

Mestrado em Engenharia Informática  
Dissertação  
Relatório Final

## Desenvolvimento de um Jogo para Aprendizagem de Polinómios

Autor :

Ricardo Nuno de Magalhães Barroca  
rbarroca@student.dei.uc.pt

Orientador :

Licínio Roque

6 de Julho de 2015



**FCTUC** DEPARTAMENTO  
**DE ENGENHARIA INFORMÁTICA**  
FACULDADE DE CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE COIMBRA



UNIVERSIDADE DE COIMBRA

Faculdade de Ciências e Tecnologia  
Departamento de Engenharia Informática

Dissertação

**Desenvolvimento de um Jogo para Aprendizagem de Polinómios**

por Ricardo Nuno de Magalhães Barroca

*Juri*

**Juri Arguente:**

*António José Nunes Mendes*

**Juri Vogal:**

*Alcides Fonseca*

**Orientador:**

*Licínio Roque*



## *Resumo*

A maioria dos jogos tem como principal objectivo proporcionar entretenimento a quem os joga. Para atingir este objectivo o desenvolvimento de cada jogo é baseado nas características idealizadas, consoante o género do mesmo, para que o jogador se sinta tentado a experimentá-lo e mantenha o interesse no mesmo. Este trabalho visou a produção de um jogo para a aprendizagem sobre polinómios, inscrito no âmbito de um projecto de doutoramento na FPCEUC. A aplicação desenvolvida segue uma variante dos jogos sérios, com o intuito de entreter o jogador, enquanto aprende sobre cálculo entre polinómios, no contexto da disciplina de Matemática no terceiro ciclo de educação.

**Palavras-Chave:** Aprendizagem, Design, Jogo, Protótipo, Polinómio, Android

## *Abstract*

The majority of the games has, as its main purpose, to provide entertainment to those who play them. To achieve this, the development of each game is based on idealized characteristics, depending on it's kind, so that the player may be tempted to try it and keep his interest in the game. This work is aimed at the production of a game for learning about polynomials, in the framework of a project of doctorate in FPCEUC. The application developed follows a serious game's variant with the aim to entertain the player, while learning about calculation between polynomials, in the context of the discipline of Mathematics in the third cycle of education.

**Keywords:** Learning, Design, Prototype, Polynomial, Android



## *Agradecimentos*

Primeiro que tudo, gostaria de reconhecer a minha gratidão ao meu orientador, o Professor Licínio Roque, por toda a sua orientação e conhecimento fornecido, bem como toda a sua paciência empregue durante todo o processo do estágio.

Gostaria também de reconhecer o grupo do CISUC por me terem recebido e por toda a ajuda por eles oferecida sempre que necessitei.

Quero também agradecer a todos os meus amigos e em especial, à República dos Galifões e todas as pessoas que dela fazem parte, que me acompanharam durante todo o processo e que me apoiaram e estiverem sempre presentes nos bons e maus momentos durante todo este trajecto.

Finalmente gostaria de agradecer todo o apoio dos meus familiares pois sem eles não estaria aqui, por me acompanharem durante todo o processo e me ajudarem com tudo o que estava ao seu alcance.



# Conteúdo

<b>Resumo</b>	<b>iv</b>
<b>Agradecimentos</b>	<b>vi</b>
<b>Conteúdo</b>	<b>viii</b>
<b>Lista de Figuras</b>	<b>xi</b>
<b>Lista de Tabelas</b>	<b>xiv</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Contexto . . . . .	1
1.2 Identificação do Problema . . . . .	2
1.3 Objetivos . . . . .	2
1.4 Motivações . . . . .	3
1.4.1 Utilizador Alvo . . . . .	3
1.4.2 Desafios a Superar . . . . .	4
1.5 Estrutura do Documento . . . . .	4
<b>2 Estado da Arte</b>	<b>6</b>
2.1 Aprendizagem . . . . .	6
2.1.1 Aprendizagem por Actividade . . . . .	6
2.1.2 Aprendizagem por Interactividade . . . . .	7
2.1.3 Aprendizagem Através de Jogos . . . . .	7
2.1.4 Matemática e Aprendizagem através de Jogos . . . . .	8
2.2 Design de Jogos . . . . .	10
2.2.1 Definindo Jogo . . . . .	10
2.2.1.1 David Parlett . . . . .	10
2.2.1.2 Clark C. Abt . . . . .	10
2.2.1.3 E. Avedon e B. Sutton-Smith . . . . .	11
2.2.1.4 Salen e Zimmerman . . . . .	11
2.2.2 Regras . . . . .	12
2.2.3 Mecânicas de Jogo . . . . .	13
2.2.4 Design e Desenvolvimento de Jogos . . . . .	14
2.2.4.1 Designer . . . . .	16
2.2.4.2 Experiência . . . . .	17
2.2.4.3 Jogo . . . . .	18

2.2.4.4	Processo . . . . .	19
2.2.4.5	Jogador . . . . .	21
2.2.4.6	Interface . . . . .	22
2.3	Design de Jogos e Aprendizagem . . . . .	23
2.3.1	Princípios de Gee . . . . .	23
2.3.2	Serious Games . . . . .	24
2.4	Análise de Aplicações Existentes . . . . .	26
<b>3</b>	<b>Metodologia</b>	<b>28</b>
3.1	Objectivos . . . . .	28
3.2	Abordagem . . . . .	29
3.3	Responsabilidades da Equipa . . . . .	30
3.4	Plano de Trabalho . . . . .	31
3.4.1	Primeiro Semestre . . . . .	31
3.4.2	Segundo Semestre . . . . .	32
<b>4</b>	<b>Análise de Requisitos</b>	<b>34</b>
4.1	Requisitos Iniciais . . . . .	35
4.1.1	Requisitos Funcionais . . . . .	35
4.1.2	Requisitos Não Funcionais . . . . .	41
4.2	Requisitos Finais . . . . .	43
4.2.1	Requisitos Funcionais . . . . .	43
4.2.2	Requisitos não Funcionais . . . . .	48
<b>5</b>	<b>Arquitectura</b>	<b>50</b>
5.1	Arquitectura Conceptual . . . . .	50
5.2	Arquitectura Final . . . . .	51
5.2.1	Modelo . . . . .	52
5.2.2	Classes de comunicação Aplicação-Servidor . . . . .	52
5.2.3	Visão Parcial - Autenticação . . . . .	53
5.2.4	Visão Parcial - Jogar . . . . .	54
5.2.5	Visão Parcial - Menu Medalhas e de Liderança . . . . .	55
5.2.6	Visão Parcial - Desafiar . . . . .	55
5.2.7	Visão Parcial - Jogar Problema Desafiado . . . . .	56
5.3	Tecnologias . . . . .	57
5.3.1	Plataforma Escolhida . . . . .	57
5.3.2	Linguagens de Programação . . . . .	57
5.3.3	Base de Dados . . . . .	58
5.3.4	Outras Tecnologias . . . . .	58
<b>6</b>	<b>TEMPOLY - Prototipagem e Desenvolvimento</b>	<b>60</b>
6.1	Mecânicas de Jogo - Modo de jogo com apenas uma operação . . . . .	62
6.2	Menu Principal . . . . .	65
6.3	Modos de jogo com diferente número de operações . . . . .	66
6.4	Recolha de Dados . . . . .	68
6.5	Servidor e Base de Dados . . . . .	69
6.6	Design de Interface Final . . . . .	70
6.6.1	Navegação Global Entre Ecrãs . . . . .	70

---

6.6.2	Menu inicial, Registo e Autenticação de Utilizadores . . . . .	71
6.6.3	Inserção do Design . . . . .	72
6.6.4	Menu de Progressão - Níveis . . . . .	73
6.6.5	Menu de Medalhas e Liderança . . . . .	74
6.6.6	Modo de desafio e respectivos menus . . . . .	76
6.6.6.1	Desafiar . . . . .	76
6.6.6.2	Meus Desafios . . . . .	79
6.7	Requisitos por Alcançar . . . . .	80
<b>7</b>	<b>Avaliação</b>	<b>82</b>
7.1	Processo . . . . .	82
7.2	Resultados e Análise . . . . .	84
7.3	Observações . . . . .	96
<b>8</b>	<b>Conclusões e Trabalho para o Futuro</b>	<b>98</b>
8.1	Conclusões . . . . .	98
8.2	Trabalho para o Futuro . . . . .	99
	<b>Bibliografia</b>	<b>100</b>

# Lista de Figuras

2.1	Desenvolvimento de um Jogo . . . . .	15
2.2	Primeiro existe um Designer . . . . .	16
2.3	O Designer cria uma Experiência . . . . .	17
2.4	A Experiência nasce através de um Jogo . . . . .	18
2.5	O Jogo nasce de uma Ideia(Processo) . . . . .	20
2.6	O Jogo é feito para um Jogador . . . . .	21
2.7	O Jogador interage através de uma Interface . . . . .	22
2.8	Interação entre Aprendizagem, Jogos e Simulação . . . . .	25
3.1	Prototipagem Evolutiva . . . . .	30
3.2	Plano para o primeiro semestre . . . . .	31
3.3	Plano inicial para o segundo semestre . . . . .	32
3.4	Plano final para o segundo semestre . . . . .	32
5.1	Arquitetura Conceptual . . . . .	50
5.2	Diagrama de Componentes . . . . .	52
5.3	Diagrama de Sequência - Autenticação . . . . .	53
5.4	Diagrama de Sequência - Jogar . . . . .	54
5.5	Diagrama de Sequência - Menu de Medalhas e de Liderança . . . . .	55
5.6	Diagrama de Sequência - Desafiar . . . . .	55

---

5.7	Diagrama de Sequência - Jogar Problema Desafiado . . . . .	56
6.1	Algoritmo de verificação de resposta . . . . .	64
6.2	Ambiente de Jogo Simples . . . . .	65
6.3	Menu Principal . . . . .	66
6.4	Ambiente de jogo com duas operações . . . . .	67
6.5	Ambiente de jogo com 3 operações sem divisão . . . . .	67
6.6	Ambiente de jogo com 3 operações com divisão . . . . .	68
6.7	Navegação entre Ecrãs . . . . .	70
6.8	Menu Inicial . . . . .	71
6.9	Menu de Registo . . . . .	72
6.10	Menu de Login . . . . .	72
6.11	Menu de Níveis - Início . . . . .	74
6.12	Menu de Níveis . . . . .	74
6.13	Menu de Medalhas . . . . .	75
6.14	Menu de Liderança . . . . .	75
6.15	Menu para selecção de desafiando . . . . .	77
6.16	Menu para selecção de tipo de desafio . . . . .	77
6.17	Menu para inserção de dados . . . . .	78
6.18	Ambiente de Jogo . . . . .	79
6.19	Menu de problemas desafiados ao jogador . . . . .	80
7.1	Vista geral de avaliação - jovens adultos . . . . .	85
7.2	Vista geral de avaliação - público alvo . . . . .	86
7.3	Tempo de cada jogador para superar primeiro nível simples - jovens adultos	87
7.4	Tempo de cada jogador para superar primeiro nível simples - público alvo	88
7.5	Tempo de cada jogador para superar primeiro nível médio - jovens adultos	89

---

7.6	Tempo de cada jogador para superar primeiro nível médio - público alvo . . . . .	89
7.7	Tempo de cada jogador para superar primeiro nível complexo - jovens adultos . . . . .	90
7.8	Tempo de cada jogador para superar primeiro nível complexo - público alvo . . . . .	90
7.9	Tempo de cada jogador para criar um desafio - público alvo . . . . .	92
7.10	Tempo de cada jogador para resolver desafio criado - jovens adultos . . . . .	93
7.11	Tempo de cada jogador para resolver desafio criado - público alvo . . . . .	93
7.12	Tempo de cada jogador para resolver desafio - jovens adultos . . . . .	94
7.13	Tempo de cada jogador para resolver desafio - público alvo . . . . .	95

# Lista de Tabelas

4.1	Requisitos Iniciais de Leitura e Sorteio de Dados . . . . .	35
4.2	Requisitos Iniciais de Movimentação de Botões . . . . .	36
4.3	Requisitos Iniciais de Verificação de Posicionamento de Botões . . . . .	37
4.4	Requisitos Iniciais de Níveis de Dificuldade . . . . .	37
4.5	Requisitos Iniciais de Verificação de Resposta e Níveis de Dificuldade . . .	38
4.6	Requisitos Iniciais de Recolha de Dados . . . . .	38
4.7	Requisitos Iniciais de Efeitos Sonoros e Feedback . . . . .	39
4.8	Requisitos Iniciais de Mérito . . . . .	39
4.9	Requisitos Iniciais de Servidor . . . . .	39
4.10	Requisitos Iniciais de Registo e Autenticação . . . . .	40
4.11	Requisitos Iniciais de Modo Criativo . . . . .	40
4.12	Requisitos Iniciais de Usabilidade . . . . .	41
4.13	Requisitos Iniciais de Fiabilidade . . . . .	41
4.14	Requisitos Iniciais de Segurança . . . . .	42
4.15	Requisitos Iniciais de Portabilidade . . . . .	42
4.16	Requisitos Iniciais de Implementação . . . . .	42
4.17	Requisitos Finais de Leitura e Sorteio de Dados . . . . .	43
4.18	Requisitos Finais de Movimentação de Botões . . . . .	43
4.19	Requisitos Finais de Verificação de Posicionamento de Botões . . . . .	44

---

4.20	Requisitos Finais de Níveis de Dificuldade . . . . .	44
4.21	Requisitos Finais de Verificação de Resposta e Níveis de Dificuldade . . . . .	45
4.22	Requisitos Finais de Recolha de Dados . . . . .	45
4.23	Requisitos Finais de Efeitos Sonoros e Feedback . . . . .	45
4.24	Requisitos Finais de Mérito . . . . .	46
4.25	Requisitos Finais de Registo e Autenticação . . . . .	46
4.26	Requisitos Finais de Servidor . . . . .	47
4.27	Requisitos Finais de Modo de Desafio . . . . .	47
4.28	Requisitos Finais de Usabilidade . . . . .	48
4.29	Requisitos Finais de Fiabilidade . . . . .	48
4.30	Requisitos Finais de Segurança . . . . .	48
4.31	Requisitos Finais de Portabilidade . . . . .	48
4.32	Requisitos Finais de Implementação . . . . .	49
4.33	Requisitos Finais de Restrições . . . . .	49
7.1	Resultados de avaliação de jovens adultos . . . . .	85
7.2	Resultados de avaliação com jovens pertencentes ao público alvo . . . . .	86

# Capítulo 1

## Introdução

Este estágio é realizado no âmbito do projecto de desenvolvimento de um jogo para aprendizagem de polinómios para dispositivos Android. O jogo é do tipo puzzle e baseia-se na resolução de problemas de cálculo de polinómios através da movimentação de operandos e operadores de forma a obter um resultado previamente apresentado ao jogador.

Este projecto está a ser desenvolvido no Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologias da Universidade de Coimbra em parceria com a Faculdade de Psicologia e Ciências da Educação da Universidade de Coimbra no âmbito do programa de mestrado em Engenharia Informática.

### 1.1 Contexto

Quando se pensa em jogos directamente se associa crianças e jovens pois jogar é, na sua grande maioria, a sua actividade favorita e é por eles que esta é mais praticada.

Cada jogo oferece uma experiência a quem o joga e embora esta experiência varie de indivíduo para indivíduo é a ela que estes dão activamente uma resposta consoante o tipo de jogo. Esta experiência é constituinte de um tipo de aprendizagem, obtida através do jogo e que varia com a forma como o jogador interage com este, podendo a aprendizagem ser ou não relevante para o mesmo.

Sabendo que os jogos são uma das componentes da aprendizagem e, sendo esta uma característica inerente aos jovens, a questão reside sobre o tema: como transformar esta experiência numa experiência significativa que não se resume apenas a simples entretenimento mas também a um tipo de entretenimento produtivo, como transformar um

jogo numa ferramenta de aprendizagem significativa em que são obtidos conhecimentos aplicáveis ao Mundo-real sem que o jogo deixe de ser, efectivamente, um jogo?

Os videojogos, num passado recente, têm ganho cada vez mais preponderância no tempo despendido pelos jovens com os mesmos. A facilidade de acesso a tecnologias móveis fez com que estas, nos tempos que correm, façam cada vez mais parte do dia-a-dia dos jovens o que levou a um crescimento exponencial no desenvolvimento de aplicações móveis dos mais diversos tipos. Este crescimento, subseqüentemente, levou também ao aumento da quantidade, diversidade e complexidade dos jogos para estas tecnologias bem como a sua crescente ligação à aprendizagem significativa.

## 1.2 Identificação do Problema

A matemática é uma disciplina da escolaridade obrigatória onde se verifica maior insuficiência de resultados. Sendo esta uma disciplina de aprendizagem progressiva ao longo do percurso académico dos alunos, bases mal consolidadas em algumas partes do mesmo podem levar a uma maior dificuldade de assimilação de novos processos matemáticos. Posteriormente, esta dificuldade de assimilação de novos processos pode levar ao desinteresse e falta de motivação por parte dos alunos em relação à matemática levando estes a criar barreiras à aprendizagem.

## 1.3 Objetivos

Como é descrito no artigo de Cândida Barros e Ana Carvalho[1] para o desenvolvimento da aplicação, pretende-se desenvolver um jogo em plataforma Android[2] que se destina a alunos do 3º ciclo do ensino básico e do ensino secundário, com os conhecimentos básicos de operações com polinómios. O jogo enquadra-se no género puzzle, tendo o jogador que resolver vários quebra-cabeças, de dificuldade crescente. Este género foi selecionado entre diversas possibilidades, uma vez que se pretende que os alunos desenvolvam o seu raciocínio e a sua capacidade de resolver problemas. O controlo do jogo é realizado de forma tátil, em qualquer smartphone que corra na plataforma na qual o jogo está inserido, neste caso Android. Em cada nível apresentado ao aluno, este deve manipular os polinómios apresentados pelo jogo, no sentido literal do termo, usando uma o mais operações elementares (adição, subtração, multiplicação e divisão), tendo como objetivo obter um resultado pré-determinado apresentado pelo jogo.

## 1.4 Motivações

Como é anteriormente descrito, para este trabalho, pretende-se desenvolver um jogo em plataforma Android. Este jogo foi desenvolvido maioritariamente para ajudar a combater o insucesso escolar na disciplina de matemática dos alunos de terceiro ciclo bem como cimentar conhecimentos a nível de operações entre polinómios. É importante que estes conhecimentos sejam bem cimentados pois estas operações são bastante utilizadas nos patamares seguintes da escolaridade obrigatória e possivelmente no ensino superior, se for esse o caso.

O jogo desenvolvido segue uma variação do que são chamados jogos sérios ( “*Serious Games*”) que são jogos destinados a um tipo de aprendizagem significativa aplicável ao Mundo-real. Como existem várias aplicações deste tipo nas mais variadas plataformas, desde aplicações web a aplicações móveis, é necessário ultrapassar metas num sentido global que não sejam atingidas por outras aplicações do mesmo género.

Os jogos em geral, para que sejam considerados um sucesso, devem ser intuitivos, motivadores, dinâmicos, etc. Para que o utilizador sinta prazer ao jogá-los o que, inevitavelmente, o levará a continuar a jogá-los tendo em conta que o objectivo é desenvolver um jogo sério. Pretende-se então que haja um equilíbrio entre entretenimento e aprendizagem para que o utilizador através de entretenimento obtenha uma aprendizagem significativa para o mesmo sem que o jogo seja confundido ou associado a um questionário ou teste.

Desta forma, se o jogo desenvolvido obter resultados que comprovam a eficácia do mesmo poder-se-á ponderar um desenvolvimento mais complexo do jogo de forma a englobar novos tipos de exercícios matemáticos bem como a adição de novas funcionalidades. Este processo levará a um enriquecimento da aplicação e assim ampliar o seu leque de utilizadores alvo e áreas em que esta é aplicável.

### 1.4.1 Utilizador Alvo

O principal utilizador alvo da aplicação são jovens do terceiro ciclo do ensino obrigatório. Sendo esta uma aplicação não só ligada à aprendizagem mas também ao treino e consolidação de processos matemáticos pode ter também como utilizadores alvo estudantes de secundário ou até mesmo do ensino superior pois existe a possibilidade de aumentar ou diminuir a dificuldade do jogo consoante o utilizador alvo. Como é possível variar a dificuldade do jogo a aplicação pode também ser aplicável como um simples processo de verificação cognitiva, através de problemas simples, em idosos para avaliação de velocidade de processos dos mesmos.

### 1.4.2 Desafios a Superar

Existem várias aplicações, especialmente aplicações web, de carácter semelhante o que impõe um alto nível de dificuldade em desenvolver uma aplicação que realmente faça a diferença junto do utilizador alvo.

A usabilidade é também uma característica que impõe um desafio ao desenvolvimento da aplicação. Não só porque é uma característica muito importante no desenvolvimento de qualquer jogo mas também porque se trata de um jogo ligado à matemática, disciplina esta à qual os jovens estudantes criam barreiras devido à falta de interesse pela mesma. É portanto necessário alcançar um elevado nível de usabilidade da aplicação de forma a facilitar a interacção do utilizador com o jogo.

## 1.5 Estrutura do Documento

Nesta secção é apresentada a estrutura do documento.

No Capítulo 2 é apresentado o Estado da Arte que está dividido em 4 secções. Na primeira secção é analisada a aprendizagem através de actividade, interatividade e através de jogos, na segunda secção é feita uma análise detalhada do que são jogos, quais as características fundamentais destes e o processo de desenvolvimento dos mesmos, na terceira secção foi analisada a aprendizagem aplicada aos jogos e finalmente na quarta secção são analisadas algumas aplicações estudadas.

No Capítulo 3 são apresentadas as metodologias para o desenvolvimento da aplicação em que na primeira secção são apresentados os objectivos a serem atingidos, na segunda secção é apresentada a abordagem escolhida para a implementação e validação da aplicação, na terceira secção são enumeradas e justificadas as tecnologias utilizadas, desde as linguagens de programação escolhidas a outras tecnologias que foram consideradas úteis e finalmente na quarta secção é enunciado o plano de trabalho proposto.

No Capítulo 4 são apresentados e analisados os requisitos inicialmente propostos de forma a serem obtidos aqueles que são os requisitos finais. Este capítulo divide-se em duas secções, a primeira secção é constituída pelos requisitos funcionais e não funcionais propostos e subsequente análise de alterações, e a segunda pelos requisitos funcionais e não funcionais finais.

No capítulo 5 é apresentada a arquitectura do sistema. Este capítulo divide-se em quatro secções, a primeira é constituída pela arquitectura conceptual onde se obtém uma visão geral da arquitectura do sistema, a segunda pela arquitectura anteriormente

proposta, a terceira pela arquitectura final e dividida pelas diversas visões que constituem a arquitectura global e, finalmente pelas tecnologias utilizadas e respectivas justificações de escolha.

No capítulo 6 é apresentado o desenvolvimento de prototipagem de todo o processo que se divide em várias secções. Inicialmente é apresentado, num plano temporal, o que foi feito em cada protótipo, genericamente, que é depois analisado, componente a componente, relativamente ao seu processo de desenvolvimento e em que estágios do processo foi desenvolvido. Finalmente são analisados os requisitos que não foram terminados.

No capítulo 7 é apresentado o processo de avaliação. Neste capítulo começa-se por evidenciar que tipo de testes foi feito e porquê para posteriormente serem enumerados todo o processo dos mesmos. Este capítulo divide-se em três secções que são eles o processo, resultados e respectiva análise.

No capítulo 8 são apresentadas as conclusões de todo o processo de estágio e as propostas para o desenvolvimento de trabalho futuro.

## Capítulo 2

# Estado da Arte

Neste capítulo são observadas algumas análises sobre o contexto em que o design de jogos está inserido e quais as suas principais influências. As abordagens e referências apresentadas foram selecionadas de forma a obter uma ideia clara sobre a importância e impacto que o desenvolvimento e design de jogos têm na área da educação e ensino.

### 2.1 Aprendizagem

#### 2.1.1 Aprendizagem por Actividade

Piaget[3], conhecido pela sua teoria do desenvolvimento psicológico de crianças, acreditava no papel da acção no desenvolvimento e noção que as crianças desenvolvem estruturas cognitivas através da acção e actividade espontânea. Hoje em dia pensa-se em como a aproximação construtivista influencia a aprendizagem. Esta aproximação defende que os indivíduos submetidos a aprendizagem têm de activamente construir conhecimento através de experiências que são significantes e são importantes para os mesmos. Os participantes constroem o seu próprio conhecimento através do teste das suas próprias ideias e conceitos baseados em experiências anteriores aplicando-as a uma nova situação. Desta forma integram os seus conhecimentos anteriores com os conhecimentos adquiridos. Esta situação verifica-se nas mais variadas situações do dia-a-dia de cada individuo

### 2.1.2 Aprendizagem por Interactividade

Quando se fala de aprendizagem é comum falar-se de interactividade pois é através da interacção (recepção de informação, interpretação e resposta) que a maior parte desta é aglomerada tanto a nível formal como informal.

Interagir é actuar reciprocamente, entre pessoas, pessoas e máquinas, pessoas e software ou até mesmo entre máquinas. Jogar implica interactividade, seja com um brinquedo, um jogo, uma pessoa ou uma ideia. Segundo o estudo de Salen e Zimmerman[4], a interactividade com jogos está directamente ligada à aprendizagem pois jogar um jogo implica tomar decisões dentro de um sistema desenhado para suportar decisões e respostas significantes, ou seja, cada acção resulta numa mudança que afecta o sistema.

A interacção ocorre a todos os níveis, desde interacção com os objectos que fazem parte do jogo, à interacção social entre os jogadores, à interacção cultural do jogo com os contextos que vão além do seu espaço de jogo. Tal como é referido, de entre os modelos de interactividade de Salen e Zimmerman[4], o modelo de interactividade cognitiva baseia-se na participação psicológica, emocional e intelectual de um participante com o sistema.

### 2.1.3 Aprendizagem Através de Jogos

A aprendizagem através dos jogos engloba os dois tipos de aprendizagem anteriormente referidos mas de uma forma camuflada ao jogador, ou seja, é incutida aprendizagem num ambiente de descontração e divertimento. Em todos os tipos de jogos existe um tipo e/ou nível de aprendizagem daí, tal como Gee[5] referiu, o jogador ser um “*learner*” (aprendiz), ou seja, o jogador está constantemente submetido a algum tipo de aprendizagem ainda que esta possa não ter significância para que seja aplicada no Mundo-real.

Os jogos de uma maneira geral podem ser bastante úteis para a aprendizagem de crianças pelas seguintes razões:

- Estimulam a espontaneidade das crianças/jovens
- Tal como Piaget[3] referiu na sua teoria do construtivismo, interesse, iniciativa, experimentação, descoberta, jogo e imaginação são fundamentais para o desenvolvimento da capacidade das crianças/jovens para aprender
- Jogar pode unir imaginação e intelecto de várias formas e ajudar as crianças a descobrir coisas e/ou obter ensinamentos sem que estes se sintam pressionados a tal

- Sendo os jogos a actividade favorita das crianças/jovens, através destes num ambiente de divertimento e desafio a aprendizagem observa-se de uma forma mais rápida

Tal como Gee[5] descreveu “*todos os aspectos do ambiente de aprendizagem (incluindo as formas em que o domínio semiótico é desenhado e apresentado) estão delineados para encorajar aprendizagem activa e crítica, não passiva*”, logo os jogadores estão em constante aprendizagem activa. Existem vários tipos de aprendizagem através de jogos sendo que para este trabalho foi focado no “*meaningful learning*” (aprendizagem significativa), tarefas significantes em jogos para que os estudantes possam obter uma aprendizagem séria através de tarefas difíceis.

#### 2.1.4 Matemática e Aprendizagem através de Jogos

Como está descrito no artigo de Claudia Groenwald[6] que visa o estudo de jogos aplicados à matemática, o uso de jogos no ensino da mesma tem o objetivo de fazer com que os adolescentes gostem de aprender a disciplina, mudando a rotina de aula de forma a aumentar o interesse do aluno envolvido. A aprendizagem através de jogos, como é o caso do dominó, palavras cruzadas, memória e outros permite que o aluno faça da aprendizagem um processo interessante e até divertido. Para isso, estes devem ser utilizados ocasionalmente para colmatar as lacunas que se produzem na atividade escolar diária. Neste sentido, a investigadora verificou que há três aspectos que por si só justificam a incorporação de jogos nas aulas. São estes: o carácter lúdico, o desenvolvimento de técnicas intelectuais e a formação de relações sociais.

Jogar não é estudar nem trabalhar, porque ao fazê-lo, o aluno aprende, sobretudo, a conhecer e compreender o mundo social que o rodeia. Os jogos são educativos, sendo assim, requerem um plano de acção que permita a aprendizagem de conceitos matemáticos e culturais de uma maneira geral. Já que os jogos em sala de aula são importantes, deve-se ocupar um horário dentro do planeamento, de modo a permitir que o professor possa explorar todo o potencial dos jogos, processos de solução, registos e discussões sobre possíveis caminhos que poderão surgir.

Keith Devlin referiu numa entrevista[7] sobre a sua obra “*Mathematics Education for a New Era: Video Games as a Medium for Learning*”[8] que a matemática é uma actividade, não é algo que se sabe, é algo que se faz ou pratica que vai de encontro ao que foi referido anteriormente na aprendizagem através de actividade. Os estudos de matemática de rua como referiu Keith Devlin demonstraram que quase todas as pessoas se podem tornar pro-eficientes na área da matemática se a procurarem no decorrer de

---

acções relevantes para elas mesmas. Para Keith Devlin, quando devidamente idealizados, os videojogos são simuladores de actividade com sistemas de aprendizagem com valor mais acentuado pois representam um estímulo muito maior aos indivíduos que fazem uso deles. Ao se aplicar videojogos num ambiente escolar de matemática, tendo em conta que quando alguém joga um jogo está a aprender algo, o desafio do educador passa a ser agora fazer com que essa aprendizagem através de jogos seja sobre matemática.

## 2.2 Design de Jogos

Nesta secção são avaliadas as estruturas de jogo no âmbito das características intrínsecas à aplicação a ser desenvolvida.

### 2.2.1 Definindo Jogo

Nesta secção são apresentadas algumas definições de investigadores relativamente ao conceito “Jogo” de forma a se poder observar como este conceito pode variar não só consoante a opinião de cada um dele mas também relativamente ao contexto em que se inserem.

As análises que se seguem foram recolhidas do estudo de Salen e Zimmerman(2003)[4] referente a alguns investigadores, das mais variadas áreas, sobre o termo “jogo”.

#### 2.2.1.1 David Parlett

Para Parlett[4] um jogo pode ser entendido como sendo formal ou informal. Este entendimento é de certa forma também referenciado por outros investigadores pois “*Play*” em inglês pode ser entendido tanto como jogar como brincar em português. Neste caso será apenas observado o tipo de jogo formal que Parlett acredita ter duas vertentes estruturais básicas que são os meios e os fins:

- **Fins:** O facto de um jogo formal ser um desafio com um final como seu objectivo, e como objectivo entende-se ganhar.
- **Meios:** As regras e materiais previamente acordados que podem levar um jogador a ganhar o jogo

Para Parlett, ambas as componentes, a ideia de ganhar e a ideia de o conseguir através de regras, são a chave para a definição de Jogo.

#### 2.2.1.2 Clark C. Abt

A análise de Abt[4] sobre Jogos enfatiza o papel activo do jogador num jogo sendo perceptíveis quatro termos chaves:

- **Actividade:** Um jogo é uma actividade, um processo, um evento

- **Tomada de decisões:** um jogo requer jogadores que tomam decisões activamente
- **Objectivos:** Tal como Parlett, jogos têm objectivos
- **Contexto limitado:** Existem regras que limitam e estruturam a actividade de jogo

### 2.2.1.3 E. Avedon e B. Sutton-Smith

Avedon e Sutton-Smith[4] têm como definição a seguinte frase:

*“Jogos são um exercício de sistemas de controlo voluntários, em que em cada um há competição entre poderes, definidos por regras de forma a produzir um resultado desequilibrado.”*

Os elementos chave desta definição são:

- **Exercício de sistemas de controlo:** implica algum tipo de actividade física ou intelectual
- **Voluntário:** A participação nos jogos é voluntária
- **Competição entre poderes:** jogos implicam um conflito entre jogadores
- **Definido por regras:** A limitação das regras mais uma vez é referenciada
- **Resultado desequilibrado:** O resultado de um jogo é um estado (objectivo) diferente do estado inicial de jogo

### 2.2.1.4 Salen e Zimmerman

Salen e Zimmerman[4], na sua definição de jogo, a qual é referenciada em outras obras estudadas tem como dísia a seguinte frase:

*“Um jogo é um sistema em que um jogador se empenha num conflito artificial, definido por regras, que retorna um resultado quantificável”*

As suas ideias primárias são portanto:

- **Sistema:** Sistemas são fundamentais na aproximação a jogos
- **Jogadores:** Um jogo é algo em que um ou mais jogadores participam activamente.

- **Artificial:** Embora os jogos cada vez mais reflectam a vida real, artificialidade é uma das características que os definem.
- **Conflito:** Todos os jogos implicam uma competição de poderes.
- **Regras:** As regras são a base da estrutura do jogo, estas limitam os que os jogadores podem ou não fazer.
- **Resultado quantificável:** No final de um jogo um jogador perde, ganha ou recebe um resultado numérico.

Depois de uma pequena análise pode-se observar que, mesmo havendo alguns parâmetros que se distinguem de definição para definição, as definições anteriores não estão muito distantes umas das outras sendo que todos concordam que as regras são um elemento fundamental num jogo, sendo estas analisadas na secção seguinte.

### 2.2.2 Regras

As regras[4] e mecânicas[9] de jogos são diferenciadas ainda que sejam conceitos relacionados porque as regras são instruções/limitações das quais o jogador tem conhecimento ou toma conhecimento à medida que obtém experiência de jogo enquanto que as mecânicas de jogo não são “visíveis” ao jogador, estas encontram-se “escondidas” no software onde o jogador não tem nenhuma interface de acesso a estas.

As regras constituem a estrutura de um jogo pois todos os jogos têm regras sendo estas uma das qualidades de jogo. Como referido anteriormente, segundo Salen e Zimmerman(2003)[4] um jogo é um sistema em que um jogador se empenha num conflito artificial, definido por regras, que retorna um resultado quantificável. Os mesmos defendem que existe um leque de regras comum a todos os jogos que são as seguintes:

- As regras limitam as acções dos jogadores
- As regras são explícitas e não ambíguas
- As regras são partilhadas por todos os jogadores
- As regras são fixas
- As regras implicam obrigatoriedade
- As regras são repetíveis

Esta lista de regras permitem manter a estrutura de um jogo sólida de forma a manter a coesão do sistema caso contrário o ambiente de jogo não seria coerente o que tornaria a sua prática pouco atractiva. Por exemplo, se as regras não fossem partilhadas por todos os jogadores implicaria desinteresse por parte de jogadores prejudicados, se as regras não fossem obrigatórias os jogadores poderiam desrespeitar a sua autoridade e fazer batota, entre outros exemplos.

### 2.2.3 Mecânicas de Jogo

Para se obter uma melhor percepção do jogo a ser desenvolvido deve-se primeiro ter a percepção de quais os parâmetros de relevância que tornam um jogo apelativo ao utilizador. Para um jogo ser apelativo este não pode ser frustrante por ser demasiado difícil ou confuso, o que poderá levar o jogador a rapidamente desistir de o jogar. Do lado oposto o jogo também não pode ser demasiado fácil evitando assim o desinteresse do jogador. Existem mecânicas[9] de jogos que nos ajudam a perceber quais os factores a ter em conta de forma a desenvolver um jogo que não contemple nenhum dos extremos anteriormente referenciados.

São em seguida descritas 5 tipos diferentes de mecânicas que se podem encontrar num jogo.

- **Física**

Esta ciência, de movimento e força, está muitas vezes ligada às mecânicas de jogo, a posição onde os elementos são colocados, a forma como se movem e em que direcção, se intersecta ou colide com outros elementos formam o leque de cálculos em muitos jogos.

- **Economia Interna**

A mecânica de transações que envolvem elementos que são colecionados, consumidos ou trocados constituem a economia interna de jogo. Estes elementos são geralmente identificados como recursos e podem tomar várias tipos sendo que, todos eles, geralmente, são quantificáveis.

- **Mecanismos de Progressão**

Em muitos jogos, o design de níveis dita a progressão do jogador no mundo de um jogo. Neste tipo de jogos o progresso do jogador é estritamente controlado por mecanismos que podem ou não se desbloqueados consoante o desempenho do jogador. Estes mecanismos podem ser entendidos por níveis, áreas, items, entre outros.

- **Manobras Táticas**

As mecânicas de jogo podem ter de lidar com o posicionamento num mapa para obter vantagens positivas ou negativas, sendo indispensáveis em jogos de estratégia. Este tipo de mecânica especifica quais as vantagens estratégicas que se podem obter de cada tipo de elemento consoante o seu posicionamento nas várias localizações possíveis.

- **Interação Social**

É cada vez mais comum os jogos terem presente nas suas mecânicas interação social ainda que nem em todos esta seja explícita. Interação social pode tomar várias formas bem como a sua interpretação, desde o chat à forma de jogar num jogo multiplayer são elementos de interação social que fazem um jogador interagir socialmente com outro.

Cada jogo tem as suas mecânicas dependendo do tipo do mesmo, muitos contemplam algumas das anteriormente referidas como outras, o que é importante referir no contexto das mecânicas de jogo prende-se ao facto de que não é por estas serem transparentes ao utilizador que acabam por ser menos importantes. Como anteriormente foi constatado, as regras constituem a limitação consciente que cada jogador tem do jogo, já as mecânicas são basicamente as regras “backend” ou de software, que permitem a coesão estrutural de cada jogo e tem influência directa no jogador sem que este se aperceba disso mesmo.

#### **2.2.4 Design e Desenvolvimento de Jogos**

Depois de devidamente avaliadas as características base de implementação que representam os alicerces de um jogo é agora exposto o design de jogos como um todo, e como este é conseguido.

Segundo Jesse Schell[10], o design de um jogo implica a tomada de muitas decisões, e são essas decisões que vão delinear os contornos de um jogo. Tanto as regras como as mecânicas de jogo são apenas uma dessas decisões e como anteriormente foi referido estas variam de jogo para jogo consoante o que é pretendido, sendo sabido que ambas têm de estar presentes, tal como muitas outras variáveis que serão agora avaliadas e estão presentes na imagem seguinte.

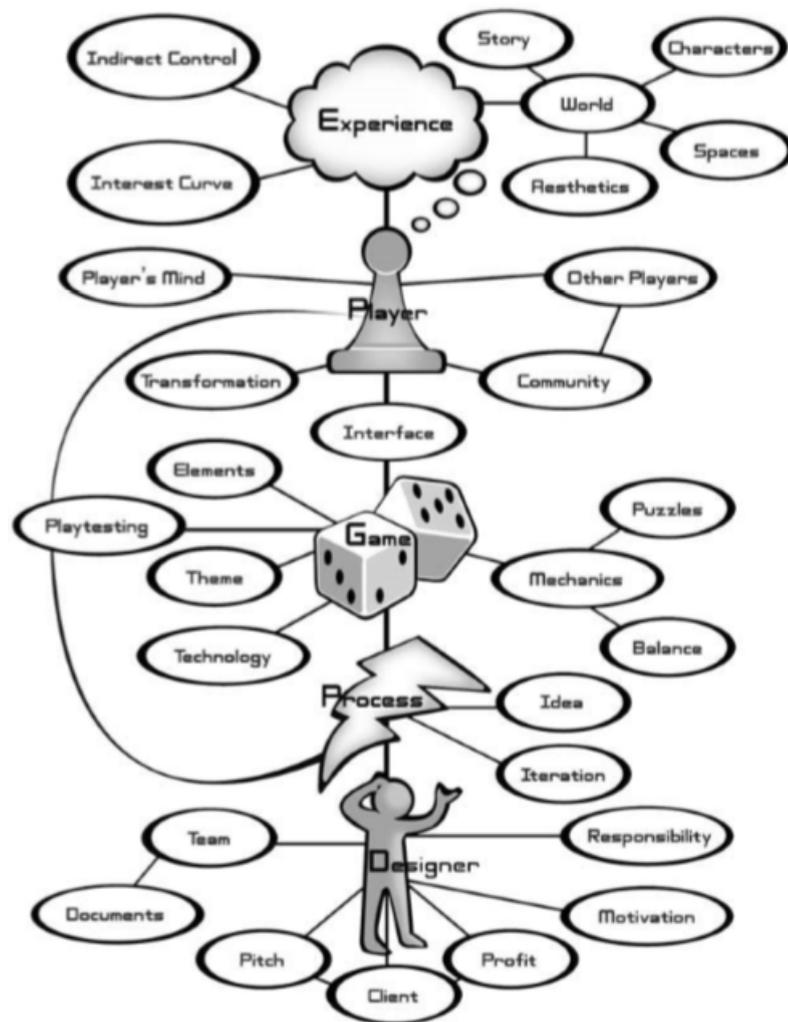


FIGURA 2.1: Esquema de Desenvolvimento de um Jogo segundo Jesse Schell[10]

Como se pode observar na imagem anterior, encontram-se detalhadamente evidenciados os elementos constituintes do processo de desenvolvimento de jogos e as suas características que variam consoante o jogo pretendido. Os 6 elementos que se encontram ligados em série na vertical e expandem os restantes nós sendo os últimos inerentes aos primeiros são os elementos base presentes em todos os videojogos e sem qualquer um deles não é possível conceber um jogo que se possa considerar um sucesso. Esses 6 elementos são constituídos pelo Designer, Processo, Jogo, Interface, Jogador e Experiência, e serão estes os elementos analisados nesta secção devido ao seu grau de importância no desenvolvimento e design de jogos.

### 2.2.4.1 Designer

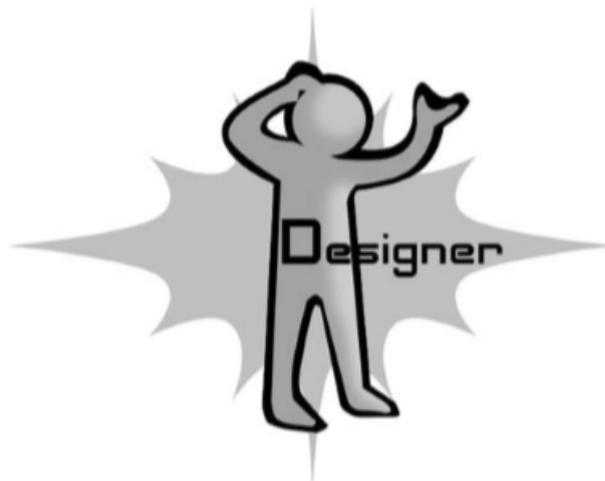


FIGURA 2.2: Primeiro existe um Designer

Como foi referido, o design de jogos é a tomada de muitas decisões e, logicamente, essas decisões recaem sobre o designer. É o designer que idealiza o que o jogo vai ser, como se vai parecer, que sensações vais transmitir, como as vai transmitir, etc, sendo que este tem de ter várias apetências sobre as mais diversas áreas na tomada de decisões. Jesse Schell[10] enumerou várias dessas competências: Animação, antropologia, arquitectura, brainstorming, negócio, cinematografia, comunicação, escrita criativa, economia, engenharia, história, gestão, matemática, música, psicologia, discurso público, design de som, escrita técnica, artes visuais. Além destas apetências existem muitas mais e, logicamente, consoante o jogo a ser desenvolvido algumas são mais importantes que outras ou podem até nem ser necessárias.

Ainda que nenhum designer consiga ter todos os conhecimentos anteriormente referidos, o que é importante referir prende-se ao facto de que um designer deve ter um vasto leque de conhecimentos, os quais vão influenciar o desenvolvimento base do jogo, o seu processo e enquadramentos, qual a sua interface, qual o utilizador alvo e que experiência este vai ter, etc. De todas estas apetências a que Jesse Schell considera mais importante não está ainda nessa lista, sendo esta apetência a capacidade de ouvir. Quando se diz capacidade de ouvir não é apenas ouvir o que é dito, é ouvir profundamente, através de interpretação e observação, ouvir através da mente. Esta apetência divide-se em 5 categorias: Equipa, audiência, jogo, cliente e a si próprio.

- **Equipa:** saber ouvir a sua equipa é fundamental pois é a equipa em conjunto que desenvolve um jogo através de todas as decisões que são tomadas. Tendo em conta que um designer não consegue dominar todas as apetências referidas, é bem

possível que uma equipa de designers consiga portanto a capacidade de ouvir e comunicar podem levar uma equipa a funcionar apenas como “um” designer.

- **Audiência:** É necessário saber ouvir as pessoas que irão jogar o jogo pois se elas não gostarem do jogo este não terá sucesso.
- **Jogo:** Não sendo no sentido literal, saber ouvir o jogo é muito importante, ou seja, conseguir perceber o que está mal só ao “ouvir” o jogo.
- **Cliente:** Um dos problemas de game design passa pela má interpretação do que o cliente quer. Saber ouvir o cliente é muito importante para conseguir perceber o que este realmente quer.
- **Consciência:** Ouvir a si próprio ou, neste caso, ter consciência e confiança nas suas capacidades é a ferramenta responsável por libertar a capacidade criativa de um designer, capacidade esta entendida por muitos como uma das mais importantes.

Agora que são conhecidas as aptências necessárias a um designer será analisado como e onde serão aplicadas essas aptências.

#### 2.2.4.2 Experiência



FIGURA 2.3: O Designer cria uma Experiência

A principal função de um designer de jogos, como muitos o pensam, é criar jogos mas, segundo Jesse Schell, a principal função de um designer de jogos não é a criação de jogos. De facto, o designer cria uma experiência e não jogos, os jogos são meramente um meio para atingir um fim, um artefacto neste caso. Tudo o que temos conhecimento advém de experiências vividas e são essas experiências que moldam o nosso conhecimento e avaliação de cada experiência seguinte. A experiência criada é muito importante pois é

a qualidade dessa experiência que vai ou não manter as pessoas a usufruir dela, ou seja, a jogar um jogo.

Ainda que o jogo esteja directamente ligado à experiência, este apenas permite que o jogador obtenha uma experiência através do mesmo, e para criar experiências é necessário ter conhecimentos que anteriormente foram referidos pois para que se consiga criar boas experiências, que levem as pessoas a usufruir delas e fazer com que continuem a querer fazê-lo mesmo depois de já as terem experienciado. É portanto necessário conhecer a mente humana e conseguir prever as suas reacções através de conhecimentos de psicologia, antropologia e design.

- **Psicologia:** Se se quer criar experiências humanas a psicologia é uma das ciências mais importantes a ter em conta pois ela permite perceber as formas como a mente humana é estimulada e como esta responde aos estímulos.
- **Antropologia:** Se o objectivo passa por estimular pessoas ter um bom conhecimento em relação às suas culturas revela-se também bastante importante. Consoante cada cultura as pessoas respondem de formas diferentes a um mesmo estímulo logo é necessário conseguir perceber qual é o público alvo e quais as variações da sua cultura.
- **Design:** Outra área que é também importante estudar é a área do design pois é através do design que o jogador vai usufruir da experiência a ser transmitida, é através deste que o jogador vai interagir.

O Designer cria a experiência e esta sim surge através do jogo.

### 2.2.4.3 Jogo



FIGURA 2.4: A Experiência nasce através de um Jogo

O objectivo do designer é, como foi referido anteriormente, criar experiências mas este não as consegue manipular directamente, consegue sim manipular o jogo que vai criar a experiência. Já foi analisado anteriormente o que é um jogo a nível mecânico, será agora analisado o que o jogo deve transparecer para criar a experiência.

- **Surpresa:** O jogo deve ser surpreendente. A transmissão de surpresa está ligada ao entretenimento, a sensação de ficar surpreso permite ao jogador usufruir desse mesmo entretenimento de forma a apreciar mais o jogo.
- **Divertimento:** O jogo deve ser divertido, é um dos alicerces de todos os jogos e é esse um dos seus principais objectivos ainda que existam vários tipos de divertimento e como estas são processados pela mente humana.
- **Despertar curiosidade:** A curiosidade motiva o jogador a jogar e mantê-lo a jogar, um jogo que não desperte curiosidade ou que não motiva um jogador a jogá-lo é um jogo inútil.
- **Valor endógeno:** A obtenção de valorização dentro do jogo é também bastante importante, o que motiva o jogador a ganhar reconhecimento à medida que avança no jogo, quais os prémios adquiridos, realização pessoal através do sucesso.
- **Resolução de problemas:** A grande maioria dos jogos implicam resolução de problemas mas esta característica deve estar presente sem que seja interpretada dessa forma pelo jogador. Um jogo é uma actividade de resolução de problemas com uma aproximação de desafio e entretenimento e é através desta actividade que o jogador obtém o valor endógeno referido anteriormente.

De uma forma geral, tal como Jesse Schell[10] resumiu, divertimento é prazer com surpresas, jogar é manipulação que satisfaz curiosidade, um jogo é um objecto com que se joga, um bom jogo é um objecto com o qual se tem divertimento ao jogar, o jogo é uma actividade de resolução de problemas.

#### 2.2.4.4 Processo

O processo engloba muitas áreas distintas tal como a engenharia de software, mitigação de riscos, etc, sendo que nesta secção foram tomados como principais objectivos fazer referência ao jogo como sendo um jogo e não como um programa.

Agora que já há uma percepção do que o jogo deve ter, este agora tem o seu desenvolvimento e esse desenvolvimento começa pela criação de uma ideia. Logicamente o produto final será bastante diferente daquele que foi inicialmente idealizado, o processo

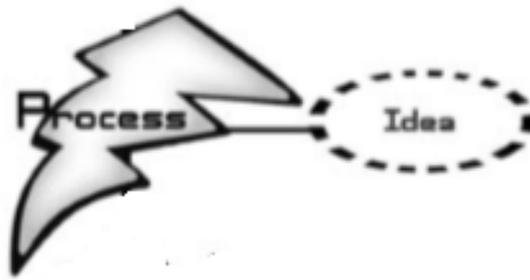


FIGURA 2.5: O Jogo nasce de uma Ideia(Processo)

de desenvolvimento de um jogo começa com uma ideia inicial totalmente diferente da ideia final. O manipular desta ideia provém da constante observação do que pode ou não ser melhorado e é este manipular que vai transmitir a experiência pretendida ao jogador. Para a criação e manipulação da ideia, Jesse Schell[10] indica as *"lens"* de inspiração infinita e para a por em prática é necessário uma mente aberta e muita imaginação pois é através desta prática que se vai oferecer ao jogador todas as experiências que este vai conseguir absorver ao jogar o jogo.

Quase todas as áreas implicam resolução de problemas e o design de jogos não é excepção, inicialmente tem de se identificar o problema a ser resolvido, ou seja, antes de se criar a ideia deve-se primeiro identificar o que há a ser resolvido de forma a estipular um objectivo a alcançar. Estabelecer claramente os objectivos a alcançar oferece três boas vantagens, oferece um espaço criativo mais amplo, ou seja, permite avaliar soluções que caso contrário não seriam ponderadas, oferece uma medida clara da qualidade da ideia proposta e finalmente aquando de desenvolvimento de um jogo em equipa permite uma melhor comunicação entre os elementos da mesma pois todos tem o mesmo conhecimento dos objectivos a serem alcançados facilitando assim o que chamamos de *"brainstorming"*, actividade esta em que todas as ideias são propostas e discutidas de forma a tentar obter um objectivo a alcançar.

Após o *"brainstorming"* onde se obtiveram várias ideias vem o escolher da ideia, o chegar a um consenso de quais os objectivos a alcançar e a criação da imagem do que será muito próximo do produto final pois à medida que o desenvolvimento é feito a ideia está sempre sujeita a alterações. Para a escolha desta ideia, Jesse Schell propôs aquilo que chama de oito filtros:

- **Impulso artístico:** "Sabe bem jogar o jogo? É estimulante?"
- **Demografia:** "Írá o público alvo gostar do jogo?"
- **Design de uma experiência:** "O jogo tem um bom design?"
- **Inovação:** "O jogo tem algo de novo?"

- **Marketing e negócio:** “Será este jogo muito vendido?”
- **Engenharia:** “É tecnologicamente possível desenvolver este jogo?”
- **Social/Comunidade:** “Terá este jogo os objectivos sociais e comunitários pretendidos?”
- **Testes de jogo:** “Os testers gostaram o suficiente do jogo?”

É fácil perceber o que cada um destes filtros significa e Schell[10] defende que um jogo, para ser considerado acabado, tem de passar por todos estes filtros sem que seja necessário uma mudança. Logicamente, dependendo do jogo as variáveis iniciais dos filtros podem ser alteradas à medida que o desenvolvimento do jogo avança sendo que é necessário ter sempre em conta que, dependendo das suas restrições, todos os filtros são superados.

#### 2.2.4.5 Jogador



FIGURA 2.6: O Jogo é feito para um Jogador

É muito importante saber identificar qual o público alvo e como motivar esse mesmo público a jogar e fazê-lo continuamente. Para que isto aconteça é necessário saber avaliar um leque de características que têm de estar presentes no jogo para que este tenha sucesso junto do público para o qual foi desenvolvido.

Segundo Jesse Schell[10], para que esta avaliação aconteça o designer tem de se focar no jogador ao invés de no jogo, saber identificar o que o jogador irá experienciar e saber observar o jogador à medida que este usufrui das suas experiências é muito importante pois permite uma rápida identificação das mecânicas que devem estar inerentes ao jogo. É através desta avaliação e observação que vai permitir ao designer prever o que o jogador vai sentir ao jogar.

O prazer está directamente ligado à experiência que se quer transmitir através de um jogo, saber identificar que tipo de prazer este transmite é outro passo importante para que o jogo satisfaça o jogador. Procurar um tipo de prazer que não seja transmitido na maioria dos jogos é um grande passo para que o jogo resulte pois o jogador gosta de experienciar sentimentos que usualmente não tem a oportunidade de experienciar, esta característica é identificada através da percepção do que se passa na mente do jogador.

#### 2.2.4.6 Interface



FIGURA 2.7: O Jogador interage através de uma Interface

Tudo o que foi anteriormente referido na avaliação de design e desenvolvimento de jogos é transmitido ao jogador através da interface. A cima de tudo esta deve ser intuitiva, o jogador gosta de rapidamente identificar o que tem de fazer num jogo o que envolve a transparência da interface. Uma interface não transparente representa uma parede entre o mundo do jogo e o jogador e dificulta a sua interacção, a interface ideal torna-se transparente ao jogador ao permitir que a sua imaginação fique profundamente submersa no mundo do jogo sem que este se aperceba.

O *feedback* é também uma característica importante de uma interface e este pode tomar várias formas que são: julgamento, recompensa, instrução, encorajamento e desafio. O *feedback* é também responsável pela experiência transmitida ao jogador, tendo em conta que o *feedback* deve ser contínuo e na grande maioria dos casos varia consoante a situação, é uma característica na qual deve ser aplicado grande empenho em todos os momentos do jogo pois irá garantir que o jogo é limpo, desafiante e gratificante.

## 2.3 Design de Jogos e Aprendizagem

Depois de devidamente analisada os tipos de aprendizagem ligada aos jogos e os aspectos a ter em conta para o desenvolvimento e design de um jogo é agora analisado como estes estão interligados e quais os requisitos necessários para que um jogo educacional (Serious Game) seja considerado um “bom jogo” como assim os identifica Gee[5] através dos seus princípios de jogo que serão seguidamente identificados.

### 2.3.1 Princípios de Gee

Na sua obra, “*What video games have to teach us*”, Gee[5] listou 36 princípios da aprendizagem através de vídeo jogos. Apenas alguns serão analisados por referência no documento de design proposto por Cândida Barros e Ana Amélia Carvalho no âmbito do desenvolvimento de um jogo ao qual este documento está associado. Como descrito anteriormente, a ideia de design de um jogo pode variar à medida que este é desenvolvido mas agora serão especificados apenas os princípios presentes na ideia inicial:

- **Princípio do subconjunto:** Este princípio baseia-se na aprendizagem crescente, ou seja, estipula que a aprendizagem se inicia numa forma simplificada do jogo, intuitivamente.
- **Princípio incremental:** Ao respeitar o princípio anterior obrigatoriamente verifica o princípio incremental que estipula que as situações de aprendizagem são ordenadas nos níveis iniciais para que os casos apresentados inicialmente possam lidar a generalizações que possam ser frutíferas para casos seguintes. Quando um jogador enfrenta casos mais complexos numa fase mais avançada de um jogo o tempo de aprendizagem é contragida pelos tipos de generalizações frutíferas que o jogador interiorizou em fases iniciais.
- **Princípio da Inteligência Material:** O pensamento, resolução de problemas e conhecimentos são “guardados” em objectos materiais e ambiente do jogo, desta forma a mente do jogador é libertada, ou seja, permite ao jogador uma melhor concentração de forma a focar-se mais no seu raciocínio e como combina os objectos através do conhecimento já guardado nos mesmos.
- **Princípio do Regime de Competência:** Este princípio estipula que um jogador sinta que o jogo é desafiante mas não impossível de forma a não gerar desinteresse.
- **Princípio Multimodal:** Significado e conhecimento são construídos através de várias modalidades (imagens, texto, símbolos, etc), não apenas palavras.

- **Princípio da Amplificação da Entrada:** De um pequeno input provêm vários outputs.
- **Princípio da Moratória Psico-social:** O jogador pode tomar riscos num espaço em que as consequências do mundo real são minimizadas
- **Princípio da Prática:** O jogador pode praticar várias vezes num contexto em que a prática não é entediante.
- **Princípio da Experimentação:** Princípio que defende que o jogador repita o ciclo de colocar, testar, aceitar ou refutar hipóteses até resolver o problema.
- **Princípio do Conhecimento Intuitivo:** O conhecimento intuitivo é construído através da prática e experiência que são repetidas à medida que o jogador joga e nem sempre de uma forma consciente.
- **Princípio do Iniciado:** Este princípio defende que o jogador não é apenas um consumidor mas também um produtor, capaz de customizar a experiência de aprendizagem através da experiência de jogo.

Foram estes os princípios de Gee identificados que estarão presentes naquele que se pensa ser o produto final e será posteriormente analisado se realmente se verificam.

### 2.3.2 Serious Games

O que são "*Serious Games*"?

O termo "*Serious Games*" apareceu inicialmente no livro "*Serious Games*" de Clark C. Abt[11]. Inicialmente quando este tipo de jogo foi estudado os vídeo jogos não eram muito comuns ainda que a definição de Abt ainda hoje se aplica à era digital.

"*Serious Games*"[12] como vídeo jogos são jogos tecnologicamente desenvolvidos da mesma forma que outros jogos mas não são desenvolvido primariamente ou exclusivamente para o divertimento do jogador. Para este tipo de jogo é mais importante ter um balanço entre aspectos de jogo que representam o factor de divertimento e os aspectos que representam o factor educacional. São neste caso jogos considerados educacionais pois o jogador que os joga deve aprender algo que possa aplicar em situações reais daí o elevado valor da experiência de aprendizagem presente nos mesmos.

De uma forma mais resumida, como referiu Zyda[13], "*Serious Games*" representam um desafio mental, jogado num computador de acordo com regras específicas que usa o entretenimento para posteriormente alcançar objectivos de treino, educação, saúde, política pública e estratégia de comunicação.

Existem vários tipos de *"Serious Games"* e cada um desses tipos foca-se num determinado objectivo para o qual o jogo é desenvolvido. O caso aqui estudado é o tipo de jogo de *"edutainment"* que traduzido significa actividade ou programa que combina elementos educacionais e de entretenimento, de *"Game-based Learning"* ou aprendizagem baseada em jogos, e jogos de treino e simulação que visam a ganhar de prática de um dado exercício[14]. Os três, ainda que de uma forma um pouco diferente, focam-se no mesmo tipo de público alvo e aprendizagem.

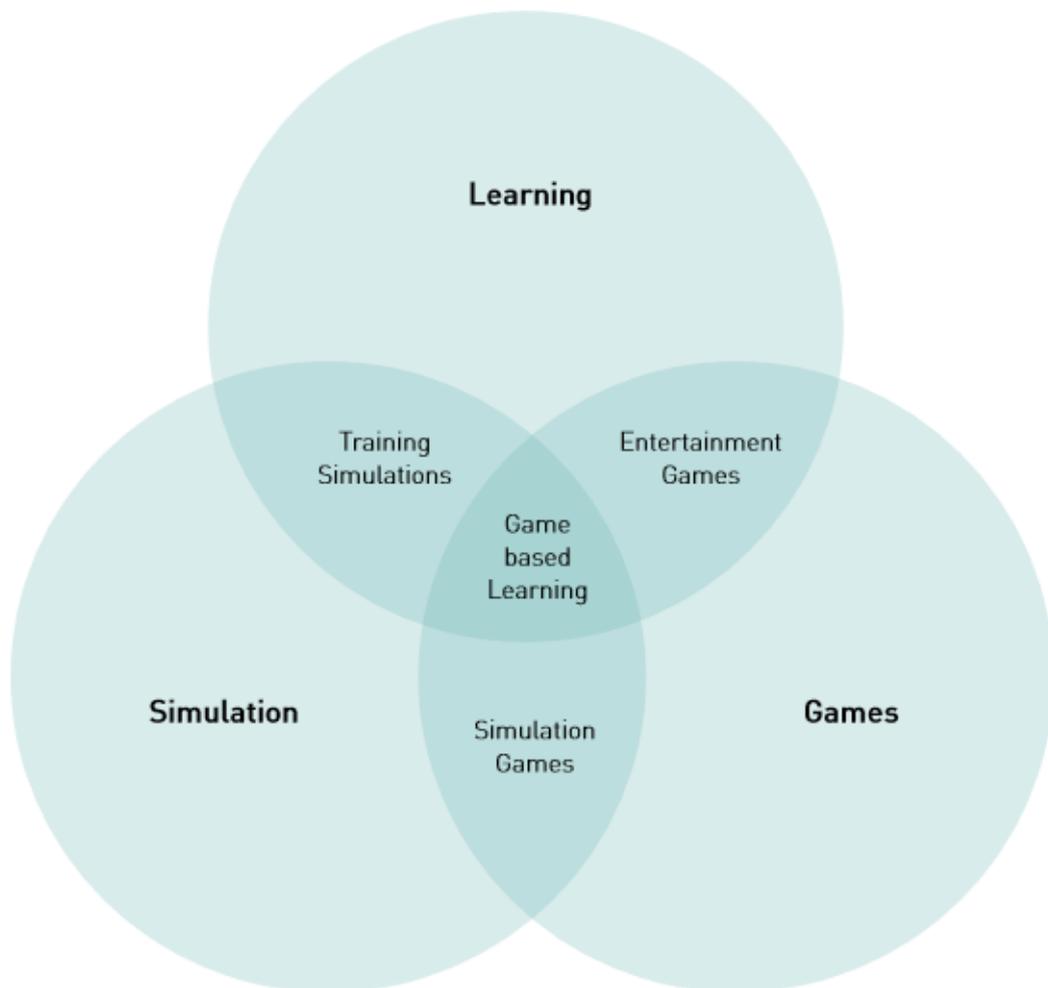


FIGURA 2.8: Interação entre Aprendizagem, Jogos e Simulação[13]

Para uma melhor percepção do que é um *"Serious Games"* serão analisadas algumas aplicações deste tipo na secção seguinte de forma a ser estabelecido um grupo de estudo para uma melhor compreensão dos objectivos a serem estabelecidos e alcançados.

## 2.4 Análise de Aplicações Existentes

Nesta secção serão revistas algumas das aplicações já existentes. A escolha das aplicações estudadas foi feita tendo em conta a aplicação que se pretende desenvolver tendo sido procuradas, no vasto leque de jogos, aquelas que partilham algumas das características que se entende serem características constituintes daquela que será a aplicação final a ser desenvolvida.

- **Math Puzzles for Toddlers**

O "*Math Puzzle for Toddlers*"[15] é um jogo para Android também do tipo puzzle em que, tal como na aplicação que se pretende desenvolver, o jogador interage através da movimentação de botões de forma a superar um exercício dado. A cada exercício proposto é apresentado ao jogador uma conta à qual o jogador tem quatro hipóteses possíveis de resposta. Se o jogador movimentar a opção correcta para a zona de resposta o exercício é superado e o jogo apresenta um novo exercício ao jogador sem que este obtenha qualquer tipo de prémio por resposta ou aglomeração de respostas correctas não existindo qualquer vertente de superação ou gratificação. Caso o jogador movimente para a zona de resposta uma opção errada, esta é devolvida à sua posição de origem e é dada ao jogador uma nova tentativa para resolver o exercício sem que haja número limite de respostas erradas por exercício. Em termos de dificuldade o jogo não apresenta um padrão coerente, ou seja, não há aparente progressão na mesma pois os exercícios são apenas sorteados aleatoriamente e não por graus de dificuldade. Pode-se concluir que este jogo partilha algumas das regras e mecânicas do jogo que se pretende desenvolver.

- **Math Maniac**

O "*Math Maniac*"[16] é um jogo Android do tipo puzzle. Neste jogo é apresentado um tabuleiro de números e um resultado no canto inferior esquerdo em que o jogador, através da adição e sem limite de números escolhidos, tem de chegar ao resultado. A cada selecção de botões correcta os botões escolhidos desaparecem e é apresentado um novo resultado para escolha de novos números até o tabuleiro ficar vazio de forma a completar o nível. O jogo tem tempo limite para cada selecção e a cada resposta correcta o jogador é premiado com pontos, ou seja, está presente a vertente gratificante de jogo e, como é composto por níveis este tem também a vertente de superação. Ainda que em termos de jogabilidade sejam um pouco diferentes este jogo apresenta várias características e mecânicas comuns à aplicação que se pretende desenvolver.

- **King of Math**

O "*King of Math*"[17] tal como o Math Puzzle for Toddlers[15] é um jogo para android do tipo puzzle em que são apresentados exercícios, neste caso contas, em que o jogador tem 4 opções em que uma é a correcta para responder. Neste jogo o jogador não arrasta o botão mas limita-se simplesmente a escolhe-lo. Ao contrário do "*Math Puzzle for Toddlers*" o "*King of Math*" é dividido por categorias de cálculo (adição, subtração,etc) e cada uma destas categorias tem diferentes níveis de dificuldade em que são atribuídas estrelas mediante o número de respostas correctas e uma pontuação consoante o desempenho do jogador num dado intervalo de tempo. Este jogo tem portanto, além de estar também incluída na área da resolução através de processos matemáticos, algumas características comuns à aplicação que se pretende desenvolver, sendo estas relacionadas com a vertente gratificante por premiação através da conclusão com sucesso de exercícios bem como a vertente de superação obtida através da subida de níveis e constante subida de dificuldade.

- **Tricky Maths**

O "*Tricky Maths*"[18] é um jogo muito semelhante ao "*King of Math*" mas tem mais opções de jogo, sendo que as que se consideram relevantes, tendo em conta a aplicação que se pretende desenvolver, o modo de jogo em que tal como o "*King of Math*" é apresentada uma conta e o jogador escolhe a opção correcta e o modo de jogo em que são apresentados os operandos e o resultado e o jogador escolhe qual o operador correcto sendo que a aplicação a desenvolver usa os dois em simultâneo.

Não constam mais análises de aplicações na avaliação anterior pois, do vasto leque de jogos testados, a maior parte apresenta características muito semelhantes às anteriormente referidas o que não justifica constante repetição de análises.

## Capítulo 3

# Metodologia

Depois de analisado o estado da arte em relação ao design de jogos e videojogos ligados à aprendizagem é agora descrita a metodologia utilizada para o desenvolvimento da aplicação.

### 3.1 Objectivos

A nível da metodologia pretende-se o desenvolvimento de um jogo que contenha um determinado número de funcionalidades que se consideram indispensáveis ao sucesso do mesmo. Este sucesso só será obtido se se provar que o jogo não só funciona correctamente e vai de encontro às funcionalidades esperadas, mas também respeite, a nível de design, utilidade e fluidez que permita ao jogador obter uma experiência agradável e significativa para o mesmo. Para atingir estes objectivos é necessário que o desenvolvimento seja feito por estágios que serão seguidamente justificados.

Como cada jogador interage de forma diferente com um jogo e cada um obtém também a sua própria experiência, logo é necessário que o desenvolvimento da aplicação seja um tipo de desenvolvimento que permita a realização de testes durante todo o seu percurso de maturação. Desta forma é permitido que, atempadamente, se possa fazer alterações para mitigar ou corrigir erros que poderiam não estar previstos. A necessidade desta contínua bateria de testes permite também a adição de funcionalidades que não estavam inicialmente listadas, como requisitos que se considerem viáveis para enriquecer a aplicação.

É também necessário que o desenvolvimento possa ser continuamente testado pois quando falamos de jogos é necessária uma opinião "*outside the box*" que possa identificar erros,

e até mesmo oferecer soluções para melhorias que possam ter sido invisíveis para o designer. Como se pretende que o jogo, o mais generalizadamente possível, se adapte da melhor forma àquele que é o seu público alvo, é de extrema importância que seja frequentemente testado durante todo o processo de desenvolvimento.

Para que seja possível realizar testes durante todo o processo é necessário definir uma priorização de requisitos de forma a que a aplicação, ainda não estando terminada, permita que estes sejam realizados. É necessário dar prioridade aos requisitos mais importantes, neste caso os requisitos funcionais e estruturais e só após estes estarem desenvolvidos se passou ao desenvolvimento dos demais.

## 3.2 Abordagem

Para atingir os objectivos estipulados na metodologia de desenvolvimento da aplicação foi utilizada uma abordagem de prototipagem evolutiva com desenvolvimento iterativo e incremental[19].

A prototipagem evolutiva consiste na concepção de um sistema que emprega diversos ciclos de projecto, implementação e avaliação até à aceitação final do software, ou seja, cada novo protótipo confirma, refina ou adiciona novos requisitos até ser atingido a totalidade dos requisitos do sistema. Este tipo de prototipagem consiste nos seguintes quatro passos: Concepção do que será o software, projecto e implementação do protótipo inicial, refinamento do protótipo até à sua aceitação e finalmente conclusão e entrega do produto final. Este modelo é adoptado no contexto em que os requisitos são instáveis e alguns podem ainda ser desconhecidos e poderão ser propostos e adicionados à medida que o protótipo é desenvolvido. Pode também haver a necessidade de teste de algoritmos antes da sua aceitação para posteriormente, se esta não for aceite, se proceder à sua substituição. Este tipo de prototipagem é mais vantajoso para este caso pois permite a diminuição da taxa de defeitos através de uma melhor definição das especificações do sistema, promove maior participação do cliente, permite maior visibilidade do progresso de desenvolvimento e é possível verificar a aceitação do sistema nos primeiros estágios do desenvolvimento.

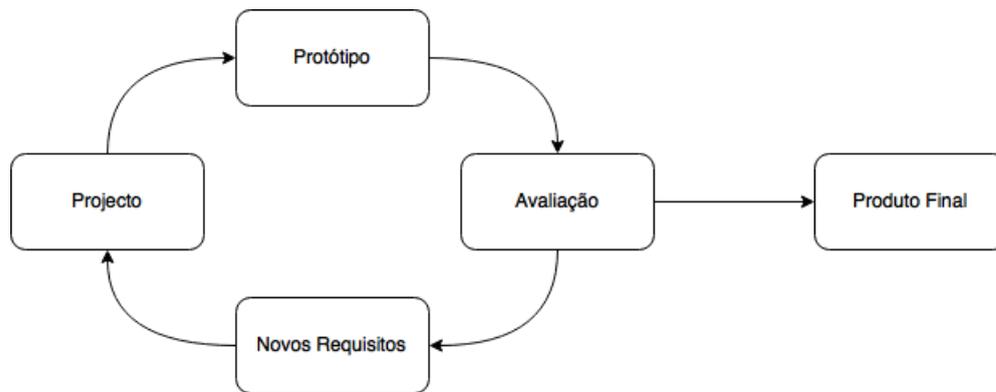


FIGURA 3.1: Prototipagem Evolutiva

O desenvolvimento foi iterativo e incremental pois este é feito, como foi referido anteriormente, por estágios, ou seja, os requisitos mais importantes ou requisitos que representam a estrutura do que será a aplicação foram implementados primeiro sendo que os demais foram acrescentados nas versões seguintes, desta forma o desenvolvimento ocorre gradualmente até à sua conclusão. Às vantagens anteriormente referidas, este tipo de desenvolvimento acrescenta a redução de riscos, detecção de problemas logo no início do desenvolvimento e auxilia a estimativa de tempo de desenvolvimento da aplicação.

### 3.3 Responsabilidades da Equipa

Tendo em conta que o design conceptual do jogo já estava feito, é necessário haver uma percepção do papel dos elementos que compõe a equipa que serão agora apresentados:

- **Investigadora**

Como está evidenciado no artigo da investigadora[1] esta é responsável pelo design do jogo. O design do jogo engloba as mecânicas de ambiente de jogo, as mecânicas de mérito e progressão através de níveis. No artigo é também referenciado o modo criativo mas com muito pouco detalhe.

- **Estagiário**

O estagiário é responsável por toda a implementação da aplicação. Este desenvolvimento divide-se em diferentes áreas:

• **O jogo** em si, baseado na ideia da investigadora

• **O servidor**, desenvolvido e idealizado pelo estagiário, e todas as funcionalidades necessárias para manter os dados persistentes, constantemente actualizados e de fácil acesso para a aplicação ser o mais coerente possível

**Toda a navegação de ecrãs** e como o jogador navega através dos mesmos, sempre com o objectivo de os manter o mais simples e intuitivos possível

O estagiário foi também responsável pela inserção da tabela de classificação na aplicação em si.

- **Equipa**

Todas as reuniões de atribuição de requisitos foram realizadas com toda a equipa presente. Estes requisitos foram discutidos por todos os elementos da mesma logo, todos os elementos são directamente responsáveis pelo produto final.

### 3.4 Plano de Trabalho

Para se poder efectuar uma gestão mais correcta de todas as metas que fazem parte do processo de estágio foi elaborado um diagrama de Gantt com o planeamento a ser dividido em duas fases (uma por cada semestre do projecto). Esta tarefa pretende organizar todos os passos do desenvolvimento assim como definir prazos para a sua conclusão.

#### 3.4.1 Primeiro Semestre

A nível de planeamento, para o primeiro semestre, como não foi efectuado o design do jogo pois este já tinha sido feito pela investigadora, procedeu-se directamente para a implementação do primeiro protótipo. Os requisitos proposto para este protótipo foram alcançados dentro dos prazos previstos, e posteriormente passou-se à realização do relatório intermédio. Para o primeiro semestre o plano idealizado foi cumprido não havendo desvios a assinalar.

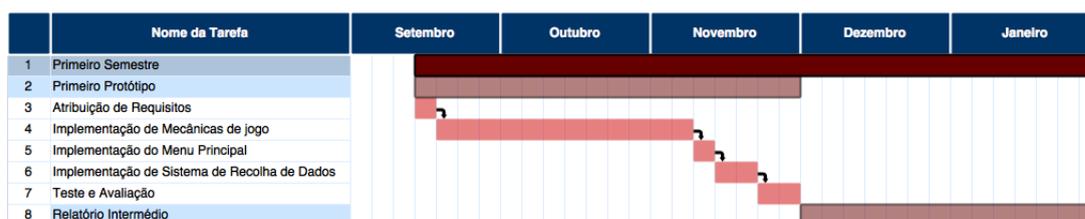


FIGURA 3.2: Plano para o primeiro semestre

### 3.4.2 Segundo Semestre

No documento intermédio foi apresentado um diagrama de Gantt com o planeamento daquilo que estava previsto ser realizado no segundo semestre e respectivas datas de implementação. Como durante o processo de desenvolvimento foi necessário alterar a prioridade de implementação de algumas funcionalidades foi necessário também mudar este plano. A nível estrutural o plano manteve-se, ou seja, foram desenvolvidos mais dois protótipo tal como se tinha planeado, o que diferiu a nível de planeamento foi o que seria implementado em cada protótipo bem como a sua duração de implementação.

Plano inicial apresentado no relatório intermédio:



FIGURA 3.3: Plano inicial para o segundo semestre

Plano final e real do segundo semestre:

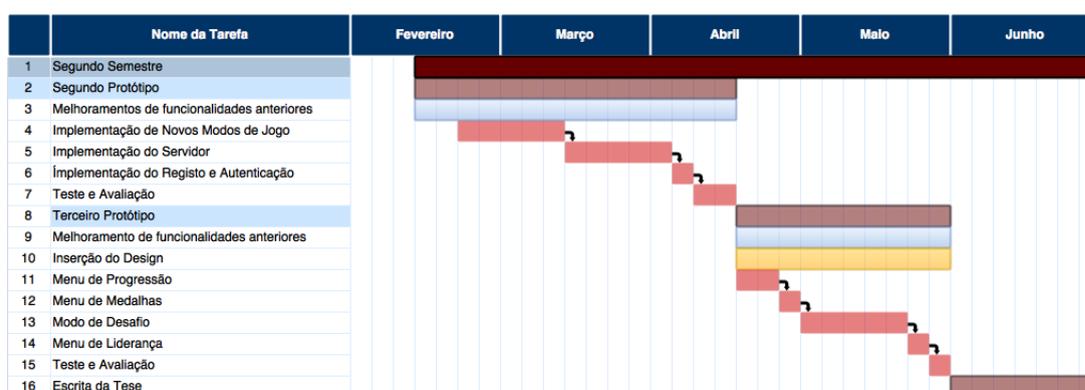


FIGURA 3.4: Plano final para o segundo semestre

Como se pode observar, temporalmente houve um pequeno desvio no terminar do segundo protótipo o que levou ao atraso do início do terceiro. Esta alteração deve-se,

em grande parte, à implementação do servidor que demorou mais tempo que o previsto devido a um conflito de tecnologias que será posteriormente explicado neste documento.

Houve também a adição da tarefa de melhoramentos nos dois protótipo, pois à medida que os testes foram sendo realizados houve necessidade de estes melhoramentos serem feitos à medida que foram identificados. Não foram especificados os melhoramentos porque são pequenos detalhes, que espalhados durante todo o desenvolvimento acabaram por roubar bastante tempo. Estes melhoramentos também são constituídos por alguns requisitos que não estavam previstos e foram posteriormente adicionados.

O terceiro protótipo conta também com a inserção de design que foi feita durante todo o processo à medida que as funcionalidades foram implementadas e não estava presente, como se pode verificar, no planeamento inicial.

## Capítulo 4

# Análise de Requisitos

Neste Capítulo serão analisados os requisitos discutidos em reunião.

A análise de requisitos foi baseada na metodologia descrita no livro “*Software Engineering*” de Ian Sommerville[20] que divide os mesmos em duas categorias:

- **Requisitos Funcionais**

Estabelecimento de serviços que o sistema deve oferecer, como o sistema deve reagir a diferentes inputs, e como o sistema se deve comportar em diferentes situações. Em alguns casos, os requisitos funcionais podem explicitamente estabelecer o que o sistema não deve fazer.

- **Requisitos não Funcionais**

São as restrições dos serviços ou funções oferecidas pelo sistema. Incluem restrições temporais, restrições no processo de desenvolvimento e restrições impostas por padrões. Os requisitos não funcionais frequentemente se aplicam ao sistema como um todo invés de serviços ou características do sistema individuais.

Para atribuir a prioridade de cada requisito recorreu-se ao uso do método de MosCoW[21] em que cada requisito pode ser classificado com uma de quatro prioridades:

- **M - MUST**

Descreve um requisito que tem de ser satisfeito na solução final para que a solução seja considerada um sucesso.

- **S - SHOULD**

Representa um item de alta prioridade que deve ser incluído na solução final se possível. Este é geralmente um requisito crítico mas um requisito que pode ser satisfeito de diferentes maneiras se necessário.

- **C - COULD**

Descreve um requisito que é considerado como desejado mas não necessário. Este tipo de requisito será incluído se houver tempo e recursos.

- **W - WOULD**

Representa um requisito em que foi acordado não ser implementado numa dada versão mas poderá ser implementado numa versão futura.

## 4.1 Requisitos Iniciais

Foram apresentados no documento intermédio aqueles que eram os requisitos propostos para o desenvolvimento da aplicação. Através da metodologia de prototipagem evolutiva e respectivas avaliações sentiu-se a necessidade de alterar a lista de requisitos bem como a sua importância e prioridade. São agora apresentados os requisitos iniciais com a explicação das respectivas alterações.

Procedeu-se também à limpeza daquilo que eram opções de design.

### 4.1.1 Requisitos Funcionais

<b>L-1</b>	<b>Leitura e Sorteio de Dados</b>	<b>Categoria</b>
L-1.1	A aplicação tem uma base de dados independente do servidor para jogo offline	M
L-1.2	Em jogo online o sistema lê a base de dados do servidor	M
L-1.3	O sistema tem de ler problemas da base de dados para todos os tipos de estruturas de exercícios	M
L-1.4	O sistema tem de sortear problemas para cada nível consoante o tipo de estrutura de exercícios associado	M
L-1.5	O sistema tem de sortear os operandos e operadores pelos botões associados a cada um para todos os tipos de estruturas de exercícios	M

TABELA 4.1: Requisitos Iniciais de Leitura e Sorteio de Dados

Para aceder à aplicação é agora necessário autenticação, implicando uma conexão à internet, logo o primeiro requisito foi alterado. Após login o jogador consegue efectivamente jogar não estando conectado à internet mas o requisito em si está mal especificado.

Tal como o requisito anterior também o segundo requisito foi alvo de alterações. O jogo lê sim uma base de dados de problemas do servidor, mas apenas aqueles que correspondem a desafios já que os restantes problemas já se encontram guardados localmente na aplicação.

Como no primeiro protótipo a verificação de resposta era feita através do posicionamento do botão com informação para resposta correcta, as posições dos operadores e operandos eram sorteados pois a aplicação lia primeiro o operador e os operandos correctos e sem este sorteio estes seriam os primeiros a ser posicionados. Este requisito foi removido pois como a verificação é feita através de cálculo não existem operadores e operandos identificados como correctos.

<b>M-2</b>	<b>Movimentação de Botões</b>	<b>Categoria</b>
M-2.1	O utilizador pode mover operandos independentemente da sua posição	M
M-2.2	O utilizador pode mover operadores independentemente da sua posição	M
M-2.3	O utilizador não pode mover o botão correspondente à resposta final	M
M-2.4	O utilizador ao mover um botão para uma posição de resposta é criado um novo botão igual e na posição de origem do botão movimentado	S
M-2.5	O utilizador pode mover botões criados	S

TABELA 4.2: Requisitos Iniciais de Movimentação de Botões

Estes requisitos sofreram uma pequena alteração, nomeadamente na prioridade dos dois últimos requisitos pois foi identificado que seria de extrema importância que fosse possível repetir botões se forma a aumentar o leque de soluções possível para um dado problema. A prioridade foi portanto alterada de S(SHOULD) para M(MUST).

<b>V-3</b>	<b>Verificação de Posicionamento de Botões</b>	<b>Categoria</b>
V-3.1	A cada movimento de botão o sistema verifica se este é posicionado numa área de resposta correspondente ao tipo de botão	M
V-3.2	Se um botão é posicionado numa área de resposta correspondente e esta estiver vazia guarda o botão em posição de resposta	M
V-3.3	Se um botão é posicionado numa área de resposta correspondente e a posição de resposta estiver ocupada o botão é devolvido à sua posição de origem	M
V-3.4	A cada posicionamento de botão em posição de resposta é verificado se todas as posições de resposta estão ocupadas	M
V-3.5	Se todas as posições de resposta se encontram ocupadas o sistema procede à verificação da solução	M

TABELA 4.3: Requisitos Iniciais de Verificação de Posicionamento de Botões

Estes requisitos não sofreram qualquer alteração.

<b>N-4</b>	<b>Níveis de Dificuldade</b>	<b>Categoria</b>
N-4.1	O jogo é composto por níveis de dificuldade	M
N-4.2	Os níveis são superados por acumulação de respostas correctas	M
N-4.3	Cada nível tem um tipo de estrutura de exercício correspondente	M
N-4.4	Cada nível tem apenas um tipo de estrutura de exercício associado	M
N-4.5	Cada tipo de estrutura de exercício pode estar associado a um ou mais níveis de dificuldade	M

TABELA 4.4: Requisitos Iniciais de Níveis de Dificuldade

A investigadora entendeu que para passar de nível apenas seria necessário resolver um exercício desse mesmo nível o que levou à remoção do segundo requisito.

<b>VR-5</b>	<b>Verificação de Respostas e subida de Nível de Dificuldade</b>	<b>Categoria</b>
VR-5.1	A verificação de resposta é feita através de cálculos matemáticos	M
VR-5.2	O utilizador tem 3 tentativas para resolver cada exercício	M
VR-5.3	Se o utilizador responder erradamente a um exercício é lhe apresentado o mesmo exercício para uma nova tentativa	M
VR-5.4	Se o utilizador responder erradamente por três vezes consecutivas a um exercício é lhe sorteado um novo exercício do mesmo nível para resolver	M
VR-5.5	Se o utilizador responder correctamente a um exercício é lhe apresentado um novo exercício do mesmo nível para resolver	M
VR-5.6	Se o utilizador responder correctamente a três exercícios do mesmo nível sobe de nível de dificuldade	M

TABELA 4.5: Requisitos Iniciais de Verificação de Resposta e Níveis de Dificuldade

Tal como referido na secção de requisitos anterior, a investigadora entendeu que para passar de nível apenas seria necessário resolver um exercício desse mesmo nível o que levou à alteração do penúltimo requisito e remoção do último e adição de um novo requisito. Este último consiste na possibilidade de um jogador jogar um nível que não seja o último nível desbloqueado onde este se encontra, e possa jogar mais do que um problema sem ser constantemente redireccionado para o menu de níveis.

<b>R-6</b>	<b>Recolha de Dados</b>	<b>Categoria</b>
R-6.1	A cada verificação de resposta o sistema guarda todos os dados referentes ao exercício respondido	M
R-6.2	Os dados são guardados em ficheiros TXT distintos consoante o tipo de estrutura de exercício	M
R-6.3	Os dados são transferidos para documentos Excel distintos consoante o tipo de estrutura de exercício para tratamento dos mesmos	M
R-6.4	Os dados recolhidos são enviados para o servidor se a aplicação estiver ligada a este	M
R-6.5	Caso a aplicação não se encontrar ligada ao servidor os dados são guardados e enviados assim que possível para este	M

TABELA 4.6: Requisitos Iniciais de Recolha de Dados

Como os dados de cada resposta são directamente enviados, a cada resposta dada, para o servidor, esta tabela de requisitos sofreu várias alterações pois já não há necessidade

de guardar dados localmente.

<b>E-7</b>	<b>Efeitos Sonoros e Feedback</b>	<b>Categoria</b>
E-7.1	Melodia de fundo sempre que a aplicação se encontrar em funcionamento	M
E-7.2	Alerta com som de notificação para resposta correcta	M
E-7.3	Alerta com som de notificação para resposta errada	M
E-7.4	Alerta com som de notificação para subida de nível	C
E-7.5	Alerta com som de notificação para jogo concluído	C

TABELA 4.7: Requisitos Iniciais de Efeitos Sonoros e Feedback

Através de avaliação e discussão entendeu-se que os dois últimos requisitos não são necessários. Chegou-se também à conclusão que seria necessário haver feedback visual daí terem sido adicionados novos requisitos a esta tabela.

<b>Me-8</b>	<b>Mérito</b>	<b>Categoria</b>
Me-8.1	Atribuição de medalhas por acumulação de respostas correctas	M
Me-8.2	Atribuição de medalhas por acumulação de respostas correctas consecutivas	M
Me-8.3	Tabela de classificação por mérito de utilizadores	C

TABELA 4.8: Requisitos Iniciais de Mérito

Na proposta apresentada pela investigadora está presente que haveria sistema de mérito, nomeadamente medalhas, mas não estava definido quantas e quais seriam. Após reunião de atribuição de requisitos foram definidas quais as medalhas que o jogador poderia conquistar, requisitos estes que foram adicionados aos requisitos finais.

<b>S-9</b>	<b>Servidor</b>	<b>Categoria</b>
S-9.1	O servidor tem uma base de dados para todos os tipo de estrutura de exercícios	M
S-9.2	O servidor tem uma base de dados com a informação de registo dos utilizadores	S
S-9.3	O servidor recebe dados da aplicação	M
S-9.4	O servidor envia dados via email para o investigador num prazo temporal	M

TABELA 4.9: Requisitos Iniciais de Servidor

Através da evolução do processo de desenvolvimento e atribuição de requisitos foram sendo especificadas certas funcionalidades que assim não se encontravam. Esta especificação de requisitos e funcionalidades levou a um maior número de requisitos a nível do servidor que estão presentes nos requisitos finais.

<b>RA-10</b>	<b>Registo e Autenticação</b>	<b>Categoria</b>
RA-10.1	O utilizador pode registar-se no servidor	S
RA-10.2	O utilizador pode autenticar-se com o servidor	S

TABELA 4.10: Requisitos Iniciais de Registo e Autenticação

Verificou-se que era necessário haver um registo de utilizador para que fosse possível guardar dados persistentes associados ao mesmo que são utilizados em toda a aplicação consoante a tarefa que o jogador executar. Foi portanto alterada a prioridade destes requisitos.

<b>MC-11</b>	<b>Modo Criativo</b>	<b>Categoria</b>
MC-11.1	O utilizador pode criar exercícios para qualquer tipo de estrutura de dados de exercícios	M
MC-11.2	Para cada exercício criado o sistema verifica se este tem solução	M
MC-11.3	Se tiver solução o exercício criado é guardado numa base de dados à parte dos restantes exercícios já presentes na aplicação e fornecidos pelo investigador consoante o seu tipo de estrutura de exercício	M
MC-11.4	O sistema fornece ao utilizador um identificador do exercício que criou	M
MC-11.5	O utilizador poderá escolher resolver exercícios específicos através da inserção de identificador na interface inicial	M

TABELA 4.11: Requisitos Iniciais de Modo Criativo

A funcionalidade do modo criativo sofreu alterações consideráveis pelo que quase todos os requisitos sofreram alterações.

Depois de ponderação entendeu-se que o jogador ao criar um desafio deveria ser este a apresentar a sua solução para garantir que o jogador cria o desafio com consciência de cálculo e resolução o que levou à alteração do segundo requisito desta tabela. Como os problemas que não são desafios e correspondem ao ambiente de jogo de progressão (níveis) se encontram guardados localmente na aplicação o terceiro requisito necessitou de alteração. Como os desafios são guardados na base de dados persistente do servidor

verificou-se que não seria necessário usar qualquer tipo de identificador pois ao aceder ao menu dos seus desafios são apresentados ao jogador todos os desafios que lhe foram feitos. Desta forma os dois últimos requisitos foram substituídos por novos requisitos.

#### 4.1.2 Requisitos Não Funcionais

Além das secções de requisitos já presentes nos requisitos iniciais foi adiciona uma nova secção aos requisitos finais, os requisitos de restrição.

<b>Us-1</b>	<b>Usabilidade</b>	<b>Categoria</b>
Us-1.1	Interface intuitiva e de fácil utilização	M
Us-1.2	Facilidade de percepção para que posição os botões devem ser movidos pelo utilizador	M
Us-1.3	Fácil percepção do resultado de respostas a exercícios	M
Us-1.4	Facilidade de percepção do desenvolver do nível de dificuldade mediante progresso no jogo	M
Us-1.5	Recolha de dados transparente por parte do sistema	M

TABELA 4.12: Requisitos Iniciais de Usabilidade

Nesta secção procedeu-se à limpeza daqueles que são opções de design e removeram os requisitos 1.2, 1.3 e 1.4.

<b>F-2</b>	<b>Fiabilidade</b>	<b>Categoria</b>
F-2.1	Caso um dispositivo não se conseguir conectar à internet(servidor) deve ser possível que o utilizador consiga utilizar a aplicação	M
F-2.2	O sistema deve ser robusto e à prova de erros	M
F-2.3	Caso ocorra um erro na aplicação todos os dados até à ocorrência do erro devem estar guardados em base de dados persistente	M
F-2.4	O tratamento de erros deve ser transparente ao utilizador	M

TABELA 4.13: Requisitos Iniciais de Fiabilidade

Como a aplicação funciona com uma base de dados na sua totalidade no lado do servidor este terá de ter ligação à internet para se poder autenticar logo o primeiro requisito foi removido. Também de se notar que é obrigatório haver ligação à internet pois o modo de desafio funciona baseado na troca de desafios entre jogadores.

<b>S-3</b>	<b>Segurança</b>	<b>Categoria</b>
S-3.1	O sistema deve garantir a confidencialidade dos dados dos utilizadores	M

TABELA 4.14: Requisitos Iniciais de Segurança

Esta secção de requisitos não sofreu qualquer alteração.

<b>Po-4</b>	<b>Portabilidade</b>	<b>Categoria</b>
Po-4.1	A aplicação deve correr em múltiplas versões Android	M
Po-4.2	Deve ser possível testar a aplicação para garantir testes de superação de metas	M

TABELA 4.15: Requisitos Iniciais de Portabilidade

Foi adicionado um novo requisito a esta secção de requisitos para que seja garantido que a aplicação se adapte a diferentes resoluções de ecrã.

<b>Im-5</b>	<b>Implementação</b>	<b>Categoria</b>
Im-5.1	O desenvolvimento deve ser feito através de prototipagem evolutiva	M
Im-5.2	O desenvolvimento deve ser feito numa linguagem conhecida pelo programador	M
Im-5.3	O desenvolvimento deve ser feito numa plataforma onde o programador tenha experiência	M
Im-5.4	As base de dados devem ser persistentes de forma a facilitar o acesso aos dados	M

TABELA 4.16: Requisitos Iniciais de Implementação

Esta secção de requisitos não sofreu qualquer alteração.

## 4.2 Requisitos Finais

Depois da análise presente na secção anterior são agora apresentados os requisitos finais.

### 4.2.1 Requisitos Funcionais

<b>L-1</b>	<b>Leitura e Sorteio de Dados</b>	<b>Categoria</b>
L-1.1	A aplicação tem uma base de dados de problemas local independente com informação para todos os tipos de problema	M
L-1.2	Em jogo online(desafio) o sistema lê a base de dados do servidor	M
L-1.3	O sistema tem de ler problemas da base de dados para todos os tipos de estruturas de problemas	M
L-1.4	O sistema tem de sortear problemas para cada nível consoante o tipo de estrutura de problemas associado	M

TABELA 4.17: Requisitos Finais de Leitura e Sorteio de Dados

<b>M-2</b>	<b>Movimentação de Botões</b>	<b>Categoria</b>
M-2.1	O utilizador pode mover operandos independentemente da sua posição	M
M-2.2	O utilizador pode mover operadores independentemente da sua posição	M
M-2.3	O utilizador não pode mover o botão correspondente à resposta final	M
M-2.4	O utilizador ao mover um botão para uma posição de resposta é criado um novo botão igual e na posição de origem do botão movimentado	M
M-2.5	O utilizador pode mover botões criados	M

TABELA 4.18: Requisitos Finais de Movimentação de Botões

<b>V-3</b>	<b>Verificação de Posicionamento de Botões</b>	<b>Categoria</b>
V-3.1	A cada movimento de botão o sistema verifica se este é posicionado numa área de resposta correspondente ao tipo de botão	M
V-3.2	Se um botão é posicionado numa área de resposta correspondente e esta estiver vazia guarda o botão em posição de resposta	M
V-3.3	Se um botão é posicionado numa área de resposta correspondente e a posição de resposta estiver ocupada o botão é devolvido à sua posição de origem	M
V-3.4	A cada posicionamento de botão em posição de resposta é verificado se todas as posições de resposta estão ocupadas	M
V-3.5	Se todas as posições de resposta se encontram ocupadas o sistema procede à verificação da solução	M

TABELA 4.19: Requisitos Finais de Verificação de Posicionamento de Botões

<b>N-4</b>	<b>Níveis de Dificuldade</b>	<b>Categoria</b>
N-4.1	O jogo é composto por níveis de dificuldade	M
N-4.2	Cada nível é superado assim que o jogador acertar um problema desse mesmo nível	M
N-4.3	Cada nível tem um tipo de estrutura de exercício correspondente	M
N-4.4	Cada nível tem apenas um tipo de estrutura de exercício associado	M
N-4.5	Cada tipo de estrutura de exercício pode estar associado a um ou mais níveis de dificuldade	M
N-4.6	O jogador pode voltar a jogar um nível anteriormente superado	M
N-4.7	Ao superar um nível o jogador é redireccionado para o menu de níveis	M

TABELA 4.20: Requisitos Finais de Níveis de Dificuldade

<b>VR-5</b>	<b>Verificação de Respostas e subida de Nível de Dificuldade</b>	<b>Categoria</b>
VR-5.1	A verificação de resposta é feita através de cálculos matemáticos	M
VR-5.2	O utilizador tem 3 tentativas para resolver cada exercício	M
VR-5.3	Se o utilizador responder erradamente a um exercício é lhe apresentado o mesmo exercício para uma nova tentativa	M
VR-5.4	Se o utilizador responder erradamente por três vezes consecutivas a um exercício é lhe sorteado um novo exercício do mesmo nível para resolver	M
VR-5.5	Se o utilizador responder correctamente a um exercício este sobe de nível	M
VR-5.6	Se o jogador estiver a jogar um nível que não seja o último desbloqueado é redireccionado para o menu de níveis apenas após a resolução de cinco problemas	M

TABELA 4.21: Requisitos Finais de Verificação de Resposta e Níveis de Dificuldade

<b>R-6</b>	<b>Rescolha de Dados</b>	<b>Categoria</b>
R-6.1	A cada resposta dada o sistema envia para o servidor todos os dados referentes ao exercício respondido	M
R-6.2	A cada resposta dada o sistema envia para o servidor a actualização de dados do jogador	M

TABELA 4.22: Requisitos Finais de Recolha de Dados

<b>E-7</b>	<b>Efeitos Sonoros e Feedback</b>	<b>Categoria</b>
E-7.1	Melodia de fundo sempre que a aplicação se encontrar em funcionamento	M
E-7.2	Feedback sonor de notificação para resposta correcta	M
E-7.3	Feedback sonor de notificação para resposta errada	M
E-7.4	Feedback visual de notificação para resposta correcta	M
E-7.5	Feedback visual de notificação para resposta errada	M

TABELA 4.23: Requisitos Finais de Efeitos Sonoros e Feedback

<b>Me-8</b>	<b>Mérito</b>	<b>Categoria</b>
Me-8.1	Atribuição de medalhas por acumulação de respostas correctas	M
Me-8.2	Atribuição de medalhas por acumulação de respostas correctas consecutivas	M
Me-8.3	Atribuição de medalhas por superação de níveis	M
Me-8.4	Atribuição de medalha através da aplicação da soma	M
Me-8.5	Atribuição de medalha através da aplicação da subtração	M
Me-8.6	Atribuição de medalha através da aplicação da multiplicação	M
Me-8.7	Atribuição de medalha através da aplicação da divisão	M
Me-8.8	Atribuição de medalha através da aplicação de duas operações matemáticas diferentes	M
Me-8.9	Atribuição de medalha através da aplicação da soma e resolução de problema em menos de 5 segundos	M
Me-8.10	Atribuição de medalha através da aplicação da subtração e resolução de problema em menos de 5 segundos	M
Me-8.11	Atribuição de medalha através da aplicação da multiplicação e resolução de problema em menos de 5 segundos	M
Me-8.12	Atribuição de medalha através da aplicação da divisão e resolução de problema em menos de 5 segundos	M
Me-8.13	Tabela de classificação por número de respostas correctas de utilizadores	C

TABELA 4.24: Requisitos Finais de Mérito

<b>RA-9</b>	<b>Registo e Autenticação</b>	<b>Categoria</b>
RA-9.1	O utilizador pode registar-se no servidor	M
RA-9.2	O utilizador pode autenticar-se com o servidor	M

TABELA 4.25: Requisitos Finais de Registo e Autenticação

<b>S-10</b>	<b>Servidor</b>	<b>Categoria</b>
S-10.1	O servidor guarda todos os dados de forma persistente	M
S-10.2	O servidor tem uma base de dados com a informação de registo dos utilizadores e respectivos dados de mérito	M
S-10.3	O servidor guarda todos os dados de cada jogador	M
S-10.4	O servidor recebe dados da aplicação	M
S-10.5	O servidor actualiza dados sempre que recebe dados	M
S-10.6	O servidor envia dados sempre que estes são pedidos pela aplicação	M
S-10.7	O servidor tem uma base de dados para os desafios criados pelos jogadores	M
S-10.8	O servidor guarda um desafio sempre que recebe um desafio da aplicação	M
S-10.9	O servidor apaga um desafio sempre que recebe informação para desafio resolvido ou falhado	M

TABELA 4.26: Requisitos Finais de Servidor

<b>MD-11</b>	<b>Modo de Desafio</b>	<b>Categoria</b>
MD-11.1	O utilizador pode desafiar qualquer utilizador registado na aplicação	M
MD-11.2	O utilizador pode criar exercícios para qualquer tipo de estrutura de dados de exercícios	M
MD-11.3	Para cada exercício criado o jogador tem de apresentar a respectiva solução	M
MD-11.4	Se tiver a solução apresentada estiver correcta o desafio é enviado para o servidor	M
MD-11.5	O utilizador pode ver todos os seus desafios	M
MD-11.6	O utilizador pode escolher um desafio para resolver	M
MD-11.7	Ao responder a um desafio, correcta ou incorrectamente, o desafio deixa de fazer parte da lista de desafios	M

TABELA 4.27: Requisitos Finais de Modo de Desafio

### 4.2.2 Requisitos não Funcionais

<b>Us-1</b>	<b>Usabilidade</b>	<b>Categoria</b>
Us-1.1	Interface intuitiva e de fácil utilização	M
Us-1.2	Recolha de dados transparente por parte do sistema	M

TABELA 4.28: Requisitos Finais de Usabilidade

<b>F-2</b>	<b>Fiabilidade</b>	<b>Categoria</b>
F-2.1	O sistema deve ser robusto e à prova de erros	M
F-2.2	Caso ocorra um erro na aplicação todos os dados até à ocorrência do erro devem estar guardados em base de dados persistente	M
F-2.3	O tratamento de erros deve ser transparente ao utilizador	M

TABELA 4.29: Requisitos Finais de Fiabilidade

<b>S-3</b>	<b>Segurança</b>	<b>Categoria</b>
S-3.1	O sistema deve garantir a confidencialidade dos dados dos utilizadores	M

TABELA 4.30: Requisitos Finais de Segurança

<b>Po-4</b>	<b>Portabilidade</b>	<b>Categoria</b>
Po-4.1	A aplicação deve correr em múltiplas versões Android	M
Po-4.2	Deve ser possível testar a aplicação para garantir testes de superação de metas	M
Po-4.3	A aplicação deve adaptar-se às diferentes resoluções de ecrãs	M

TABELA 4.31: Requisitos Finais de Portabilidade

<b>Im-5</b>	<b>Implementação</b>	<b>Categoria</b>
Im-5.1	O desenvolvimento deve ser feito através de prototipagem evolutiva	M
Im-5.2	O desenvolvimento deve ser feito numa linguagem conhecida pelo programador	M
Im-5.3	O desenvolvimento deve ser feito numa plataforma onde o programador tenha experiência	M
Im-5.4	As base de dados devem ser persistentes de forma a facilitar o acesso aos dados	M

TABELA 4.32: Requisitos Finais de Implementação

<b>Res-6</b>	<b>Restrições</b>	<b>Categoria</b>
Res-6.1	O desenvolvimento do servidor tem de ser implementado em PHP	M
Res-6.2	A base de dados tem de ser MySQL	M
Res-6.3	O desenvolvimento tem de ser feito em plataforma Android	M

TABELA 4.33: Requisitos Finais de Restrições

## Capítulo 5

# Arquitectura

Este capítulo está dividido em 3 secções. Na primeira é apresentada uma vista conceptual da arquitectura, na segunda é apresentada a arquitectura final com as diferentes visões parciais da arquitectura e as respectivas explicações detalhadas, e finalmente são apresentadas as tecnologias utilizadas para desenvolver a aplicação e as respectivas razões de escolha.

Inicialmente, como se pode verificar no capítulo 4, havia uma lista de requisitos inicial que difere da lista de requisitos final. Esta lista sofreu estas alterações através da prototipagem e da procura de melhores soluções para a aplicação em si. Desta forma também a arquitectura da aplicação sofreu alterações pois foram encontradas soluções que facilitam a interacção do jogador com o jogo e a aplicação em si.

### 5.1 Arquitectura Conceptual

Para um melhor entendimento do sistema é primeiramente explicado como o sistema funciona de uma forma conceptual.

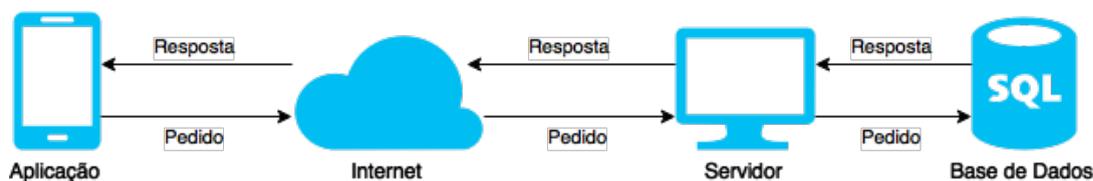


FIGURA 5.1: Arquitectura Conceptual

O utilizador acede à aplicação através do seu telemóvel. À medida que o jogador interage com a aplicação, seja através da realização de tarefas, seja através do ambiente de jogo, esta envia, sempre que necessário, para o servidor os pedidos referentes à tarefa realizada

pelo jogador. Estes pedidos são feitos via HTTP e podem visar um pedido, inserção, actualização ou remoção de dados na base de dados, através do servidor, que realiza todas estas operações.

## 5.2 Arquitectura Final

Para representar a arquitectura final recorreu-se a dois tipos de diagramas:

- Diagrama de componentes[22] onde está representada a arquitectura global da aplicação, o seu modelo e análise à forma como o sistema comunica na relação Aplicação-Servidor. Foi escolhido este tipo de diagrama pois o seu propósito é representar a relação estrutural entre componentes de um sistema.
- Diagramas de sequência[23], que é aplicado em visões parciais da arquitectura onde são apresentados e explicados mais detalhadamente os processos de organização sequenciais da arquitectura da aplicação e como estes estão ligados entre si. Foi escolhido este tipo de diagrama porque este permite demonstrar, de forma sequencial ordenada, a interacção entre objectos ou classes. Para este caso concreto usou-se os diagramas de sequência para demonstrar como estão relacionadas as diferentes tarefas que são feitas dentro da aplicação e como estas interagem com o servidor.

### 5.2.1 Modelo

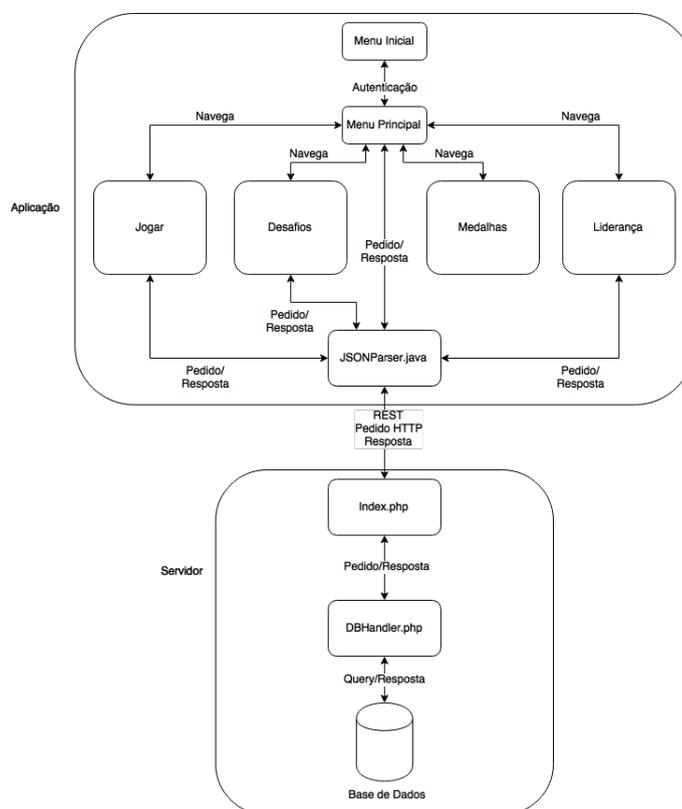


FIGURA 5.2: Diagrama de Componentes

Este modelo é responsável por representar o fluxo de informação presente no sistema. Pretendeu-se, assim, apresentar a estrutura de uma forma mais aprofundada, como é transportado o fluxo de informação dentro do sistema da aplicação, como este está ligado ao servidor e, ainda, como a informação que é enviada para este é processada. A nível do servidor, na arquitectura, é evidenciada a totalidade das suas componentes, já na aplicação as componentes não foram apresentadas com muito detalhe pois nas secções seguintes serão apresentadas, uma a uma, detalhadamente.

### 5.2.2 Classes de comunicação Aplicação-Servidor

- **JSONParser.java**

Esta classe é responsável por enviar todos os pedidos feitos pela aplicação para o servidor, bem como receber as respostas do mesmo. Estes pedidos são feitos através de request HTTP, sendo estes POST ou GET.

- **index.php**

Esta ficheiro é responsável por executar todos os pedidos feitos pela aplicação ao servidor. Mediante cada pedido este ficheiro, através dos seus métodos, chama os métodos do ficheiro DBHandler.php para proceder à realização da query pretendida. Após a query ser feita e receber os dados, o ficheiro index.php analisa os dados obtidos e envia uma resposta para a aplicação, consoante os mesmos.

- **DBHandler.php**

Este ficheiro é responsável por fazer todas as queries necessárias à base de dados de forma a adicionar, remover, obter ou actualizar informações consoante o pedido feito pela aplicação.

São agora analisadas as visões parciais da arquitectura anteriormente explicada. Note-se que em qualquer parte do sistema o jogador pode sempre voltar à página anterior daí esta funcionalidade não ser referenciada nas descrições das visões parciais.

### 5.2.3 Visão Parcial - Autenticação

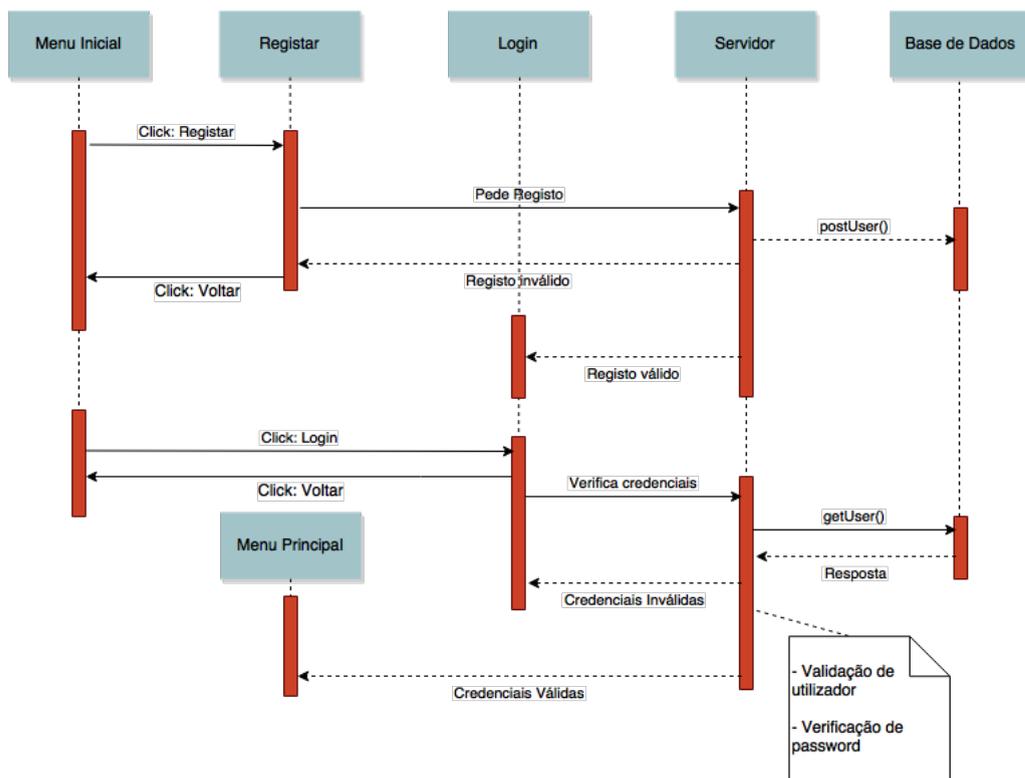


FIGURA 5.3: Diagrama de Sequência - Autenticação

A figura 5.3 representa o diagrama de sequência referente ao registo e login de utilizadores. Ao fazer um registo válido a aplicação envia os dados para o servidor através de

request HTTP POST de forma a que sejam guardados persistentemente sendo o jogador redireccionado para o menu de login de forma a autenticar-se. Ao fazer login, seja através do menu inicial ou depois do registo, o jogador insere os seus dados, a aplicação envia os dados para o servidor através de request HTTP POST (o index.php tem de receber a password e descriptá-la) para verificação de autenticação e se o utilizador estiver na base de dados e a password de autenticação estiver correcta, este é redireccionado para o menu principal, caso contrário é apresentada uma mensagem de dados inválidos. Ao ser autenticado o servidor envia para a aplicação todos os dados do jogador presentes na base de dados através de request HTTP GET.

### 5.2.4 Visão Parcial - Jogar

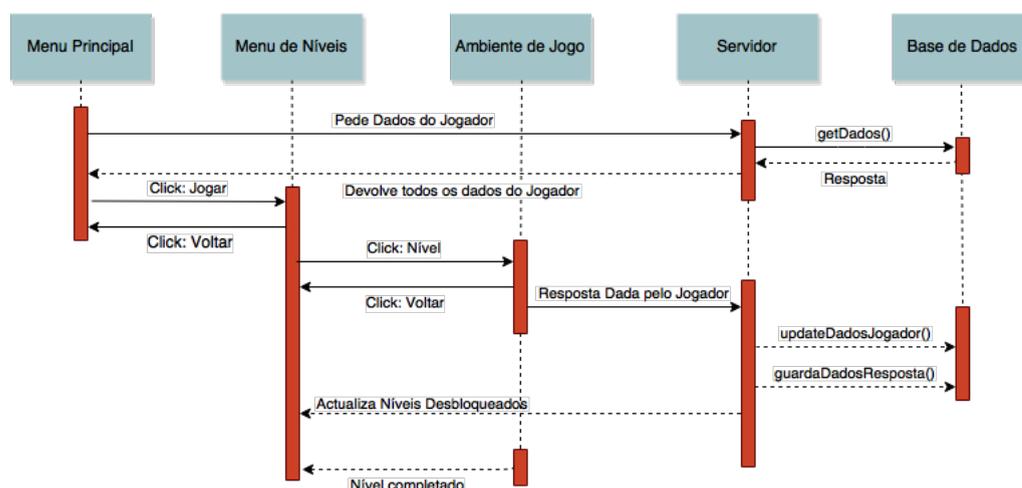


FIGURA 5.4: Diagrama de Sequência - Jogar

A figura 5.4 representa o diagrama de sequência referente ao processo para modo de jogo single player através do sistema de progressão. O jogador acede ao menu de níveis para escolha do nível que pretender desde que este esteja desbloqueado. Ao seleccionar o nível é apresentado um problema do respectivo nível e, ao receber uma resposta, a aplicação irá enviar os dados para o servidor através de request HTTP POST. Os dados actualizados são os referentes a alguma meta de mérito que o jogador possa superar, os dados guardados são referentes à resposta do utilizador para investigação dos mesmos.

### 5.2.5 Visão Parcial - Menu Medalhas e de Liderança

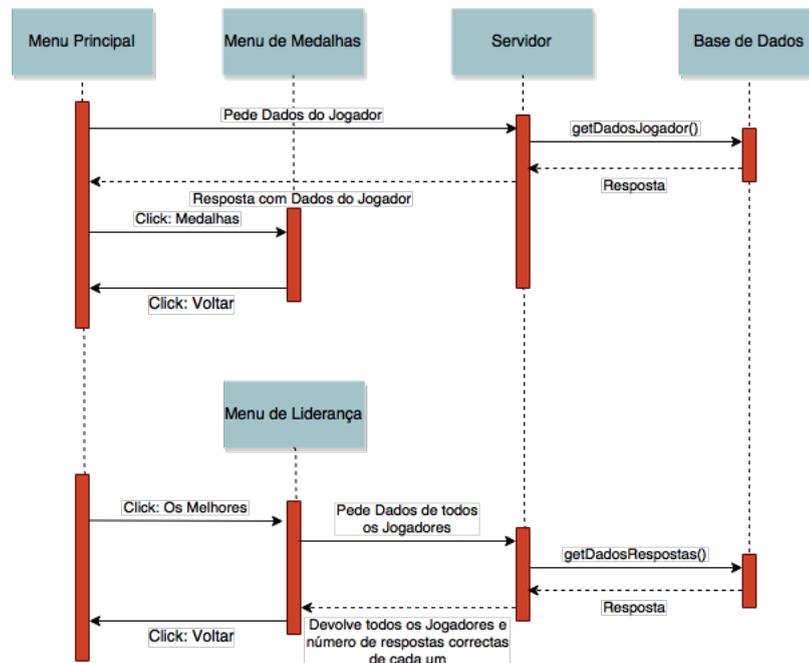


FIGURA 5.5: Diagrama de Sequência - Menu de Medalhas e de Liderança

A figura 5.5 representa os diagramas de sequências referente ao acesso aos menus de medalhas e de liderança. Ao seleccionar o menu de medalhas o sistema apresenta as medalhas conquistadas consoante os dados de mérito do mesmo já obtidos no menu principal após autenticação. Ao seleccionar o menu de liderança o sistema pede os dados de todos os jogadores através de request HTTP GET, nomeadamente o nome de utilizador e total de respostas correctas de forma a gerar a tabela de liderança.

### 5.2.6 Visão Parcial - Desafiar

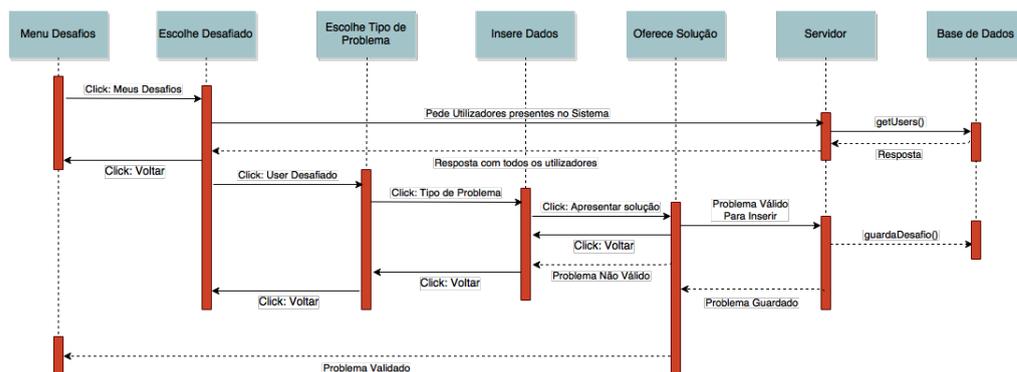


FIGURA 5.6: Diagrama de Sequência - Desafiar

A figura 5.6 representa o diagrama de sequência referente a todo o processo necessário para o jogador criar um desafio. Ao seleccionar o menu de desafio e posteriormente a opção de desafiar a aplicação pede ao servidor todos os nomes de utilizador presentes na base de dados, através de request HTTP GET, para que o jogador possa escolher um jogador para desafiar. Após escolher o desafiando é pedido que o jogador escolha o tipo de jogo e, em seguida, é pedido ao utilizador que insira os dados pretendidos e que apresente a solução para esses mesmos dados. Se a solução apresentada for correcta a aplicação envia o problema para o servidor guardar na base de dados, através de request HTTP POST, com todos os identificadores anteriormente seleccionados de forma a ficar disponível para ser jogado. Finalmente o jogador é redireccionado para o menu de desafios. Se a solução não estiver correcta o jogador volta ao menu para nova inserção de dados de forma a corrigir os mesmos.

### 5.2.7 Visão Parcial - Jogar Problema Desafiado

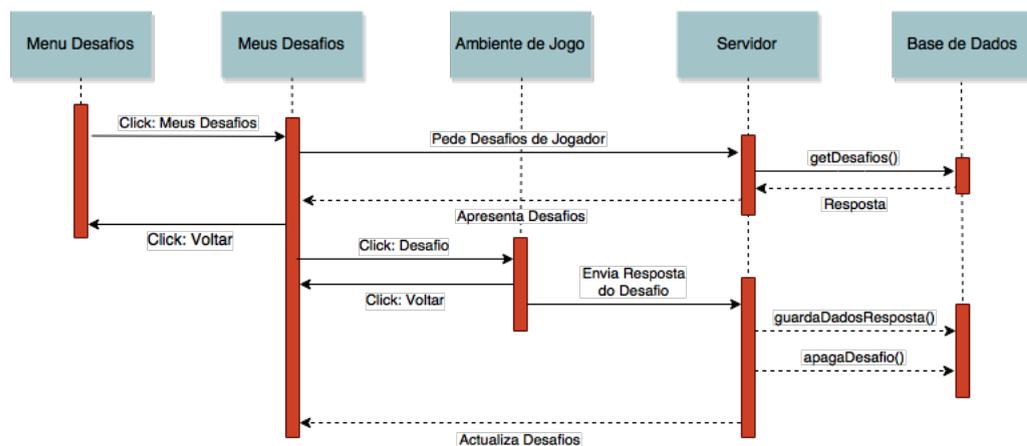


FIGURA 5.7: Diagrama de Sequência - Jogar Problema Desafiado

A figura 5.7 representa o diagrama de sequência referente a todo o processo necessário para resolver um desafio. Ao seleccionar o menu de desafio e posteriormente a opção de *Meus Desafios*, a aplicação irá pedir ao servidor todos os desafios feitos ao jogador, se estes existirem, através de request HTTP GET. Caso existam, estes serão apresentados e o jogador poderá seleccionar aquele que pretender. Após oferecida uma solução correcta ou errada, através de request HTTP POST, o desafio é removido da base de dados, e finalmente o jogador é redireccionado para o menu dos seus desafios.

## 5.3 Tecnologias

Nesta secção são apresentadas as diversas tecnologias escolhidas para o desenvolvimento da aplicação proposta e as razões da escolha dessas mesmas tecnologias.

### 5.3.1 Plataforma Escolhida

A plataforma escolhida foi a plataforma Android[2]. Esta escolha prende-se não por razões tecnológicas mas sim porque foi uma das vontades da cliente, que justificou a sua escolha ao identificar que o Android é a plataforma mais comum nos *smartphones* vendidos em Portugal, na actualidade. Trata-se também de uma plataforma *opensource* o que facilita o desenvolvimento através do vasto leque de documentação existente e na qual o estagiário já tinha experiência.

### 5.3.2 Linguagens de Programação

As linguagens escolhidas foram o Java[24] para implementação de funcionalidades da aplicação, XML para a implementação de interfaces e PHP para a implementação do servidor.

Para a implementação da aplicação foi escolhida a linguagem JAVA pois é a linguagem em que a maior parte das aplicações Android estão implementadas ainda que seja possível fazê-lo noutras linguagens tal como C Sharp, C++, entre outras. Esta é também a linguagem com maior suporte da Google e é nesta que existe maior número de APIs para o desenvolvimento do tipo de aplicações pretendido. A ferramenta utilizada para o desenvolvimento da aplicação é o Eclipse[25] com o *plugin* desenvolvido pela Google, Android Developer Tools[26] (ainda que a Google agora suporte o Android Studio), e este já vem com a linguagem Java por *default* para o desenvolvimento de aplicações Android, e o mesmo se aplica ao XML para a implementação da interface.

Para o desenvolvimento do servidor foi inicialmente utilizada a linguagem JAVA recorrendo-se à arquitectura REST com recurso a servidor Apache Tomcat. Posteriormente foi alterado para linguagem PHP devido ao tipo de servidores alojados nos Serviços Informáticos da Faculdade de Ciências e Tecnologias da Universidade de Coimbra (Servidores Apache HTTP).

### 5.3.3 Base de Dados

A base de dados para este projecto foi feita de forma a otimizar o acesso a recursos. Para alcançar este objectivo a base de dados foi dividida em duas partes distintas: Uma base de dados independente que é transferida para o dispositivo e guardada neste aquando da sua instalação, e uma base de dados no servidor acedida pela aplicação com frequência para adição, actualização e/ou remoção de dados.

**Base de dados do dispositivo:** Esta base de dados consiste em ficheiros com a informação para os problemas consoante o nível. Esta foi definida com o objectivo de ter todos os problemas de jogo *single player*, e que estes possam ser manipulados sem recurso a requests ao servidor poupando assim possíveis problemas de tráfego bem como a falta de acesso do utilizador ao servidor. Inicialmente foi proposto o uso de SQLite para base de dados persistente nos dispositivos mas como foi decidido que a base de dados de exercícios estaria já inserida na aplicação não se considerou necessário o recurso à utilização de base de dados persistente SQLite[27], base de dados esta considerada a ideal para dispositivos móveis[28]. Cada nível tem um ficheiro distinto associado, o que torna o consumo de recursos por parte do dispositivo bastante reduzido.

**Base de dados no servidor:** Esta base de dados é uma base de dados persistente, que guarda todas as informações de cada utilizador, sejam elas credenciais, níveis, medalhas, entre outras, a cada jogada. A escolha da base de dados limitou-se ao tipo de base de dados dos serviços informáticos da FPCEUC que é o MySQL. Neste caso não se tornou um obstáculo pois o MySQL[29] oferece várias vantagens para guardar dados persistentes, sendo que as principais para a aplicação desenvolvida são:

- Elevada performance caso vários utilizadores acedam ao mesmo tempo à base de dados.
- Elevada disponibilidade que garante actualizações imediatas de forma a manter a informação sempre actualizada.
- Elevada protecção de dados que garante a protecção das credenciais de cada jogador de forma a mantê-las confidenciais.

### 5.3.4 Outras Tecnologias

Como ferramenta de desenvolvimento foi escolhido o Eclipse para todo o desenvolvimento com a ajuda do plugin desenvolvido pela Google, Android Developer Tools (ADT), como foi anteriormente referido, que é integrado com a IDE do Eclipse. Este *plugin* oferece

acesso a muitas ferramentas que ajudam ao desenvolvimento de aplicações Android. O ADT oferece acesso à GUI, a muitas linhas de comando da ferramenta SDK bem como à ferramenta de design de UI que permite rápida prototipagem, design e construção de aplicações e interfaces de utilizador. A escolha recaiu também sobre o Eclipse porque é a ferramenta em que o estagiário tinha mais experiência para o desenvolvimento de aplicações Android.

A nível de comunicação entre cliente e servidor recorreu-se à arquitectura REST[30] (*Representational State Transfer*). REST é um estilo de arquitectura de sistemas ligados em rede que consiste em clientes e servidor. Ao invés de se usar mecanismos complexos, tais como o CORBA, RPC ou SOAP, para fazer a conexão entre as máquinas, é usado HTTP simples para fazer chamadas entre as mesmas. As aplicações REST usam pedidos HTTP para criar ou actualizar dados, ler dados e apagar dados, através das operações CRUD (*Create, Read, Update e Delete*)[31]. As motivações por trás dos serviços web REST é igual às motivações da tecnologia em si, tornar processos complexos em processos mais simples, pois mesmo sendo simples, a arquitectura REST tem todas as ferramentas necessárias para serviços web. Muitos investigadores defendem que o uso de SOAP e XML é demasiado pesado pois o HTTP tem capacidade suficiente para as aplicações comunicarem entre si.

O REST aproveita na sua totalidade os protocolos e padrões da *World Wide Web* e é mais simples que os tradicionais serviços web baseados em SOAP. Com o surgimento da cloud-computing e o crescente interesse para aplicações *web-hosting*, tecnologias baseadas em REST podem ajudar tanto no desenvolvimento de interfaces de utilizador ricas e que chamam servidores remotos, bem como no desenvolvimento de servidores reais para manipular estruturas de dados em uma aplicação de cliente, como é o caso, ( escrito em qualquer linguagem ) ou directamente no navegador[32].

## Capítulo 6

# TEMPOLY - Prototipagem e Desenvolvimento

O jogo TEMPOLY foi um jogo que esteve a ser desenvolvido durante todo este programa de estágio, resultado da parceria entre a FPCEUC e a FCTUC, mais precisamente o DEI, no âmbito da investigação de Doutoramento em Ciências de Educação da Doutoranda Cândida Barros, supervisionada pela Professora Doutora Ana Amélia Carvalho. Visto que este projecto teve as suas origens numa investigação preliminar no âmbito do doutoramento da Cândida Barros, o conceito do jogo estava, à partida, limitado às soluções apresentadas pela mesma. Esta foi também umas das razões pela qual não se realizou prototipagem em papel pois esta já tinha sido realizada, bem como a avaliação com o público alvo e a concepção do jogo, procedendo-se assim de imediato à implementação das funcionalidades.

Através do desenvolvimento das funcionalidades foi possível corrigir a aplicação através da adição/alteração/remoção das mesmas, à medida que o processo de prototipagem avançou e revelou lacunas que não tinham sido previstas nos testes anteriores à implementação.

Uma das funcionalidades em que houve mais liberdade para adicionar e manipular foram os ecrãs de menus pois estes não estavam todos definidos à partida, mais precisamente, os menus do modo de desafio. Este facto deveu-se também à pouca especificação do que era pretendido para o modo de desafio, o que deu também mais liberdade no desenvolvimento da funcionalidade em si.

Para que um jogo seja fácil de utilizar, sem que se esteja efectivamente a jogar, é necessário ter uma boa organização de tarefas que o jogador desenvolva dentro da aplicação. Todas as tarefas oferecidas ao jogador estão organizadas e agrupadas de forma a que

este consiga, intuitivamente, aceder a todas elas sem dificuldade de forma a estas não criarem obstáculos ao jogador. Desta forma, para os menus, optou-se por criar um tipo de navegação simples, inserindo pouca informação nos mesmos, para não confundir o jogador, e de uma forma sequencial de maneira a que este não se "perca" na aplicação tornando a experiência uma experiência frustrante. Desta forma simplificou-se os menus o mais possível para que o jogador consiga facilmente chegar onde quer.

O processo de desenvolvimento, que foi iniciado desde o início do primeiro semestre, foi fiel à abordagem evidenciada no Capítulo 3, abordagem esta de prototipagem evolutiva com recurso a desenvolvimento iterativo e incremental. Desta forma, ao longo do processo de prototipagem foi dada prioridade aos requisitos que se consideraram mais importantes.

A nível de prototipagem o desenvolvimento foi dividido em três ciclos de forma a atingir a solução final. O primeiro protótipo foi desenvolvido e apresentado no documento intermédio e nesse mesmo documento foram evidenciadas as funcionalidades que seriam implementadas no segundo e terceiro protótipo. À medida que o desenvolvimento progrediu sentiu-se a necessidade de alterar a prioridade de algumas tarefas alterando assim a ordem do desenvolvimento efectuado daquele que foi proposto. Estas alterações foram efectuadas maioritariamente para otimizar o desenvolvimento, sendo que foi dada prioridade a certas funcionalidades que, após reconsideração, acabou por ser alterada.

De um modo geral, o que foi feito não variou muito do que foi proposto. É agora enunciado o que foi feito em cada protótipo e por ordem de implementação dentro de cada um deles.

- **Primeiro Protótipo**

- Implementação da mecânica de jogo - modo de jogo com apenas uma operação

- Implementação do menu principal

- Recolha de Dados

- **Segundo Protótipo**

- Implementação de modos de jogo com maior número de operações

- Implementação do servidor

- Implementação do menu inicial, registo e autenticação de utilizadores

- **Terceiro Protótipo**

- Inserção do Design

- Implementação do menu de progressão - níveis

Implementação do menu de medalhas

Implementação do modo de desafio e respectivos menus

Implementação do menu de liderança

Note-se que durante todo este processo muitas funcionalidades foram constantemente sofrendo melhorias e/ou alterações, especialmente nas mecânicas de jogo, sendo que o seu desenvolvimento foi contínuo durante todo o processo de estágio. Todos estes pontos expandem um vasto leque de funcionalidades que serão melhor detalhadas nas respectivas secções. Este detalhe é relativo à solução final de cada um bem como as figuras apresentadas.

## 6.1 Mecânicas de Jogo - Modo de jogo com apenas uma operação

As mecânicas de jogo são as regras que oferecem coerência ao jogo ainda que transparentes ao jogador. O jogo é um jogo do tipo puzzle em que são apresentados ao jogador diversos operadores e operandos que este poderá utilizar através do seu arrastamento no ecrã para posições predefinidas de forma a chegar a um resultado previamente apresentado ao mesmo.

Estas mecânicas foram desenvolvidas para desenvolver aquele que seria o primeiro ambiente de jogo. Este é o ambiente de jogo mais simples e é jogado nos 10 primeiros níveis. A cada nível o número de operadores varia entre 2 e 3 sendo que a divisão nunca está presente e os operandos variam entre 3 e 4 opções possíveis.

Para resolver este tipo de problema o jogador tem de mover dois operandos e um operador, um de cada vez, e posicioná-los nas respectivas posições de resposta. Quando todas as posições de resposta estão preenchidas o sistema procede à verificação da resposta oferecida e procede ao início da animação respectiva.

- **Controlo de Botões**

Os botões são controlados, como anteriormente foi referido neste documento, de forma táctil através do arrastamento dos mesmos. Cada tipo de botão, seja ele um operador (botão redondo) ou um operando (botão rectangular), tem que ou deve ser posicionado em localizações específicas para o mesmo. Para que o jogador tenha uma mais fácil percepção de onde posicionar os botões, nas localizações de posicionamento para resposta foram colocados botões iguais para o utilizador arrastar os botões até eles.

Para facilitar o posicionamento dos botões foi dado um raio de detecção em que, se o botão for largado dentro desse mesmo raio, é imediatamente posicionado na respectiva posição de resposta. Caso o utilizador queira trocar a sua resposta basta remover o botão o suficiente para este ficar fora desse raio e ser devolvido à sua posição de origem. Numa situação em que um jogador largue um botão numa localização de resposta já ocupada não é permitido que este seja colocado nessa posição, sendo imediatamente devolvido à sua posição de origem.

- **Método para Efectuar Cálculo**

Inicialmente, no primeiro protótipo, este método era inexistente. A verificação de resposta não era feita através de cálculos mas sim através da verificação do posicionamento de botões correctos em zonas de resposta. Este primeiro método provou-se ineficiente pois invalidava os caminhos múltiplos de resolução da matemática, pois era possível apresentar uma solução correcta mas identificada como errada pelo sistema. Desta forma, para o segundo protótipo, deixou de se fazer esta verificação e substituiu-se pelo cálculo da solução apresentada pelo jogador e comparação com o resultado.

Este método procede à verificação de cálculo sempre que um botão é posicionado numa posição de resposta. Se um botão for posicionado mas existir algum campo de resposta vazio não é efectuado qualquer cálculo e retorna *false*, caso contrário procede ao cálculo entre os botões inseridos pelo jogador através de uma classe auxiliar. Se a resposta dada pelo jogador for igual ao resultado previamente apresentado retorna *true*, caso contrário retorna *false*.

- **Método para Verificar Resposta**

Este método verifica o resultado proveniente do método anteriormente descrito e passa à tomada de decisão sobre o que a aplicação deverá fazer consoante o input. Este método apenas faz a sua verificação se todos os campos de resposta estiverem devidamente preenchidos.

**Resposta Correcta:**

Inicialmente foi definido que o jogador teria de oferecer três respostas correctas para subir de nível, ou seja, se o jogador respondesse correctamente a um exercício ser-lhe-ia apresentado novo exercício do mesmo nível até este atingir as 3 respostas correctas para proceder ao nível seguinte. Posteriormente, a cliente considerou que seria necessária apenas uma resposta correcta para passar de nível. Se o jogador se encontrar no último nível que tem desbloqueado este subirá de nível e será redireccionado para o menu de níveis onde o novo nível estará agora desbloqueado. Se o jogador estiver a jogar um nível que não seja o último nível que tem desbloqueado, aquando um problema correctamente resolvido ser-lhe-á apresentado um novo

problema desse nível pois não faria sentido, neste caso, estar a ser redireccionado para o menu de níveis.

### Resposta Errada:

Caso a resposta seja errada, o problema proposto ao jogador será repetido até que este, ou o resolva e suba de nível, ou atinja 3 respostas erradas consecutivas e ser-lhe-á apresentado um novo problema do mesmo nível.

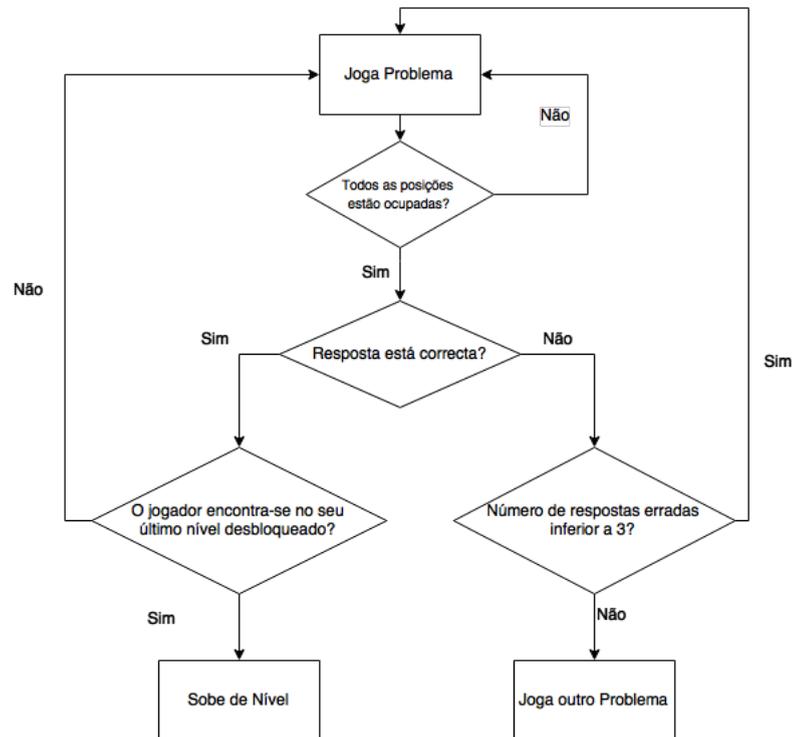


FIGURA 6.1: Algoritmo de verificação de resposta

- **Reutilização de Botões**

Este era um dos requisitos propostos e foi desenvolvido para o segundo protótipo. Esta funcionalidade permite ao jogador repetir um botão as vezes que entender.

- **Expoentes a Sobrescrito**

Inicialmente os polinómios estavam em formato "Coeficiente \* x ^ Grau". Sugeriu-se e foi alterado para o terceiro protótipo que o grau de cada polinómio se encontrasse a sobrescrito e sem o acento circunflexo.

- **Enquadramento dos polinómios ao tamanho das caixas**

À medida que a prototipagem avançou e foram inseridos problemas com maior complexidade notou-se que alguns polinómios eram demasiado grandes para o tamanho dos botões pelo que, para o terceiro protótipo, implementou-se um método

que permitiu definir o tamanho da letra de cada polinómio consoante o número de caracteres do mesmo.

É agora apresentada aquela que é a solução final do primeiro ambiente de jogo implementado.



FIGURA 6.2: Ambiente de Jogo Simples

## 6.2 Menu Principal

Este menu começou por ser o menu inicial mas após ser implementado o registo e autenticação de jogadores com o servidor passou a ser o menu que dá acesso, dentro do jogo, a todas as tarefas disponíveis para o jogador, após autenticação. Este foi o menu que mais alterações sofreu durante todo o processo de desenvolvimento pois à medida que se adicionaram funcionalidades à aplicação, é através deste menu que o jogador acede às mesmas. Este menu ao ser iniciado pede ao servidor todas as informações do jogador e guarda-as localmente para serem usadas e actualizadas quando necessário.



FIGURA 6.3: Menu Principal

### 6.3 Modos de jogo com diferente número de operações

A nível de mecânicas de jogo, todos os ambientes de jogo seguidamente enunciados, respeitam as mecânicas previamente apresentadas na descrição das mecânicas de jogo, variando apenas no número de opções que o jogador tem de preencher para chegar ao resultado previamente apresentado.

Inicialmente apenas o ambiente de jogo com duas operações foi proposto sendo que se considerou uma mais valia para a aplicação implementar-se mais um ambiente de jogo. Tal como proposto, este requisito foi atingido no segundo protótipo.

- **Problema com duas operações:**

Este ambiente de jogo é um pouco mais complexo que o anterior e é jogado do nível 11 ao nível 15. A cada nível o número de operadores varia entre 2 e 3 sendo que a divisão nunca está presente e os operandos variam entre 3 e 4 opções possíveis.

Para resolver este tipo de problema o jogador tem agora de mover três operandos e dois operadores, um de cada vez, e posicioná-los nas respectivas posições de resposta. Quando todas as posições de resposta estão preenchidas o sistema procede à verificação da resposta oferecida e procede ao início da animação respectiva.



FIGURA 6.4: Ambiente de jogo com duas operações

- **Problema com três operações sem divisão:**

Este é o ambiente de jogo mais complexo que o anterior e é jogado do nível 16 ao nível 20. A cada nível o número de operadores varia entre 2 e 3 sendo que a divisão nunca está presente e os operandos variam entre 3 e 4 opções possíveis.

Para resolver este tipo de problema o jogador tem de mover quatro operandos e três operadores, um de cada vez, e posicioná-los nas respectivas posições de resposta. Quando todas as posições de resposta estão preenchidas o sistema procede à verificação da resposta oferecida e procede ao início da animação respectiva.



FIGURA 6.5: Ambiente de jogo com 3 operações sem divisão

- **Problema com três operações com divisão:**

Este é o ambiente de jogo mais complexo e é jogado do nível 21 ao nível 25. Em todos os níveis o jogador tem 4 operadores e 4 operandos disponíveis para movimentar.

Para resolver este tipo de problema o jogador tem de mover quatro operandos e três operadores, um de cada vez, e posicioná-los nas respectivas posições de resposta. Quando todas as posições de resposta estão preenchidas o sistema procede à verificação da resposta oferecida e procede ao início da animação respectiva. Caso a divisão seja utilizada e se se verificar a existência de resto este é apresentado.



FIGURA 6.6: Ambiente de jogo com 3 operações com divisão

## 6.4 Recolha de Dados

A recolha de dados foi também um dos requisitos que sofreu alterações durante todo o processo.

Inicialmente, no primeiro protótipo, a cada resposta a um problema apresentado ao utilizador, o sistema guardava numa string os dados na seguinte ordem: nível correspondente, operadores disponíveis, operandos disponíveis, operador(es) utilizados, operandos utilizados e finalmente um identificador de resposta correcta ou errada com uma formatação definida para ser facilmente migrado para um ficheiro Excel. Esta string era posteriormente guardada localmente no dispositivo num ficheiro de texto e, a determinada altura do jogo (ainda em fase de testes), era enviado via mail para uma conta definida pela investigadora. O problema desta solução residiu no facto de que vários mails seriam enviados correspondentes a um mesmo jogador.

Após a implementação do servidor e este se encontrar totalmente funcional deixou de se guardar a String com os dados recolhidos num ficheiro local e passou-se a enviá-la, a cada

resposta, directamente para o servidor onde esta é guardada num ficheiro identificado com o nome de utilizador de cada jogador. Esta mudança permitiu um acesso aos dados recolhidos mais limpo, directo e simples por parte da investigadora sem que esta tenha de ir mail a mail e recolher os dados de cada jogador.

Foi também adicionado à String de recolha de dados enviada a cada resposta o tempo que cada jogador demorou a resolver o problema de forma a que se possa fazer uma análise mais detalhada do desempenho do jogador.

## 6.5 Servidor e Base de Dados

O servidor foi feito com o intuito de guardar dados numa base de dados persistente o desempenho de cada jogador de forma a garantir a viabilidade de funcionalidades que foram posteriormente implementadas. Para atingir este objectivo, inicialmente, definiu-se que seria necessário que um jogador teria de se registar e autenticar para interagir com a aplicação. Estes requisitos foram atingidos, como previsto, no desenvolvimento do segundo protótipo. De referir que ao se registar, a password de cada utilizador é encriptada através de Password Hashing de forma a garantir a confidencialidade da mesma.

Durante todo o desenvolvimento do terceiro protótipo, à medida que as necessidades para atingir os requisitos se verificaram, foram sendo adicionadas funcionalidades de comunicação com a aplicação bem como a adição, actualização ou remoção de dados da base de dados consoante as tarefas executadas pelo jogador.

Também durante todo o processo, quando se considerou necessário, foram sendo adicionados novos tipos de colunas à tabela já existente de utilizadores que foi inicialmente criada para o registo bem como a criação de uma nova tabela onde foi possível guardar os dados dos exercícios dos desafios criados pelos jogadores.

## 6.6 Design de Interface Final

### 6.6.1 Navegação Global Entre Ecrãs

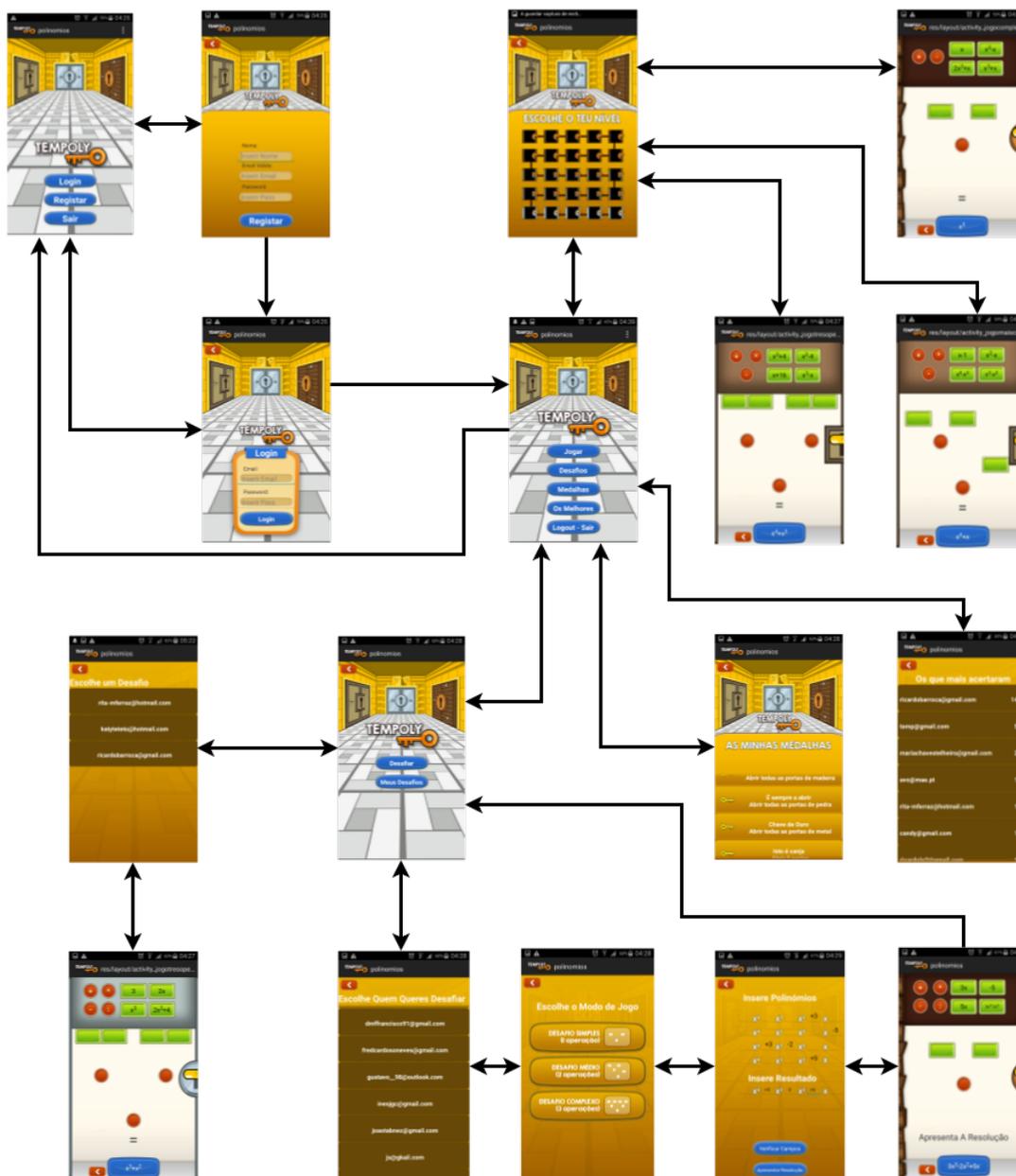


FIGURA 6.7: Navegação entre Ecrãs

Na Figura 6.7 é apresentada a navegação que o jogador pode fazer através de todos os ecrãs da aplicação. Toda esta navegação acaba por ser um resumo do que será em seguida explicado, funcionalidade após funcionalidade, e apresenta todas as tarefas que o jogador pode realizar dentro da aplicação.

### 6.6.2 Menu inicial, Registo e Autenticação de Utilizadores

Estes ecrãs são bastante simples, no menu inicial são oferecidas 3 opções ao jogador sendo elas registar, login e sair. Este menu acaba por ser indispensável para garantir que o utilizador tenha de se registar e/ou fazer login de forma a que este esteja presente na base de dados do servidor para registos de progressão e mérito, para poder desafiar e ser desafiado, estar presente na leader board e finalmente para obtenção e registo de dados para investigação.



FIGURA 6.8: Menu Inicial

Ao seleccionar a opção de registo este navegará para o ecrã de registo e é pedido ao jogador que insira 3 dados obrigatórios sendo ele nome, email que é validado através da obrigatoriedade de componentes (Ex: "algo@algo.algo"), ainda que seja possível registar-se sem que este efectivamente exista, e password. O email é a chave única da base de dados portanto caso este já esteja registado não será permitido ao utilizador registar-se. Ao ser feito o registo com sucesso o jogador é adicionado à base de dados, a sua password é enviada por mail caso este a perca, e finalmente navegará para o ecrã de login.

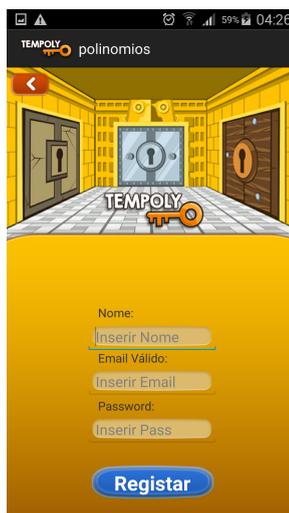


FIGURA 6.9: Menu de Registo

Ao seleccionar a opção de Login este navegará para o ecrã de login e é pedido que insira o seu email de registo e password. Se efectuar login com sucesso o jogador navegará para o Menu Principal.



FIGURA 6.10: Menu de Login

### 6.6.3 Inserção do Design

Para o desenvolvimento do design recorreu-se à contratação de um profissional que acompanhou o desenvolvimento do terceiro e último protótipo, fornecendo o material que se considerou necessário.

- **Tema**

Como o jogo é baseado na superação dos problemas apresentados decidiu-se adaptar um tema de portas. Com este tema pretendeu-se criar um ambiente de superação através do destrancar de portas à medida que o jogador progride no jogo sendo que inicialmente estas se encontram trancadas. A aplicação, em todos os seus ecrãs de navegação e no ambiente de jogo em si, está ligado ao tema descrito.

- **Animações**

Existem dois tipos de animações em ambiente de jogo, um tipo de animação que visa o movimento de botões e um tipo de animação que vai de encontro ao tema definido.

#### **Movimentação de Botões**

Estas animações têm como elemento base a movimentação dos operandos até ao respectivo operador, passando por trás deste, sendo este processo repetido se o modo de jogo tiver mais de uma operação, e finalmente o resultado final desce até ao resultado previamente apresentado pelo jogo. Se o resultado for correcto o botão com o resultado apresentado pelo jogador "entra" dentro do resultado previamente apresentado pelo sistema, caso contrário o botão do resultado final não "entra" ficando ligeiramente a cima do resultado previamente apresentado pelo jogo de forma a que o jogador possa verificar que são diferentes.

#### **Abrir e Fechar de Portas e Feedback**

Ao entrar num nível desbloqueado é apresentada uma animação de uma porta a abrir e se errar a porta é fechada numa forma representativa de que não conseguiu mantê-la aberta ao mesmo tempo que é emitido o som de uma porta a ser trancada. Caso contrário esta mantém-se aberta, é apresentada uma imagem verde com um visto, é emitido um som facilmente identificável como uma resposta correcta e o jogador procede para o problema ou nível seguinte dependendo do nível que está a jogar.

#### **6.6.4 Menu de Progressão - Níveis**

O jogo TEMPOLY é um jogo baseado no progresso e constituído por níveis. Para jogar, sem que seja um desafio, o jogador tem de passar obrigatoriamente pelo menu de níveis para seleccionar que nível pretende jogar. Este menu é constituído por uma grelha de níveis representados por portas em que os níveis desbloqueados estão representados com portas abertas e os bloqueados com portas fechadas não sendo possível para o jogador seleccionar os últimos referidos.

Ao registrar-se na aplicação um jogador tem apenas a primeira porta aberta, como está demonstrado na seguinte imagem.



FIGURA 6.11: Menu de Níveis - Início

Através da superação de níveis o jogador desbloqueia as seguintes. Qualquer porta que já esteja desbloqueada, mesmo não sendo o último nível desbloqueado, pode ser selecionado caso o jogador queira jogar um problema desse nível. Eis um exemplo de um jogador já num patamar mais avançado.



FIGURA 6.12: Menu de Níveis

### 6.6.5 Menu de Medalhas e Liderança

No menu de medalhas são apresentadas todas as medalhas que o jogador já conquistou através da superação de metas, sejam elas em ambiente de jogo ou através de tarefas

realizadas dentro da aplicação. Neste menu não há qualquer pedido ao servidor pois todas as informações necessárias já tinham sido pedidas no menu principal, até porque estas são apenas referentes ao próprio jogador.



FIGURA 6.13: Menu de Medalhas

O menu de liderança é semelhante ao anterior mas neste menu são apresentados todos os utilizadores da aplicação sendo organizados de forma decrescente de acordo com o respectivo número total de respostas correctas já alcançadas. Este menu ao ser iniciado faz um pedido ao servidor de forma a obter todos os utilizadores e respectivo número de respostas ordenando-as para serem apresentadas.



Os que mais acertaram	
user1@gmail.com	155
user2@gmail.com	112
user6@gmail.com	103
user3@gmail.com	81
user5@gmail.com	74
user4@gmail.com	67
temp@gmail.com	51

FIGURA 6.14: Menu de Liderança

### 6.6.6 Modo de desafio e respectivos menus

Depois da implementação das mecânicas de jogo, este foi provavelmente o requisito mais dispendioso pois envolve um grande leque de funcionalidades. A solução final desta funcionalidade não difere muito daquela que foi proposta no processo de criação do desafio mas foi melhorada noutros aspectos.

Ao entrar no menu de desafios são apresentadas duas opções ao jogador: Desafiar e Meus Desafios, em que o primeiro, como o nome indica, permitirá ao jogador entrar no fluxo de tarefas para desafiar outro jogador através da criação de um desafio, e o segundo que permite ao jogador visualizar e escolher um desafio para resolver caso tenha algum.

#### 6.6.6.1 Desafiar

O processo para desafiar foi desenvolvido de forma a ser o mais intuitivo e simples possível. Desta forma foram criados menus sequenciais para que, em cada um, o jogador não tivesse de realizar muitas tarefas. A implementação foi desta maneira feita para que esta funcionalidade não se tornasse muito confusa e difícil, e assim, frustrante para o jogador.

As figuras do processo e análise, que é descrito em seguida, correspondem ao mesmo fluxo de tarefas, estando dispostas sequencialmente para se perceber todo o processo.

- **Escolher Jogador para Desafiar**

Ao escolher a opção para desafiar este é o primeiro menu a ser apresentado ao jogador. Neste menu é apresentada uma lista com todos os jogadores presentes na aplicação através de um pedido ao servidor (à exceção do próprio) e é pedido que o jogador selecione quem pretende desafiar. Esta solução foi adicionada à proposta inicial pois facilita todo o processo em si e permite à aplicação fazer toda a gestão do processo de desafios e assim simplificar as tarefas do utilizador.



FIGURA 6.15: Menu para selecção de desafiando

- **Escolhe Tipo de Desafio**

Após seleccionar um jogador para desafiar o jogador navega para este ecrã onde lhe é pedido que escolha o tipo de problema que pretende criar. Existem três opções: Desafio simples (apenas uma operação), desafio médio (problema de duas operações) e desafio complexo (problema de três operações).



FIGURA 6.16: Menu para selecção de tipo de desafio

- **Inserir Dados**

Ao seleccionar o tipo de jogo pretendido o jogador navega para este ecrã e é apresentada uma mensagem de alerta que diz especificamente como o jogador deve inserir os dados nos respectivos campos.

Após fechar a mensagem de alerta é pedido a este que insira os polinómios de cálculo e o respectivo resultado pretendido. Tanto os polinómios como o resultado

para inserir estão numa estrutura em que as variáveis já se encontram presentes e o jogador apenas tem de preencher campos numéricos da forma como foi descrito na mensagem de alerta. Assim foi possível reduzir a quantidade de informação que o jogador tem de inserir de forma a tornar mais fácil para o jogador apresentar um desafio elegível para ser validado, de forma a que seja permitido ao jogador navegar para o ecrã seguinte e apresentar a solução para o mesmo.

Para que um problema seja elegível para validação, para que seja apresentada a sua solução, tem de ser preenchido pelo menos um campo de cada polinómio de cálculo e do resultado. Caso haja algum polinómio ou resultado vazio e o jogador pressionar o botão para que o sistema proceda à verificação da validade dos campos é apresentada uma mensagem de alerta para preencher campos em falta. Se todos os campos estiverem preenchidos mas algum for inválido é-lhe apresentada uma mensagem para isso mesmo, caso contrário o sistema valida os campos e torna visível um botão que levará o jogador ao próximo ecrã.

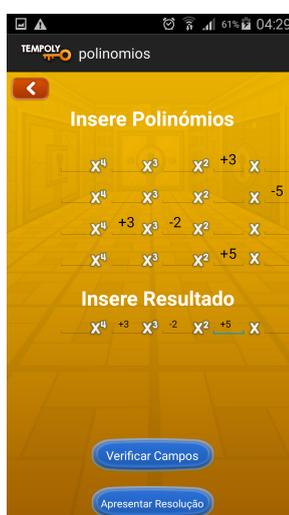


FIGURA 6.17: Menu para inserção de dados

- **Apresenta Solução**

Dependendo do tipo de desafio que o jogador seleccionou este será levado para um modo de jogo diferente (uma, duas ou três operações), e é agora pedido que o jogador apresente a solução em ambiente de jogo, da mesma forma como se o jogador estivesse efectivamente a jogar. São dadas três oportunidades ao jogador para resolver o seu desafio. Se responder com sucesso o desafio é enviado para o servidor e guardado na respectiva tabela de dados e pronto a ser jogado pelo desafiado, se o jogador errar três vezes este será redireccionado para o ecrã anterior de forma a poder corrigir os dados que inseriu de forma a conseguir criar um problema para o qual consiga oferecer uma solução válida.

A nível de implementação foi considerado se se deveria criar um método para verificar se um desafio tem solução sem que seja pedido ao jogador para a apresentar. Após alguma consideração pareceu-me mais coerente se fosse pedido ao jogador para apresentar a sua própria resposta independentemente do problema ter uma ou mais soluções. O desenvolvimento foi feito desta forma para garantir que o problema foi criado com consciência de cálculo por parte do jogador, anulando assim a aleatoriedade na criação de desafios garantindo que o jogador que o criou tem conhecimentos matemáticos para o poder superar.



FIGURA 6.18: Ambiente de Jogo

#### 6.6.6.2 Meus Desafios

Esta funcionalidade não estava proposta na solução inicial mas, como foi descrito anteriormente, para oferecer maior otimização da interação do jogador com a aplicação deixou-se toda a gestão de desafios ser feita pela aplicação/servidor.

Ao selecionar esta opção é apresentado ao jogador uma lista todos os desafios que lhe foram feitos e este pode selecionar qualquer um deles para resolução, caso tenha algum. Ao selecionar um desafio o jogador navega para o ambiente de jogo correspondente ao desafio que foi feito por outro jogador.



FIGURA 6.19: Menu de problemas desafiados ao jogador

Ao entrar no ambiente de jogo para resolver o desafio são oferecidas ao jogador três possibilidades para o fazer. Caso acerte, o jogador navega novamente para o menu dos seus desafios onde o problema resolvido já não constará da lista pois, ao ser resolvido, o problema é imediatamente removido da tabela de dados correspondente. O mesmo verifica caso o jogador erre três vezes o mesmo problema. Em caso de erro o jogador é informado do número tentativas que ainda tem para resolver o desafio proposto.

## 6.7 Requisitos por Alcançar

São agora analisados os requisitos que não foram alcançados na sua totalidade.

Um grande problema que se observou durante o estágio foi a dificuldade de especificação do que realmente se pretendia. Como a prototipagem é um processo dinâmico como foi referido anteriormente, os requisitos são constantemente alterados de forma a refinar e melhorar a aplicação. Esta refinação aliada à grande dificuldade em clarificar o que se pretendia por parte da investigadora levou a uma interpretação errada do funcionamento pretendido para o ambiente de jogo. O ambiente de jogo foi desenvolvido objectivamente e sempre no âmbito de alcançar os objectivos que se pensavam ser os pretendidos mas estes não foram clarificados com antecedência suficiente para ser viável a respectiva alteração já num estágio final do processo de estágio. Durante quase todo o processo a interpretação do que se pretendia para o ambiente de jogo passava por haver diferentes tipos de problema (uma, duas ou três operações), cada um deles um modo de resolução independente e aplicados a níveis distintos. Percebeu-se demasiadamente tarde que a investigadora pretendia que o jogador, de uma forma dinâmica, pudesse

resolver um problema usando qualquer um dos caminhos, ou seja, usando uma, duas ou três operações.

Devido ao acumular de novos requisitos que surgiram nas fases finais do desenvolvimento onde este já continha o design, além de ter sido testado já numa fase próxima do seu estado final não tive tempo para tratar, na sua totalidade, o suporte para diferentes tipos de ecrã. A aplicação, como jogo em si, funciona correctamente em diferentes dispositivos, apenas o menu de níveis, como é um ecrã que contém um número considerável de botões, o seu ajuste aos diferentes ecrãs não foi tratado, pelo que dispositivos com menores resoluções fazem com que os botões não se encontrem devidamente posicionados sobre a linha de progressão, podendo haver a possibilidade dos últimos botões ficarem comprimidos.

# Capítulo 7

## Avaliação

Para testar a aplicação recorreu-se à realização de um inquérito de usabilidade que visa verificar se as funcionalidades implementadas vão de encontro aos objectivos pretendidos.

### 7.1 Processo

Para a realização dos testes anteriormente referidos foram feitas entrevistas individuais que passam pela submissão de um leque de testes a cada entrevistado e a respectiva recolha de dados consoante o seu desempenho/avaliação. Estes testes foram feitos a um grupo de indivíduos pertencentes ao público alvo da aplicação e a outro grupo de indivíduos não pertencente ao mesmo, mais especificamente, a jovens adultos (estudantes universitários).

Em todos os testes foi pedido ao utilizador para executar uma tarefa específica e, para cada métrica, foram recolhidos dados referentes ao parâmetros de tempo de duração da tarefa, erros e dificuldades do testados para realizá-la, observações do entrevistador e a avaliação dada à funcionalidade, numa escala de 1(Muito fraco) a 5 (Muito bom), pelos testados. São agora apresentadas as métricas dos testes.

- **Identificar o Registo e Login**

Esta métrica teve como objectivo perceber se o jogador consegue com facilidade identificar o que fazer no menu inicial, mais precisamente, identificar que é necessário um registo.

- **Entrar no modo de jogo**

Para esta métrica pretendeu-se perceber se o jogador, após autenticação consegue identificar directamente como se dirigir ao modo de jogo.

- **Superar um nível simples**

Com este teste, pretendeu-se perceber se o jogador, no seu primeiro contacto com o ambiente de jogo, consegue rapidamente identificar o que cada elemento do jogo significa e o que o mesmo tem de fazer para superar um nível de dificuldade simples.

- **Superar um nível médio**

Com este teste, pretendeu-se perceber se o jogador, já com alguma experiência com as mecânicas de jogo, consegue identificar a mudança no sistema de jogo de forma a superar este tipo de nível.

- **Superar um nível complexo**

Com este teste, pretendeu-se perceber se o jogador, agora com mais experiência com as mecânicas de jogo, consegue identificar a mudança no sistema de jogo de forma a superar este tipo de nível.

- **Visualizar as suas medalhas**

Neste teste pretendeu-se perceber se o jogador consegue identificar onde se encontra o menu onde estão as medalhas conquistadas pelo mesmo, e se consegue identificar o que estas significam.

- **Visualizar a tabela de classificação**

Neste teste pretendeu-se perceber se o jogador consegue identificar onde se encontra o menu com a tabela de classificação, e os elementos que a constituem.

- **Criar um desafio**

Para este teste, o jogador foi submetido a todo o processo para a criação de um desafio que engloba a navegação através de três menus. Pretendeu-se identificar se o jogador conseguia identificar o que fazer em cada um desses passos.

- **Resolver o desafio criado**

Neste teste pretendeu-se perceber se o jogador consegue identificar que é o seu desafio que lhe é apresentado e que deverá resolvê-lo para que este seja validado pelo sistema.

- **Resolver um desafio**

Foi também pedido ao jogador que identificasse onde se encontram os seus desafios e, após identificação, que resolvesse um deles.

- **Design e outras observações**

Finalmente foi pedido ao jogador para classificar o design e foram também pedidas outras observações que este quisesse oferecer.

## 7.2 Resultados e Análise

Os testes foram feitos a dois grupos distintos, o primeiro constituído por cinco jovens adultos estudantes universitários na Universidade de Coimbra e o segundo constituído por 6 menores, estudantes do terceiro ciclo da escolaridade obrigatória, pertencentes ao público alvo definido para a aplicação.

Primeiramente são apresentadas duas tabelas com as avaliações de cada um dos grupos em relação à tarefa a que foram submetidos de forma a se perceber, de uma forma geral, onde estes apresentaram mais dificuldade, para posteriormente ser analisada métrica a métrica. Para uma melhor percepção da facilidade/dificuldade na realização das tarefas, por parte dos jogadores, são apresentados gráficos de desempenho (tarefa/tempo) nas métricas em que se considerou ser necessário e relevante tendo em conta os resultados. As tarefas mais simples de realização quase imediata (intervalo de tempo entre 1 a 2 segundos) não se considerou oportuno a utilização de gráficos.

### Resultados de avaliação

A avaliação do jogo pelos utilizadores de teste foi bastante importante porque permitiu encontrar alguns problemas que não foram detectados durante a implementação do jogo assim como permitiu também verificar que alguns dos objectivos propostos para este jogo foram cumpridos.

Como foi referido anteriormente, foi pedido a cada jogador para classificar de 1 (muito fraco) a 5 (muito bom) cada tarefa no âmbito de facilidade de percepção de processos e realização da mesma. Nas seguintes tabelas estão apresentados os resultados dessa avaliação em que, na coluna da esquerda estão listadas as tarefas, e as seguintes 5 colunas correspondem à avaliação, onde o número que se encontra nas células corresponde ao número de ocorrências para cada classificação consoante a tarefa.

<b>Teste</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
Identificar o Registo e Login	0	0	0	1	4
Entrar no modo de jogo	0	0	0	0	5
Superar um nível simples	0	0	2	0	3
Superar um nível médio	0	0	0	0	5
Superar um nível complexo	0	0	1	1	3
Visualizar as suas medalhas	0	0	0	0	5
Visualizar a tabela de classificação	0	0	0	0	5
Criar um desafio	0	1	0	4	0
Resolver o desafio criado	0	0	0	1	4
Resolver um desafio	0	0	0	0	5
Design	0	0	0	3	2

TABELA 7.1: Resultados de avaliação de jovens adultos



FIGURA 7.1: Vista geral de avaliação - jovens adultos

Como se pode observar na tabela 7.1 e no gráfico da figura 7.1, a avaliação por parte do grupo de jovens adultos foi bastante positiva havendo apenas uma classificação negativa, 3 satisfatórias, 10 boas e 41 muito boas. Pode-se portanto concluir que, de uma maneira geral, a aplicação apresenta boas condições de uso ainda que hajam tarefas a ser melhoradas como é o caso da criação de um desafio, que será detalhadamente analisada, tal como todas as outras tarefas, nesta secção.

Teste	1	2	3	4	5
Identificar o Registo e Login	0	0	0	0	6
Entrar no modo de jogo	0	0	0	0	6
Superar um nível simples	0	0	0	2	4
Superar um nível médio	0	0	0	1	5
Superar um nível complexo	0	0	0	2	4
Visualizar as suas medalhas	0	0	0	0	6
Visualizar a tabela de classificação	0	0	0	0	6
Criar um desafio	0	0	5	1	0
Resolver o desafio criado	0	0	0	1	5
Resolver um desafio	0	0	0	0	6
Design	0	0	0	0	6

TABELA 7.2: Resultados de avaliação com jovens pertencentes ao público alvo

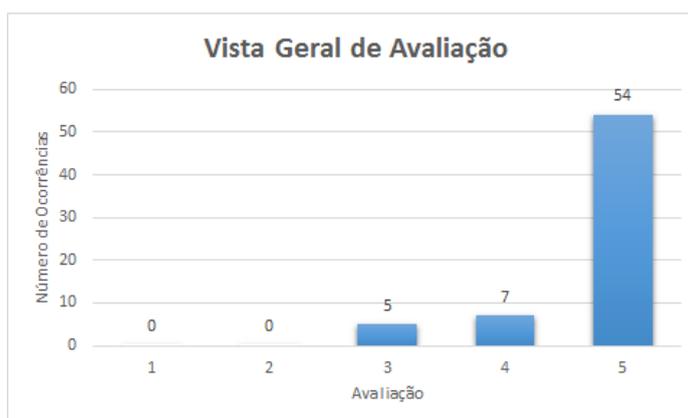


FIGURA 7.2: Vista geral de avaliação - público alvo

Ao se observar a tabela 7.2 e a figura 7.2, o grupo pertencente ao público alvo atribuiu mais avaliações de 5 valores e menos de 3 e 4 valores sendo que não houve nenhuma ocorrência de avaliação negativa. Pode-se concluir que esta avaliação é melhor que a anterior, o que é um dado muito positivo, pois sendo este o grupo de testes pertencente ao público alvo, é de todo o interesse que seja a estes testados que a aplicação mais se enquadre e agrade. Tal como na avaliação dos jovens adultos, a tarefa onde o público alvo mais dificuldades identificou, foi a criação de desafios.

São agora analisados mais pormenorizadamente cada uma das tarefas anteriormente apresentadas.

## Identificar o Registo e Login

Nesta tarefa não é apresentada a duração da tarefa pois os resultados iriam variar consoante o número de caracteres dos dados inseridos pelo jogador e não a dificuldade da tarefa em si. A nível de usabilidade todos os testados identificaram de imediato o que fazer. Percebe-se que tenha sido este o resultado pois é uma tarefa que está presente em quase todas as aplicações.

Houve apenas uma avaliação inferior a 5 (4 neste caso), e a observação do testado prendeu-se ao facto de, no registo, não ser pedido para confirmar a password de forma a evitar erros de registo.

## Entrar no modo de jogo

Tal como na tarefa anterior, nesta tarefa também não são apresentados resultados de duração da mesma pois todos os jogadores identificaram de imediato como aceder ao modo de jogo, neste caso jogar um nível. Este resultado também é um resultado esperado pois o botão "Jogar" foi propositadamente posicionado no topo da lista de tarefas do menu principal, daí uma identificação imediata do testado. Após seleccionar a opção "Jogar" também todos identificaram de imediato o que fazer em seguida, neste caso seleccionar o único nível desbloqueado para aceder ao modo de jogo. Este resultado também é esperado pois muitos jogos que funcionam com mecânicas de progressão funcionam da mesma forma, o que resultou numa unanimidade na avaliação em relação a esta tarefa que foi de 5 valores.

## Superar um nível simples

Esta tarefa é responsável pela primeira interacção do jogador com o ambiente de jogo e é, provavelmente, a tarefa mais importante dos testes de usabilidade que foram realizados.

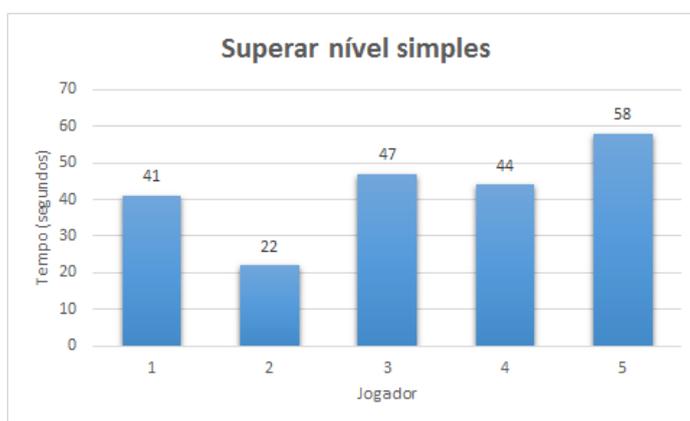


FIGURA 7.3: Tempo de cada jogador para superar primeiro nível simples - jovens adultos

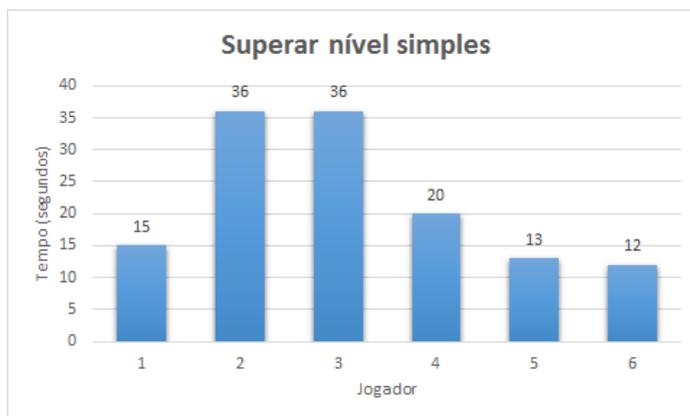


FIGURA 7.4: Tempo de cada jogador para superar primeiro nível simples - público alvo

- Média de tempo dos jovens adultos: **42,4** segundos
- Média de tempo do público alvo: **22** segundos

Observando os dois gráficos e as respectivas médias pode-se observar que a média de tempo do grupo pertencente ao público alvo, além de ser inferior em quase metade do valor, é também igual à resolução mais rápida do grupo de jovens adultos. Pode concluir-se que os elementos pertencentes ao público alvo tiveram mais facilidade em identificar as mecânicas de jogo e aplicaram-nas mais rapidamente. Existe sempre o factor da maior ou menor dificuldade em cálculos matemáticos de cada pessoa, o que considero não se aplicar a este caso pois o primeiro nível é, propositadamente, extremamente simples.

A nível da avaliação oferecida pelos testados, nos jovens adultos houve duas ocorrências de três valores e as restantes duas de cinco valores, já no grupo de público alvo houve duas ocorrências de quatro valores e quatro de cinco valores. Estas avaliações vão de encontro à dificuldade demonstrada por cada grupo para interagir com o jogo e conseguir atingir uma solução final.

## Superar um nível médio

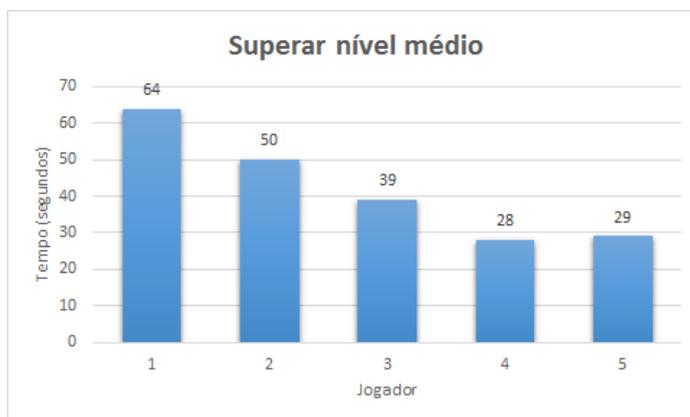


FIGURA 7.5: Tempo de cada jogador para superar primeiro nível médio - jovens adultos

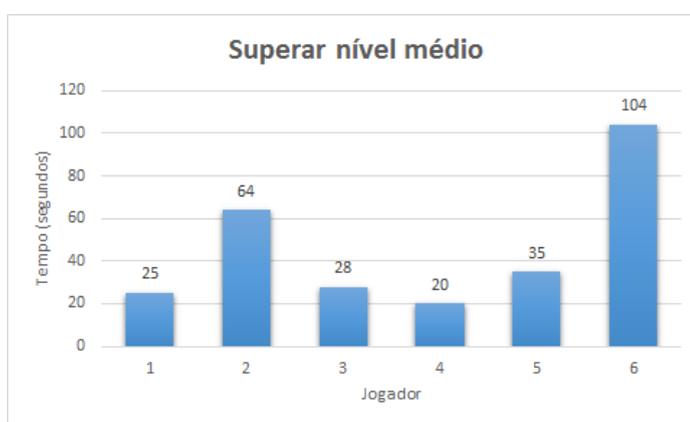


FIGURA 7.6: Tempo de cada jogador para superar primeiro nível médio - público alvo

- Média de tempo dos jovens adultos: **42** segundos
- Média de tempo do público alvo: **46** segundos

Através das informações obtidas nos dois gráficos anteriores, de uma forma geral através da média, pode-se observar que o tempo médio dos jovens adultos foi bastante semelhante ao do tipo de nível anterior, ao contrário do tempo médio dos elementos do público alvo que subiu acentuadamente, neste caso, para mais do dobro e é agora superior ao do grupo de jovens adultos.

Ao se observar mais detalhadamente, no gráfico do público alvo, pode-se constatar que há dois valores que se destacam dos demais, sendo que um deles apresenta uma discrepância muito acentuada. A razão deste resultado pode ser atribuída a duas causas distintas: o jogador demorou mais tempo para identificar a nova estrutura de jogo, ou o maior número de operações matemáticas implicaram, neste caso específico, uma

maior dificuldade para atingir o resultado pretendido ou, muito provavelmente, ambas em simultâneo. Se este resultado não estivesse presente a média seria substancialmente inferior e também inferior à dos jovens adultos, mas ainda assim superior à media do modo de jogo anterior do próprio grupo.

Os resultados do grupo de jovens adultos diminui muito ligeiramente. Tendo em conta que o número de operações a aplicar é maior, o nível de dificuldade matemática é igualmente maior logo, este resultado é fruto de uma maior familiaridade com as mecânicas de jogo.

A nível de avaliações por parte dos testados, todos avaliaram esta tarefa com 5 valores à excepção de um elemento do grupo do público alvo que ofereceu 4 valores à mesma. Analisando estas avaliações percebe-se que os jovens adultos tiveram bastante mais facilidade em lidar com as mecânicas de jogo no novo ambiente de jogo bem como os jogadores do público alvo, à excepção de uma ocorrência.

### Superar um nível complexo



FIGURA 7.7: Tempo de cada jogador para superar primeiro nível complexo - jovens adultos



FIGURA 7.8: Tempo de cada jogador para superar primeiro nível complexo - público alvo

- Média de tempo dos jovens adultos: **77,8** segundos
- Média de tempo do público alvo: **114,67** segundos

Para superar este tipo de nível, conforme se pode constatar, as médias de ambos os grupos subiram acentuadamente, especialmente as médias do público alvo que foi mais de duas vezes superior, comparativamente ao modo de jogo anteriormente analisado.

A nível de usabilidade não se registou qualquer dificuldade por parte dos jogadores para oferecer uma solução daí concluir-se que estes resultados tomaram estes valores devido à dificuldade que os diferentes públicos apresentaram a nível de cálculo. Tendo em conta que os jovens adultos são todos alunos do ensino superior, uma vez que interiorizaram as mecânicas de jogo, é normal que numa maior complexidade de cálculo apresentem melhor desempenho que alunos do terceiro ciclo.

Ao observar-se mais detalhadamente os gráficos pode-se perceber que existe uma variação mais acentuada em cada um deles. No entanto, são variações de desempenho superiores às restantes, o que pode ser explicado pela maior facilidade dos jogadores em questão em processos de cálculo matemático já que nenhum jogador testado demonstrou qualquer dificuldade com as mecânicas de jogo mas sim na matemática em si. Como este é um jogo de progressão em que é aplicada uma dificuldade crescente à medida que os níveis vão sendo superados, estes resultados são os resultados esperados.

A nível de avaliação, os jovens adultos atribuíram uma avaliação com 3 valores, uma com 4, e três com 5, já os jogadores do público alvo atribuíram duas avaliações de 4 valores e as restantes quatro com 6 valores. Nota-se um ligeiro decrescer na qualidade das avaliações, de uma certa forma porque ao haver mais botões no ecrã, a sua área de deteção para posicionamento é menor, o que torna um pouco mais difícil posicionar os botões, tal como foi identificado por alguns testados.

### **Visualizar Medalhas**

Esta tarefa foi identificada e realizada com bastante facilidade por todos os jogadores testados num intervalo de tempo bastante reduzido. As avaliações dos jogadores justificam o que foi anteriormente referido pois todos atribuíram cinco valores a esta tarefa.

### **Visualizar tabela de classificação**

Tal como na tarefa anterior, esta tarefa também foi realizada com bastante sucesso por todos os jogadores e mais uma vez todos atribuíram uma avaliação de 5 valores à mesma.

## Criar um desafio

Esta é a tarefa mais complexa do jogo e foi nela que todos os jogadores apresentaram maior dificuldade, especialmente na inserção de dados. A tarefa de criação de um desafio divide-se em 5 subtarefas distintas:

- Identificar e seleccionar menu do modo de desafio no menu principal
- Identificar e seleccionar para desafiar
- Identificar e seleccionar jogador para desafiar
- Identificar e seleccionar modo de jogo pretendido para o desafio
- Inserir dados

As primeiras quatro subtarefas foram realizadas por todos os jogadores quase de imediato. Todos identificaram com facilidade o que tinham de fazer em cada uma delas no âmbito da realização da tarefa que lhes foi proposta, desafiar um jogador. A grande dificuldade verificou-se na inserção de dados devido à complexidade que implica inserir valores de forma a compor polinómios válidos.

Nesta tarefa apenas são analisados os dados referentes ao tempo de duração recolhidos com o público alvo porque os dados recolhidos com o grupo de jovens adultos não têm a precisão necessária para serem analisados. Esta situação deve-se ao facto de terem sido feitos por aproximação.



FIGURA 7.9: Tempo de cada jogador para criar um desafio - público alvo

Como se pode observar no gráfico da figura 7.9, o tempo despendido pelo público alvo na inserção de dados até conseguir oferecer uma proposta que seja validada pelo sistema é bastante elevado. As aproximações feitas com o grupo de jovens adultos esteve também

inserido num intervalo de tempo bastante comprido. Esta dificuldade deve-se ao facto de haver muitos campos de inserção de dados presentes no ecrã o que torna a tarefa um pouco confusa porque nem todos identificaram o facto de não terem que preencher todos os campos. Todos conseguiram finalizar o processo mas foi através de processo de tentativas. De referir também que existe uma mensagem de alerta que contém uma explicação para se inserir dados que foi ignorada pela maior parte dos jogadores, o que dificultou o processo. Todos os jogadores identificaram que, depois de pela primeira vez conseguirem inserir os dados com sucesso, conseguiriam repetir o processo com bastante mais facilidade, o que leva a concluir que um melhoramento a nível de intuitividade do processo resultará em melhores desempenhos.

A dificuldade evidenciada é justificada nas avaliações oferecidas por todos os jogadores pois esta é claramente a tarefa de menos qualidade para os utilizadores. O grupo de jovens alunos, embora sem resultados por razões justificadas anteriormente, teve uma ocorrência de 2 valores e quatro de 4 valores na avaliação, já o público alvo registou cinco ocorrências de 3 valores e uma de 4.

### Resolver desafio criado



FIGURA 7.10: Tempo de cada jogador para resolver desafio criado - jovens adultos

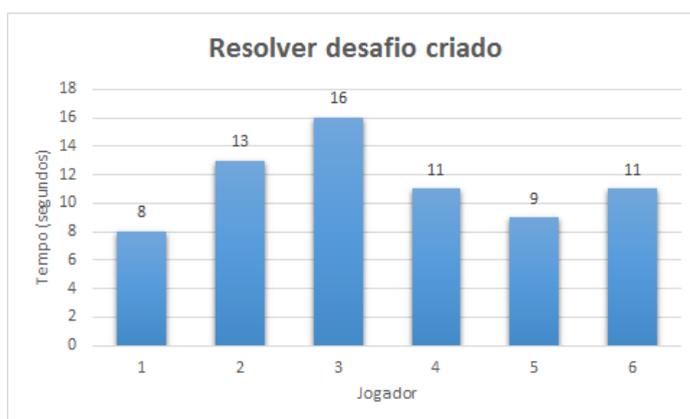


FIGURA 7.11: Tempo de cada jogador para resolver desafio criado - público alvo

- Média de tempo dos jovens adultos: **10,25** segundos
- Média de tempo do público alvo: **11,33** segundos

Primeiro que tudo, como se pode observar, há um valor no gráfico da figura 7.10 que se encontra a 0. Este resultado verificou-se porque o jogador criou o desafio aleatoriamente ao inserir dados tanto nos polinómios como no resultado, logo quando foi pedido ao mesmo para apresentar a solução de forma a ser finalizada a criação do desafio, este não a conseguiu apresentar pois não tinha resolução. Desta forma, a média apresentada para os jovens adultos é referente apenas ao restantes valores obtidos.

Como se pode observar através das médias, os dois grupos apresentam valores semelhantes e bastante inferiores ao valores obtidos em testes anteriores. Estes resultados verificam-se por três razões: os jogadores criaram (à excepção de um) os problemas com consciência de cálculo e resolução, facilmente perceberam que o problema apresentado era constituído pelos dados que tinham inserido e, finalmente, demonstraram grande facilidade em dominar as mecânicas de jogo derivado da experiência obtida.

A nível de avaliações, no grupo de jovens adultos houve uma ocorrência de 4 valores e quatro de 5 valores, no grupo do público alvo houve também uma ocorrência de 4 valores e as restantes(cinco) de 5 valores. Estas avaliações vão de encontro aos resultados obtidos pois demonstram a facilidade que tiveram em identificar o que lhes foi pedido.

### Resolver um desafio



FIGURA 7.12: Tempo de cada jogador para resolver desafio - jovens adultos



FIGURA 7.13: Tempo de cada jogador para resolver desafio - público alvo

- Média de tempo dos jovens adultos: **14** segundos
- Média de tempo do público alvo: **18,33** segundos

Nesta tarefa, tal como na tarefa anterior, também se verifica um valor a 0 mas desta vez este valor verificou-se porque um dos testados do grupo de jovens adultos errou propositadamente as três tentativas que são fornecidas para resolver o desafio de forma a perceber como o sistema se comportava com essa situação.

A nível de comparação de médias pode-se observar que a média do grupo do público alvo ainda é consideravelmente maior considerando o comprimento do intervalo tempo. Se se observar mais detalhadamente pode-se também verificar que esse mesmo grupo tem um valor mais de duas vezes superior a qualquer outro e sem este valor, a média do grupo seria ligeiramente inferior a 14, ou seja, muita próxima, e até mesmo inferior, da média do outro grupo.

Numa análise mais detalhada ao valor acima referido, tentou-se perceber a razão que levou a esta discrepância. A razão mais provável é a dificuldade do problema matemático proposto porque como os desafios não funcionam como os níveis (nível de dificuldade crescente) pois cada desafio tem a dificuldade incutida no mesmo definida pelo desafiador. Esta é a justificação mais plausível pois o jogador que registou estes dados não revelou qualquer dificuldade a nível de usabilidade e identificou com facilidade o que lhe foi apresentado até porque já tinha tido uma experiência com o tipo de jogo apresentado.

Quanto à avaliação de usabilidade desta tarefa a opinião foi unânime, todos os jogadores de cada grupo ofereceram 5 valores à mesma o que permite concluir que esta tarefa foi concluída por todos os jogadores com enorme facilidade.

## Design

No que toca ao design visual da aplicação, depois de concluídas todas as tarefas foi pedido a cada jogador para dar a sua opinião e avaliação ao mesmo.

O grupo de jovens adultos, na sua grande maioria identificou o design enquadrado com o público alvo pretendido, havendo mais um ou outro reparo positivo em relação à ideia das portas. Neste grupo houve três ocorrências de 4 valores e duas de 5 valores.

Já o grupo pertencente ao público alvo ofereceu uma diversidade de opiniões bastante maior e de uma forma menos formal, ou seja, acabaram por adjectivar mais do que o grupo anterior. De entre as opiniões oferecidas são agora apresentadas algumas delas: Consideraram agradável, muitos acharam engraçado o tema das portas, fechaduras e templo, consideraram original, que não é complicado e é fácil de compreender. Neste grupo as avaliações de 1 a 5 valores foram todas de 5 valores.

Em jeito de análise, e tendo em conta que o jogo foi feito e desenhado para jovens do terceiro ciclo, a nível de design os resultados vão de encontro ao se pretendia inicialmente, acima de tudo satisfazer o seu público alvo e que se verifica através das avaliações do grupo pertencente ao mesmo. As avaliações oferecidas pelos jovens adultos também se compreendem especialmente através das suas observações. Todos classificaram positivamente o design e fizeram referência que, para o seu público alvo, se enquadra bem, mas para os mesmos já deixa de ser algo tão atractivo ou que estimule os mesmos.

### 7.3 Observações

Foi pedido aos jogadores entrevistados, após a realização dos testes que oferecessem reparos e observações caso considerassem relevante. Estas observações tiveram *feedback* positivo e métricas a melhorar, mas como foram recolhidas no âmbito de melhorar a aplicação para esta ir de encontro às necessidades do utilizador são listadas as que foram identificadas como algo a melhorar. Algumas dessas observações foram as seguintes:

- Considera que a sensibilidade dos botões no último nível é pequena (dificuldade em posicioná-los)
- Considera que criar o desafio é a parte mais difícil da aplicação (nomeadamente a inserção de dados) mas todos identificaram que conseguiriam repetir o processo com muito mais facilidade
- Considera que podia haver uma melhor explicação de como inserir dados para desafiar

- Considera que deveria haver um pequeno tutorial que explicasse como se joga ao registrar
- Considera que ao posicionar-se um botão sobre uma posição já ocupada o botão anteriormente posicionado deveria ser substituído pelo novo botão arrastado

Estas observações recolhidas podem e devem ser interpretadas como trabalho para o futuro no sentido de moldar a aplicação o mais possível ao seu público alvo bem como melhorar a sua usabilidade.

## Capítulo 8

# Conclusões e Trabalho para o Futuro

### 8.1 Conclusões

A aplicação foi implementada com sucesso e encontra-se totalmente funcional. As avaliações recolhidas e analisadas no capítulo anterior demonstram, na sua grande maioria, resultados bastante positivos que justificam o sucesso da mesma a nível de usabilidade.

No que toca a objectivos para a aplicação e o jogo em si, quase todos foram alcançados dentro do que se idealizou que seria a aplicação. Como o processo de prototipagem é um processo dinâmico, que implica alterações durante todo o seu percurso, acabaram por se verificar algumas alterações na formulação do ambiente de jogo, que perto do final deste mesmo processo, implicaram a não conclusão dessas mesmas alterações. No entanto a aplicação foi sempre desenvolvida com uma ideia fixa daquilo que se pensou que era o pretendido, ideia esta correspondente ao produto final. Pode-se então concluir que a aplicação foi desenvolvida na sua totalidade e preparada para testes de campo.

A nível pessoal foi-me permitido aprofundar alguns conhecimentos, que foram recolhidos durante todo o processo de licenciatura e mestrado, que já tinha no âmbito da formulação de todo um processo de engenharias, bem como aprender muitos mais. A oportunidade que me foi dada neste estágio, e mesmo o lidar com uma cliente, ofereceu-me um pouco de visão daquilo que é o mercado de trabalho e o quão sensível a informação dada por um cliente pode ser, de forma a oferecer o que este pretende. Foi-me também permitido lidar com a necessidade e importância que tem a estruturação de todo um projecto, de forma a se obter um desenvolvimento mais objectivo e sem alterações, entre outras métricas,

que tornaram todo este estágio um verdadeiro artefacto para ser utilizado durante todo o meu futuro profissional.

## 8.2 Trabalho para o Futuro

Dada a intenção que motivou o desenvolvimento do jogo descrito anteriormente, é naturalmente importante realizar testes de campo de forma a perceber o seu impacto numa amostra suficientemente grande para se concluir se esta, efectivamente, é responsável por um melhoramento significativo no desempenho dos alunos, em cálculos entre polinómios. Caso esses testes apresentem resultados positivos considera-se viável um melhoramento da aplicação em si, através da inserção de mais funcionalidades e a correcção daquelas que não foram atingidas na sua totalidade.

Existem também melhorias a implementar na mecânica do jogo. De facto, háverá uma mais-valia em torná-la dinâmica, ou seja, em qualquer problema o jogador pode arrastar o número de operadores e operandos que entender para resolver um problema, de forma a ir de encontro à ideia da investigadora. Ainda que considere que a nível de usabilidade teria de ser muito bem trabalhado, pois pode tornar-se confuso.

A adaptação a diferentes resoluções de ecrãs também pode ser melhorada, nomeadamente no menu dos níveis, como foi identificado anteriormente.

A inserção de dados para criar um desafio deve ser melhorada pois está um pouco rígida e, a nível de usabilidade, acaba por ser a tarefa mais difícil para os utilizadores realizarem como foi comprovado nos testes de usabilidade.

Caso os testes revelem resultados promissores, e de forma a aumentar a sua base de potenciais utilizadores, outras plataformas deverão ser consideradas como iOS ou Windows Phone.

# Bibliografia

- [1] Cândida Barros e Ana Carvalho. Jogo mobile para o estudo de operações com polinómios: uma proposta. *Aprendizagem Online - Atas do III Congresso Internacional das TIC na Educação*, 2014.
- [2] Android Website, 1 2015. URL <https://android.com/>. Data da última consulta: 06/07/15.
- [3] Maria Roussou. Learning by doing and learning through play: An exploration of interactivity in virtual environments for children. Technical report, University College London, 2004.
- [4] Katie Salen e Eric Zimmerman. *Rules of Play: Game Design Fundamentals*. MIT Press, 2003.
- [5] James Paul Gee. *What Video Games Have to Teach Us About Learning and Literacy?* Palgrave Macmillan, 2003.
- [6] Claudia Groenwald. Utilizando curiosidades e jogos matemáticos em sala de aula. 2013. URL <http://www.somatematica.com.br/artigos/a1/>. Data da última consulta: 06/07/15.
- [7] Jordan Shapiro. Video games are the perfect way to teach math, says stanford mathematician. *Forbes*, Agosto 2013.
- [8] Keith Devlin. *Mathematics Education for a New Era: Video Games as a Medium for Learning*. A. K. Peters, Ltd., 201.
- [9] Ernest Adams e Joris Dormans. *Game Mechanics: Advanced Game Design*. New Riders, 2003.
- [10] Jesse Schell. *The Art of Game Design: A Book of Lenses*. Morgan Kaufmann, 2008.
- [11] Clark C. Abt. *Serious Games*. Viking Press, 1970.

- 
- [12] Serious Games University Website, 1 2015. URL <https://seriousgameuniversity.com/>. Data da última consulta: 27/01/15.
- [13] Mary Ulicsak e Martha Wright. Games in education: Serious games. Technical report, Futurelab, 2010.
- [14] Serious Games, 1 2015. URL [http://www.seriousgameuniversity.com/index.php?option=com\\_content&view=article&id=22&Itemid=149](http://www.seriousgameuniversity.com/index.php?option=com_content&view=article&id=22&Itemid=149). Data da última consulta: 27/01/15.
- [15] Jogo Math Puzzles for Toddlers, 1 2015. URL <https://play.google.com/store/apps/details?id=co.romesoft.toddlers.puzzle.math>. Data da última consulta: 27/01/15.
- [16] Jogo Math Maniac, 1 2015. URL <https://play.google.com/store/apps/details?id=de.math.maniac>. Data da última consulta: 27/01/15.
- [17] Jogo King of Math, 1 2015. URL <https://play.google.com/store/apps/details?id=com.troubi.kingofmath>. Data da última consulta: 27/01/15.
- [18] Jogo Tricky Maths, 1 2015. URL <https://play.google.com/store/apps/details?id=com.rohitporwal.trickymaths>. Data da última consulta: 27/01/15.
- [19] Ciclos de Engenharia de Software, 1 2015. URL <http://www.ebah.pt/content/ABAAABu1wAL/ciclos-engenharia-software>. Data da última consulta: 27/01/15.
- [20] Ian Sommerville. *Software Engineering*. Addison-Wesley, 2013.
- [21] Priorização de Requisitos, 1 2015. URL <http://businessanalystlearnings.com/ba-techniques/2013/3/5/moscow-technique-requirements-prioritization>. Data da última consulta: 27/01/15.
- [22] UML basics: The component diagram, 1 2015. URL <http://www.ibm.com/developerworks/rational/library/dec04/bell/>. Data da última consulta: 06/07/15.
- [23] UML basics: The sequence diagram, 1 2015. URL <http://www.ibm.com/developerworks/rational/library/3101.html>. Data da última consulta: 06/07/15.
- [24] Oracle Website, 1 2015. URL <https://oracle.com/>. Data da última consulta: 06/07/15.
- [25] Eclipse website, 1 2015. URL <https://eclipse.org/>. Data da última consulta: 06/07/15.

- 
- [26] Android Developer Tools - Eclipse, 1 2015. URL <http://developer.android.com/tools/sdk/eclipse-adt.html>. Data da última consulta: 27/01/15.
- [27] Android Storage Options, 1 2015. URL <http://developer.android.com/guide/topics/data/data-storage.html#db>. Data da última consulta: 27/01/15.
- [28] SQLite Features, 1 2015. URL <http://sqlite.org/features.html>. Data da última consulta: 27/01/15.
- [29] MySQL Website, 1 2015. URL <https://www.mysql.com/>. Data da última consulta: 06/07/15.
- [30] Servidor REST e Android, 1 2015. URL <http://programmerguru.com/android-tutorial/android-restful-webservice-tutorial-part-1/>. Data da última consulta: 06/07/15.
- [31] Learn REST: A tutorial, 1 2015. URL <http://rest.elkstein.org/2008/02/what-is-rest.html>. Data da última consulta: 06/07/15.
- [32] Pawel Glowacki. Why use "rest" architecture for web services? URL <http://edn.embarcadero.com/article/40467>. Data da última consulta: 06/07/15.
- [33] SQLite Website, 1 2015. URL <https://sqlite.org/>. Data da última consulta: 06/07/15.
- [34] Vogella Website, 1 2015. URL <https://vogella.com/>. Data da última consulta: 06/07/15.
- [35] Android Developers Website, 1 2015. URL <https://developer.android.com/>. Data da última consulta: 06/07/15.
- [36] Balsamiq Website, 1 2015. URL <https://balsamiq.com/>. Data da última consulta: 06/07/15.
- [37] Textstudio Website, 1 2015. URL <https://textstudio.sourceforge.net/>. Data da última consulta: 06/07/15.