

ESCOLA SECUNDÁRIA DE CAMÕES

CURSO PROFISSIONAL
TÉCNICO DE GESTÃO E PROGRAMAÇÃO DE SISTEMAS
INFORMÁTICOS



PROVA DE APTIDÃO PROFISSIONAL (RELATÓRIO)

DETECTIVE MAOLI

Unique Thapa

NÚMERO:14 TURMA: 12N

Yili Huang

NÚMERO:18 TURMA: 12N

Professor(es) orientador(es):

Alexandre Barão
Nuno Padeiro

julho, 2025.

Esta página foi deixada em branco intencionalmente

Resumo

Este trabalho apresenta o **Detective MaoLi**, um jogo de investigação eletrónico, apoiado por uma página *web*, realizado no âmbito do curso de **Técnico de Gestão e Programação de Sistemas Informáticos** da **Escola Secundária de Camões**.

Motivados pela escassez de títulos no género **detetive** e pelo entusiasmo pessoal por narrativas de mistério, os autores propuseram-se criar uma experiência imersiva que combina **vídeo, diálogo interativo e ramificações narrativas**.

Neste relatório, define-se o contexto e a motivação do projeto, enuncia-se o problema, a oferta limitada de videojogos detectivescos e estabelecem-se seis objetivos principais, desde a investigação teórica até à realização de protótipos funcionais.

De igual modo, procede-se ao enquadramento teórico, abordando conceitos de videojogos, metodologias de análise (casos de uso, modelo de domínio, requisitos funcionais e não funcionais) e tecnologias adotadas, como *Unity*, HTML5, CSS3, *JavaScript* e *Microsoft Copilot*.

São apresentadas as principais interações dos atores do sistema (casos de uso) e o modelo de domínio que identifica as entidades *Website* e *Game*, seguido pela arquitetura técnica do jogo (baseada no motor *Unity*) e da página de suporte (estrutura cliente-servidor com *MySQL*).

Definem-se requisitos funcionais e não funcionais para ambos os componentes (jogo e página de suporte), prevendo aspetos de desempenho, compatibilidade e usabilidade, entre outros.

O protótipo é apresentado de modo a evidenciar diversos aspetos. Em concreto, o jogo implementa menus, sistema de vidas, vídeos de introdução e ramificações narrativas em nós de diálogo, enquanto o *website* disponibiliza informação sobre enredo, mecânicas de jogo, vídeo demonstrativo e opção de *download*.

Palavras-chave: jogo eletrónico, detetive maoli, *unity*, tgpsi

Esta página foi deixada em branco intencionalmente

Notações

2D	Bidimensional
3D	<i>Tridimensional</i>
AES	<i>Advanced Encryption Standart</i>
C#	<i>C Sharp</i>
CSS	<i>Cascade Style Sheet</i>
HTML	<i>Hypertext Markup Language</i>
JS	<i>JavaScript</i>
MySQL	<i>My Structured Query Language</i>
RPG	<i>Role-Playing-Game</i>
TGPSI	Técnico de Gestão e Programação de Sistemas Informáticos
TPP	<i>First Person Perspective</i>
UML	<i>Unified Modelling Language</i>

Esta página foi deixada em branco intencionalmente

Agradecimentos

Gostaríamos de expressar a nossa profunda gratidão à **Escola Secundária de Camões**, cuja formação ao longo destes três anos nos proporcionou competências essenciais que têm sido, e continuarão a ser, fundamentais para o nosso percurso académico e profissional.

Ao nosso orientador, professor e mentor, o Professor Doutor **Alexandre Barão**, agradecemos do fundo do nosso coração, foi ele quem, em tempo livre e de forma totalmente voluntária, nos guiou, reorientou e apoiou a cada passo, garantindo que este relatório se tornasse realidade. Estendemos também o nosso reconhecimento ao Professor **Nuno Padeiro**, cujo vosso apoio constante ao longo do curso contribuiu para o nosso desenvolvimento técnico e pessoal.

Um agradecimento especial ao Professor Doutor **Rui Miguel Pascoal**, pelas horas de aconselhamento por ocasião do estágio de um dos autores na Universidade Europeia em FCT. A sua disponibilidade e *expertise* foram importantes para resolvermos determinados desafios do desenvolvimento do jogo.

Tendo trabalhado em grupo, ainda assim, gostaríamos de agradecer de seguida a título individual.

Yili Huang

Manifesto o meu agradecimento à minha família e aos meus amigos, que dedicaram tempo precioso para participar nos testes do jogo *Detective MaoLi*. Em particular, à minha mãe, cujo apoio incondicional e confiança foram a base que me permitiu levar este projeto até ao fim.

Unique Thapa

Gostaria de expressar a minha mais profunda gratidão à minha família e amigos que me apoiaram quando mais precisei durante o desenvolvimento deste projeto. Um agradecimento especial à minha mãe e ao meu pai, cujo apoio e incentivo inabaláveis estiveram comigo em cada passo do caminho. Estou também extremamente feliz com o meu amigo, Yili Huang, cuja ajuda e dedicação desempenharam um papel crucial na concretização deste trabalho.

A todos, o nosso sincero obrigado.

Esta página foi deixada em branco intencionalmente

Índice

1.	Introdução	1
1.1.	Contexto e Motivação	1
1.2.	Problema.....	1
1.3.	Objetivos	1
1.4.	Cronograma	2
1.5.	Resultados alcançados	3
1.6.	Estrutura do documento.....	3
2.	Enquadramento teórico	5
2.1.	Videojogos	5
2.2.	Metodologias.....	11
2.2.1.	Casos de Uso.....	11
2.2.2.	Modelo de Domínio	13
2.2.3.	Requisitos funcionais / não funcionais	14
2.3.	Tecnologias utilizadas.....	15
2.3.1.	<i>Unity</i>	15
2.3.2.	HTML5.....	16
2.3.3.	CSS3.....	16
2.3.4.	<i>JavaScript</i>	17
2.3.5.	<i>Microsoft Copilot</i>	18
3.	Projeto Detective MaoLi	21
3.1.	Casos de Uso do Projeto Detective MaoLi	21
3.2.	Modelo de Domínio do Projeto Detective MaoLi	22
3.3.	Arquitetura do Sistema	22
3.3.1.	Arquitetura do Jogo	22
3.3.2.	Arquitetura da página de suporte	23
3.3.	Requisitos	24
3.4.1.	Requisitos funcionais do Jogo.....	24
3.4.2.	Requisitos funcionais da página de suporte	25
3.4.3.	Requisitos não funcionais do Jogo.....	25
3.4.4.	Requisitos não funcionais da página de suporte	26
3.5.	Protótipo.....	27
3.5.1.	Jogo (Unity).....	27
3.5.1.1.	Características gerais	27
3.5.1.2.	Sistemas de menu e cenários de jogo	28
3.5.1.3.	Exemplos de código-fonte	32

3.5.2.	Página de suporte aos jogadores (HTML/CSS)	34
3.5.2.1.	Características gerais	35
3.5.2.2.	Esquema navegacional.....	35
3.5.2.3.	Exemplos de código-fonte	38
3.6.	Testes de Produto.....	39
4.	Conclusões	41
4.1.	Discussão	41
4.2.	Limitações.....	42
4.3.	Trabalho futuro.....	42
	Referências	43

Índice de Figuras

Figura 1 Cronograma do Projeto	2
Figura 2 Caso de uso do Sistema de Reservas Online	12
Figura 3 Modelo de Domínio UML do sistema “Escola”	13
Figura 4 Casos de uso UML (Detective MaoLi).....	21
Figura 5 Modelo de Domínio UML (Detective MaoLi).....	22
Figura 6 Arquitetura do jogo desenvolvido com Unity	23
Figura 7 Arquitetura da página de suporte	23
Figura 8 Interface inicial do jogo	28
Figura 9 Painel de configurações	28
Figura 10 Cenário SeasonDirectory	29
Figura 11 Cenário EpisodeDirectory	29
Figura 12 Vídeo de introdução no cenário S1Ep1	30
Figura 13 Interface para início de diálogo	31
Figura 14 Interface de fim do diálogo (S1Ep1StodyNode1)	31
Figura 15 Vídeo de transição para o nó seguinte.....	32
Figura 16 Novo nó, nova imagem de fundo e novo painel	32
Figura 17 Código-fonte (após início de episódio).....	33
Figura 18 Código-fonte (criar Slots no Hierarchy do Unity)	33
Figura 19 Código-fonte (características do avatar)	34
Figura 20 Código-fonte (visibilidade da UI)	34
Figura 21 Esquema navegacional da página de suporte	35
Figura 22 Main Page da página de suporte	36
Figura 23 Home Page da página de suporte.....	36
Figura 24 Story da página de suporte	37
Figura 25 Gameplay da página de suporte	37
Figura 26 Contact da página de suporte	38
Figura 27 Código-fonte (Navbar).....	38
Figura 28 Código-fonte (Video).....	39
Figura 29 Código-fonte (Styling)	39

Prova de Aptidão Profissional

Esta página foi deixada em branco intencionalmente

x

Índice de Tabelas

Tabela 1 Géneros de videojogos e respetivas características	10
Tabela 2 Principais componentes dos diagramas de casos de uso	12
Tabela 3 Principais diferenças entre os Requisitos Funcionais e Não Funcionais...	15
Tabela 4 Requisitos Funcionais do jogo.....	25
Tabela 5 Requisitos Funcionais da página de suporte.....	25
Tabela 6 Requisitos Não Funcionais do Jogo	26
Tabela 7 Requisitos Não Funcionais da página de suporte	27

Esta página foi deixada em branco intencionalmente

1. Introdução

Neste capítulo, apresentam-se os principais elementos introdutórios que servem de base para a compreensão do trabalho realizado. Na Secção 1.1, são descritos o **contexto e a motivação** do projeto, destacando a relevância e os fatores que impulsionaram o seu desenvolvimento. Na Secção 1.2, é apresentado o **problema** que se pretende abordar, incluindo os desafios e as necessidades identificadas.

De seguida, na Secção 1.3, apresenta os **objetivos** do trabalho, definindo de forma clara as metas a atingir. Na Secção 1.4, apresenta o **cronograma**, detalhando a organização temporal das atividades realizadas ao longo do projeto.

A Secção 1.5 são descritos os **resultados alcançados**, fornecendo uma visão geral sobre os progressos realizados. Finalmente, na Secção 1.6, descreve-se a estrutura do documento, oferecendo ao leitor uma visão geral da organização e conteúdo dos capítulos subsequentes.

1.1. Contexto e Motivação

O projeto atual centra-se no desenvolvimento de um **jogo de detetive**. Para usufruir melhor da experiência do jogo, é necessário um **website de suporte**, para permitir aos visitantes (jogadores), compreender melhor o enredo e até encontrar pistas em falta. Os autores deste projeto são grandes fãs de filmes de mistério, e por isso foram motivados a iniciar este projeto de informática na área referida.

1.2. Problema

Apesar do crescimento e reconhecimento gradual da indústria de jogos do tipo *detective*, na nossa perspetiva, verifica-se uma escassez dos mesmos.

Devido a este problema, este projeto visa contribuir com mais um título para o ecossistema de jogos do tipo *detective*.

1.3. Objetivos

Considerando o contexto e motivação, perante o problema identificado, definiu-se um conjunto de objetivos principais:

- **O1** – Analisar o enquadramento teórico do projeto;
- **O2** - Definir os casos de uso do projeto, i.e., identificação dos principais atores, processos e fronteiras do sistema;
- **O3** - Elaborar o modelo de domínio que traduza as relações estáticas das principais classes do projeto;

- **O4** - Definir a arquitetura do sistema;
- **O5** - Definir os requisitos das aplicações;
- **O6** - Realizar o protótipo aplicacional (jogo e página web).

1.4. Cronograma

Nesta secção ilustra-se o cronograma de tarefas executadas relativas ao projeto (Diagrama de Gantt, Figura 1). Assim, observa-se no cronograma:

- **Tarefa 1**, *Definição de Contexto/ Motivação*;
- **Tarefa 2**, *Definição de Problema/Objetivos*;
- **Tarefa 3**, *Estudo sobre o Enquadramento Teórico*;
- **Tarefa 4**, *Definição de Casos de Uso/Modelo de Domínio*;
- **Tarefa 5**, *Arquitetura do Sistema*;
- **Tarefa 6**, *Definição de Requisitos*;
- **Tarefa 7**, *Desenvolvimento do Protótipo*;
- **Tarefa 8**, *Testes do Produto*; e,
- **Tarefa 9**, *Elaboração do Relatório*.

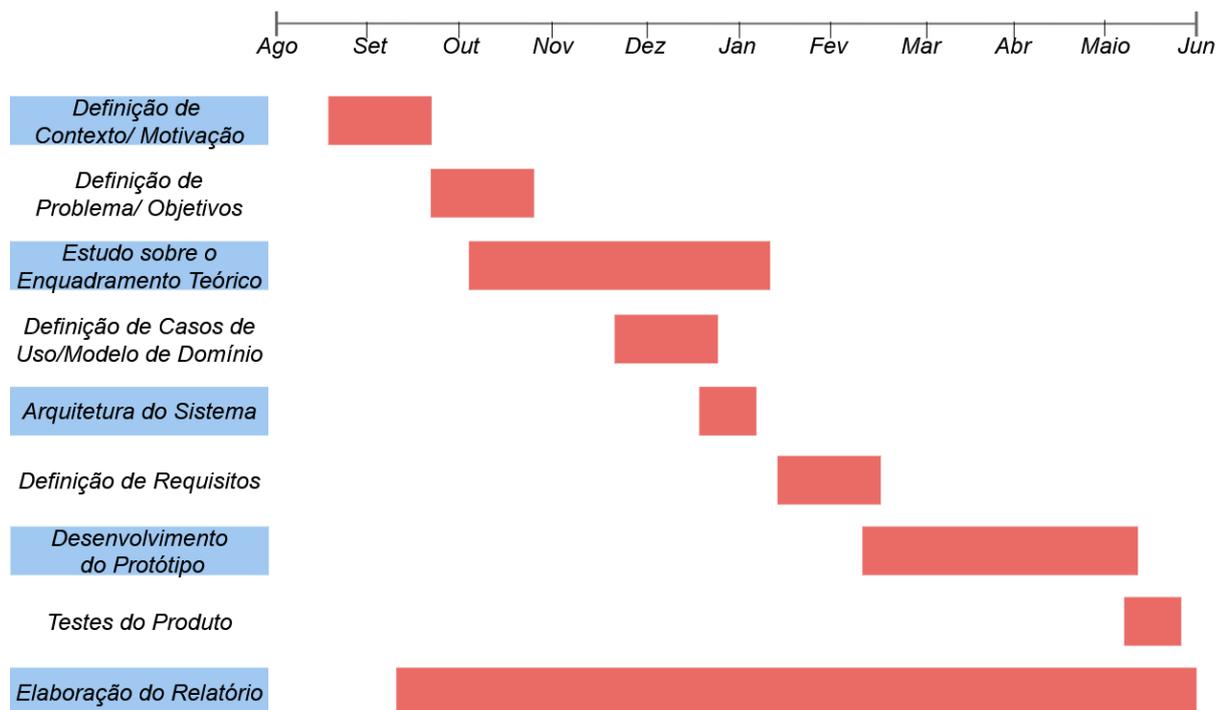


Figura 1 Cronograma do Projeto

1.5. Resultados alcançados

Os principais resultados alcançados consistem na elaboração formal do presente relatório, desenvolvimento do jogo Detective Maoli, e, finalmente, desenvolvimento da página *web* para suporte aos jogadores.

1.6. Estrutura do documento

Este documento encontra-se organizado em capítulos. Assim, o capítulo 1 refere-se à **introdução**, incluindo as seguintes secções: 1.1 **Contexto e Motivação**, 1.2 **Problema**, 1.3 **Objetivos**, 1.4 **Cronograma**, 1.5 **Resultados Alcançados** e 1.6 **Estrutura do Documento**.

O Capítulo 2 aborda o **enquadramento teórico**, dividido nas secções: 2.1 **Modelo de Negócio**, 2.2 **Metodologias** e 2.3 **Tecnologias Utilizadas**.

O Capítulo 3 detalha o **Projeto Detective MaoLi** e inclui as secções: 3.1 **Casos de Uso do Projeto Detective MaoLi**, 3.2 **Modelo de Domínio do Projeto Detective MaoLi**, 3.3 **Arquitetura do Sistema**, 3.4 **Requisitos**, 3.5 **Protótipo** e 3.6 **Testes do Produto**.

Finalmente, o Capítulo 4 refere-se às **conclusões** preliminares do projeto, que contem as secções: 4.1 **Discussão**, 4.2 **Limitações** e 4.3 **Trabalho Futuro**.

Prova de Aptidão Profissional

Esta página foi deixada em branco intencionalmente

2. Enquadramento teórico

Neste capítulo apresenta-se o enquadramento teórico relativo a **videojogos** e **metodologias** de análise utilizadas no processo de desenvolvimento deste projeto.

2.1. Videojogos

O modelo de negócio associado ao mercado de videojogos gera receitas muito significativas a nível mundial.

Os videojogos são pacotes de *software* interativos com uma interface gráfica, que permite ao jogador controlar personagens e/ou elementos dentro de um ambiente virtual. Este ambiente digital promove uma experiência dinâmica, imersiva e orientada por objetivos onde o jogador atua como a entidade ativa e interage através de uma interface visual e periféricos. Para alcançar esse propósito, os videojogos apresentam várias características, que incluem [CG, 24]:

- **Interatividade** - O jogador executa ações e comandos que influenciam a narrativa e a dinâmica do jogo possibilitando uma experiência ativa e personalizada;
- **Gráficos** - Os videojogos utilizam componentes visuais que variam em qualidade e estilo, desde gráficos bidimensionais (2D) até ambientes tridimensionais (3D). Além disso, existem gráficos que combinam os elementos de ambos os estilos (2.5D);
- **Sons** - A música e os efeitos sonoros complementam a experiência do videojogo, contribuindo para a criação de atmosferas e reforçando emoções e reações às ações do jogador;
- **Objetivos** - Cada jogo estabelece metas e/ou desafios que orientam o jogador, acrescentam uma estrutura motivacional e um sentido de progresso;
- **Regras** - As limitações e permissões de ações são definidas por um conjunto de regras, que estruturam a experiência e garantem a coerência do universo do jogo. De resto, existe uma área científica que se denomina *Teoria dos Jogos* [TM+,24].

2.1.1. Plataformas para Videojogos

As plataformas de videojogos são os dispositivos onde os jogos são executados e determinam a qualidade da experiência e a forma de interação. As três principais plataformas para videojogos incluem computadores, consolas e dispositivos móveis, cada uma com características únicas que influenciam o desempenho, os gráficos e o tipo de controlo [JC, 24]:

- **Computadores (PCs)** - oferecem uma plataforma versátil e altamente configurável, permitindo jogos com gráficos de alta resolução e uma grande diversidade de controlos, como teclados, ratos e comandos. Os computadores permitem ampla personalização de *hardware*, o que pode melhorar significativamente o desempenho gráfico e a capacidade de processamento, sendo ideais para jogos mais pesados computacionalmente, que exigem alta qualidade visual e maior capacidade de recursos (e.g. memória RAM);
- **Consola de Videojogos** - *PlayStation 5, Xbox Series X, Nintendo Switch2 e Steam Deck* são exemplos de consolas usadas especificamente para videojogos, desenhados para oferecer uma experiência otimizada e consistente. Com *hardware* padronizado e comandos dedicados, permite ao jogador uma utilização simplificada, sem necessidade de configurações complexas;
- **Dispositivos Móveis** - *Smartphones e tablets*, são dispositivos móveis que possibilitam uma experiência de jogo portátil, com interações baseadas em ecrãs táteis e por vezes sensores de movimentos. Embora sejam menos poderosos que computadores e consolas, tipicamente, nos dispositivos móveis as sessões são mais curtas, idealmente para jogos casuais e rápidos. Estas plataformas são apreciadas pela conveniência e portabilidade, permitindo a experiência de jogo *anytime / anywhere*.

2.1.2. Videojogos *online* vs. *offline*

Os modos de jogo *online* e *offline*, oferecem experiências distintas e são adequados a diferentes perfis e preferências dos jogadores.

No modo *offline*, o jogador não depende de uma ligação à *Internet* para desfrutar do jogo. Normalmente este modo de jogo é mais focado na experiência individual e na exploração de elementos internos do mesmo, com a narrativa e desafios estabelecidos. Este modo de jogo é ideal para quem prefere um ritmo próprio, concentrando-se na progressão e na imersão sem depender de fatores externos (e.g. acesso à *Internet*).

Por outro lado, no modo de jogo *online*, é necessária ligação à *Internet* e pode proporcionar uma experiência de jogo coletiva e interativa, permitindo que os jogadores interajam entre si em tempo real. Este modo de jogo é normalmente caracterizado pela possibilidade de colaboração ou competição com outros jogadores, que frequentemente podem ser de outras regiões. Além disso os videojogos *online* são habitualmente atualizados com novos conteúdos e desafios, oferecendo aos jogadores uma experiência diversificada e em expansão. Todavia, a dependência de uma ligação à *Internet* pode implicar o risco de problemas de conectividade, segurança, latência e interrupções que podem afetar a fluidez da experiência [JC, 24].

2.1.3. Videojogos 2D, 3D e 2.5D

As dimensões gráficas dos videojogos, que incluem 2D, 3D e 2.5D, influenciam significativamente como o jogador irá interagir com o ambiente do jogo.

Videojogos de duas dimensões (2D) operam no eixo horizontal e vertical, proporcionando uma jogabilidade linear e normalmente simplificada. Esta estética gráfica permite um foco claro na mecânica de jogo e na narrativa, sendo mais comum em videojogos de género plataforma e de *puzzle*, onde a clareza visual e a acessibilidade são cruciais.

Por outro lado, os videojogos de três dimensões (3D) oferecem uma representação mais realista e imersiva, por operarem em três eixos – horizontal, vertical e de profundidade. Esta dimensão adicional permite a criação de ambientes complexos e dinâmicos. Videojogos com gráficos 3D são amplamente utilizados em géneros como ação, aventura e simulação, que acentua a sensação de presença e imersão no mundo virtual.

Contudo, a criação de videojogos 3D requer um *hardware* mais avançado, o que poderá limitar ou restringir o acesso de certos jogadores. A complexidade visual inerente a estes videojogos também implicará frequentemente um maior tempo de desenvolvimento, que resulta em experiências de utilização mais detalhadas e aprofundadas.

Como referido anteriormente (Secção 2.1.1), o conceito dos gráficos 2.5D, surge como uma intersecção entre o 2D e o 3D, combinando os elementos de ambas. Os videojogos 2.5D utilizam gráficos tridimensionais, mas limitando a jogabilidade a um plano bidimensional, assim proporcionando uma estética visual que preserva a simplicidade do 2D enquanto oferece a profundidade e os detalhes do 3D. Normalmente esta abordagem é frequentemente utilizada em jogos de género plataforma. Este tipo de conceito gráfico permite que os jogadores desfrutem de um ambiente visualmente atrativo, sem perder a clareza das mecânicas dos videojogos 2D [JC, 24].

2.1.4. Tipos de videojogos

A tabela seguinte resume diversos géneros e tipos de videojogos [CG, 4]. O género de vídeo jogo deste projeto é **Puzzle**.

Prova de Aptidão Profissional

Género	SubGénero	Descrição	Exemplo
Ação	Ação/Aventura	Exploração e combate	Uncharted Tomb Raider
	Arena shooter	Combate em arenas fechadas com ritmo rápido	Quake Unreal Tournament
	Beat' em up	Progresso em fase com combate contra múltiplos inimigos	Street of Rage Double Dragon
	Hack and Slash	Combate corpo-a-corpo com combos	Devil My Cry Bayonetta
	Hero Shooter	Personagens com capacidades únicas	Paladins Overwatch
Ação	FPS	Tiro na primeira pessoa	Halo Valorant
Aventura	Point and Click	Resolução de Puzzles e Narrativas	Monkey Island Myst
	Text Adventure	Jogo narrativo usando texto como principal interação	Zork
	Visual Novel	Baseado em textos e ilustrações, com escolhas que afetam a narrativa	Doki Doki Literature Club
	Walking Simulator	Exploração e narrativa sem grandes desafios	FireWatch Gone home
Battle Royale	N/A	<i>Multiplayer</i> onde o último jogador ou equipa vence	Fortnite PUBG
Corrida	Corrida Arcade	Corrida com física simplificada	Need for Speed Mario Kart
	Simulação de corrida	Corrida realista e com física precisa	Forza Motorsport iRacing
Desporto	Desporto Arcade	Versão menos realista e mais acessível	NBAJam Rocket League
	Simulação de Desportos	Simulação realista do desporto	Madden NFL Pro Evolution Scorer
Estratégia	Moba	Combate de equipa em mapas divididos em rotas	League of Legends Dota 2
	RTS (Real Time Strategy)	Estratégia em tempo real, com construções de base e combate	StarCraft Age of Empires

Prova de Aptidão Profissional

Género	SubGénero	Descrição	Exemplo
	TBS (Turn Based Strategy)	Estratégia baseada em turnos	Civilization Heroes of Might and Magic
	Tower Defense	Defesa de pontos com torres e armadilhas	Plants vs. Zombies Blood TD
Musical e Rítmico	Jogos de Música e ritmo	Sincronização com música	Guitar Hero Just Dance
Party e Casual	Jogos casuais	Jogos simples e acessíveis	Bejeweled Crossy Road
	Party Games	Jogos para grupos ou família, com foco em diversão e interação	Mario Party JackBox Party Pack
Plataforma	Metroidvania	Exploração e aquisição de competências para avançar	Holo Knight Castlevania: Symphony of the Night
	Plataforma 2D	Ação e saltos em cenários 2D	Super Mario Bros. Celest
	Plataforma 3D	Exploração e saltos em cenários tridimensionais	Super Mario 64 Crash Bandicoot
Puzzle	Logic Puzzle	Baseado em lógica	Portal The Witness
	Physics-Based Puzzle	Puzzle que envolve físico para resolver problemas	Angry Birds World of Goo
Puzzle	Quebra-Cabeças Clássicos	Resolução de problemas	Tetris Candy Crush
RPG	JRPG (Japanese RPG)	Narrativa linear e combate baseado em turnos	Final Fantasy Persona
	MMORPG	RPG <i>online</i> de mundo aberto para milhares de jogadores	World of Warcraft Guild Wars
	RPG de Ação	Combate em tempo real com progressão de personagens	The Witcher Dark Souls
	RPG de Ocidental	Liberdade de escolha e mundo aberto	Skyrim Fallout
RPG	Roguelike e Roguelite	Mapas diferentes com grau de dificuldade acima da média	Hades The Binding of Isaac

Género	SubGénero	Descrição	Exemplo
	Tactical RPG	RPG com combate em mapas táticos e por turnos	Fire Emblem XCOM
	RPG Narrativo	Jogos com foco em diálogos e decisões investigativas, muitas vezes com elementos de RPG.	Disco Elysion The Council
Simulação	Simulador de Construção/Gestão	Gerir recursos, cidades ou negócios	SimCity RollerCoaster Tycoon
	Simulador de Desporto	Simula desportos realistas como futebol e basquetbol	FIFA NBA 2K
	Simulador de Veículos	Condução visual	Fight Simulador Gran Turismo
	Simulador de Invetigação	Jogos que simulam o trabalho de um detetive, como resolver casos, analisar pistas, ect.	Her Story Telling Lies
	Simulador de vida	Simula o dia a dia	The Sims Animal Crossing
Terror	Horror de Ação	Mistura elementos de terror com ação imensa	Dead Space The Evil Within
	Horror de Investigação	Focado em mistérios com elementos de terror e suspense	Phasmonphobia Call of Cthulhu
	Psychological Horror	Terror psicológico e narrativo	Amnesia Layers of Fear
	Survival Horror	Enfatizar a sobrevivência e atmosfera tensa	Resident Evil Silent Hill
VR e AR	Realidade Aumentada (AR)	Elementos do jogo integrados com o mundo real	Pokémon GO
	Realidade Virtual (VR)	Jogos desenvolvidos para óculos VR. com imersão total	Beat Saber Half-Life: Alyx

Tabela 1 Géneros de videojogos e respetivas características

2.1.5. E-Sports e aspetos de competitividade dos videojogos

Competições com base em jogos eletrónicos, também conhecidos como *e-sports*, referem-se a competições profissionais de videojogos de diversos géneros (ver tabela da secção anterior). Onde jogadores individuais ou equipas competem em videojogos

com uma componente competitiva de alto nível. *E-Sports* envolve uma vasta gama de géneros de jogos, como jogos de estratégia em tempo real, jogos de combate, jogos de tiro na primeira pessoa (FPS), jogos de luta, entre outros.

Estes tipos de competições podem ocorrer tanto em plataformas *online* como também em eventos presenciais. Frequentemente são transmitidas ao vivo, atraindo grandes audiências em várias partes do mundo.

No contexto competitivo *e-sports*, a prática envolve a organização de torneios e ligas, onde os participantes podem ser jogadores amadores e/ou jogadores profissionais que competem por prémios monetários e prestígio. A competição em *e-sports* é um campo onde competências técnicas, trabalho em equipa, capacidade de estratégia e reações rápidas são determinantes para o sucesso. Além disso, o ambiente competitivo dos *e-sports* é sustentado por uma infraestrutura robusta, que pode incluir: treinadores, analistas, gestores de equipas e outros profissionais que suportam o desenvolvimento dos jogadores e das equipas.

Os *e-sports*, ao longo dos últimos anos, tornou-se um fenómeno mundial, com uma crescente profissionalização do setor e uma enorme base de fãs. Plataformas como a *Twitch* e o *Youtube*, eventos como *League of Legends World Championship* do jogo *League of Legends* e *Major Championships* do jogo *CS2*, contribuem para visibilidade e o crescimento no cenário competitivo dos videojogos.

Torneios e competições de *e-sports* são reconhecidos pelo seu alto nível de organização, e interação entre jogadores, público e patrocinadores, sendo atualmente uma das formas de entretenimento mais consumidas na *Internet* a nível mundial.

2.2. Metodologias

Nas secções que se seguem, são apresentadas as **metodologias** de análise utilizadas neste projeto. Encontram-se organizadas da seguinte forma: na secção 2.2.1, são descritos os **casos de uso**, que detalham os cenários específicos de interação entre os utilizadores e o sistema; na secção 2.2.2, é introduzido o **modelo de domínio**, que fornece uma visão estruturada dos principais conceitos e relações do sistema; e, por fim, na secção 2.2.3, são especificados os **requisitos funcionais** e **não funcionais**, que definem as capacidades esperadas do sistema e eventuais restrições que devem ser cumpridas.

2.2.1. Casos de Uso

Os diagramas de casos de uso (em UML) são representações gráficas utilizadas em engenharia de *software*, em contexto de análise de sistemas. Estes diagramas descrevem as interações entre os atores (utilizadores ou sistemas externos) e o sistema em desenvolvimento. O seu objetivo é demonstrar as principais funcionalidades do sistema sobre a perspetiva do utilizador.

A tabela seguinte descreve os principais componentes de um diagrama de casos de uso (adaptado de [CG, 4]).

Componente	Descrição
Ator	Representa uma entidade externa que interage com o sistema (pode ser uma pessoa, ou outro sistema, e.g. <i>hardware</i>). Podem existir relações de herança entre os atores
Caso de Uso	Representa uma funcionalidade ou comportamento do sistema que gera valor ao ator. Normalmente é representado por uma oval e inclui um verbo e um substantivo.
Relação	Linhas que conectam atores e casos de uso. Existem relações de inclusão, extensão e generalização.
Sistema	Pode ser representado por um retângulo que inclui os casos de uso do sistema (define os limites do sistema)

Tabela 2 Principais componentes dos diagramas de casos de uso

A figura seguinte ilustra um exemplo de diagrama de casos de uso para um sistema *online* de reserva de bilhetes de avião (Fonte: [EDIO, 24]). Como se observa, existem dois atores: 1) Cliente (*Customer*); e, 2) Administrador (*Admin*). Cada processo associado ao ator é caracterizado por um verbo e substantivo. O cliente pode efetuar reservas, pesquisas e cancelamentos. O administrador pode efetuar cancelamentos e atualizações da informação de voo. Observa-se ainda, o título e a fronteira do sistema (representada por um retângulo).

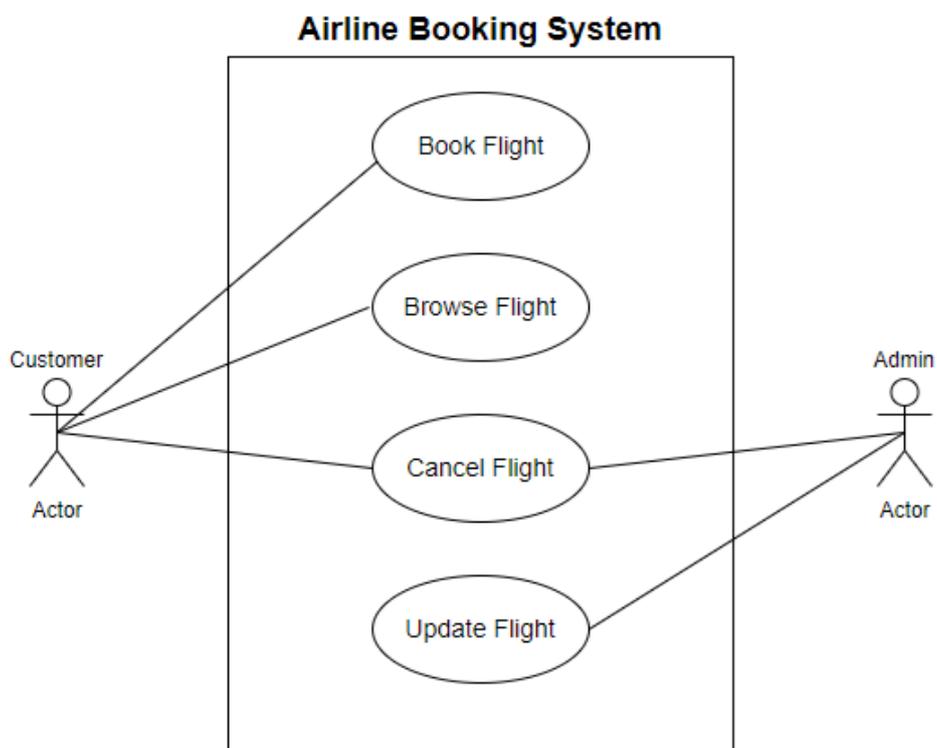


Figura 2 Caso de uso do Sistema de Reservas Online

2.2.2. Modelo de Domínio

O Modelo de Domínio é uma representação conceptual das entidades relevantes dentro de um determinado sistema ou área de interesse. Ele fornece uma visão clara dos principais conceitos e relacionamentos que compõem o domínio, funcionando como uma base fundamental para a compreensão e desenvolvimento de um sistema.

Como referido anteriormente, o Modelo de Domínio traduz as relações estáticas entre as principais classes do sistema e não se preocupa com a implementação técnica ou operações detalhadas, mas sim com os elementos essenciais do problema a ser resolvido. Ele é frequentemente representado por meio de diagramas de classes UML (*Unified Modeling Language*), uma linguagem poderosa que – entre outros aspetos - permite visualizar entidades, seus atributos, e os relacionamentos entre elas [QN, 24].

No contexto de desenvolvimento de *software*, um Modelo de Domínio é utilizado para capturar o vocabulário do domínio e as interações entre os conceitos, o que facilita o entendimento entre a equipe técnica e as partes interessadas. Ele serve como um mapa conceptual que define as fronteiras do sistema e ajuda a alinhar as expectativas entre programadores e *stakeholders*. A criação de um modelo de domínio robusto permite evitar ambiguidades e lacunas de entendimento, além de ser um ponto de partida essencial para as fases de *design* e implementação do projeto [QN, 24].

A figura seguinte ilustra um diagrama de classes para o sistema "Escola" (Exemplo).

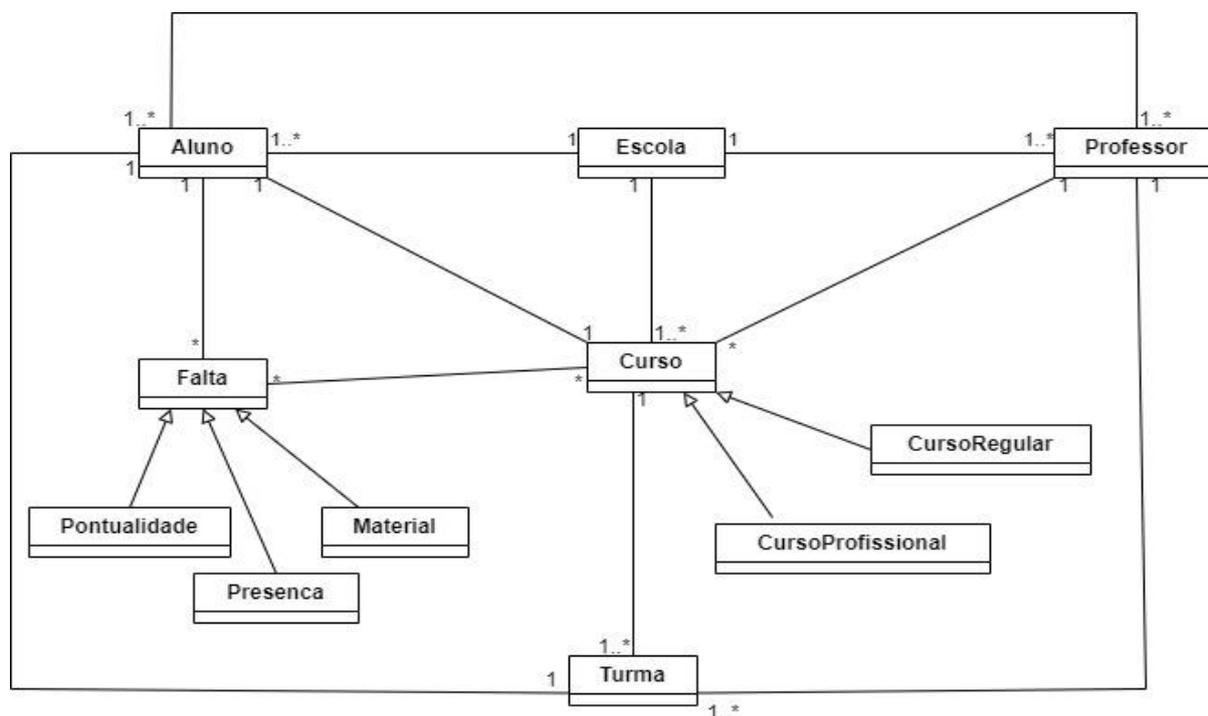


Figura 3 Modelo de Domínio UML do sistema "Escola"

Como se observa, o sistema é composto por várias classes principais que representam entidades no domínio educativo, como Escola, Aluno, Professor, Turma, e, Curso. As classes estão interligadas através de relacionamentos que refletem suas

interações no sistema. A escola é responsável por manter cursos, agregar professores, agregar turmas, e gerir alunos. A classe Aluno representa um ou mais estudantes que estão matriculados na escola. Os alunos estão associados a uma turma e participam em diferentes cursos. Cada aluno está sujeito a um controle de presença, que pode ser registado através da classe Falta. Existem três tipos de faltas: Pontualidade, Presença e Material. A classe Professor representa um ou mais docentes que pertencem à escola. Cada professor pode ser responsável por diversas turmas, que por sua vez têm vários alunos. As turmas também são caracterizadas em função do tipo de curso a que estão associadas, por exemplo, Curso Regular ou Curso Profissional.

2.2.3. Requisitos funcionais / não funcionais

Os requisitos funcionais definem o que o sistema deve fazer para resolver problemas e satisfazer as necessidades dos utilizadores. Incluem as ações, entradas e saídas, fluxos de trabalho, cálculos e comportamentos que o sistema deve ter de forma clara e objetiva. Eis alguns exemplos de requisitos funcionais [FV, 24]:

- **Inserir dados num formulário;**
- **Realizar compras;**
- **Selecionar opções num menu;**
- **Permitir que o utilizador procure produtos por categoria ou nome.**

Os requisitos não funcionais envolvem aspetos tal como: usabilidade, desempenho, confiabilidade e manutenção [FV, 24]. Eis alguns exemplos de requisitos não funcionais:

- **Tipo de Sistema Operativo;**
- **Hardware a ser utilizado;**
- **Disponibilidade do sistema;**
- **Os dados do utilizador serem encriptados usando algoritmos AES-256** (algoritmo de criptografia que usa uma chave de 256 bits para converter texto numa cifra [PG,22]).

Tipicamente, em cenários reais de análise, os requisitos são numerados para facilitar a sua identificação e acompanhamento (e.g. RF01, RF02... RNF01, RNF02, ...).

A tabela seguinte ilustra diferenças entre requisitos funcionais e não funcionais (adaptada de) [OS, 24].

Aspeto	Requisitos Funcionais	Requisitos não Funcionais
Foco	O que o sistema deve fazer.	Como o sistema se deve comportar.
Objetivo	Descrever as funcionalidades.	Definir restrições e qualidades.
Exemplo	Autorizar um registo de um utilizador.	O registo ser realizado em menos de 2 segundos.
Visibilidade	Geralmente visíveis pelo utilizador.	Raramente vistos pelo utilizador.

Tabela 3 Principais diferenças entre os Requisitos Funcionais e Não Funcionais

2.3. Tecnologias utilizadas

Este capítulo apresenta as tecnologias utilizadas durante o processo de desenvolvimento deste projeto. Em concreto, a secção 2.3.1 **Unity**, 2.3.2, **HTML5**, 2.3.3 **CSS3**, 2.3.4 **JavaScript**, e finalmente, a secção 2.3.5 **Microsoft Copilot**.

2.3.1. Unity

Unity é um motor de jogos multiplataforma e um ambiente de desenvolvimento integrado, utilizado para criar jogos 2D e 3D interativos. Lançado oficialmente em 2005 pela *Unity Technologies*, o *Unity* distingue-se pela sua capacidade de gerar experiências visuais impressionantes e ambientes virtuais imersivos.

Os principais recursos do *Unity* incluem:

- **Desenvolvimento multiplataforma** - Permite exportar jogos para diversas plataformas, como computadores, dispositivos móveis, consolas e até para a *web*;
- **Editor visual intuitivo** - Dispõe de ferramentas de *design* e prototipagem que facilitam o desenvolvimento, permitindo o arrastar e largar de componentes;
- **Integração com linguagens de programação** - Suporta a linguagem C#, possibilitando a criação de *scripts* para funcionalidades personalizadas;
- **Asset Store** - Oferece uma ampla variedade de recursos, *plugins* e ativos que aceleram o desenvolvimento;
- **Ferramentas avançadas** - Incorpora funcionalidades para renderização, física, animações e efeitos, essenciais para a criação de jogos modernos.

Para além de ser amplamente utilizado no desenvolvimento de jogos, o *Unity* aplica-se também em simulações, formações e aplicações interativas, demonstrando a sua versatilidade. [WEL_Unity, 24]

2.3.2. HTML5

HTML5 (*Hypertext Markup Language*, versão 5) é a versão mais recente da linguagem de marcação usada na *World Wide Web* [W3C, 24], projetada para estruturar e apresentar conteúdo na *Internet*.

Inicialmente proposto pela *Opera Software*, o HTML5 apresenta mudanças significativas em relação às versões anteriores, com ênfase na semântica, acessibilidade e na capacidade de integrar multimédia de forma mais eficiente.

O seu principal objetivo é melhorar a experiência do utilizador, mantendo a legibilidade e acessibilidade, ao mesmo tempo em que suporta recursos multimédia modernos, como áudio, vídeo e gráficos 2D, sem depender de *plugins* externos [WEL_HTML5, 24].

Os principais recursos de HTML5 incluem:

- **Multimédia** - suporte nativo para conteúdo de áudio e vídeo através das *tags* <audio> e <video>, eliminando a necessidade de *plugins* como o *Adobe Flash*;
- **Conteúdo gráfico** - introdução da *tag* <canvas>, permitindo desenhar gráficos dinâmicos diretamente no navegador;
- **Semântica** - novos elementos como <header>, <footer>, <article> e <section>, que enriquecem a estrutura semântica do conteúdo e facilitam a indexação pelos motores de busca;
- **Acessibilidade** - melhorias na acessibilidade e compatibilidade com dispositivos móveis e de baixo consumo, como *smartphones* e *tablets*;
- **APIs** - integração de novas APIs que permitem o desenvolvimento de aplicações *web* mais complexas, incluindo armazenamento local e manipulação gráfica;
- **Desempenho e compatibilidade** - HTML5 visa melhorar a interoperabilidade entre navegadores e o desempenho de aplicações *web*, bem como permitir que aplicações móveis baseadas em HTML5 sejam executados sem a necessidade de *software* adicional.

O HTML5 também pode ser usado para criar aplicações *web responsive* que funcionam em diferentes plataformas e dispositivos, tornando-o uma boa escolha para a criação de aplicações móveis e interfaces interativas [WEL_HTML5, 24].

2.3.3. CSS3

CSS3 é a terceira versão de *Cascading Style Sheets*. Trata-se de uma linguagem usada para definir a aparência e o *design* de páginas da *web*.

Com o mecanismo CSS3, podem criar-se estilos mais avançados, como transições suaves, animações, efeitos visuais e controle de *layout*. Esta evolução trouxe maior

flexibilidade e criatividade ao *design* de *sites*, otimizando o desempenho e reduzindo a complexidade do código [WEL _CSS3, 24].

O CSS3 é usado para estilizar páginas *web*, oferecendo uma ampla gama de recursos que melhoram a estética e interatividade do conteúdo. Além de dispor de mecanismos que facilitam a reutilização, entre as suas principais funções são:

- **Design responsive** - adaptar *layouts* para diferentes tamanhos de ecrã e dispositivos;
- **Animações e transições** - criar efeitos visuais dinâmicos sem a necessidade de *JavaScript*;
- **Estilo avançado** - adicionar sombras, transparências, gradientes e cantos arredondados;
- **Personalização de layout** - controlar a disposição dos elementos na página com precisão;
- **Redução de recursos externos** - substituir imagens de fundo e outros elementos gráficos por propriedades CSS3 nativas, otimizando o carregamento da página;

O CSS3 suporta efeitos 2D e 3D, como rotação, movimento e transição, tornando-se uma ferramenta poderosa para *designers* e programadores.

A capacidade da CSS3 de criar animações em duas e três dimensões destacou-se como uma das suas maiores inovações, uma vez que são recursos amplamente utilizados para criar *sites* interativos, experiências imersivas e até mesmo campanhas de publicidade *online*.

Navegadores modernos como *Google Chrome*, *Safari*, *Mozilla Firefox*, *Opera* e *Microsoft Edge* suportam a maioria das propriedades CSS3, o que facilita a adoção da linguagem em projetos *web* contemporâneos [WEL _CSS3, 24].

2.3.4. *JavaScript*

JavaScript é uma linguagem de programação versátil e poderosa utilizada para adicionar interatividade e comportamento dinâmico a páginas *web*. É executada no *browser*, permitindo aos programadores criar elementos interativos e responder às ações do utilizador sem a necessidade de recarregar a página [WEL _JavaScript, 24].

Com o *JavaScript*, é possível criar funcionalidades avançadas, como atualizações em tempo real, animações, validações de formulários e carregamento assíncrono de conteúdos. Esta evolução trouxe maior flexibilidade e criatividade ao desenvolvimento *web*, otimizando a experiência do utilizador e simplificando a complexidade do código.

O *JavaScript* é utilizado para controlar o comportamento das páginas *web* e oferece uma vasta gama de recursos que melhoram a usabilidade e a interatividade. Entre os seus principais recursos estão:

- **Manipulação de eventos** – responder a ações do utilizador, como cliques, digitação ou *scroll*;
- **Validação de formulários** – verificar os dados de entrada antes de os enviar para o servidor;
- **Animações e efeitos** – criar efeitos visuais dinâmicos, frequentemente em combinação com CSS;
- **Manipulação de DOM** – alterar o conteúdo e a estrutura da página dinamicamente;
- **Comunicação assíncrona** – carregar dados do servidor sem atualizar a página (AJAX);

O *JavaScript* suporta efeitos visuais 2D e 3D quando combinado com HTML5 *Canvas* ou *WebGL*, permitindo a criação de jogos interativos, simulações e experiências *web* imersivas. Isto tornou o *JavaScript* uma ferramenta essencial para programadores e designers.

Uma das maiores inovações do *JavaScript* é a sua capacidade de criar aplicações em tempo real, responsivas e interativas, desde simples formulários *web* a aplicações *web* completas e serviços *online*.

Navegadores modernos como o *Google Chrome*, *Safari*, *Mozilla Firefox*, *Opera* e *Microsoft Edge* oferecem um amplo suporte ao *JavaScript*, tornando-o uma linguagem padrão para projetos *web* contemporâneos [WEL _JavaScript, 24].

2.3.5. *Microsoft Copilot*

Microsoft Copilot é um assistente de inteligência artificial integrado na suíte *Microsoft 365*, concebido para potenciar a produtividade dos utilizadores. Ao combinar algoritmos avançados de linguagem natural com os dados das aplicações *Microsoft*, o *Copilot* auxilia na criação de documentos, na análise de dados e na automatização de tarefas, promovendo uma experiência de trabalho mais eficiente e colaborativa.

Os principais recursos do *Microsoft Copilot* incluem:

- **Integração com *Microsoft 365***: Opera de forma integrada com aplicações como *Word*, *Excel*, *PowerPoint* e *Outlook*, oferecendo sugestões contextuais em tempo real;
- **Geração de conteúdo**: Auxilia na redação de textos e na criação de apresentações, simplificando processos criativos e otimizando o fluxo de trabalho;
- **Automatização de tarefas**: Permite a execução automatizada de comandos e a análise de dados, facilitando a realização de tarefas rotineiras;

- **Assistência contextual:** Adapta-se ao ambiente de trabalho e às necessidades específicas do utilizador, oferecendo soluções personalizadas;
- **Aprendizado contínuo:** Evolui com base nas interações e no *feedback* dos utilizadores, aperfeiçoando constantemente o seu desempenho e utilidade.

Com a implementação do *Microsoft Copilot*, a *Microsoft* pretende transformar a forma de trabalhar, integrando a inteligência artificial de forma transparente e intuitiva no quotidiano dos seus produtos [WEL_Microsoft_Copilot, 24].

Esta página foi deixada em branco intencionalmente

3. Projeto Detective MaoLi

Este capítulo apresenta os principais aspetos técnicos e funcionais do sistema desenvolvido. São descritos os Casos de Uso (secção 3.1), o Modelo de Domínio (3.2) e a Arquitetura do Sistema (3.3), bem como os Requisitos Funcionais e Não Funcionais (3.4).

A seguir apresenta-se o Protótipo (3.5). Assim, é apresentado o Jogo (3.5.1), com características gerais, sistemas de menu e cenário, bem como exemplos de código e a página de suporte aos jogadores (3.5.2), identificando características gerais, esquema navegacional e exemplos de código.

3.1. Casos de Uso do Projeto Detective MaoLi

No diagrama seguinte, observa-se o caso de uso que reúne atores e respetivos processos principais do *website* e do jogo Detective MaoLi.

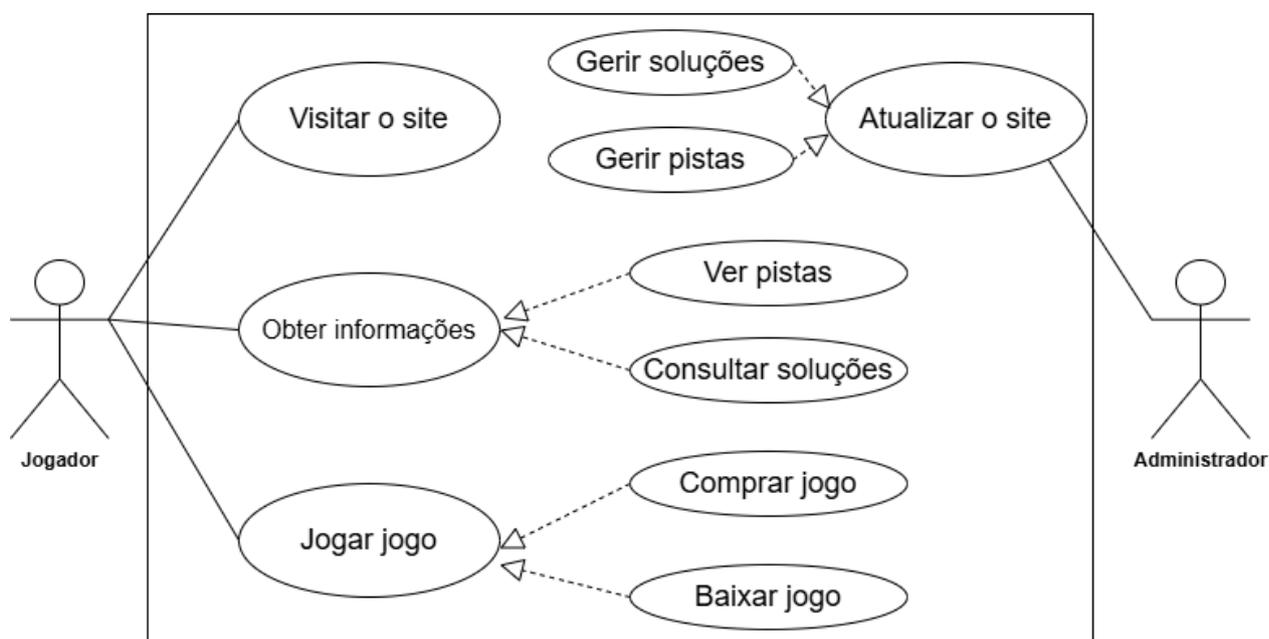


Figura 4 Casos de uso UML (Detective MaoLi)

O ator **Jogador** pode visitar o *website* e obter informações tais como **ver pistas** e **consultar soluções**. Pode ainda **jogar** caso esteja interessado e tenha realizado a compra do jogo.

O ator **Administrador** pode **atualizar o website**, que inclui **gerir pistas e/ou soluções**.

3.2. Modelo de Domínio do Projeto Detective MaoLi

No diagrama seguinte, apresenta-se o modelo de domínio deste projeto. As classes centrais são: **Website** e **Game**.

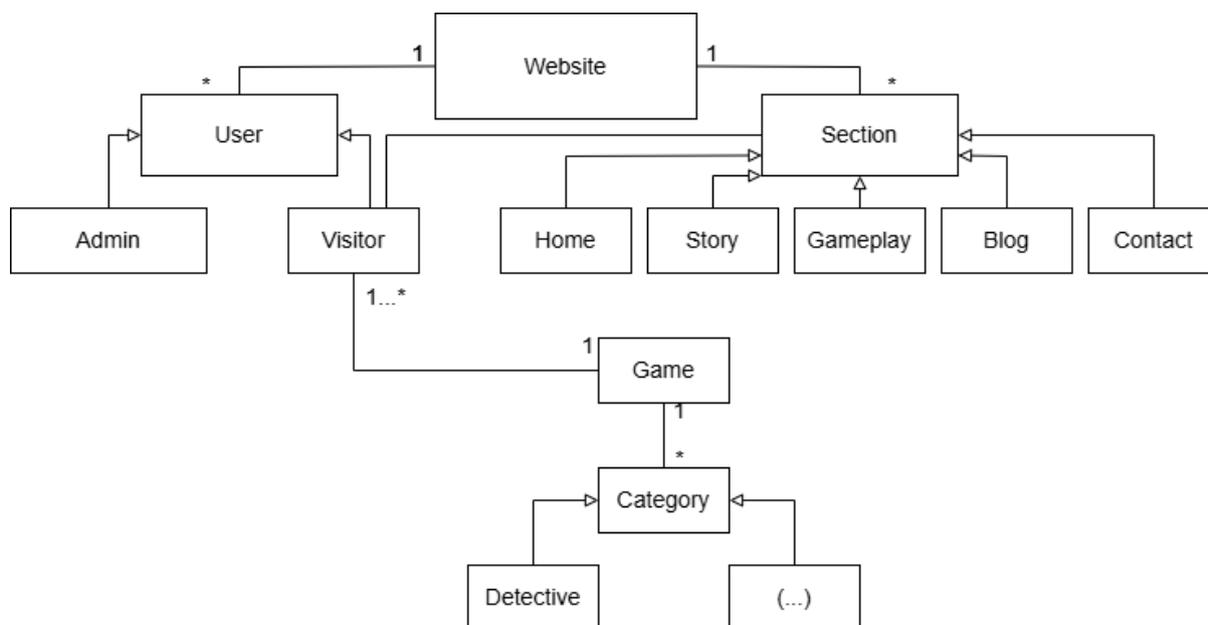


Figura 5 Modelo de Domínio UML (Detective MaoLi)

O **Website** tem vários utilizadores. Os utilizadores podem ser de dois tipos: **Admin** e **Visitor**. Cada visitante (classe **Visitor**) pode interagir com o jogo (**Game**). Por sua vez, cada jogo tem uma categoria. Existem diversos tipos de categoria, por exemplos **Detective**.

Salienta-se que o **Website** tem várias secções. As secções podem ser de vários tipos, em concreto: **Home**; **Story**; **Gameplay**; **Blog**; **Contact**.

3.3. Arquitetura do Sistema

3.3.1. Arquitetura do Jogo

Na arquitetura do projeto a **App** (aplicação final) contém o **Game**, composto por **Scenes** (cenas). Em cada cena, há vários **Game objects**, e cada objeto de jogo tem **Components** como **Transform** (posição/rotação), **Physics** (física), **Scripting** (lógica) e **Audio** (som). Tudo isto assenta na estrutura do **Unity Framework**, que inclui pacotes, o trabalho do programador e o motor principal (**Unity Core Engine**). Por fim, a camada **Cross-Platform** mostra a capacidade de compilar o jogo para várias

plataformas. A imagem seguinte ilustra de forma resumida a arquitetura de um projeto *Unity*.

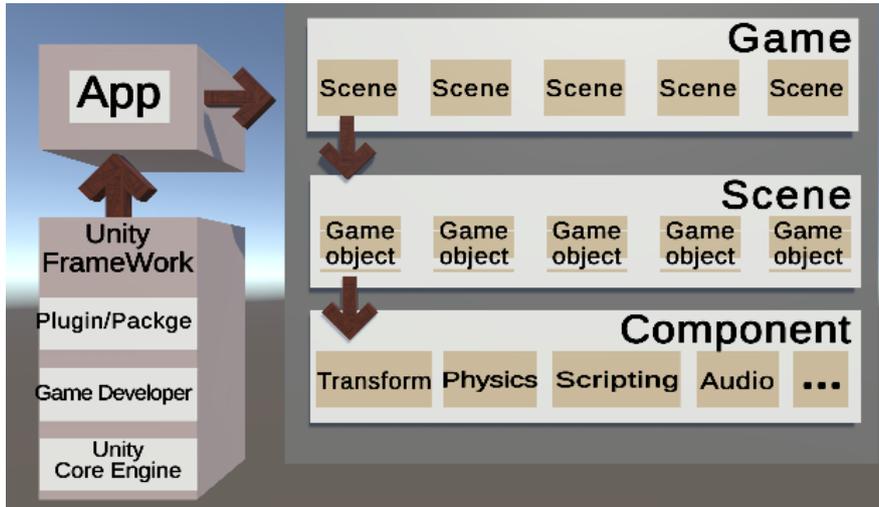


Figura 6 Arquitetura do jogo desenvolvido com Unity

3.3.2. Arquitetura da página de suporte

A página de suporte deste projeto baseia-se numa arquitetura de aplicação *web* tradicional onde o **cliente** (*browser*) envia um **pedido** via protocolo **HTTP** para o **servidor**, que atua como intermediário e encaminha o pedido para o **servidor aplicacional**. Este servidor, utilizando tecnologias como **PHP, HTML, CSS e MySQL**, processa o pedido, interage com a **base de dados (BD)**, se necessário, e gera uma resposta que é enviada de volta para o cliente via **HTTP**. Este fluxo ilustra a estrutura clássica das aplicações *web* dinâmicas, com uma clara separação entre a lógica do cliente e do servidor. Assim, a imagem seguinte representa a arquitetura referida.

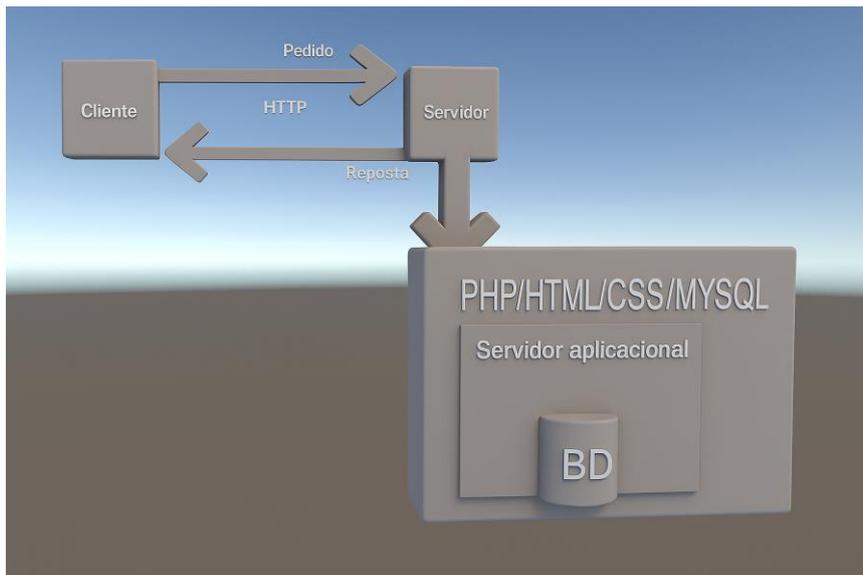


Figura 7 Arquitetura da página de suporte

3.3. Requisitos

Nesta secção apresentam-se os requisitos do jogo e da página de suporte.

3.4.1. Requisitos funcionais do Jogo

A tabela seguinte identifica os requisitos funcionais do Jogo.

ID	Descrição
RFJ01	Implementar menu principal com 4 botões Iniciar Jogo, Continuar, Definições e Sair.
RFJ02	Implementar painel de definições para ajustar o volume de música de fundo (BGM) e efeitos sonoros (SFX) em tempo real.
RFJ03	Adicionar a função de alterar a resolução do ecrã.
RFJ04	Adicionar a função de seleccionar o idioma de preferência (Inglês, português de Portugal, português de Brasil, chinês simplificado, chinês tradicional e japonês).
RFJ05	Adicionar música de fundo em <i>Loop</i> e imagem estática em ecrã completo no menu e na interface de jogo.
RFJ06	Implementar o cenário <i>SeasonDirectory</i> com botões de temporada que conduzam ao respetivo diretório de episódios.
RFJ07	Implementar navegação lateral de capas semitransparentes no cenário <i>EpisodeDirectory</i> .
RFJ08	Permitir que o jogador inicie o episódio selecionado através de duplo clique na capa central do cenário <i>EpisodeDirectory</i> .
RFJ09	Reproduzir vídeo de introdução, após um Episódio do Jogo foi iniciado.
RFJ10	Adicionar a função de ocultar a interface durante a reprodução do qualquer vídeo.
RFJ11	Adicionar a função de saltar o vídeo quando pressionar a tecla Espaço mais de 3 segundos durante a reprodução do vídeo e, restaurar a interface.
RFJ12	Implementar sistema de vidas.
RFJ13	Adicionar a mecanismo de castigo, escolhas erradas reduzem uma vida, e, ao esgotar, reproduz-se vídeo de falha e reinicia-se o episódio.
RFJ14	Reproduzir vídeos de falha e reduzir uma vida quando o jogador seleciona uma opção errada.
RFJ15	Criar sistema de diálogo de múltiplas linhas, onde cada linha de texto é apresentada sequencialmente.
RFJ16	Adicionar a função de realçar o avatar do orador atual.
RFJ17	Permitir a navegação no diálogo através de botões <i>Prev/Next</i> e das teclas de seta “←” e “→”.

Prova de Aptidão Profissional

RFJ18	Apresentar botões de opção ao fim do diálogo para que o jogador escolha.
RFJ19	Reproduzir o vídeo associado à opção correta e avançar automaticamente para o próximo nó de diálogo.
RFJ20	No final de cada episódio, exibir caixa de diálogo de gravação com as opções “Guardar e Continuar” ou “Continuar sem Guardar”.
RFJ21	Permitir pausar o jogo, pressionar “ESC” abre o menu de pausa, permitindo retomar ou regressar ao menu principal.

Tabela 4 Requisitos Funcionais do jogo

3.4.2. Requisitos funcionais da página de suporte

A tabela seguinte apresenta os principais requisitos funcionais da Página de suporte.

ID	Descrição
RFWS01	O utilizador tem acesso a um menu navegacional entre páginas. Este menu deve apresentar as opções: <i>home, story, game play, contact</i> .
RFWS02	O utilizador tem acesso a um vídeo demonstrativo sobre o jogo.
RFWS03	O utilizador tem um botão para baixar o jogo.
RFWS04	O utilizador pode visitar uma página inicial com uma introdução aos jogos de detetives.
RFWS05	O utilizador pode visualizar uma página de "jogabilidade" explicando a mecânica do jogo.
RFWS06	O utilizador pode regressar à página principal a partir de qualquer página utilizando ou clicando no logótipo.

Tabela 5 Requisitos Funcionais da página de suporte

3.4.3. Requisitos não funcionais do Jogo

A tabela seguinte apresenta os requisitos não funcionais do Jogo.

ID	Descrição
RNFJ01	Desempenho - O jogo deve apresentar tempos de carregamento curtos e transições suaves entre cenas, garantindo uma experiência fluida mesmo em computadores com <i>hardware</i> modesto.

Prova de Aptidão Profissional

ID	Descrição
RNFJ02	Compatibilidade - A aplicação deve ser executada em múltiplas plataformas (<i>Windows, macOS, Linux</i> e possivelmente em dispositivos móveis) com desempenho consistente e sem perda de funcionalidades.
RNFJ03	Usabilidade - A interface e os controlos devem ser intuitivos, permitindo que o utilizador compreenda rapidamente as mecânicas de jogo, reduzindo a necessidade de tutoriais extensos.
RNFJ04	Fiabilidade - O sistema deve evitar falhas e, caso ocorram erros, estes não devem comprometer o progresso do utilizador. A integridade dos dados de jogo (ficheiros de salvaguarda) deve ser preservada.
RNFJ05	Segurança - As informações de progresso, incluindo os ficheiros de salvaguarda, devem ser armazenadas de forma segura, prevenindo perdas e adulterações.
RNFJ06	Escalabilidade - A arquitetura deve permitir a adição de novos conteúdos (capítulos, personagens, missões) sem exigir grandes alterações no núcleo do jogo, facilitando futuras expansões.
RNFJ07	Manutenibilidade - O código deve ser bem estruturado e documentado, de modo a possibilitar a implementação de atualizações e correções de erros com facilidade, garantindo um ciclo de desenvolvimento sustentável.

Tabela 6 Requisitos Não Funcionais do Jogo

3.4.4. Requisitos não funcionais da página de suporte

A tabela seguinte apresenta os requisitos não funcionais da Página de suporte.

ID	Descrição
RNFWS01	A página deve ser compatível com todos os principais navegadores (por exemplo, <i>Chrome, Firefox, Safari, Edge</i>).
RNFWS02	A página deve ter um design responsivo para utilizadores de <i>desktop</i> e dispositivos móveis.
RNFWS03	A página deve carregar em 3 segundos para uma experiência de utilizador ideal.
RNFWS04	A página deve utilizar uma fonte legível e uma formatação clara para facilitar a navegação.
RNFWS05	A página deve ser legível em condições de pouca luz (tema escuro).
RNFWS06	A página deve ter um <i>layout</i> simples e intuitivo para ajudar os utilizadores a encontrar informação rapidamente.
RNFWS07	A página deve parecer limpa e organizada.
RNFWS08	Todos os botões e <i>links</i> devem funcionar corretamente.

ID	Descrição
RNFWS09	A página não deve ter menus confusos.
RNFWS10	O texto e as imagens devem estar bem espaçados.
RNFWS11	A página deve funcionar mesmo que as imagens não carreguem.
RNFWS12	A página não deve utilizar demasiadas cores brilhantes ou chamativas.
RNFWS13	A página não deve fazer <i>scroll</i> para os lados.
RNFWS14	A página deve ser silenciosa (sem sons de reprodução automática).
RNFWS15	A página deve ser atualizada regularmente para manter a informação relevante.

Tabela 7 Requisitos Não Funcionais da página de suporte

3.5. Protótipo

Apresentam-se secções referentes ao protótipo na componente do jogo e da página de suporte.

3.5.1. Jogo (Unity)

3.5.1.1. Características gerais

O jogo está estruturado em episódios centrados num processo de investigação, nos quais os jogadores recolhem **provas** e dialogam com **NPCs**.

Cada **episódio** começa com um **vídeo de introdução**, seguido pelos **diálogos**, que dispõem de **áudio** e, nos momentos-chave, de **vídeos** específicos. A narrativa de cada episódio organiza-se em vários **nós**: ao termo de cada diálogo, o jogador escolhe a opção correta, se acertar, avança para o nó seguinte; se errar, perde uma vida e, caso seja a última, é reproduzido um vídeo de derrota e o episódio termina. Todos os episódios têm uma **linha narrativa principal**, mas as escolhas do jogador podem desbloquear um **ramo oculto**, mais extenso, com informação adicional e **pistas** sobre as personagens.

Como a história é inspirada numa banda desenhada, os episódios devem ser jogados pela ordem, o que enriquece a experiência e facilita a resolução dos casos subsequentes. Se for detetado um ficheiro de gravação, o jogador pode escolher retomar episódios já concluídos ou continuar no mais recente.

Existe ainda um menu de opções para ajustar o volume dos vídeos, do áudio e dos efeitos sonoros, guardar o progresso e sair do jogo.

3.5.1.2. Sistemas de menu e cenários de jogo

A imagem seguinte ilustra a interface inicial do jogo assim que este é iniciado.



Figura 8 Interface inicial do jogo

Ao clicar em “Configuração” (*Settings*), é exibido o painel do menu de definições, onde pode ajustar o **volume** dos efeitos sonoros e da música, definir o **idioma** e aceder a outras opções. A imagem que se segue ilustra este painel.



Figura 9 Painel de configurações

Ao clicar em “Novo Jogo” (*New Game*), o jogador acede ao cenário *SeasonDirectory*, no cenário é apresentado um **menu** com as imagens correspondentes às **temporadas disponíveis**. A imagem que se segue ilustra este cenário.



Figura 10 Cenário *SeasonDirectory*

Ao clicar numa das imagens, por exemplo na que tem a descrição “*Season 1*”, o jogador acede ao cenário *EpisodeDirectory 1* (correspondente à *Season 1*), ao entrar no cenário *EpisodeDirectory 1*, é apresentado outro **menu** com as imagens correspondentes aos **episódios disponíveis**. Ao fazer **duplo clique** numa dessas imagens (por exemplo a primeira), inicia o episódio. A imagem que se segue ilustra este cenário.



Figura 11 Cenário *EpisodeDirectory*

Ao entrar no cenário *S1Ep1*, é produzido imediatamente o **vídeo de introdução (IntroClip)**. A imagem que se segue ilustra uma captura de ecrã desse vídeo de introdução.

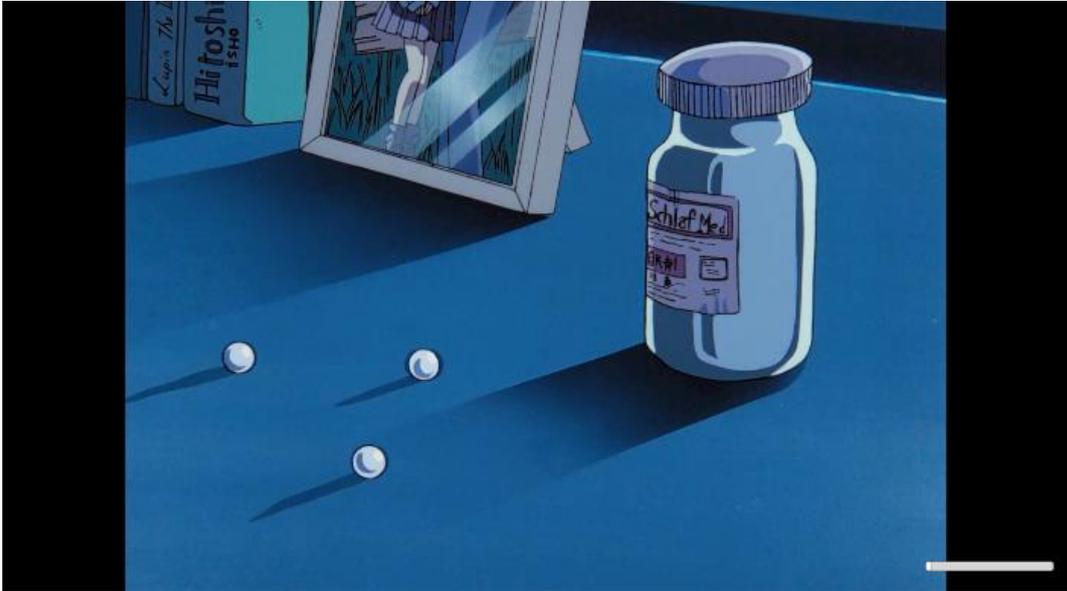


Figura 12 Vídeo de introdução no cenário *S1Ep1*

Após a reprodução do vídeo, surge de imediato uma **imagem de fundo (1)**, o **avatar do primeiro interlocutor (2)**, um **painel de diálogo (3)** e um **botão para avançar para a fala seguinte (4)**. A imagem que se segue ilustra as situações referidas.



Figura 13 Interface para início de diálogo

No final da conversa, surgem **botões de escolha**. No nó ilustrado na imagem seguinte, há apenas um botão (único e correto), que serve exclusivamente para informar o jogador e avançar para o próximo nó.



Figura 14 Interface de fim do diálogo (S1Ep1StodyNode1)

Ao clicar no botão, é reproduzido um **vídeo de transição** para o próximo nó/contexto, onde surgirão **novos diálogos** e **escolhas**. A imagem que se segue ilustra uma captura de ecrã desse vídeo de transição.



Figura 15 Vídeo de transição para o nó seguinte

No final, o jogador terá opções para novos diálogos até acabar a história. A imagem seguinte ilustra uma nova imagem do fundo e respetivo novo painel.



Figura 16 Novo nó, nova imagem de fundo e novo painel

3.5.1.3. Exemplos de código-fonte

Nesta secção ilustram-se vários exemplos de código-fonte. O bloco de código seguinte demonstra como é reproduzido o vídeo de introdução (*IntroClip*) previamente guardado no *Unity*; em simultâneo, são desencadeadas outras ações em segundo plano, como alterar a imagem de fundo, esticá-la para cobrir todo o ecrã e garantir que todos os elementos da UI fiquem posicionados por baixo do vídeo.

```
private IEnumerator RunEpisode()
{
    if (introClip != null)
        yield return StartCoroutine(VideoManager.Instance.PlayClip(introClip));

    foreach (var node in storyNodes)
    {
        if (node.background != null)
        {
            backgroundImage.gameObject.SetActive(true);
            backgroundImage.sprite = node.background;

            var rt = backgroundImage.rectTransform;
            rt.anchorMin = Vector2.zero;
            rt.anchorMax = Vector2.one;
            rt.offsetMin = Vector2.zero;
            rt.offsetMax = Vector2.zero;

            backgroundImage.transform.SetAsFirstSibling();

            Canvas.ForceUpdateCanvases();
            yield return null;
        }
    }
}
```

Figura 17 Código-fonte (após início de episódio)

O bloco de código seguinte cria vários “slots” no *Unity* que permitem designar quem assume o papel de protagonista e quem é convidado especial, bem como definir o nome e o avatar de cada personagem.

```
using UnityEngine;

[CreateAssetMenu(fileName = "Character_", menuName = "Game/CharacterData")]
public class CharacterData : ScriptableObject
{
    [Tooltip("角色显示名称 (可选)")]
    public string characterName;

    [Tooltip("头像 Sprite, 用于对话 UI")]
    public Sprite portrait;

    [Tooltip("是否为主角, 主角始终显示在左侧")]
    public bool isMainCharacter = false;
}
```

Figura 18 Código-fonte (criar Slots no Hierarchy do Unity)

O bloco de código seguinte começa por definir ambos os avatares (esquerdo e direito) como semitransparentes e ligeiramente mais escuros. Se a linha de diálogo tiver um orador e este for o protagonista, o avatar da esquerda passa a exibir o retrato do orador, recupera a cor original e deixa de ser semitransparente; caso contrário (isto é, se houver orador, mas este não for o protagonista), aplica-se o mesmo ao avatar da direita.

```
private void UpdateAvatars(DialogueLine line)
{
    Color dim = new Color(1f, 1f, 1f, 0.5f);
    leftAvatar.color = dim;
    rightAvatar.color = dim;

    if (line.speaker != null && line.speaker.isMainCharacter)
    {
        leftAvatar.sprite = line.speaker.portrait;
        leftAvatar.color = Color.white;
        rightAvatar.gameObject.SetActive(false);
    }
    else if (line.speaker != null)
    {
        rightAvatar.sprite = line.speaker.portrait;
        rightAvatar.color = Color.white;
        rightAvatar.gameObject.SetActive(true);
    }
}
```

Figura 19 Código-fonte (características do avatar)

O bloco de código seguinte atribui os *sprites* opcionais aos avatares esquerdo e direito, define *canvasGroup.alpha* para 1 e ativa *interactable* e *blocksRaycasts*, garantindo que o painel fique visível e interativo. Por fim, o método *Hide()* inverte estas definições, tornando o painel invisível e sem resposta a qualquer *input*.

```
public void Show(string text, Sprite leftSprite = null, Sprite rightSprite = null)
{
    narrativeText.text = text;
    leftAvatar.sprite = leftSprite;
    rightAvatar.sprite = rightSprite;
    canvasGroup.alpha = 1f;
    canvasGroup.interactable = true;
    canvasGroup.blocksRaycasts = true;
    narrativeText.rectTransform.anchoredPosition = Vector2.zero;
}

public void Hide()
{
    canvasGroup.alpha = 0f;
    canvasGroup.interactable = false;
    canvasGroup.blocksRaycasts = false;
}
```

Figura 20 Código-fonte (visibilidade da UI)

3.5.2. Página de suporte aos jogadores (HTML/CSS)

3.5.2.1. Características gerais

O *site* inclui as seguintes páginas principais: **Home**, **Story**, **Gameplay** e **Contact**.

Na página principal (**Main Page**), há uma breve introdução e a opção de descarregar o jogo. Além disso, existe um *menu* de navegação com as opções **Home**, **Story**, **Gameplay** e **Contact**, permitindo que o utilizador aceda facilmente a cada página ao clicar nas respetivas opções:

- **Home** - O utilizador encontrará informações sobre o jogo, algumas pistas e outros detalhes intrigantes;
- **Story** - Esta página apresenta histórias sobre os casos que estarão no jogo;
- **Gameplay** - Aqui, o utilizador poderá ver informações sobre como o jogo funciona, incluindo um vídeo explicativo para uma melhor compreensão;
- **Contact** - Esta secção contém informações sobre os administradores responsáveis pelo *site* e pelo jogo, incluindo contactos como *email*.

3.5.2.2. Esquema navegacional

A imagem seguinte ilustra a estrutura do menu de seleção de episódios no jogo *Detective Maoli*. Cada nó (de 1 a 5) será detalhado nesta secção.

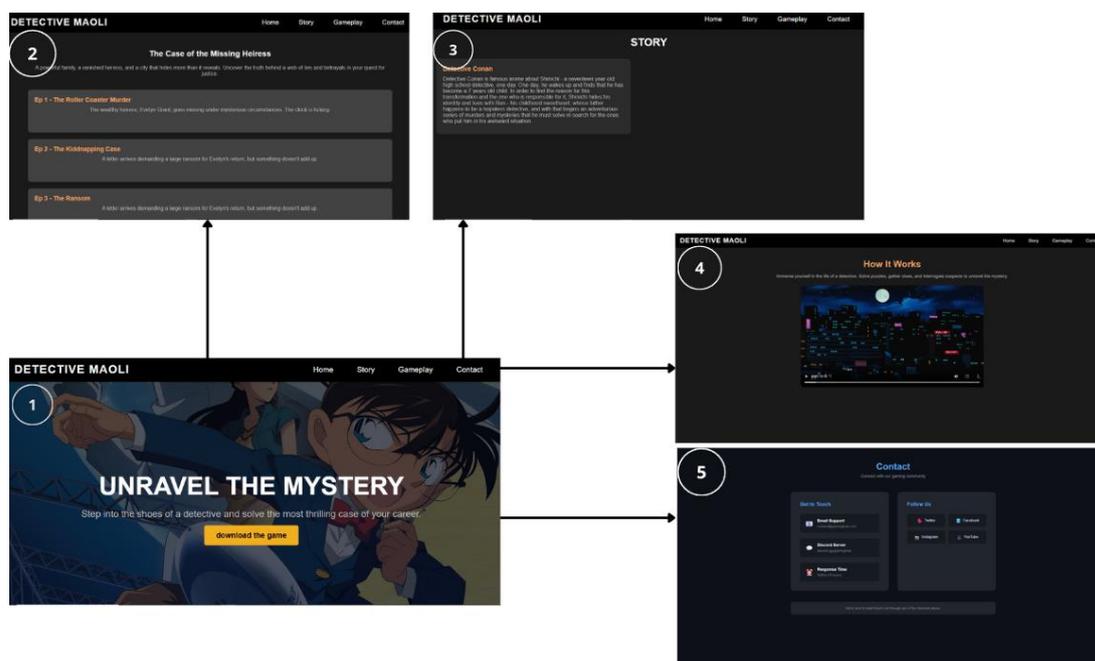


Figura 21 Esquema navegacional da página de suporte

Descrevem-se os cinco nós de informação identificados na imagem anterior:

1. **Página principal** do jogo **Detective Maoli**, destacando o título, um **botão de download** e um **menu de navegação** para **Home, Story, Gameplay e Contact**. Serve como ponto de entrada, guiando os jogadores de forma intuitiva;



Figura 22 Main Page da página de suporte

2. **Página inicial** do jogo **Detective Maoli** apresenta um design simples e intuitivo. O título do jogo está no topo, seguido por um menu de navegação com opções como **Story, Gameplay e Contact**;

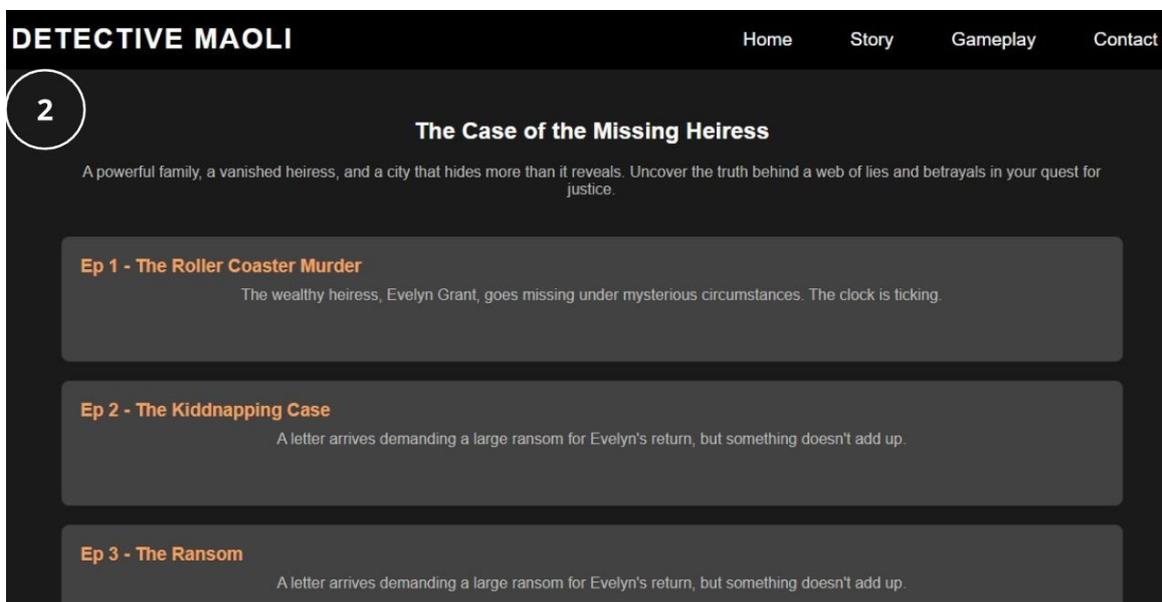


Figura 23 Home Page da página de suporte

3. A **página de Story** do **Detective Maoli** apresenta a história principal, **"The Case of the Missing Heiress"**, centrada no desaparecimento de uma herdeira e numa cidade cheia de segredos;

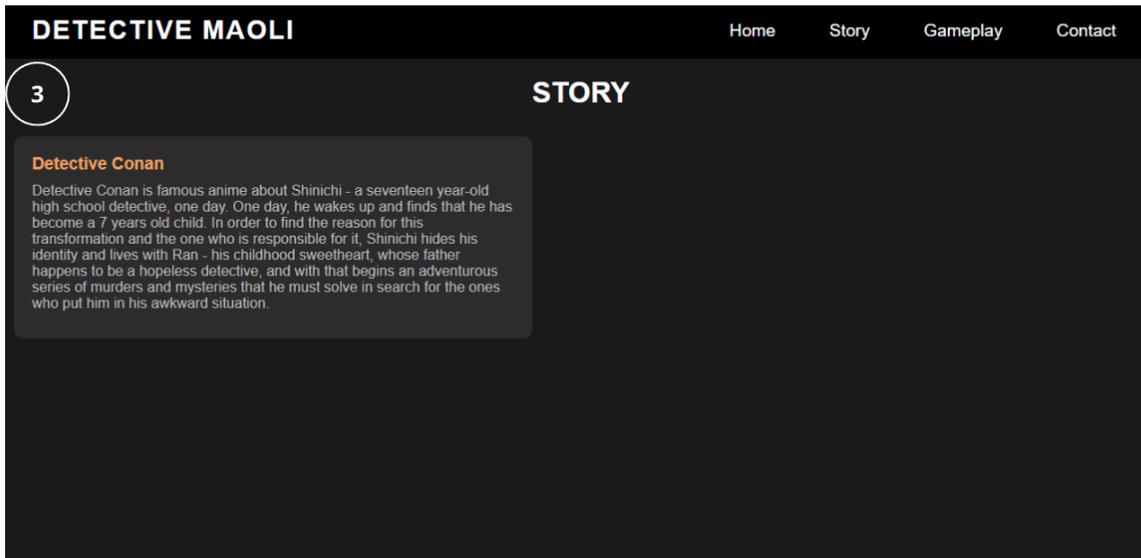


Figura 24 Story da página de suporte

4. A página de **Gameplay** do **Detective Maoli** demonstra uma página com um vídeo para os utilizadores perceberem como funciona o jogo;

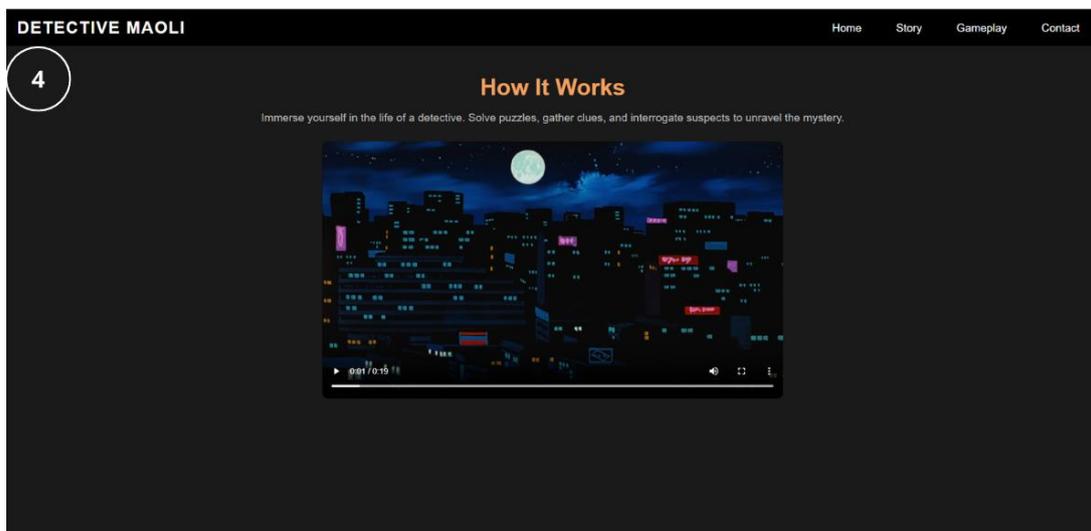


Figura 25 Gameplay da página de suporte

5. A página de **Contact** do **Detetive Maoli** demonstra uma página com informações como o e-mail do administrador do site e do jogo.

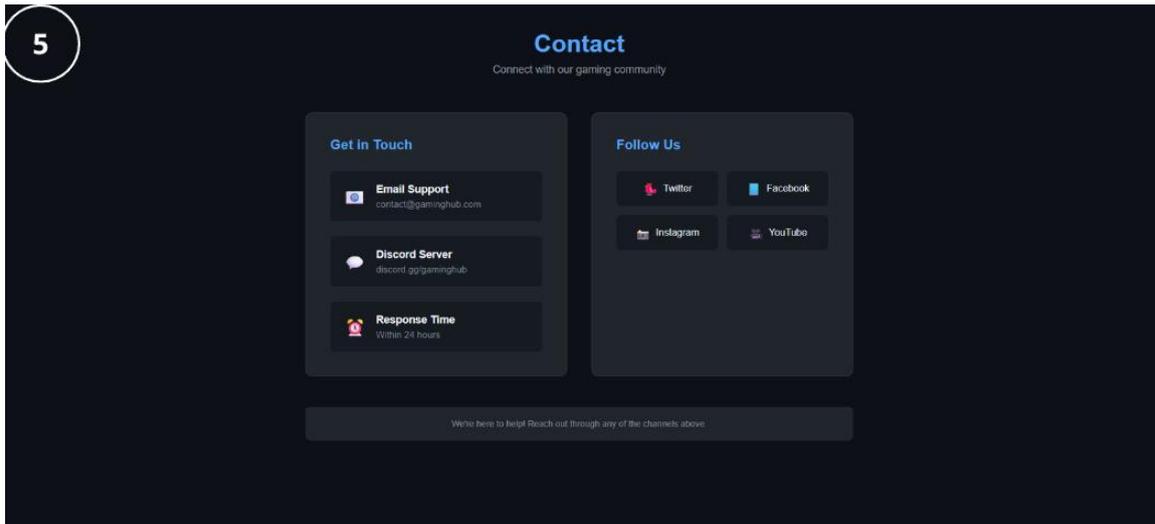


Figura 26 Contact da página de suporte

3.5.2.3. Exemplos de código-fonte

Nesta seção, a título de exemplo, é ilustrado o código-fonte criado para a barra de navegação, que inclui um logótipo e links para diferentes páginas. O elemento `<nav>` contém uma `<div>` com a classe "logo", que liga a "index.html" como "Detective Maoli". Além disso, uma lista não ordenada (`<ul class="nav-links">`) organiza os links de navegação dentro de elementos ``, direcionando os utilizadores para "home.html" (Início), "story.html" (História), "gameplay.html" (Jogabilidade) e "contact.html" (Contato), facilitando a navegação no site.

```

<!-- Navbar -->
<nav class="navbar">
  <div class="logo">
    <a href="index.html">Detective Maoli</a>
  </div>
  <ul class="nav-links">
    <li><a href="home.html">Home</a></li>
    <li><a href="story.html">Story</a></li>
    <li><a href="gameplay.html">Gameplay</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</nav>
    
```

Figura 27 Código-fonte (Navbar)

Nesta imagem, a título de exemplo, é ilustrado o código-fonte criado para incorporar um vídeo teaser. Uma `<div>` com a classe "video-container" contém um elemento

<video> com o atributo controls, permitindo controlos de reprodução. A tag <source> especifica o ficheiro de vídeo (src=".mp4") e o formato (type="video/mp4"). Caso o navegador não suporte a tag <video>, uma mensagem alternativa, "O seu navegador não suporta a tag de vídeo", será exibida.

```
<!-- Video Teaser -->
<div class="video-container">
  <video controls>
    <source src=".mp4" type="video/mp4">
    Your browser does not support the video tag.
  </video>
</div>
```

Figura 28 Código-fonte (Video)

Na imagem seguinte, a título de exemplo, é ilustrado o código-fonte criado para estilizar uma secção de história usando CSS. A classe *story-section* aplica um fundo com um *gradiente linear* (rgba(4,9,30,0.7)) sobre uma imagem (*detective-conan-6.jpg*), com posicionamento centrado (*background-position: center*) e cobertura total (*background-size: cover*). A secção tem uma posição relativa (*position: relative*), um preenchimento de *50px*, uma largura máxima de *1200px* e está centralizada com *margin: 0 auto*.

```
/* Story Section Styling */
.story-section {
  background-image: linear-gradient(rgba(4,9,30,0.7), rgba(4,9,30,0.7)), url(detective-conan-6.jpg);
  background-position: center;
  background-size: cover;
  position: relative;
  padding: 50px;
  max-width: 1200px;
  margin: 0 auto;
}
```

Figura 29 Código-fonte (Styling)

3.6. Testes de Produto

Sob o ponto de vista de desenvolvimento de *software* os testes de produto (relativos ao jogo e página de suporte) são essenciais. Por exemplo, ao nível de testes

modulares e testes de usabilidade. Estes testes serão realizados em âmbito de trabalho futuro (ver secção 4.3).

4. Conclusões

Os protótipos aplicativos demonstraram que todos os requisitos funcionais foram cumpridos. Neste capítulo apresenta-se a discussão relativa ao desenvolvimento do projeto e o trabalho futuro.

4.1. Discussão

Este trabalho permitiu consolidar conhecimentos adquiridos ao longo do curso bem como desenvolver novas competências no domínio de desenvolvimento de *software*.

Considerando o contexto e motivação, perante o problema identificado:

“Apesar do crescimento e reconhecimento gradual da indústria de jogos do tipo detective, na nossa perspetiva, verifica-se uma escassez dos mesmos. Devido a este problema, este projeto visa contribuir com mais um título para o ecossistema de jogos do tipo detective.”, definiram-se seis objetivos principais:

- **O1** – Analisar o enquadramento teórico do projeto;
- **O2** - Definir os casos de uso do projeto, i.e. identificação dos principais atores, processos e fronteiras do sistema;
- **O3** - Elaborar o modelo de domínio que traduza as relações estáticas das principais classes do projeto;
- **O4** - Definir a arquitetura do sistema;
- **O5** - Definir os requisitos das aplicações;
- **O6** - Realizar o protótipo aplicativo.

Assim, relativamente a:

- **O1** – Como se demonstrou no Capítulo 2, foram estudadas e incluídas no presente relatório os temas: videojogos, metodologias, e tecnologias utilizadas;
- **O2** - Como se demonstrou na secção 3.1, foram definidos os casos de uso do projeto, identificados os principais atores e processos associados, bem como as fronteiras do sistema;
- **O3** - Como se demonstrou na secção 3.2, foi elaborado o modelo de domínio que traduz as principais relações estáticas das classes do projeto;
- **O4** - Como se demonstrou no Capítulo 3, secção 3.3, foi definida a arquitetura do sistema para o jogo e para a página *web* de suporte aos jogadores;

- O5 - Como se demonstrou na secção 3.4, foram definidos os requisitos funcionais e não funcionais das aplicações realizadas;
- O6 - Como se demonstrou na secção 3.5 foram realizados dois protótipos aplicativos (para o jogo e página *web* respetivamente).

Conclui-se deste modo que os objetivos inicialmente propostos foram alcançados.

4.2. Limitações

Este trabalho apresenta limitações, por exemplo, a seleção de idioma (português de Portugal ou do Brasil) não foi implementada.

Se por um lado a página de suporte tem uma interface *responsive*, por outro lado, a interface do jogo não é ainda *responsive*, ou seja, a resolução é fixa.

Todavia, foi escolhida uma resolução que garante a compatibilidade com a maior parte dos ecrãs.

Finalmente, por se tratar de um primeiro protótipo, o jogo só está disponível para PCs com *Windows* na versão 10 ou 11.

4.3. Trabalho futuro

No futuro, pretende-se resolver os problemas referidos na secção 4.2 e realizar testes modulares e de usabilidade.

Outro objetivo consiste no lançamento do projeto na plataforma *online Steam* e, em fases seguintes, adicionar novos episódios e cenários, implementar efeitos visuais e animações nas personagens, criar um servidor para gerir atualizações e recolher o feedback dos jogadores através de uma área de discussão onde os mesmos possam partilhar a sua experiência. Ou seja, ações que visam melhorar continuamente a experiência de jogo.

Referências

- [CG, 4] ChatGPT, Versão 4
- [EB_MYSQL,24] MySQL | Definition, History, & Facts | Britannica, Encyclopædia Britannica, Inc, <https://www.britannica.com/topic/MySQL>, [online] acedido em 2/12/2024.
- [EDIO, 24] What is a use case diagram? Educative Inc, <https://www.educative.io/answers/what-is-a-use-case-diagram>, <https://www.educative.io/answers/what-is-a-use-case-diagram>, [online] acedido em 25/9/2024.
- [FV, 24] Valente, Francisco. Requisitos Funcionais/Não Funcionais. Relatório de PAP. E.S.Camões (2024)
- [JC, 24] Corbelli, Johnny. Modelo de Negócio. Relatório de PAP. E.S. Camões (2024)
- [MW, 24] Requisitos funcionais e não funcionais o que são? Mestres da Web, artigo escrito por Fernando Cunha <https://mestresdawe.com.br/tecnologias/requisitos-funcionais-e-nao-funcionais-o-que-sao> [online] acedido em 22/10/2024
- [MySQL, 24] MySQL, <https://www.mysql.com/>, [online] acedido em 5/12/2024.
- [OS, 24] Sow, Oumar. Modelo de Domínio. Relatório de PAP E.S.Camões (2024)
- [PG, 22] What is AES-256 Encryption & How Does it Work? Progress Software Corporation, escrito por Victor Kananda a 22/6/2022 <https://www.progress.com/blogs/use-aes-256-encryption-secure-data> [online] acedido em 23/10/2024
- [QN, 24] Modelo de Domínio de Problema e de Solução, Quati Network, https://quatinetwork.wordpress.com/wp-content/uploads/2012/04/modelo_dominio.pdf, [online] acedido em 25/10/2024.
- [VP, 24] Pires, Valdir Modelo de Negócio. Relatório de PAP. E.S. Camões (2024)
- [W3C, 24] World Wide Web Consortium, <https://www.w3.org/>, [online] acedido em 5/12/2024.
- [W3PHP_FORM, 24] PHP Form Validation, W3Schools, Refsnes Data, https://www.w3schools.com/PHP/php_form_validation.asp, [online] acedido em 26/9/2024.
- [WEL_CSS3, 24] CSS3 – Wikipédia, a enciclopédia livre, <https://pt.wikipedia.org/wiki/CSS3>, [online] acedido em 30/11/2024.
- [WEL_HTML5, 24] HTML5 – Wikipédia, a enciclopédia livre, <https://pt.wikipedia.org/wiki/HTML5>, [online] acedido em 30/11/2024.

Prova de Aptidão Profissional

- [WEL_PHP, 24] PHP – Wikipédia, a enciclopédia livre, <https://pt.wikipedia.org/wiki/PHP>, [online] acedido em 30/11/2024.
- [WEL_MYSQL, 24] MySQL – Wikipédia, a enciclopédia livre, <https://pt.wikipedia.org/wiki/MySQL>, [online] acedido em 2/12/2024.
- [CP, 25] https://www.conanpedia.com/TV1_%E4%BA%91%E9%9C%84%E9%A3%9E%E8%BD%A6%E6%9D%80%E4%BA%BA%E4%BA%8B%E4%BB%B6
- [DCW, 25] https://www.detectiveconanworld.com/wiki/Roller_Coaster_Murder_Case [online] acedido em 2/12/2024.