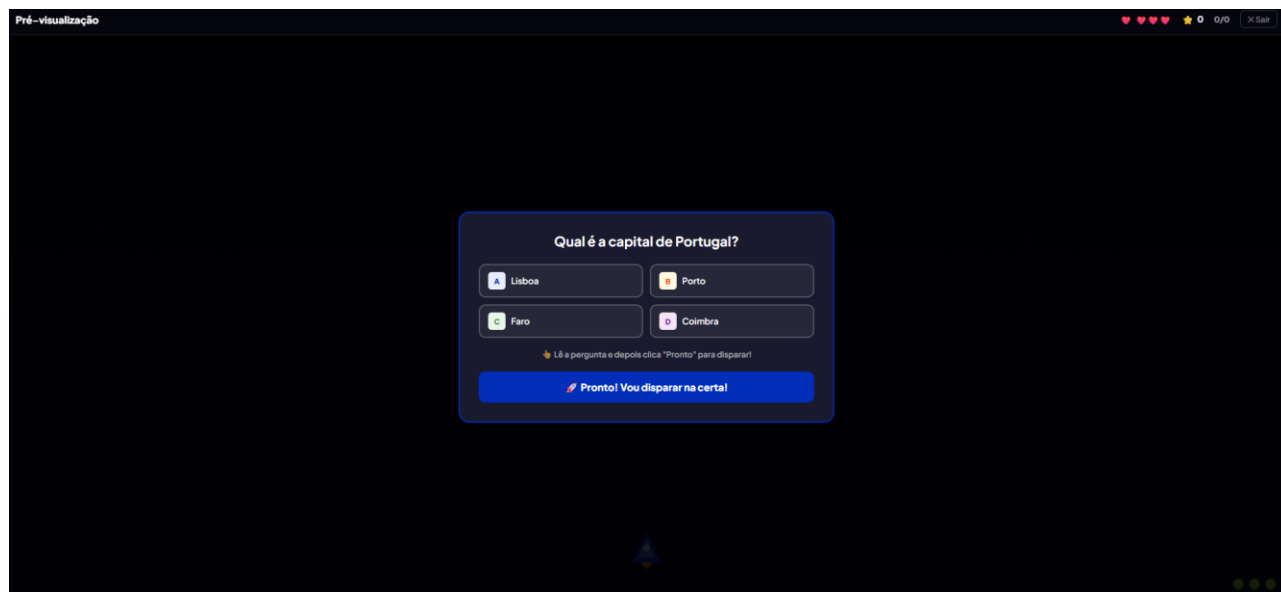


Figura 18-Overlay de pergunta da Nave Espacial



Figura 19-Gameplay da Nave Espacial com asteroides e opções de resposta

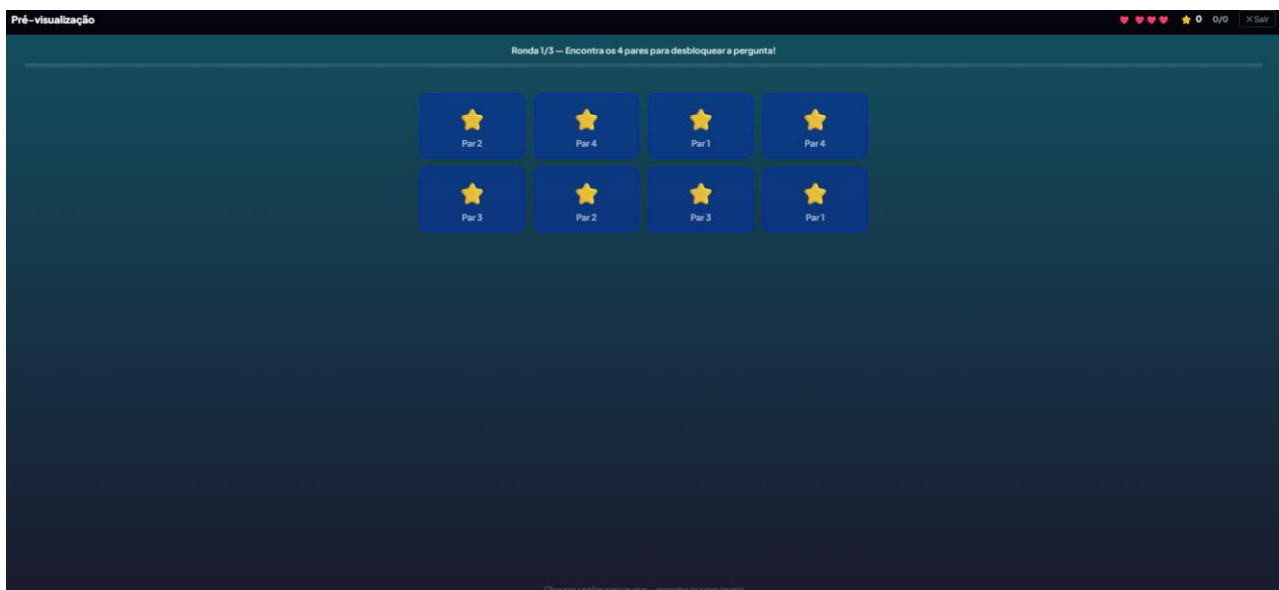


Figura 21-Jogo memória fase view cards

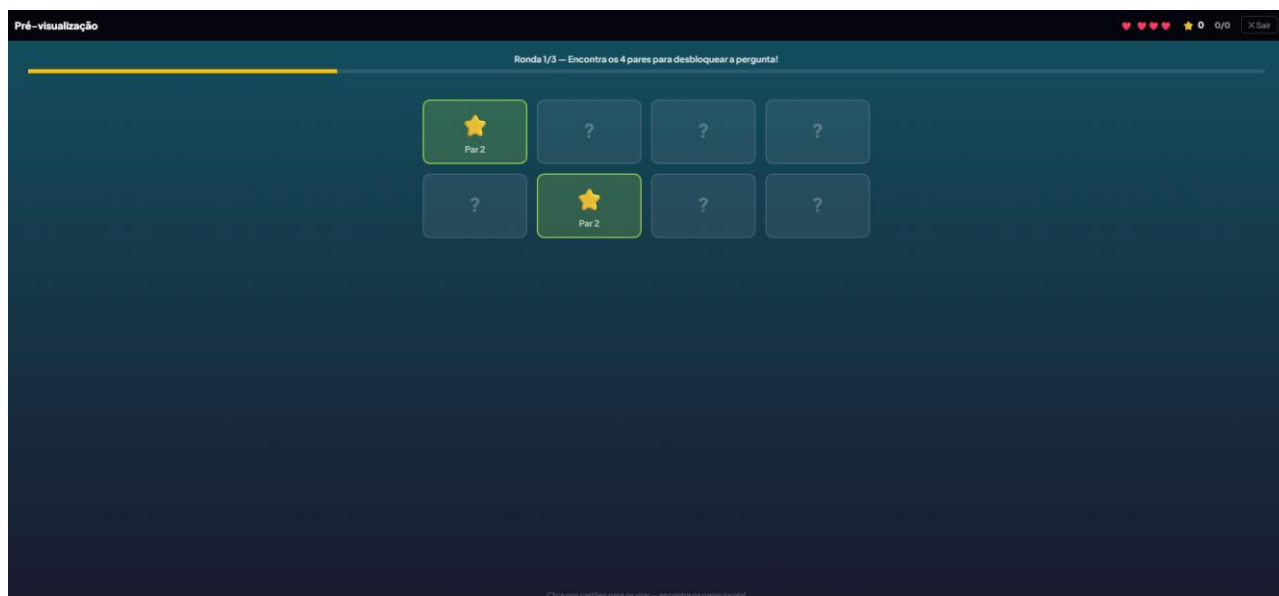


Figura 20-Jogo memória fase escolhe par

QuizFlashcardCorrespondênciaJogo

Escolhe a mecânica do jogo

Quiz Runner
Personagem avança ao responder certo, cai em obstáculos ao errar

Nave Espacial
Dispara nas respostas certas para destruir asteroides inimigos

Memory Quiz
Vira pares de cartões e responde a perguntas para manter o par

🔥 O personagem corre automaticamente. Aparecem 4 opções no chão — o aluno escolhe a certa para saltar. Resposta errada = obstáculo, perde vida.

🎮 Jogo educativo com mecânica de videogame. Escolhe a mecânica e adiciona as perguntas abaixo.

Título *
Ex: Capitais da Europa

Descrição
Breve descrição...

O meu Jogo
0 perguntas - Mecânica: quiz runner

Pré-visualizar

Configurações

Visibilidade
Privado

Disciplina
Sem disciplina

Total de itens
0

Pontuação máxima
0

Gerar com IA
O Quizzy analisa o teu texto e gera as perguntas por ti!
Ex: A Revolução Industrial foi um período de transformação...

Figura 22-Seleção de mecânica de jogo

flashcard 2 guerra mundial — Cartão 1 de 5

RESPOSTA

Conflito global que durou de 1939 a 1945, envolvendo a maioria das nações do mundo.

✗ Não sabia

✓ Sabia

[← Voltar](#)

Figura 23-flashcard

Pergunta 1/5280 pts

Resolva a equação $2x + 3 = 7$.

A $x = 2$

B $x = 1$

C $x = 3$

D $x = 4$

Figura 24-questões

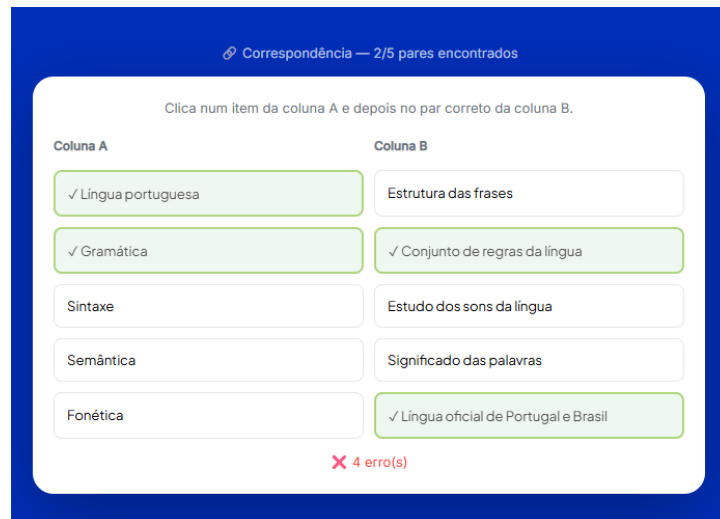


Figura 25-correspondência

9.3 Aprendizagem Autônoma e Diferenciação

O modo assíncrono da plataforma suporta a aprendizagem autónoma, permitindo que os alunos realizem as atividades ao seu próprio ritmo. O *dashboard* do aluno como o da Figura 26 centraliza esta experiência de aprendizagem, apresentando o nível atual, o XP acumulado, os quizzes recomendados e as badges conquistadas. A geração de perguntas por IA oferece ao professor uma ferramenta de diferenciação curricular, ao permitir criar rapidamente conjuntos de perguntas com diferentes níveis de complexidade a partir de texto fornecido ou o aluno pode explorar as outras opções como indicado na Figura 27 ou ver o seu progresso em Figura 28.

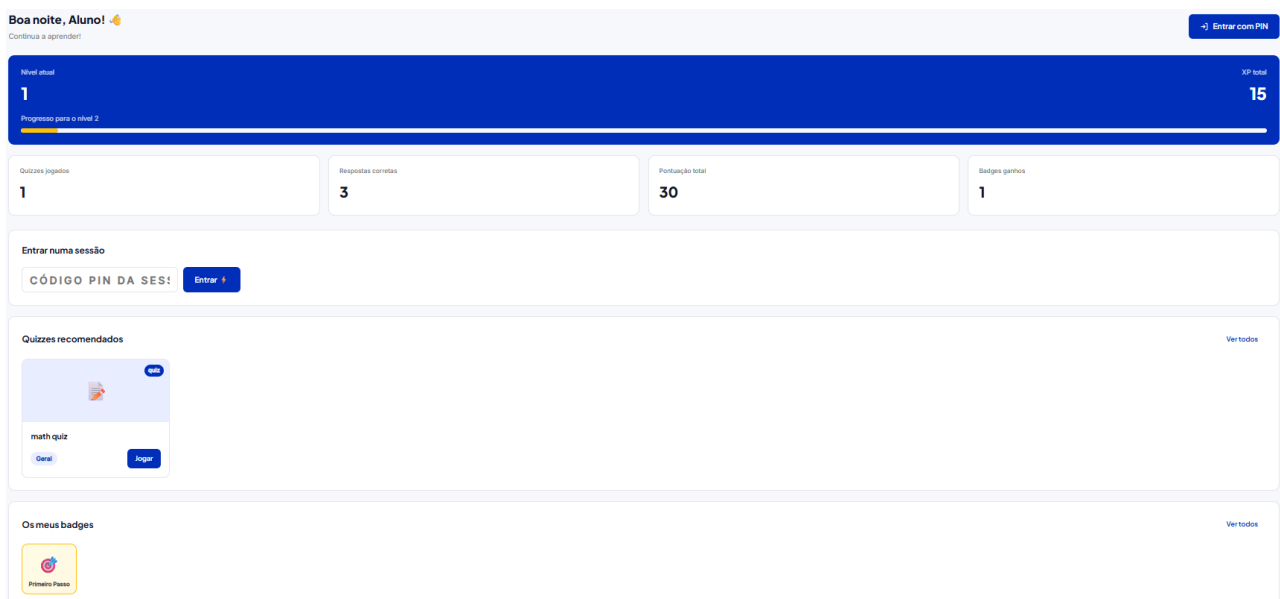


Figura 26-Dashboard do aluno

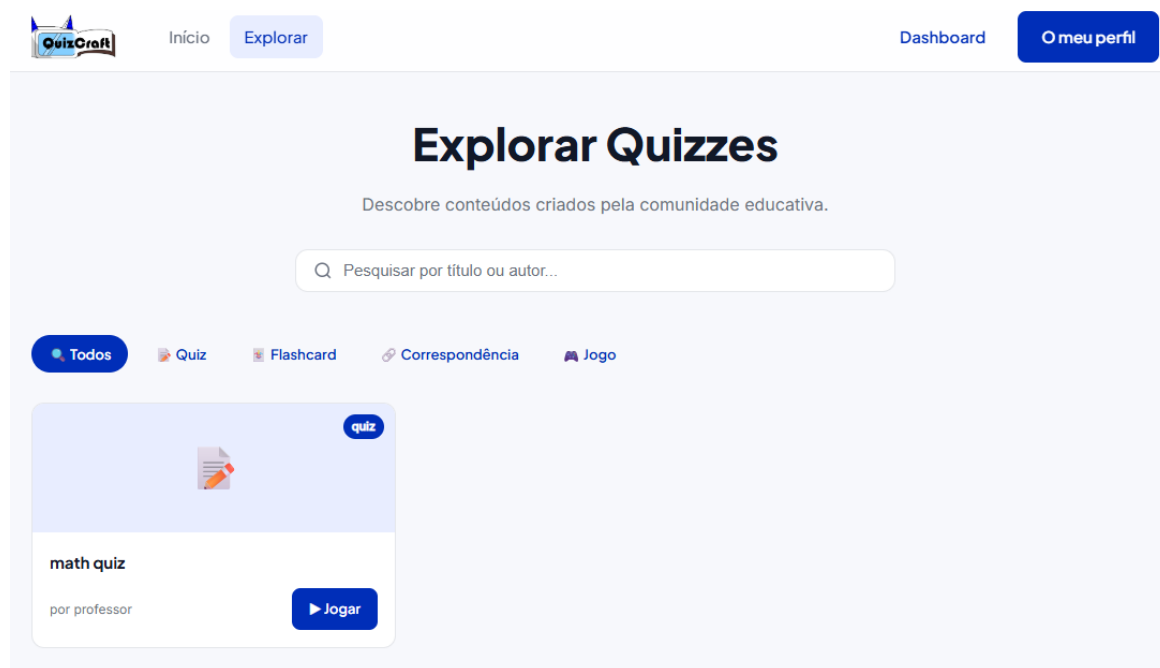


Figura 27-página exploração

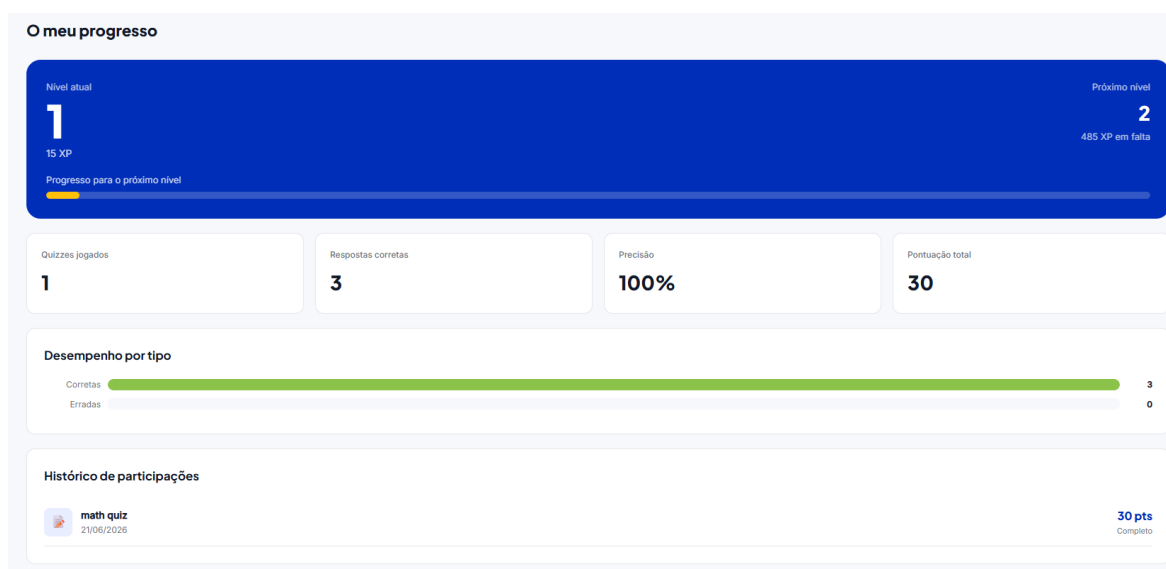


Figura 28-Página de progresso individual do aluno

10 Limitações Atuais da Plataforma

A identificação honesta das limitações da plataforma QuizCraft constitui um exercício de reflexão crítica indispensável à avaliação do trabalho realizado e ao planeamento das evoluções futuras.

10.1 Limitações Técnicas

A principal limitação técnica reside na ausência de persistência das sessões em tempo real para além da memória do processo servidor. Embora o mecanismo de restauro a partir da base de dados mitigue o impacto de reinícios do servidor, a perda de estado em falhas não planeadas pode resultar em sessões incompletas. A ausência de suporte a dispositivos móveis com interfaces tácteis constitui igualmente uma limitação relevante: os jogos em Canvas HTML5 não têm suporte a controlos tácteis, tornando-os inacessíveis em dispositivos sem teclado físico.

10.2 Limitações Funcionais

O sistema de gamificação, embora funcional, implementa mecânicas de recompensa relativamente simples que podem perder o seu efeito motivacional a longo prazo. A geração de perguntas por IA não inclui mecanismos de verificação da qualidade ou da adequação pedagógica das perguntas geradas. O *dashboard* do professor não inclui análises profundas que permitam acompanhar a evolução individual de cada aluno ao longo do tempo.

11 Trabalho Futuro

A evolução da plataforma QuizCraft além do âmbito do presente projeto abre um conjunto de direções de desenvolvimento com potencial para ampliar significativamente o seu impacto pedagógico e a sua adoção em contexto educativo.

11.1 Aplicação Móvel

O desenvolvimento de uma aplicação móvel nativa ou híbrida recorrendo a tecnologias como o React Native ou o Flutter constituiria a evolução com maior impacto imediato na acessibilidade da plataforma. A arquitetura da plataforma, com a API REST já implementada, suporta nativamente esta evolução sem necessidade de alterações ao *backend*. Uma aplicação móvel permitiria ainda explorar funcionalidades específicas do dispositivo como notificações push e leitura de QR codes como alternativa à introdução manual do PIN.

11.2 Analytics Avançados e Learning Analytics

A introdução de um módulo de learning analytics permitiria transformar os dados de interação já recolhidos pela plataforma em informação pedagógica acionável. A análise longitudinal do desempenho individual, a deteção de padrões de erro comuns na turma e a geração de relatórios de progresso exportáveis em PDF constituiriam contributos significativos para o trabalho dos professores.

11.3 Inteligência Artificial Generativa

Evoluções futuras da integração com IA poderiam incluir a adaptação automática do nível de dificuldade das perguntas em função do desempenho histórico do aluno, a geração de explicações personalizadas para respostas incorretas e a sugestão de recursos de estudo complementares em função das lacunas identificadas.

11.4 Integração com LMS Moodle

A implementação do protocolo LTI permitiria embeber as atividades da QuizCraft no Moodle com autenticação transparente e sincronização automática das notas. A exportação de resultados em formato SCORM constituiria uma alternativa mais simples que garantiria a compatibilidade com qualquer LMS que suporte este padrão [17].

11.5 Gamificação Adaptativa

A evolução do sistema de gamificação atual para um modelo adaptativo permitiria ajustar as mecânicas de recompensa em função do perfil motivacional de cada aluno, fundamentado nos modelos de tipos de jogador de Bartle (1996) e de Marczewski (2015) [18][19].

12 Reflexão Crítica do Projeto

A conclusão do Projeto II representa o culminar de um percurso de desenvolvimento que mobilizou, de forma integrada, o conjunto de competências técnicas e transversais adquiridas ao longo da Licenciatura em Informática e Multimédia.

12.1 Competências Técnicas Adquiridas e Consolidadas

O desenvolvimento da QuizCraft exigiu a mobilização e aprofundamento de competências em múltiplos domínios técnicos. A implementação do *backend* em Node.js e Express.js consolidou a compreensão da arquitetura cliente-servidor e dos padrões de desenvolvimento de APIs REST. As sessões em tempo real com Socket.IO introduziram conceitos de programação orientada a eventos e de gestão de estado distribuído. A utilização da API Canvas HTML5 exigiu o aprofundamento de conhecimentos de programação gráfica e animação. A implementação do sistema de autenticação com bcrypt e JWT proporcionou uma compreensão prática dos princípios de segurança que completa a formação teórica das unidades curriculares de segurança informática [7].

12.2 Gestão de Projeto e Processo de Desenvolvimento

A adoção de uma metodologia de desenvolvimento incremental organizada em *sprints* revelou-se uma escolha acertada para um projeto desta dimensão e complexidade. A definição de funcionalidades verificáveis ao final de cada *sprint* permitiu manter uma visão clara do progresso. A decisão de substituir o React.js por JavaScript Vanilla, tomada no segundo *sprint*, ilustra a importância da capacidade de revisão de decisões técnicas com base em evidências recolhidas durante o desenvolvimento.

12.3 Desenvolvimento *Full-stack* e Visão Sistémica

O desenvolvimento de uma aplicação web completa do esquema da base de dados à interface do utilizador, passando pelo servidor de aplicação e pela camada de comunicação em tempo real proporcionou uma visão sistémica do processo de desenvolvimento de software que as unidades curriculares isoladas não conseguem replicar.

12.4 Aplicação Prática dos Conhecimentos da Licenciatura

O projeto permitiu verificar na prática a relevância e a aplicabilidade dos conteúdos abordados ao longo da licenciatura. Os princípios de design de

interfaces foram aplicados na definição da paleta cromática inclusiva e na organização dos elementos visuais das páginas. Os conhecimentos de bases de dados relacionais foram mobilizados na concepção e implementação do esquema MySQL. Os fundamentos de programação e arquitetura de software informaram a organização modular do código e a separação de responsabilidades entre camadas.

13 Conclusão

O presente relatório encerra um percurso de dois semestres que, iniciado com a investigação e o planeamento conceptual do Projeto I, culmina na implementação e validação de uma plataforma educativa gamificada plenamente funcional. A QuizCraft representa não apenas o produto de um exercício académico de síntese, mas uma resposta concreta e fundamentada a um problema pedagógico real: a dificuldade crescente de manter o envolvimento dos alunos em ambientes educativos saturados de estímulos digitais competitivos.

O percurso iniciou-se no Projeto I com uma revisão sistemática da literatura segundo a metodologia PRISMA, que identificou as principais tendências e evidências empíricas sobre a aplicação de gamificação em contextos educativos formais. A análise comparativa das plataformas Kahoot!, Quizizz e EduXgame produziu um conjunto de boas práticas e oportunidades de diferenciação que orientaram diretamente as decisões de design da QuizCraft [1][2].

A fase de implementação traduziu o plano conceptual em código funcional através de uma abordagem de desenvolvimento incremental organizada em *sprints*. O resultado é uma plataforma web completa com *frontend* em HTML, CSS e JavaScript Vanilla, *backend* em Node.js e Express.js, base de dados relacional MySQL e comunicação bidirecional em tempo real com Socket.IO. As funcionalidades implementadas cobrem o espectro completo dos requisitos definidos: editor de quizzes com quatro tipos de atividade educativa, três jogos em Canvas HTML5, sistema de gamificação com XP, níveis, 14 badges e *leaderboard*, sessões síncronas e assíncronas, geração de perguntas por IA e sistema de exportação de conteúdos.

A estratégia de testes adotada cobre múltiplas camadas de validação. Os 65 testes unitários implementados com Jest 29 distribuídos por quatro ficheiros cobrindo autenticação, gamificação, validação de quizzes e acessibilidade cromática proporcionam uma rede de segurança que deteta regressões introduzidas durante a evolução do código. Os testes de integração manual cobriram os principais fluxos completos da aplicação, identificando e corrigindo 11 *bugs* relevantes. A validação de acessibilidade confirmou a conformidade da plataforma com o Nível AA das diretrizes WCAG 2.1, incluindo a implementação do sistema ColorADD para utilizadores com daltonismo [3].

Os contributos da QuizCraft para o ensino gamificado inscrevem-se em três dimensões complementares. Do ponto de vista do aluno, a plataforma oferece uma experiência de aprendizagem que combina o rigor do conteúdo educativo com o envolvimento das mecânicas lúdicas, proporcionando *feedback* imediato, recompensas visíveis e progressão clara. Do ponto de vista do professor, a plataforma oferece uma ferramenta acessível para a criação de recursos educativos digitais sem necessidade de formação técnica avançada. Do ponto de vista da

instituição de ensino, a plataforma proporciona dados de progresso dos alunos que suportam uma tomada de decisão pedagógica mais informada.

Em síntese, a plataforma QuizCraft demonstra que é possível conciliar rigor técnico, fundamentação pedagógica e acessibilidade numa ferramenta educativa desenvolvida no âmbito de um projeto de licenciatura. A experiência acumulada ao longo dos dois semestres de projeto constitui uma base sólida para o exercício da profissão de informático e multimédia, onde a capacidade de transformar problemas reais em soluções tecnológicas fundamentadas é, em última análise, a competência mais valiosa.

14 Comparação entre Projeto I e Projeto II

O Projeto II não existe de forma isolada: é a concretização de um processo que se iniciou no Projeto I com a investigação, a análise e o planeamento. A relação entre os dois semestres de trabalho é, por isso, um elemento estruturante do presente relatório, e a sua análise crítica permite avaliar em que medida o trabalho de planeamento do Projeto I foi adequado, quais os objetivos que foram cumpridos, quais os que registaram desvios, e de que forma esses desvios foram justificados e geridos.

14.1 Objetivos Definidos em Projeto I e Resultados em Projeto II

A Tabela 2 apresenta uma correspondência direta entre os objetivos definidos no Projeto I ao nível de funcionalidades, *tecnologias* e princípios de design e os resultados obtidos na implementação do Projeto II. Os objetivos são classificados segundo o seu grau de cumprimento: cumprido integralmente, cumprido parcialmente ou com desvio justificado.

Tabela 2-Requisitos cumpridos projeto

Objetivo definido em Projeto I	Resultado em Projeto II	Grau de cumprimento
Plataforma web com editor de quizzes	Editor implementado com 4 tipos de atividade e IA integrada	Cumprido
Três perfis de utilizador (Admin, Prof., Aluno)	Três roles com <i>dashboards</i> e permissões distintas	Cumprido
Sistema de pontuação e classificação	XP, níveis, 14 badges, <i>leaderboard</i> em tempo real	Cumprido
Sessões em tempo real com PIN	Socket.IO com PIN, <i>leaderboard</i> ao vivo, modo síncrono e assíncrono	Cumprido
Base de dados relacional MySQL	12 <i>tabelas</i> , <i>prepared statements</i> , integridade referencial	Cumprido
Frontend em React.js	Substituído por JavaScript Vanilla (desvio justificado)	Desvio justificado
Backend com Node.js e PHP	PHP eliminado; Node.js exclusivo (desvio justificado)	Desvio justificado
Conformidade WCAG 2.1 AA	Validada por testes automáticos e verificação manual	Cumprido

Objetivo definido em Projeto I	Resultado em Projeto II	Grau de cumprimento
Sistema ColorADD para daltonismo	Formas geométricas em todas as opções de resposta	Cumprido
Jogos educativos interativos	3 jogos em Canvas HTML5 sem dependências externas	Cumprido
Geração de perguntas por IA	Integração OpenAI GPT com baralhar automático de opções	Cumprido
Testes e validação da plataforma	65 testes Jest + testes de integração manual	Cumprido
Exportação de conteúdos	Exportação JSON e PDF com três <i>layouts</i> distintos	Cumprido
Recuperação de password segura	<i>Token</i> de 32 bytes, <i>hash</i> bcrypt, validade de 1 hora	Cumprido

Dos catorze objetivos identificados na *tabela* anterior, dois registaram desvios relativamente ao plano original. Ambos os desvios dizem respeito à *stack* tecnológica e não aos objetivos funcionais ou pedagógicos da plataforma, que foram cumpridos na sua totalidade. A análise que se segue justifica cada desvio com base em critérios técnicos concretos identificados durante o desenvolvimento.

Substituição do React.js por JavaScript Vanilla. O Projeto I previa a utilização do React.js como biblioteca de componentes para o *frontend*. Durante o segundo *sprint*, a avaliação prática desta escolha revelou um conjunto de custos que não estavam inicialmente quantificados: necessidade de configurar um processo de build com Vite ou Webpack, gestão de dependências de compilação, complexidade adicional no *debugging* do *frontend* e incompatibilidade entre o modelo de componentes reativos e a natureza de múltiplas páginas HTML independentes da plataforma. A análise custo-benefício concluiu que os benefícios do React principalmente a reutilização de componentes e a gestão recativa do estado não compensavam estes custos para uma aplicação com o nível de interatividade da QuizCraft. A substituição por JavaScript Vanilla permitiu manter a mesma estrutura de páginas prevista no design do Projeto I, sem overhead de compilação, com carregamento mais rápido e com uma arquitetura de módulos simples que se revelou suficientemente expressiva para todos os casos de uso.

Eliminação do PHP do *stack* tecnológico. O Projeto I previa uma arquitetura híbrida com Node.js para o servidor de aplicação e PHP para determinadas funcionalidades de *backend*. Esta combinação, comum em plataformas web académicas, foi descartada *logo* no primeiro *sprint* por razões de coerência e

simplicidade arquitetural. A utilização exclusiva de Node.js elimina a necessidade de manter dois ambientes de execução distintos, dois conjuntos de dependências e duas linguagens no mesmo projeto. Dado que o Node.js cobre nativamente todas as funcionalidades que o PHP teria desempenhado incluindo operações de base de dados, geração de emails, *hashing* de passwords e geração de *tokens*, a eliminação do PHP não resultou em nenhuma perda funcional, apenas numa simplificação arquitetural que beneficiou a manutenibilidade e a consistência do código.

É importante sublinhar que os dois desvios descritos anteriormente são exclusivamente tecnológicos e não afetaram em nenhum momento os objetivos pedagógicos do projeto, que permaneceram inalterados desde o Projeto I. A plataforma QuizCraft continua a ser, na sua versão final, uma ferramenta destinada a apoiar professores na criação de recursos educativos gamificados e a proporcionar aos alunos experiências de aprendizagem motivadoras, com *feedback* imediato e progressão visível. Todos os tipos de atividade educativa previstos no Projeto I foram implementados, o sistema de gamificação foi concretizado com um nível de detalhe superior ao inicialmente previsto, e os princípios de acessibilidade e inclusão foram verificados de forma rigorosa.

Esta distinção entre desvios tecnológicos e manutenção dos objetivos funcionais e pedagógicos é relevante porque demonstra que as alterações introduzidas durante o desenvolvimento foram consequência de uma maior compreensão técnica do problema, e não de uma redução de ambição ou de incapacidade de executar o plano original. Os desvios foram, em ambos os casos, decisões de maturidade técnica que produziram um resultado melhor do que o inicialmente previsto.

15 Resultados da Implementação

Este capítulo apresenta e analisa os resultados concretos obtidos na implementação da plataforma QuizCraft, organizando-os em indicadores quantitativos, análise qualitativa e discussão relativamente aos objetivos definidos no Projeto I. O objetivo é proporcionar uma visão objetiva e verificável do que foi construído, distinguindo claramente o que foi alcançado do que ficou como proposta de evolução futura.

A Tabela 3 apresenta os principais indicadores mensuráveis da plataforma implementada, organizados por dimensão do sistema. Estes valores são verificáveis diretamente no código-fonte e na estrutura da base de dados do projeto.

Tabela 3-Indicadores quantitativos da plataforma QuizCraft

Dimensão	Valor	Descrição
Páginas HTML	17	Páginas <i>frontend</i> estáticas servidas pelo Express
Rotas REST	9	Grupos de rotas: auth, quizzes, turmas, sessões, <i>dashboard</i> , progresso, badges, IA, admin
Tabelas na base de dados	12	Esquema relacional com integridade referencial
Tipos de atividade educativa	4	Quiz, <i>Flashcard</i> , Correspondência, Jogo educativo
Jogos educativos em Canvas	3	Quiz Runner, Nave Espacial, Memory Quiz
Badges disponíveis	14	5 de participação, 5 de pontuação, 4 de nível
Eventos Socket.IO	10	host:create/start/question/end, player:join/answer, session:start/question// <i>leaderboard</i> /end
Ficheiros de testes	4	auth, gamification, quiz, accessibility
Testes unitários implementados	65	100% PASS em Jest 29
Bugs identificados e corrigidos	11	Detetados em testes de integração manual
Perfis de utilizador	3	Administrador, Professor, Aluno
Categorias de badges	3	Participação, Pontuação, Nível

Dimensão	Valor	Descrição
Formatos de exportação	2	JSON (reimportável) e PDF (3 <i>layouts</i>)
Algoritmo de <i>hash</i>	bcrypt (12 rounds)	Configurável sem invalidar passwords existentes

15.1 Resultados Qualitativos

Para além dos indicadores mensuráveis, a análise qualitativa da plataforma permite avaliar dimensões que não são diretamente capturáveis por métricas numéricas, mas que são igualmente relevantes para a avaliação do produto desenvolvido.

Estabilidade da plataforma. Os testes de integração manual, conduzidos com múltiplos utilizadores em simultâneo durante o desenvolvimento dos *sprints* 5 a 9, não revelaram crashes do servidor, fugas de memória visíveis ou degradação progressiva de desempenho em sessões de duração normal (10 a 30 minutos). O mecanismo de restauro de sessão a partir da base de dados, implementado em resposta a um *bug* identificado nos testes, eliminou o único vetor de instabilidade crítico identificado durante o desenvolvimento.

Completude funcional. Todos os fluxos de utilização previstos nos casos de uso do Projeto I foram implementados e verificados. Um utilizador com o perfil de Professor consegue, sem assistência técnica, criar um quiz, adicionar perguntas de múltiplos tipos, gerar perguntas adicionais por IA, criar uma turma, inscrever alunos, lançar uma sessão em tempo real, acompanhar os resultados em direto e consultar a análise detalhada após o fim da sessão. Um utilizador com o perfil de Aluno consegue entrar com PIN, responder às perguntas, ver a sua posição no *leaderboard*, consultar o seu progresso acumulado e explorar quizzes públicos autonomamente.

Integração dos módulos. A plataforma demonstra uma integração coerente entre os seus módulos. O sistema de gamificação atualiza-se automaticamente após cada sessão tanto em tempo real como no modo assíncrono sem necessidade de Ação manual. As badges são avaliadas e desbloqueadas no momento correto, com notificação *toast* ao utilizador. A exportação em PDF reflete sempre o estado atual do quiz, incluindo perguntas geradas por IA. A barra lateral ajusta-se dinamicamente ao role do utilizador sem inconsistências entre páginas.

Cumprimento dos requisitos de acessibilidade. A paleta cromática foi validada programaticamente através de 16 testes Jest que calculam os rácios de contraste reais de cada combinação de cores. Todas as combinações críticas passam o Nível AA do WCAG 2.1. O sistema ColorADD está implementado em todas as perguntas de escolha múltipla. Os elementos HTML5 semânticos são utilizados de forma consistente em todas as páginas [20][6].

15.2 Discussão dos Resultados

A relação entre os resultados obtidos e os objetivos definidos no Projeto I é, em geral, de correspondência direta e, em alguns casos, de superação das expectativas iniciais. O sistema de gamificação implementado vai além do previsto no plano original: as 14 badges organizadas em três categorias, a fórmula de XP com bônus de velocidade e o *leaderboard* em tempo real representam uma implementação mais completa do que a descrita nos requisitos iniciais. Os três jogos educativos em Canvas HTML5 desenvolvidos sem dependências de *frameworks* externas excedem igualmente o nível de detalhe técnico que seria razoável esperar de um protótipo académico.

O único domínio em que os resultados ficam abaixo do inicialmente previsto é o da validação com utilizadores reais. A ausência de testes formais com professores e alunos do ensino básico e secundário significa que a adequação pedagógica da plataforma, embora fundamentada na revisão da literatura do Projeto I, não foi empiricamente verificada com o público-alvo. Esta limitação é reconhecida explicitamente e abordada na secção de validação com utilizadores.

Do ponto de vista técnico, os resultados demonstram que é possível construir uma plataforma educativa funcional, segura e acessível com um *stack* tecnológico relativamente simples Node.js, Express, MySQL, JavaScript Vanilla e Socket.IO desde que as decisões de arquitetura sejam tomadas com rigor e as responsabilidades entre camadas sejam claramente delimitadas. A ausência de *frameworks* de *frontend* complexas não se traduziu em limitações funcionais visíveis; pelo contrário, simplificou o *debugging*, reduziu o tempo de carregamento das páginas e eliminou uma categoria inteira de problemas de compatibilidade entre versões de dependências.

16 Limitações e Evoluções Futuras de Segurança

A implementação de segurança descrita nos capítulos anteriores cobre os vetores de ataque mais críticos para uma plataforma web educativa no contexto de um projeto acadêmico. No entanto, uma análise honesta da postura de segurança atual da QuizCraft deve reconhecer que existem mecanismos de proteção adicionais comuns em aplicações web em produção que não foram implementados nesta fase, quer por constrangimentos de tempo, quer por estarem fora do âmbito de um projeto de licenciatura. Esta secção identifica essas lacunas, explica as razões da sua ausência e descreve como poderiam ser integradas numa evolução futura da plataforma.

Rate limiting. A plataforma não implementa atualmente qualquer limite ao número de pedidos que um cliente pode efetuar num determinado intervalo de tempo. Esta ausência cria vulnerabilidade a ataques de força bruta sobre os endpoints de autenticação particularmente `/api/auth/login`, a enumeração de utilizadores e a sobrecarga do servidor por clientes mal configurados ou maliciosos. A solução mais direta seria a integração da biblioteca `express-rate-limit`, que permite definir janelas temporais e limites de pedidos por IP ou por utilizador autenticado, com resposta HTTP 429 (Too Many Requests) quando o limite é excedido. A configuração recomendada para o endpoint de `login` seria um máximo de 10 tentativas por IP em 15 minutos, com progressão exponencial do bloqueio em caso de violações repetidas.

Proteção CSRF. A autenticação baseada em JWT transmitida no cabeçalho `Authorization` protege naturalmente contra a maioria dos ataques de CSFR, dado que os browsers não incluem automaticamente cabeçalhos personalizados em pedidos cross-origin. No entanto, se a plataforma evoluísse para armazenar o JWT em cookies `HttpOnly` uma prática mais segura em relação a ataques XSS seria necessário implementar proteção CSRF explícita através de *tokens* sincronizados ou do padrão de verificação `SameSite`. A biblioteca `csrf` (atualmente descontinuada) ou padrões mais modernos como o `Double Submit Cookie` poderiam ser adotados nesse cenário.

CSP. A plataforma não define atualmente uma política de segurança de conteúdo (CSP) nos cabeçalhos HTTP das respostas. O CSP é uma camada de defesa em profundidade contra ataques XSS que permite ao servidor especificar quais as origens de scripts, folhas de estilo e outros recursos que o browser deve considerar confiáveis, bloqueando a execução de conteúdo injetado por um atacante mesmo que a sanitização de inputs falhe num caso específico. A implementação de um CSP adequado para a QuizCraft seria relativamente simples dado que todos os recursos estáticos são servidos pelo mesmo domínio mas requer um levantamento cuidadoso de todas as origens de recursos externos utilizados (fontes, ícones, CDNs de bibliotecas) para evitar quebras de funcionalidade.

Helmet.js. A biblioteca Helmet.js é um conjunto de *middlewares* Express que configura automaticamente um conjunto de cabeçalhos HTTP de segurança recomendados, incluindo X-Content-Type-Options, X-Frame-Options, Strict-Transport-Security e Referrer-Policy. A sua integração na plataforma reduziria a exposição a várias categorias de ataques web comuns sem exigir configuração manual extensa. A ausência do Helmet não representa uma vulnerabilidade crítica para uma plataforma académica em ambiente de desenvolvimento local, mas seria um requisito obrigatório numa implantação em produção acessível publicamente.

Auditoria de logs e monitorização de acessos. A plataforma não regista atualmente eventos de segurança relevantes — como tentativas de autenticação falhadas, acesso a rotas protegidas sem *token* válido, ou modificações a dados de outros utilizadores. A implementação de um sistema de *logging* estruturado, com biblioteca como Winston ou Pino para Node.js, permitiria criar um registo aditável de eventos de segurança que é essencial para a deteção de tentativas de intrusão e para a resposta a incidentes. Numa arquitetura de produção, estes *logs* seriam integrados com um sistema de monitorização centralizado como o ELK *Stack* ou uma solução gerida como o Datadog ou o Sentry.

A ausência destes mecanismos num projeto académico é compreensível e, em certa medida, esperada: a plataforma QuizCraft foi desenvolvida e testada em ambientes de desenvolvimento local, sem exposição à internet pública. No entanto, a sua identificação explícita neste relatório serve dois propósitos: demonstrar consciência da diferença entre segurança académica e segurança de produção, e fornecer um roteiro concreto para a evolução da plataforma em direção a uma implantação segura num ambiente escolar real.

17 Validação com Utilizadores

Uma das questões que qualquer avaliador experiente colocará ao analisar uma plataforma educativa é: foi testada com o público-alvo? A resposta honesta, no caso da QuizCraft, é que não foram realizados testes formais com professores e alunos do ensino básico e secundário durante o período de desenvolvimento do Projeto II. Este capítulo explica as razões desta ausência, discute as suas implicações para a avaliação do trabalho e apresenta o protocolo de validação que seria adequado numa fase seguinte do projeto.

É importante distinguir entre dois tipos de validação que, por vezes, são confundidos: a validação técnica e a validação com utilizadores. A validação técnica cobrindo a correção funcional, a segurança, a acessibilidade e o desempenho foi realizada de forma extensiva através dos 65 testes unitários Jest, dos testes de integração manual e da verificação de conformidade com o WCAG 2.1. A validação com utilizadores, que avalia a usabilidade, a adequação pedagógica e a aceitação por parte dos utilizadores finais, é uma dimensão diferente que requer *metodologias* próprias e que, por constrangimentos temporais, não foi possível realizar no âmbito do presente projeto.

17.1 Constrangimentos Temporais e Âmbito do Projeto

O desenvolvimento do Projeto II decorreu ao longo de um semestre académico, com a totalidade do tempo disponível dedicada ao design, implementação, teste e documentação da plataforma. A realização de um estudo de validação com utilizadores reais que implicaria recrutamento de participantes, aprovação ética, condução de sessões, análise de dados e escrita de um relatório de usabilidade representa um esforço de investigação com uma dimensão equivalente a um trabalho de projeto separado, e estava claramente fora do âmbito temporal e de recursos do presente semestre.

Esta situação não é incomum no desenvolvimento de protótipos académicos. A literatura sobre design de sistemas educativos distingue habitualmente entre a fase de desenvolvimento (onde o foco é a implementação técnica e a verificação de requisitos) e a fase de avaliação (onde o produto é testado com utilizadores reais em contexto de uso real). O presente relatório corresponde à primeira fase; a segunda constituiria trabalho futuro de investigação aplicada.

17.2 Protocolo de Validação Proposto

Uma validação adequada da plataforma QuizCraft com utilizadores reais envolveria, idealmente, dois grupos de participantes distintos professores e alunos, com *metodologias* adaptadas às suas necessidades específicas. A proposta que se segue descreve um protocolo de validação que poderia ser conduzido numa colaboração com uma ou mais escolas do ensino básico ou secundário.

SUS. O SUS é um questionário estandardizado de dez itens, desenvolvido por John Brooke em 1986 e amplamente utilizado na avaliação de usabilidade de sistemas de software. Cada item é avaliado numa escala de Likert de cinco pontos, produzindo uma pontuação global entre 0 e 100 que permite classificar a usabilidade do sistema e comparar os resultados com *benchmarks* estabelecidos na literatura. A utilização do SUS com professores e alunos, imediatamente após uma sessão de utilização da plataforma, forneceria uma medida quantitativa da usabilidade percebida que complementaria os dados qualitativos das entrevistas. Scores acima de 68 são considerados acima da média; scores acima de 80 indicam boa usabilidade. Como proposta de validação futura, seria expectável que a QuizCraft atingisse um score SUS superior a 70, dado o investimento deliberado na simplicidade da interface e na redução de barreiras à criação de conteúdo.

Think-aloud protocol. O protocolo de verbalização simultânea (*think-aloud*), proposto por Nielsen (1993), consiste em pedir aos participantes que verbalizem os seus pensamentos enquanto realizam um conjunto de tarefas predefinidas com o sistema. Esta metodologia é particularmente eficaz para identificar problemas de usabilidade que não seriam capturáveis por questionários ou métricas de desempenho, como confusão terminológica, expectativas incorretas sobre o comportamento do sistema ou dificuldades em localizar funcionalidades. Para a QuizCraft, as tarefas a testar com professores incluiriam a criação de um *quiz* com múltiplos tipos de pergunta, o lançamento de uma sessão em tempo real e a interpretação dos resultados. Com alunos, as tarefas incluiriam a entrada numa sessão com PIN, a realização de um jogo educativo e a consulta do progresso pessoal.

Entrevistas semiestruturadas. As entrevistas semiestruturadas permitem explorar, de forma aprofundada e flexível, a experiência dos utilizadores com a plataforma, incluindo aspetos que os participantes considerem relevantes, mas que não estejam cobertos pelos questionários. Com professores, as entrevistas focariam a adequação da plataforma ao contexto pedagógico real, a comparação com ferramentas já utilizadas (como o Kahoot! ou o Quizizz), as limitações identificadas e as funcionalidades consideradas mais valiosas. Com alunos, o foco seria na motivação produzida pelas mecânicas de gamificação, na clareza das interfaces e na experiência de aprendizagem proporcionada pelos diferentes tipos de atividade.

Os resultados de um tal estudo de validação permitiriam não apenas avaliar a plataforma atual, mas também orientar prioridades de desenvolvimento futuro com base em evidência empírica recolhida junto dos utilizadores reais. Esta é, aliás, a abordagem recomendada pela literatura sobre design iterativo de sistemas educativos: desenvolver, avaliar, refinar repetidamente em ciclos progressivos que aproximem o produto das necessidades reais dos seus utilizadores.

18 Contributos do Projeto

A plataforma QuizCraft é, no seu contexto imediato, um projeto de licenciatura. Mas os seus contributos estendem-se para além do exercício académico que lhe deu origem, e a sua análise sob diferentes perspetivas permite compreender o seu valor real e potencial.

Do ponto de vista técnico, a QuizCraft demonstra que é possível construir uma plataforma web funcional, segura e acessível com um *stack* tecnológico *open-source* maduro e bem documentado, sem dependência de *frameworks* proprietárias ou de serviços pagos. A utilização exclusiva de Node.js, Express, MySQL, Socket.IO e JavaScript *Vanilla* complementada pela API da OpenAI para geração de conteúdo produz uma arquitetura que é simultaneamente simples de compreender, simples de manter e suficientemente expressiva para suportar um conjunto de funcionalidades que inclui comunicação em tempo real, jogos interativos e IA generativa.

A implementação dos três jogos educativos em Canvas HTML5 sem dependências de motores de jogo externos é, em particular, uma demonstração técnica com valor próprio. A criação de um jogo de plataforma, um *shooter* espacial e um jogo de memória a partir de primitivas da API Canvas com animações, gestão de estados, sistemas de vidas e integração com perguntas educativas exigiu um nível de compreensão da programação gráfica e do modelo de eventos do browser que vai consideravelmente além do que é habitual num projeto de licenciatura.

Do ponto de vista pedagógico, a QuizCraft representa uma resposta concreta a um problema identificado na revisão da literatura do Projeto I: a ausência de ferramentas que combinem a simplicidade de criação de conteúdo do Kahoot! com a flexibilidade e a variedade de tipos de atividade do EduXgame, sem as limitações de acesso e de personalização dessas plataformas comerciais.

A diversidade de tipos de atividade *quiz* clássico, *flashcards*, correspondência e jogos *arcade* responde a diferentes estilos de aprendizagem e mantém a variedade necessária para sustentar o interesse dos alunos ao longo do tempo. O *feedback* imediato, implementado após cada resposta em todos os tipos de atividade, está fundamentado na investigação de Hattie e Timperley (2007) sobre os fatores com maior impacto na aprendizagem. O modo assíncrono suporta a aprendizagem autónoma fora do contexto de sala de aula, respondendo a necessidades identificadas nos estudos revistos no Projeto I.

A integração de IA generativa para criação de perguntas é um contributo pedagógico com implicações práticas imediatas: reduz significativamente o tempo que os professores precisam de dedicar à criação de recursos educativos digitais, tornando a adoção da gamificação acessível a docentes sem experiência prévia com ferramentas digitais avançadas. Embora a qualidade das perguntas geradas exija revisão humana, o processo de geração-revisão é substancialmente mais rápido do que a criação de raiz.

Do ponto de vista acadêmico, o projeto contribui para a área de *Tecnologias Educativas* com uma implementação concreta de um conjunto de princípios que permanecem, com demasiada frequência, ao nível da teoria ou do protótipo não funcional. A combinação de uma revisão sistemática da literatura seguindo a metodologia PRISMA (Projeto I) com a implementação e validação técnica de uma plataforma funcional (Projeto II) representa uma progressão metodológica completa, da investigação à aplicação, que é rara no contexto de projetos de licenciatura.

A documentação detalhada das decisões técnicas incluindo os desvios ao plano original e a sua justificação, a estratégia de testes com 65 casos verificáveis e a análise de acessibilidade com rácios de contraste calculados programaticamente constituem uma base de referência para trabalhos futuros na mesma área. O código-fonte do projeto, disponível no repositório GitHub, é um recurso académico em si mesmo: um exemplo comentado de arquitetura *full-stack* com comunicação em tempo real, autenticação segura e integração de IA, desenvolvido num contexto educativo e com fins educativos.

19 Referências

- [1] Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., & Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ*, 372, n71. <https://doi.org/10.1136/bmj.n71>
- [2] Wang, A. I., & Lieberoth, A. (2016). The effect of points and audio on concentration, engagement, enjoyment, learning, motivation, and classroom dynamics using Kahoot! In *Proceedings of the 10th European Conference on Games Based Learning* (pp. 738–745). Academic Conferences International.
- [3] W3C. (2018). *Web Content Accessibility Guidelines (WCAG) 2.1*. World Wide Web Consortium. <https://www.w3.org/TR/WCAG21/>
- [4] Royce, W. W. (1970). Managing the development of large software systems. *Proceedings of IEEE WESCON*, 26(8), 1–9.
- [5] Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide*. Scrum.org. <https://scrumguides.org/>
- [6] Simões, M. (2010). *ColorADD: Sistema de Identificação de Cores para Daltônicos* [Dissertação de doutoramento]. Universidade do Minho.
- [7] Provos, N., & Mazières, D. (1999). A future-adaptable password scheme. *Proceedings of the USENIX Annual Technical Conference*, 32–43.
- [8] Jones, M., Bradley, J., & Sakimura, N. (2015). *JSON Web Token (JWT)*. RFC 7519. IETF. <https://tools.ietf.org/html/rfc7519>
- [9] OWASP. (2021). *SQL Injection*. OWASP Top Ten. https://owasp.org/www-community/attacks/SQL_Injection
- [10] OWASP. (2021). *Cross Site Scripting (XSS)*. OWASP. <https://owasp.org/www-community/attacks/xss/>
- [11] Nielsen, J. (1993). *Usability Engineering*. Academic Press.
- [12] Brooke, J. (1996). SUS: A 'quick and dirty' usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & A. L. McClelland (Eds.), *Usability Evaluation in Industry* (pp. 189–194). Taylor & Francis.
- [13] Prensky, M. (2001). *Digital Game-based learning*. McGraw-Hill.
- [14] Werbach, K., & Hunter, D. (2012). *For the Win: How Game Thinking Can Revolutionize Your Business*. Wharton Digital Press.
- [15] Deci, E. L., & Ryan, R. M. (1985). *Intrinsic Motivation and Self-Determination in Human Behavior*. Plenum Press.
- [16] Hattie, J., & Timperley, H. (2007). The power of *feedback*. *Review of Educational Research*, 77(1), 81–112. <https://doi.org/10.3102/003465430298487>
- [17] IMS Global Learning Consortium. (2019). *IMS LTI Specification*. <https://www.imsglobal.org/activity/learning-tools-interoperability>

- [18] Bartle, R. (1996). Hearts, clubs, diamonds, spades: Players who suit MUDs. *Journal of MUD Research*, 1(1). <http://www.mud.co.uk/richard/hcds.htm>
- [19] Marczewski, A. (2015). *Even Ninja Monkeys Like to Play: Gamification, Game Thinking and Motivational Design*. Blurb.
- [20] Rose, D. H., & Meyer, A. (2002). *Teaching Every Student in the Digital Age: Universal Design for Learning*. ASCD.

20 Anexo A

Tabela 4-Sprints de trabalho

Sprint	Objectivo	Funcionalidades produzidas
S1	Fundações da plataforma	Autenticação JWT + bcrypt, registo com roles, <i>dashboards</i> base, sidebar dinâmica
S2	Editor de quizzes	Editor de escolha múltipla, V/F e resposta aberta, validação de perguntas, gestão de quizzes
S3	Actividades educativas	Flashcards frente/verso, exercícios de correspondência, exportação JSON/PDF
S4	Turmas e sessões	CRUD de turmas, código de convite, gestão de alunos por turma
S5	Sessões em tempo real	Socket.IO, PIN de sessão, <i>leaderboard</i> em direto, modo em tempo real
S6	Modo assíncrono e jogos	Sessões assíncronas, <i>Quiz Runner</i> em Canvas HTML5
S7	Gamificação completa	Sistema XP e níveis, 14 badges com notificações <i>toast</i> , Nave Espacial e <i>Memory Quiz</i>
S8	IA	Geração de perguntas via OpenAI API, integração com mascote Quizzzy, baralhar opções
S9	<i>Polish</i> e testes	<i>Dashboard</i> de resultados, recuperação de password, tutorial de boas-vindas, 65 testes Jest

