

Instituto Superior de Engenharia do Porto

## **Jogos Sérios para Lean Manufacturing**

**“Jogo: Método 5S”**

**Duarte Sousa Ferreira Gomes**

Dissertação para obtenção do grau de Mestre em

**Engenharia Informática**

Área de especialização em

**Sistema Gráficos e Multimédia**

**Orientador:** Engenheiro Carlos Miguel Miranda Vaz de Carvalho do Departamento de Engenharia Informática do Instituto Superior de Engenharia do Porto

**Co-orientador:** Engenheiro Manuel Joaquim Pereira Lopes do Departamento de Engenharia Mecânica do Instituto Superior de Engenharia do Porto

**Júri:**

Presidente:

João Paulo Jorge Pereira, Professor Adjunto, ISEP

Vogais:

António Fernando Vasconcelos Cunha Castro Coelho, Professor Auxiliar, FEUP

Carlos Miguel Miranda Vaz de Carvalho, Professor Adjunto, ISEP

Manuel Joaquim Pereira Lopes, Professor Adjunto, ISEP

Porto, Outubro 2011



*Aos meus Pais e Irmãos*



## Nota Prévia

Jogos Sérios, ou *Serious Games*, podem ter ainda outras designações como jogos de simulação (*Gaming Simulations*), jogos digitais baseados em educação (*digital game-based learning*) ou “jogos que se devem jogar” (games you have to play) [Derryberry, 2007].

O termo “Jogo Sério” é usado para qualquer jogo real, ou digital, cujo objectivo é a aprendizagem de conteúdos didácticos.



## Resumo

A crise económica global que se faz sentir leva a que surja uma necessidade de aumentar a competitividade das empresas, através da melhoria de performance dos seus colaboradores.

Nesse sentido, *Lean Manufacturing* é uma área que merece a atenção das empresas e dos seus colaboradores, uma vez que os seus principais objectivos são o aumento da produtividade, a redução dos desperdícios e a optimização dos recursos disponíveis.

Uma ferramenta do *Lean* que merece especial destaque é o Método 5S, que permite aumentar a produtividade através do aumento da organização do posto de trabalho.

Uma vez que usar postos de trabalho a mais, de modo a que os colaboradores pratiquem e aprendam a aplicar as regras do 5S, fica dispendioso em tempo de trabalho e custos acrescidos pela paragem da linha produção, surge a necessidade de criação de um Jogo Sério que ajude na aprendizagem destes temas.

Um breve estudo do mercado revelou que esta necessidade não está a ser devidamente respondida. Existem algumas entidades que se dedicam ao ensino e partilha deste tipo de conhecimentos, sendo poucas as que têm ferramentas digitais disponíveis.

Esta dissertação propõe um Jogo Sério que pretende dar resposta a esta necessidade, permitindo pôr em prática o Método 5S, servindo ainda como ferramenta motivadora, para que o colaborador aprofunde o seu conhecimento na matéria.

Esta abordagem prática encoraja o jogador a aprender por tentativa e erro. Informações mostradas ao longo do jogo permitem uma aprendizagem sem haver uma sobrecarga cognitiva.

A avaliação do jogo, realizada por um grupo de alunos que frequentam o Mestrado em Engenharia Mecânica, demonstrou que a nossa abordagem é eficiente e resulta na aprendizagem dos conceitos 5S.

**Palavras-chave:** Jogos Sérios, *Lean*, *Lean Manufacturing*, Método 5S.



## Abstract

The global economic crisis that is felt is leading to the emergence of a need to increase business competitiveness, by improving the performance of its employees.

In this sense, *Lean Manufacturing* is an area that deserves the attention of employers and their employees, since its main objectives are to increase productivity, reduce waste and optimize available resources.

A *Lean* tool that deserves special attention is the 5S Method, which can improve productivity, by increasing the organization of the workplace.

Since the use of real workstations, so that employees can practice and learn to apply the 5S rules, is expensive in time and costs increase due to stopping the production line, there is a need to create a Serious Game that helps learning these subjects.

A brief survey of the market showed that this need is not being adequately answered. There are several organizations dedicated to teach and share this type of knowledge and few of this have digital tools available.

This dissertation proposes a Serious Game that aims to respond to this need, by allowing in practice the 5S Method, still serving as a motivational tool for the developer to deepen their knowledge in the field.

This practical approach encourages the player to learn by trial and error. Information displayed throughout the game will allow learning, without a cognitive overload to the player.

The game evaluation, held by a group of students working to get a Master's Degree in Mechanical Engineering, has shown that our approach is efficient and results in learning 5S concepts.

**Key Words:** Serious Games, *Lean*, *Lean Manufacturing*, 5S Method.



## Agradecimentos

Começo por agradecer aos meus Pais e Irmãos, pelo apoio, motivação e suporte incansável que me deram durante estes meses de trabalho. Sem eles não conseguia chegar onde cheguei e agradeço-lhes do fundo do coração o que têm feito por mim.

Agradeço ao Senhor Prof. Doutor Carlos Vaz de Carvalho, pela sua orientação, suporte e aconselhamento que me facultou ao longo deste trabalho. É, sem dúvida, a pessoa responsável pelo sucesso deste projecto.

Ao Senhor Engenheiro Pereira Lopes, pela pertinência das suas sugestões e ensinamentos na área de conhecimento desta Dissertação.

Ao Lisarte Barbosa, pela sua amizade. As horas que trabalhámos juntos e as ideias partilhadas foram sem dúvida uma mais-valia na execução deste projecto.

Ao David Gouveia, pela amizade, pelas suas sugestões, que se revelaram bastante pertinentes e pelo auxílio nos momentos difíceis deste projecto.

Ao Duarte Lopes, agradeço o companheirismo, a calorosa recepção no ambiente de trabalho e as palavras de encorajamento.

Sem a vossa ajuda não teria conseguido realizar este trabalho.

A todos, muito obrigado!



# Índice

## Conteúdo

Nota Prévia.....	v
Resumo.....	vii
Abstract.....	ix
Agradecimentos.....	xi
Índice.....	xiii
Lista de Figuras.....	xvii
Lista de Tabelas.....	xix
Lista de Gráficos.....	xix
Lista de Código.....	xix
Lista de Abreviaturas.....	xxi
1. Introdução.....	1
1.1. Objectivos.....	1
1.2. Motivação.....	2
1.3. Organização da Dissertação.....	2
2. Contexto e Estado da Arte.....	5
2.1. Definição de Jogos Sérios.....	8
2.2. Jogos Sérios na Aprendizagem.....	9
2.3. Jogos Sérios na Gestão.....	11
2.3.1. Tipos de plataformas utilizadas.....	11
2.4. Jogos Sérios em <i>Lean Manufacturing</i> .....	13
2.4.1. Exemplos de jogos digitais de <i>Lean Manufacturing</i> .....	14
2.4.2. O que é o Método 5S?.....	15
2.4.3. Utilização de Jogos Sérios para ensinar o Método 5S.....	16
2.5. Tecnologias a utilizar.....	17
2.5.1. Escolha da tecnologia a usar para a implementação.....	18
3. Metodologia de Investigação e Desenvolvimento.....	23

3.1.	Processo de Desenvolvimento.....	23
3.1.1.	Iniciação .....	24
3.1.2.	Definição .....	25
3.1.3.	Especificação e Design.....	26
3.1.4.	Desenvolvimento.....	28
3.1.5.	Entrega.....	28
3.1.6.	Avaliação.....	29
4.	Concepção .....	31
4.1.	Ideia do jogo.....	31
4.1.1.	Concepção dos cenários da aplicação.....	32
4.1.1.2.	Cenário 2 - Posto de trabalho fabril.....	36
4.1.1.3.	Cenário 3 - Posto de trabalho num escritório de uma empresa .....	37
4.1.1.4.	Cenário 4 - Ambiente do computador .....	39
4.2.	Desenho gráfico dos cenários .....	40
4.2.1.	Cenário 1 .....	41
4.2.2.	Cenário 2 .....	44
4.2.3.	Cenário 3 .....	47
4.2.4.	Cenário 4 .....	49
4.3.	Desenho da estrutura do jogo e cenário escolhido .....	50
4.3.1.	Diagrama da estrutura do jogo .....	51
4.3.2.	Núcleo de funcionalidades .....	51
4.3.2.1.	Interacções.....	51
4.3.3.	Eventos e variáveis.....	52
4.3.4.	Interface com o utilizador.....	52
4.3.5.	Lista de materiais.....	52
5.	Implementação do Trabalho.....	55
5.1.	Explicação do menu do jogo .....	55
5.2.	Explicação do quarto cenário – Ambiente de Computador .....	60
5.2.1.	Explicação dos ecrãs iniciais do cenário .....	60

5.2.2.	Explicação técnica do funcionamento do jogo .....	65
5.2.3.	Explicação das áreas do cenário – Cronometro e botão salvar.....	70
5.2.4.	Explicação das áreas do cenário – Área de entrega do documento .....	73
5.2.5.	Explicação das áreas do cenário – Janela de informações.....	74
5.2.6.	Explicação das áreas do cenário - Submenu.....	74
5.2.7.	Explicação dos objectos do cenário - Zonas.....	77
5.2.8.	Explicação dos objectos do cenário – Documentos.....	82
5.2.9.	Explicação dos objectos do cenário – Atalho.....	86
5.2.10.	Explicação dos objectos do cenário – Pastas.....	87
5.2.11.	Explicação dos objectos do cenário – Reciclagem.....	91
5.2.12.	Explicações teóricas da programação das classes do cenário.....	104
6.	Avaliação do Trabalho .....	109
6.1.	Instrumentos de recolha de dados.....	109
6.1.1.	Método de avaliação utilizado.....	109
6.2.	Questionário para a avaliação do Jogo .....	109
6.3.	Parâmetros a avaliar .....	110
6.4.	Apresentação dos resultados.....	110
6.5.	Discussão dos resultados obtidos .....	112
6.5.1.	Comentários feitos verbalmente .....	118
6.5.2.	Apontamentos escritos nas folhas do questionário.....	119
6.5.3.	Avaliação final .....	119
7.	Conclusões .....	121
7.1.	Retrospectiva do trabalho desenvolvido.....	121
7.2.	Revisão do Objectivos e Contribuições da Dissertação .....	121
7.3.	Auto-avaliação do trabalho realizado.....	122
7.4.	Dificuldades Sentidas .....	122
7.5.	Resultados .....	122
7.6.	Desenvolvimentos futuros .....	123
7.7.	Considerações finais.....	123

7.8. Enceramento.....	123
8. Referencias Bibliográficas .....	125
Bibliografia .....	125
9. Anexos.....	129
9.1. Anexos A - Diagramas Use-Case dos cenários .....	129
9.2. Anexos B – Questionário de Avaliação.....	132
9.3. Anexos C – Slides da apresentação da avaliação do jogo .....	133

## Lista de Figuras

Ilustração 1 – Área de trabalho do <i>Adventure Game Studio</i> .....	19
Ilustração 2 – Ambiente de trabalho do <i>Wintermute Engine</i> .....	20
Ilustração 3 – Ambiente de trabalho do <i>Visionaire Studio</i> .....	21
Ilustração 4 – Diagrama da Estrutura de Desenvolvimento do Projecto.....	24
Ilustração 5 – Ecrã com a vista do Armazém desorganizado .....	41
Ilustração 6 – Cena da secretária organizada/desorganizada .....	42
Ilustração 7 – Cena com mais corredores do Armazém.....	42
Ilustração 8 – Cena com a prateleira seleccionada.....	43
Ilustração 9 – Vista aérea para organização estrutural do Armazém .....	44
Ilustração 10 – Cena com a caixa de ferramentas, bancada e parede.....	44
Ilustração 11 – Exemplo de uma caixa de ferramentas e bancada de trabalho .....	45
Ilustração 12 – Cena com a caixa de ferramentas e gavetas.....	45
Ilustração 13 – Cena com uma das gavetas aberta, da caixa de ferramentas .....	46
Ilustração 14 – Cena da bancada de trabalho e da parede de ferramentas .....	46
Ilustração 15 – Secretária desorganizada .....	47
Ilustração 16 – Exemplo de postos de escritório desorganizados .....	47
Ilustração 17 – Exemplos de ferramentas de organização.....	47
Ilustração 18 – Secretária organizada.....	48
Ilustração 19 – Exemplos de postos de escritório organizados .....	48
Ilustração 20 – Cena para a criação da raiz de ficheiros .....	49
Ilustração 21 – Ambiente de trabalho antes de ser organizado .....	49
Ilustração 22 – Exemplos de ambientes de trabalho desorganizados.....	50
Ilustração 23 – Exemplo de um ambiente de trabalho organizado.....	50
Ilustração 24 – <i>Wireframe</i> dos ecrãs do jogo .....	51
Ilustração 25 – Primeiro ecrã do jogo .....	55
Ilustração 26 – Ecrã de Créditos .....	56
Ilustração 27 – Tabela de pontuações do jogo .....	57
Ilustração 28 – Área de consulta dos materiais de estudo.....	57
Ilustração 29 – Menu de escola de cenário a jogar.....	58
Ilustração 30 – Primeiro ecrã: Ecrã de carregamento do cenário.....	60
Ilustração 31 – Ecrã de autenticação (esquerda por preencher, direita preenchido) .....	61
Ilustração 32 – Listagem das classes criadas para o quarto cenário.....	66
Ilustração 33 – Ecrã de área do jogo .....	69
Ilustração 34 – Área do cronómetro e botão salvar o jogo.....	70
Ilustração 35 – Área de entrega de um documento .....	73

Ilustração 36 – Submenu aberto com o botão direito do rato.....	74
Ilustração 37 – Exemplo de uma Zona chamada Salvar .....	77
Ilustração 38 – Processo de criação de Zona (Criar Forma) .....	78
Ilustração 39 – Exemplos de documentos que se podem adicionar ao cenário.....	82
Ilustração 40 – Visualização do conteúdo de um documento .....	83
Ilustração 41 – Exemplo de um Atalho.....	86
Ilustração 42 – Exemplos de pastas que se podem adicionar ao cenário .....	87
Ilustração 43 – Visualização do conteúdo de uma pasta.....	87
Ilustração 44 – Reciclagem quando cheia (à esquerda) e vazia (à direita).....	91
Ilustração 45 – Ecrã de resultados da primeira etapa .....	94
Ilustração 46 – Ecrã com o diagrama de <i>spaghetti</i> .....	95
Ilustração 47 – Exemplo de um diagrama de <i>spaghetti</i> .....	96
Ilustração 48 – Ecrã com o registo das acções do jogador .....	98
Ilustração 49 – Ecrã do cenário (2ª etapa).....	99
Ilustração 50 – Ecrã de organização (2ª etapa) sem janela de informação.....	99
Ilustração 51 – Ecrã que confirma que foi salvo o método de organização .....	100
Ilustração 52 – Terceiro ecrã de objectivos.....	101
Ilustração 53 – Ecrã do cenário do jogo (3ª etapa) sem janela de informação .....	102
Ilustração 54 – Terceiro e último ecrã de resultados finais .....	102
Ilustração 55 – Diagrama Use-Case do cenário 1 .....	129
Ilustração 56 – Diagrama Use-Case do cenário 2 .....	130
Ilustração 57 – Diagrama Use-Case do cenário 3 .....	131
Ilustração 58 – Diagrama Use-Case do cenário 4 .....	131
Ilustração 59 – Questionário de avaliação preenchido pelos alunos .....	132
Ilustração 60 – Apresentação para avaliação do jogo (1/5).....	133
Ilustração 61 – Apresentação para avaliação do jogo (2/5).....	134
Ilustração 62 – Apresentação para avaliação do jogo (3/5).....	135
Ilustração 63 – Apresentação para avaliação do jogo (4/5).....	136
Ilustração 64 – Apresentação para avaliação do jogo (5/5).....	137

## Lista de Tabelas

Tabela 1 – Tipos de jogos e descrições .....	6
Tabela 2 – Jogos divididos por critérios e descrição.....	8
Tabela 3 – Respostas ao questionário .....	110
Tabela 4 – Respostas ao questionário (pontuações finais) .....	111

## Lista de Gráficos

Gráfico 1 – Gráfico circular para a 1ª afirmação.....	112
Gráfico 2 – Gráfico circular para a 2ª afirmação.....	112
Gráfico 3 – Gráfico circular para a 3ª afirmação.....	113
Gráfico 4 – Gráfico circular para a 4ª afirmação.....	113
Gráfico 5 – Gráfico circular para a 5ª afirmação.....	114
Gráfico 6 – Gráfico circular para a 6ª afirmação.....	114
Gráfico 7 – Gráfico circular para a 7ª afirmação.....	115
Gráfico 8 – Gráfico circular para a 8ª afirmação.....	115
Gráfico 9 – Gráfico circular para a 9ª afirmação.....	116
Gráfico 10 – Gráfico circular para a 10ª afirmação.....	116
Gráfico 11 – Gráfico circular para a 11ª afirmação.....	117
Gráfico 12 – Gráfico circular para a 12ª afirmação.....	117
Gráfico 13 – Gráfico circular para a 13ª afirmação.....	118

## Lista de Código

Código 1 – Função que carrega os diferentes <i>swfs</i> dos cenários.....	59
Código 2 – Exemplo de como manipular um <i>SharedObject</i> .....	63
Código 3 – Exemplo de uma classe cujo objecto pode ser guardado em <i>SharedObject</i> .....	64
Código 4 – Função que gera um número aleatório .....	65
Código 5 – Adicionar e remover elementos ao stage.....	67
Código 6 – Código para cronometração do tempo.....	70
Código 7 – Verificar as condições finais do jogo .....	71
Código 8 – Código para salvar o jogo.....	72
Código 9 – Código necessário para abrir o submenu .....	75
Código 10 – Funções chamadas pelo código referente ao submenu .....	76
Código 11 – Criar uma nova zona 1ª Parte .....	80

Código 12 – Criar uma nova zona 2ª Parte .....	81
Código 13 – Mudar o nome a um elemento (documento) 2ª Parte .....	85
Código 14 – Mudar o nome a um elemento (documento) 3ª Parte .....	85
Código 15 – Criar uma pasta na raiz do cenário .....	89
Código 16 – Criar uma pasta dentro de outra.....	90
Código 17 – Adicionar um elemento à reciclagem .....	93
Código 18 – Gravar uma posição do cursor no <i>array</i> das posições .....	96
Código 19 – Criar o diagrama de <i>spaghetti</i> .....	97
Código 20 – Guardar a pontuação na base de dados.....	104

## Lista de Abreviaturas

AMF – Action Message Format

AS3 – ActionScript 3

FEUP – Faculdade de Engenharia do Porto

FPS – First-person shooter

GILT – Graphics, Interaction and Learning Technologies

IA – Inteligência Artificial

ISEP – Instituto Superior de Engenharia do Porto

JIT – Just in Time

JPEG ou JPG - Photographic Experts Group

LLA – Lean Learning Academy

LPG – Lean Production Game

PC – Personal Computer

PHP – Hypertext Preprocessor ou originalmente Personal Home Page

PNG – Portable Network Graphics ou PNG's Not GIF

RPG – Role-playing game

RTS – Real-time strategy

SMED – Single Minute Exchange of Die

SO – Sistema Operativo

SWF – Shockwave Flash

TPS – Toyota Production System

UMinho – Universidade do Minho

XML – Extensible Markup Language



## 1. Introdução

*"Production is not the application of tools to materials, but logic to work."*

**Peter F. Drucker**

A melhor solução para as empresas sobreviverem durante a crise económica, que marca a actualidade, é focarem-se na eficiência da produção e na redução de custos. Isto é o que a filosofia *Lean Manufacturing* pretende alcançar. O benefício máximo com *Lean Manufacturing* é obtido considerando todos os seus elementos (princípios, ferramentas, mentalidade) como um sistema, tendente ao desempenho cabal e consistente de boas práticas empresariais.

São estes alguns dos motivos que levam as empresas a sentir a necessidade de formar continuamente os seus empregados nas áreas mencionadas. Por outro lado, as Instituições de Ensino Superior devem ser capazes de melhorar a empregabilidade dos seus alunos, através da introdução de processos, como o *Lean Manufacturing*, nos seus cursos.

Esta tese pretende criar e testar um ambiente de jogo de simulação que permita transmitir de forma motivadora, os conceitos e práticas de *Lean Manufacturing*. Este ambiente de produção simulado, *Lean Production Game*, tornará o curso mais interessante, resultando em alunos mais motivados, com melhor rendimento de estudo e, conseqüentemente, com uma melhoria nas suas capacidades pessoais e profissionais.

### 1.1. Objectivos

O *Lean Production Game* é um jogo sério em ambiente simulado que, pelas suas características, permite a concretização do seguinte objectivo geral: Aumentar a prestação e rendimento dos formandos na aprendizagem das matérias, tornando os cursos mais motivadores.

Como objectivos específicos, temos:

- Fazer um levantamento dos pontos fortes dos Jogos Sérios com mais sucesso no mercado. É prioritário analisar os jogos existentes, qual a sua função, a melhor forma de transmitir informação, como avaliar o desempenho do jogador, entre outros aspectos;
- Identificar as necessidades actuais, para as quais o *software* existente não dá resposta;
- Criar e validar uma ferramenta de treino em condições seguras, que incentive o aluno a melhorar numa situação simulada de produção, formando-o nos conceitos *Lean* e preparando-o para realizar melhor as suas funções.

## 1.2. Motivação

A tomada de consciência da crescente importância dos meios audiovisuais e informáticos na formação das pessoas a nível educacional, profissional e pessoal, levou-me à decisão de, na minha tese, intervir na área da gestão através da criação de Jogos Sérios e didácticos.

A nível pessoal, a escolha deste tema para a tese baseou-se na minha preferência pela área da Gestão. Os conhecimentos sobre como reduzir custos, como aumentar a organização e a produtividade podem ser transpostos para o campo da Engenharia Informática. Para além disso, seja pela optimização de processos, ou pela gestão de pessoas e equipas, é necessário o conhecimento sobre como ser produtivo, para se ser um profissional qualificado.

Esta tese encontra-se inserida num projecto internacional em que Portugal participa através do centro de investigações GILT (*Graphics, Interaction and Learning Technologies*) do ISEP (Instituto Superior de Engenharia do Porto).

Tem como objectivo o desenvolvimento de ferramentas, conceitos didácticos e técnicas que facilitam, num ambiente de produção simulado, a aplicação de conceitos *Lean*. Com este projecto, não só se ajuda Instituições de Ensino Superior a ensinar conceitos *Lean Manufacturing* a estudantes, aumentando assim a sua empregabilidade, como permite às empresas uma melhor formação dos seus funcionários na prática diária destes [Martens, et al., 2010]. Pretende-se, ainda, melhorar a eficiência de produção e redução de custos das empresas, tornando-as mais competitivas.

Fazer parte de um projecto que decorre à escala europeia, além de ser motivante, é deveras enriquecedor, quer a nível pessoal, quer profissional.

## 1.3. Organização da Dissertação

Esta dissertação encontra-se organizada em sete capítulos.

No primeiro capítulo, inicia-se o tema do projecto com a Introdução. São elaborados os seguintes pontos: Contextualização e Enquadramento, Questões de Investigação, Objectivos, Motivação e Organização da Dissertação.

No segundo capítulo, debate-se o Contexto e Estado da Arte, fazem-se referências às tecnologias utilizadas, situação actual dos Jogos Sérios na área de gestão, entre outros tópicos.

O terceiro capítulo é reservado para as Metodologias de Investigação e Desenvolvimento e outros tópicos, relacionados como o processo de desenvolvimento.

No quarto capítulo deste documento, descreve-se a Concepção. Neste é proposto um conceito para o projecto, um jogo.

A Implementação do Trabalho situa-se no quinto capítulo. Neste, são referidos os passos que se seguiram para implementar o jogo que foi proposto na Concepção.

No sexto capítulo, é feita a Avaliação do trabalho, onde será discutido, entre outros tópicos, os métodos de avaliação usados e o grau de sucesso do trabalho.

No sétimo e último capítulo, faz-se a conclusão do projecto, respondendo as suas perguntas, concluindo sobre os objectivos, fazendo uma auto-avaliação do trabalho realizado e desenvolvendo sobre o futuro do projecto.

As referências bibliográficas de todo o material consultado para realizar esta dissertação, bem como os Anexos, estão presentes no final da mesma.



## 2. Contexto e Estado da Arte

*“Be open to the amazing changes which are occurring in the field that interest you.”*

**Leigh Steinberg**

Apesar de existirem jogos de computador há mais tempo do que computadores pessoais, só com o recente sucesso de vários jogos *online* é que o mundo se apercebeu do seu valor e da capacidade de envolver as pessoas que estes possuem.

O avanço tecnológico contribuiu grandemente para um maior desenvolvimento dos jogos, permitindo produzi-los cada vez mais realistas, dinâmicos e interactivos.

Um exemplo de um jogo digital que gerou bastante lucro e controvérsia pelos seus conteúdos, foi o *GTA4*<sup>1</sup> (*Grand Theft Auto IV*). De acordo com a *wired.com*<sup>2</sup> e estudos feitos em Maio de 2008, uma semana depois do lançamento, este jogo vendeu mais de seis milhões de cópias em todo o mundo, com uma receita total de quinhentos milhões de euros.

Estes valores confirmam o estudo realizado por Prensky para a *Games2train*, em New York [Prensky, 2003]. Prensky começa por fazer um contraste entre o local de ensino, a Escola, em que as matérias são descritas como aborrecidas, de onde resulta desinteresse pelos estudantes, face ao mundo dos jogos que os cativam a passar horas seguidas a jogar. Este facto contraria a ideia geral de que uma pessoa tem um período de atenção curto. De certo modo, este está relacionado com a motivação.

Prensky defende, ainda, que o sucesso dos jogos não advém dos seus conteúdos violentos, mas do facto de oferecerem oportunidades ao jogador de aprender, seja ao conduzir carros de alta cilindrada, a pilotar aviões, a combater em guerras, a gerir parques de divertimentos ou a construir civilizações.

Os jogos ajudam os jovens a tomar decisões, a compreender sistemas complexos, a procurar estratégias e, graças aos sistemas de multi-jogador, a aprender a colaborar e interagir uns com os outros.

Alguns jogos de sucesso são conhecidos até por pessoas que não possuem computador. Por exemplo: *Second Life*<sup>3</sup>, *World of Warcraft*<sup>4</sup>, *StarCraft*<sup>5</sup>, *Counter-Strike*<sup>6</sup> e *Farmville*<sup>7</sup>. Estes são, apenas, alguns casos de sucesso que atraíram empresas e um sem número de jogadores para as suas plataformas de entretenimento.

---

<sup>1</sup> <http://www.rockstargames.com/IV/>

<sup>2</sup> <http://www.wired.com/gamelifelife/2008/05/grand-theft-aut/>

<sup>3</sup> <http://secondlife.com/>

<sup>4</sup> <http://us.battle.net/wow/en/>

<sup>5</sup> <http://starcraft.org/>

<sup>6</sup> <http://store.steampowered.com/css>

<sup>7</sup> <http://www.farmville.com/>

Segundo Derryberry [Derryberry, 2007], a popularidade dessas experiências de imersão deve-se inteiramente à convergência dos mundos virtuais, jogos, redes sociais e aplicações ricas para *internet* (RIAs).

No que diz respeito aos tipos de jogos disponíveis, a lista é bastante extensa, existem jogos para praticamente todos os gostos, o que resulta numa compilação dos tipos mais comuns de jogos:

**Tabela 1 – Tipos de jogos e descrições**

<b>Tipo de jogos</b>	<b>Descrição</b>
<b>Action</b>	Jogos de acção que dão ênfase a desafios físicos, como a coordenação entre o olhar e as mãos e o tempo de reacção.
<b>Action-adventure</b>	Idêntico ao jogo de acção, mas com um enredo histórico.
<b>Adventure</b>	Jogos de aventura, em que o jogador assume o papel da personagem principal, numa história interactiva.
<b>Arcade</b>	Jogos da década de 80, jogados em máquinas colocadas em estabelecimentos de convívio.
<b>Card and Board Games</b>	Jogos de cartas e de tabuleiro. Esta categoria envolve uma grande parte de jogos tradicionais conhecidos.
<b>Children</b>	Jogos desenvolvidos para crianças. Permitem incentivar a aprendizagem.
<b>Fighting</b>	Jogos de combate. Geralmente, o jogador pode enfrentar outro jogador, ou uma personagem controlada pela IA do computador, num combate.
<b>First Person Shooters - FPS (Shooter)</b>	Jogos em que o jogador controla a personagem na primeira pessoa e manuseia armas, de modo a eliminar os seus adversários.
<b>Mini Games</b>	Jogos pequenos e fáceis de concluir. Muito comuns e populares na <i>internet</i> .
<b>Music</b>	Jogos em que o jogador pode tocar instrumentos musicais, compôr músicas, etc.

<b>Party</b>	Jogos criados para serem jogados em grupo. Jogos de diversão e entretenimento para festas.
<b>Plataforming</b>	Jogos de plataformas. Para chegar ao fim do jogo, o jogador tem de orientar a sua personagem, saltando pelas plataformas dos diferentes níveis.
<b>Play-and-learn</b>	Jogos desenvolvidos para ensinar certos conteúdos, enquanto se jogam.
<b>Puzzle</b>	O jogador tem de juntar peças de uma maneira lógica, de modo a chegar à solução pretendida.
<b>Racing</b>	Jogos de corridas. Os mais comuns são de automóveis ou ciclomotores.
<b>Real-Time Strategy (RTS)</b>	Jogo de estratégia (sem turnos), em que as acções têm de ser tomadas em tempo real, sendo o seu efeito logo sentido.
<b>Role Playing Games (RPGs)</b>	Jogo geralmente por turnos, em que o jogador assume o papel de uma ou várias personagens, numa história fictícia. Este pode controlar a narrativa do jogo.
<b>Simulation</b>	Jogo que simula ambientes, em que o jogador controla algo a que normalmente não tem acesso, como: comboios, aviões, etc.
<b>Third Person Shooters</b>	Semelhante aos FPS, a visualização da personagem é na terceira pessoa. Geralmente, estas têm mais formas de interacção, como estilos de combate corpo-a-corpo com os adversários.
<b>Strategy</b>	Jogo geralmente por turnos, em que o jogador tem de estabelecer uma estratégia para conquistar os seus objectivos.
<b>Quizzes</b>	Jogo em que o jogador tenta responder ao maior número de perguntas correctamente. Servem para avaliar e desenvolver os conhecimentos deste.

## 2.1. Definição de Jogos Sérios

Apesar da maior parte dos jogos serem desenhados para entretenimento, há bastantes que são criados com outros propósitos. Estes podem ser catalogados pela sua finalidade ou pelo conteúdo que apresentam. Sendo assim, segue-se uma lista de jogos organizada segundo estes critérios:

**Tabela 2 – Jogos divididos por critérios e descrição**

<b>Jogos divididos por critérios</b>	<b>Descrição</b>
<b>Adult video games</b>	São jogos que possuem bastante conteúdo sexual e, por esse motivo, são só para adultos.
<b>Advergames</b>	Jogos criados com o objectivo de publicitar um produto, serviço, ou organização.
<b>Art games</b>	Jogos desenvolvidos para dar ênfase à Arte ou despertar alguma reacção no jogador.
<b>Casual games</b>	Jogos destinados, ou jogados por pessoas que não o fazem frequentemente. Caracterizados por terem qualquer tipo de jogabilidade ou género, regras simples e por não precisarem que o jogador se comprometa a jogar até ao fim.
<b>Educational games</b>	Jogos que são desenvolvidos para ensinar uma determinada matéria.
<b>Electronic Sports games</b>	Jogos dedicados a qualquer desporto.
<b>Exergame</b>	Jogos que são desenvolvidos, para que o jogador pratique exercício físico. Estes jogos dependem da tecnologia, que detecta o movimento do corpo ou reacções.
<b>Serious games</b>	Jogos desenvolvidos sem outro propósito que não o do divertimento. Usados por empresas, para treinar e melhorar a performance dos seus funcionários.

Isto não significa que um Jogo Sérico não seja classificado, também, como *Adventure*. Este factor depende, ainda, do *Gameplay* (forma de jogar) e do jogo em si.

Há autores que sugerem outras taxonomias, como Derryberry, que organiza os jogos em três grupos: *Casual Games*, *Serious Games* e *Advergimes* [Derryberry, 2007].

Um *Serious Game*, ou Jogo Sérió, é desenhado com um único objectivo: melhorar algum aspecto da aprendizagem. Ao contrário dos outros tipos de jogos, pretende-se transmitir conteúdos educativos ao jogador, deixando o entretenimento para segundo plano. Consiste num ambiente simulado, onde o jogador é o interveniente principal.

Pretende-se que este execute tarefas, processos e tire conclusões, de forma a aumentar a sua produtividade e os seus conhecimentos, sem correr riscos ou sem desencadear na vida real uma sucessão de eventos que seriam negativos para o negócio.

Este tipo de jogo é utilizado em inúmeras áreas: engenharia, saúde, educação, defesa militar, gestão de emergências, exploração científica, entre muitas outras.

Podem ser usados para vários fins, seja formação de empresas, campanhas sociais ou marketing.

Segundo Prensky [Prensky, 2003], as forças militares americanas usam mais de cinquenta jogos diferentes de computador e de vídeo para ensinar vários tópicos, desde estratégia a táticas militares. Um desses jogos é o *America's Army: Operations*<sup>8</sup>, um jogo de recruta, lançado em 2002, que tem mais de dois milhões de utilizadores registados, em que quase metade concluiu o treino virtual básico.

De acordo com o projecto e-VITA<sup>9</sup>, quanto mais realistas forem os cenários de um Jogo Sérió, mais valorosa é a experiência, uma vez que se pode rapidamente transformar em situações de vida ou negócio real.

Este projecto defende, ainda, que as informações e as sensações experienciadas, através deste tipo de jogos, permitam ao jogador redefinir a sua percepção e aumentar a atenção ou memória, a fim de facilitar mudanças no comportamento da aprendizagem prática.

## 2.2. Jogos Sérios na Aprendizagem

Os Jogos Sérios têm por objectivo treinar e formar indivíduos, apresentando-lhes cenários, para que se preparem para a execução das suas tarefas.

Existem inúmeras vantagens para usar jogos como plataforma de aprendizagem. Na sua apresentação: "*Gaming can make a better world*", McGonigal<sup>10</sup> [McGonigal, 2010] afirma que, em Fevereiro de 2010, três biliões de horas por semana foram passadas a jogar jogos *online* e as suas investigações mostraram que os jogadores estão mais motivados e inspirados a fazer o que

---

<sup>8</sup> <http://www.americarmy.com>

<sup>9</sup> <http://www.evitaproject.eu/> - Projecto que promove e investiga a inovação orientada para a pedagogia, e que é co-financiado pela DG Educação e Cultura da Comissão Europeia, ao abrigo do Programa de Aprendizagem ao Longo da Vida (Lifelong Learning Programme), KA 3.

<sup>10</sup> *Designer* de jogos, investigadora e autora, especializada em jogos universais e jogos de realidade alternativa (ARGs).

é necessário, a colaborar, a interagir entre si e a recuperar melhor dos seus fracassos no mundo virtual do que no mundo real.

Estes mesmos jogadores, no mundo real, não se sentem tão à vontade e tão autoconfiantes, para enfrentarem os problemas. Sentem-se desmotivados, derrotados e várias vezes deprimidos. Estes sentimentos não existem no jogo, o que leva a concluir que os jogos podem ser a solução para vários problemas na aprendizagem, transmissão de conteúdos e até mesmo de problemas da nossa sociedade [McGonigal, 2010].

Os jogos colocam o jogador no limite das suas capacidades, com objectivos ambiciosos mas atingíveis, o que o motiva a tentar dar o seu melhor e a evoluir. Nos jogos, estes objectivos são traçados mediante as capacidades actuais do jogador. Existe a sensação de que este está prestes a atingir o objectivo, prestes a passar ao próximo nível. É esta percepção que o motiva e o leva a passar horas a jogar, mantendo um alto grau de concentração. É esta sensação que se pretende que os Jogos Sérios despertem nos jogadores, para melhorar o processo de aprendizagem.

Ainda nesta apresentação, McGonigal [McGonigal, 2010] defende que, durante este tempo todo passado a jogar, o Ser Humano está a mudar aquilo de que é capaz, evoluindo e, por inércia, fazendo evoluir a Sociedade. Os aspectos que estão a ser melhorados são: optimismo, pró-actividade na solução de problemas, melhoramentos nas relações sociais, aumento da produtividade e compreensão do significado das tarefas exigidas.

McGonigal conclui, dizendo que os jogadores são indivíduos esperançosos e que possuem grandes capacidades. Essa é a atitude que se pretende que o jogador tenha, enquanto aprende jogando.

Entre inúmeras vantagens na utilização de jogos na aprendizagem, destacam-se as mais relevantes:

- Desenvolver capacidades intelectuais e motoras;
- Perceber a aplicabilidade das matérias abordadas;
- Motivar e aumentar a atenção do jogador;
- Aumentar o relacionamento e colaboração entre indivíduos;
- Tornar o jogador mais optimista e criativo na solução de problemas;
- Permitir uma evolução em factores chaves da produtividade;
- Criar uma forte vontade de superar as dificuldades e problemas que surjam;
- Permitir criar experiência numa área de formação;
- Aumentar a confiança nas suas capacidades;
- Criar um sentido de segurança e de realização ao evoluir no jogo;
- Permitir que o jogador seja mais ousado na procura de soluções;
- Criar um sentido de rivalidade saudável entre jogadores;
- Reforçar relações pessoais.

Apesar de todos os estudos e pesquisas feitos sobre este tema, os jogos são, ainda hoje, vistos pela sociedade como métodos de distração dos jovens e como ferramentas contraproducentes na aprendizagem. Este preconceito está a ser difícil de ultrapassar e parece ser o principal factor responsável por não haver mais investigação na área.

### **2.3. Jogos Sérios na Gestão**

Um Jogo de Gestão é uma forma de simulação vocacionada para o treino do exercício de funções de Gestão. Este tipo de jogos pode ser aplicado às várias funções: Planeamento, Organização, Liderança e Controlo; bem como, aos vários níveis de Gestão: Nível Estratégico, Nível Tático e Nível Operacional.

Podem ter vários modelos de utilização ou objectivos, contudo, são sempre inspirados em modelos matemáticos. Estes jogos permitem a criação de cenários, em que se testam as capacidades de resposta do jogador, instruindo-o para que, se acontecerem na realidade, possa estar preparado.

Tornam, assim, possível treinar os jogadores sem efeitos secundários. O desenrolar destes jogos depende, apenas, do jogador. As suas acções desencadeiam um conjunto de efeitos, calculados como sendo a realidade para aquela escolha.

Os principais objectivos de jogos deste género são desenvolver capacidades de decisão, análise, criar um conhecimento da interligação de diversos factores e aumentar a colaboração entre grupos.

#### **2.3.1. Tipos de plataformas utilizadas**

No que diz respeito às plataformas utilizadas por estes jogos, podem dividir-se em Jogos Reais ou Jogos Digitais. Segue-se uma descrição destas duas plataformas.

##### **2.3.1.1. Jogos reais**

Os jogos reais não precisam de nenhuma plataforma electrónica para serem jogados. Podem ser todo o tipo de jogo que satisfaça a condição anterior, como: jogos de tabuleiro, puzzles, jogos infantis jogados no recreio ou qualquer jogo em que haja interacção física entre o jogador e objectos do mundo real.

Na área de Gestão, quando se menciona jogos reais, faz-se referência ao tipo de jogos que simulam os ambientes de produção de uma forma realista e são vocacionados para a aprendizagem de conteúdos relacionados com a produção.

Há interacção entre jogadores num ambiente real, como uma sala ou escritório. Devem ser formados grupos semelhantes aos grupos existentes no seu trabalho, dividir tarefas e produzir

um determinado produto. Este produto pode ser qualquer objecto, fornecendo as peças individuais que o constituem, por exemplo: uma ficha eléctrica ou uma lanterna.

Os jogadores seguem as instruções, usando os materiais que vêm com este, para produzir o maior número de produtos possíveis num determinado intervalo de tempo, tendo em conta vários aspectos como: qualidade, trabalho, performance, etc.

Como o objectivo deste tipo de jogos se centra na optimização das acções do jogador, após a primeira ronda, são feitas avaliações na performance de todos os jogadores e são debatidos os problemas com que estes se depararam durante a produção, como unidades produzidas com defeito ou problemas na linha de montagem. Novas estratégias de produção e sugestões são fornecidas, para optimizar a produção, antes da ronda seguinte começar.

No final, compara-se a primeira ronda com a última, para se ter uma ideia concreta, fundamentada com resultados do aumento de produtividade e de quão efectivas foram as medidas implementadas.

Estes jogos podem ter quantas rondas a pessoa responsável pelo jogo quiser. Alguns destes jogos dão suporte a cursos, que chegam a durar dezoito semanas. Ao jogar, criam-se metodologias de trabalho, que ajudam a melhorar a produção, gerando lucros e diminuindo os desperdícios.

### **2.3.1.2. Jogos digitais**

Jogos digitais são os líderes de vendas no mercado. Existem inúmeras plataformas que permitem a proliferação deste tipo de jogos, desde os computadores pessoais, sejam eles PC ou Mac, até consolas, como *PlayStation 3*<sup>11</sup> da *Sony* ou *Xbox*<sup>12</sup> da *Microsoft*, *Wii*<sup>13</sup> da *Nintendo* ou outras consolas portáteis.

Estes são os que mais têm tido sucesso, devido ao leque de vantagens que oferecem. Para além de serem mais dinâmicos, fazem uso de sistemas gráficos para criar mundos virtuais, utilizam todos os elementos multimédia, apelam e entretêm o jogador de uma forma que os jogos físicos não conseguem.

Esta plataforma foi a escolhida, para realizar este projecto.

Para se jogar este tipo de jogos, é necessária uma plataforma electrónica.

---

11 <http://pt.playstation.com/ps3/>

12 <http://www.xbox.com/pt-pt/>

13 <http://pt.wii.com/>

## 2.4. Jogos Sérios em *Lean Manufacturing*

*Lean Manufacturing*, *Lean Production*, ou o termo mais utilizado *Lean* (a partir de 1990), é uma filosofia de produção, que baseia o seu objectivo na melhoria da qualidade dos produtos para o cliente, através da redução dos desperdícios e, conseqüentemente, do tempo e custo de produção.

*Lean* centra as suas atenções em melhor servir o cliente com produtos e serviços com qualidade, optimizando a eficiência dos processos, decidindo o que é necessário e acessório nestes. É baseado em ideias como o Taylorismo, o Fordismo, a Eficiência do Movimento e na sabedoria popular do tempo de poupança [Strategos.inc, 2008].

O *Lean Manufacturing* baseia-se num sistema de produção criado por uma grande empresa de automóveis, a *Toyota*, com o nome *Toyota Production System*, ou *TPS*, criada em 1945. Este sistema de produção consistia na redução de desperdícios, tendo como finalidade melhorar o valor para o cliente, de várias perspectivas.

A *Toyota* identificou sete tipos de desperdícios. Estes podem surgir por excesso de produção, produção defeituosa, tempos de espera na linha de montagens, problemas no transporte do produto, entre outros.

*Lean Manufacturing* é tida como uma versão polida, resultante de tentativas para melhorar a eficiência construídas a partir do trabalho de líderes, como Taylor ou Ford. Esta filosofia, ainda hoje, está a ser definida pelas entidades que a utilizam.

A ELSE Inc.<sup>14</sup> definiu algumas regras para implementar *Lean Manufacturing*:

1. Descartar ideias fixas ou convencionais sobre produção;
2. Pensar “Como fazer?” e não “Porque não pode ser feito de uma determinada maneira?”;
3. Não arranjar desculpas. Questionar as práticas correntes;
4. Não procurar a perfeição. Produzir logo, ainda que apenas 50% dos objectivos sejam cumpridos;
5. Corrigir erros apenas uma vez;
6. Não gastar dinheiro. Criatividade antes do capital;
7. Sabedoria é criada, quando se depara com as dificuldades;
8. Perguntar “Porquê?” cinco vezes e procurar as raízes da causa do problema;
9. Procurar a sabedoria de dez pessoas em vez do conhecimento de uma;
10. Existe uma infinita capacidade de melhorar tudo.

---

<sup>14</sup> Implement-Lean-Manufacturing.com

O *Lean Manufacturing* possui muitas ferramentas para a optimização da produtividade. Alguns exemplos [ThroughPut Solutions, 2011]:

- SMED: *Single Minute Exchange of DIE*;
- TPM: *Total Productive Maintenance*;
- 5S *Method*;
- KanBan: *Work Signaling System*;
- 2-Bin: *Materials Replenishment System*;
- *Error & Mistake-Proofing*;
- *Level-Loading (Heijunka)*;
- *Inventory Reduction*;
- *Kaizen Events (Kaizen Blitzes ou Improvement Events)*;
- *Continuous Improvement (e Lean Culture Change)*;
- JIT – *Just in Time*.

Peterson, Jim & Smith, Roland afirmam, ainda, que esta filosofia é usada por várias empresas ou organizações, como a *Hewlett-Packard* e a *Boing*. Ambas procuraram esta ferramenta, para melhorar a segurança e desempenho da empresa, melhorar a qualidade de comunicação, reduzir as reclamações, o tempo de atendimento, bem como melhorar a qualidade geral da produção [Peterson & Roland, 1998].

#### 2.4.1. Exemplos de jogos digitais de *Lean Manufacturing*

Uma vez que a pesquisa por jogos cujo tema fosse *Lean Manufacturing*, no geral, não devolveu grandes resultados, procuraram-se jogos digitais que tivessem por objectivo facilitar a aprendizagem ou permitissem o treino de ferramentas de *Lean Manufacturing*.

A pesquisa revelou um elevado número de jogos reais, o que leva a crer que esta é a plataforma de eleição actual para transmitir conteúdos *Lean*.

Quanto a jogos digitais, um bom exemplo, com o *Lean* como tema, é o *Lean Bicycle Factory*<sup>15</sup>. Produzido por uma empresa Sueca, este jogo possui como objectivo que o jogador melhore o processo de produção de bicicletas. Ao iniciar o jogo, o jogador é confrontado com o cenário típico de produção, neste caso, bicicletas. Os operários movem-se sozinhos e são facilmente perceptíveis as fases comuns de produção, montagem, pintura e controlo de qualidade. O jogador apercebe-se das melhorias que pode realizar, para optimizar o fluxo de produção, e a cada ronda escolhe duas alterações de uma lista (reduzir *stock*, vender material

---

<sup>15</sup> <http://ludosity.com/library/lean-game/>

desnecessário, contratar peritos, melhorar o controlo de qualidade, entre outras). Cada alteração tem um custo, por isso, é essencial que sejam feitas decisões cuidadas, para se obter mais lucro.

Contudo, trata-se de um jogo limitado, no sentido em que possui objectivos pouco ambiciosos e fáceis de alcançar. Cada acção deste está categorizada com um custo, o jogador terá de escolher aquela que irá permitir atingir o objectivo de aumentar a produtividade.

No que diz respeito aos jogos digitais, os resultados mostram que há uma necessidade, já abordada nesta dissertação, de jogos que simulem ambientes reais que permitam a aprendizagem de conteúdos de *Lean Manufacturing*. Pode concluir-se, então, que essa parece ser a plataforma ideal, para a implementação do que se propõem de uma forma inovadora.

#### 2.4.2.O que é o Método 5S?

Peterson, Jim & Smith Roland definiram o 5S como uma ferramenta de trabalho, que permite desenvolver um planeamento sistemático de classificação, ordem, limpeza, permitindo assim um incremento na produtividade, segurança, clima organizacional, motivação dos funcionários e conseqüente melhoria da competitividade. Caracteriza-se por uma procura constante de valor e satisfação do cliente. Todos os processos que não contribuem para este objectivo são considerados desperdícios [Peterson & Roland, 1998].

O nome desta ferramenta provém das cinco palavras japonesas que definem a metodologia de trabalho:

- *Seiri*, que significa Senso de Utilização;
- *Seiton*, que significa Senso de Ordenação;
- *Seisō*, que significa Senso de Limpeza;
- *Seiketsu*, que significa Senso de Saúde;
- *Shitsuke* ou *Shuukan* que significa Senso de Autodisciplina.

Os propósitos da metodologia 5S são de melhorar a eficiência, através da distribuição adequada de materiais (separar o que é necessário do desnecessário), organização, limpeza, identificação de materiais/espacos, a manutenção e melhoria do próprio 5S.

Estes são os principais benefícios da implementação da metodologia 5S:

1. Maior produtividade, pela redução de tempo na procura de um determinado objecto. Só ficam no ambiente de trabalho os objectos necessários;
2. Redução de despesas e melhor aproveitamento de materiais. A acumulação excessiva de materiais tende à desorganização;
3. Melhoria da qualidade de produtos e serviços;
4. Menos acidentes do trabalho;
5. Maior satisfação das pessoas com o trabalho.

Este método é usado, para organizar o espaço de trabalho e treinar os funcionários.

Na primeira fase do processo, é necessário remover todos os materiais e ferramentas desnecessárias, sendo apenas preservado o que for importante para a tarefa a executar. Todo o resto deve ser removido do posto de trabalho.

A segunda fase consiste em melhorar o fluxo de trabalho e simplificar a disposição das ferramentas e equipamentos no posto de trabalho. Estes devem estar acessíveis a todos os funcionários nas respectivas áreas em que devem ser utilizados, de modo a melhorar a sua eficiência.

A terceira fase é dedicada à limpeza sistemática do posto de trabalho. É uma prática que deve ser realizada diariamente, para manter o posto de trabalho em boas condições.

Na quarta fase, dá-se ênfase à standardização. Uniformização dos conceitos e processos estabelecidos nas duas primeiras fases. Criar referências visuais para estas práticas, melhorar e corrigir comportamentos que não estejam explicitados formalmente.

A última fase é a sustentação. É necessário manter as quatro fases anteriores em vigor, através de formações, de modo a continuar a seguir estas práticas e evitar cair na antiga rotina.

A implementação deste método não é fácil, havendo vários problemas que podem surgir e pôr em causa o melhoramento de performance que este pretende atingir. Alguns destes problemas podem ser: falta de compreensão desta metodologia, falta de cooperação por parte dos colaboradores ou falha na transmissão de informação necessária para a correcta aplicabilidade deste.

#### **2.4.3. Utilização de Jogos Sérios para ensinar o Método 5S**

Já foram referidas as vantagens da utilização de jogos na aprendizagem. Neste tópico, verifica-se a sua aplicabilidade no ensino do Método 5S.

A melhor maneira de ensinar conceitos de *Lean Manufacturing* é através de jogos. O Método 5S não é excepção. A maior parte da formação e treino nesta área é obtida através de jogos de produção.

Reforçar as matérias ensinadas com jogos é uma boa maneira de manter o interesse e concentração por parte dos jogadores. Os jogos mantêm os jogadores envolvidos tornando-os activos na sua instrução e reforçando o conceito de aprendizagem pela prática.

É fácil detectar, através de jogos deste género, colaboradores que estão mais aptos a liderar equipas e a desempenharem bem o seu papel numa equipa. Basta, para isso, verificar a sua prestação no jogo. Um dos propósitos do jogo é fazer com que os jogadores, num ambiente simulado idêntico à realidade, identifiquem um problema e as possíveis causas que o possam ter desencadeado.

De acordo com Mcmeans [Mcmeans, Advantages Of Using 5s Simulation Games, 2011], o jogador desenvolve a capacidade de antecipar problemas, de análise de uma determinada situação e de tomar uma decisão sobre o problema que enfrenta. Pode tirar conclusões sobre a eficácia desta decisão, sem que haja um risco real de uma má decisão trazer consequências graves para o negócio. Por fornecerem esta segurança, este tipo de jogos são mais acessíveis economicamente e menos arriscados do que fazer a mesma experiência na realidade.

Mcmeans [Mcmeans, Advantages Of Using 5s Simulation Games, 2011] defende, ainda, que estes jogos promovem um ambiente divertido de aprendizagem e organização de equipas e que os participantes aprendem a realizar tarefas individuais, partilhar ideias, maximizar o seu potencial aprendendo com as capacidades e limitações uns dos outros.

Grande parte dos Jogos Sérios desenvolvidos para ensinar ou exercitar práticas do Método 5S são jogos reais<sup>16</sup>. Os jogadores juntam-se numa sala para criar um produto e são feitos ajustes na linha de produção simulada, para ensinar e treinar estes conteúdos.

O foco de atenção está nos jogos digitais, simulações gráficas para aprendizagem dos mesmos conteúdos, em ambientes virtuais. Isto não significa que as ideias base de jogos reais não possam ser utilizadas para criar jogos digitais. A tendência será começar a produzir jogos digitais de simulação inspirados em jogos reais já utilizados há muitos anos e que têm provas dadas do seu sucesso na aprendizagem.

A pesquisa não revelou nenhum Jogo Sérico digital desenvolvido para o Método 5S na área de *Lean Manufacturing*. O que confirma a análise no sentido de haver uma necessidade não satisfeita.

## 2.5. Tecnologias a utilizar

Para a implementação de jogos, está disponível no mercado um leque variado de ferramentas que auxiliam o desenvolvimento dos vários tipos existentes.

Apesar de haver esta diversidade nas ferramentas, não existe nenhuma em concreto que tenha dado provas da sua supremacia para o desenvolvimento de Jogos Sérios.

O motivo pelo qual não existem ferramentas para a criação deste género de jogos deve-se ao facto destes serem recentes. Como estes ainda não deram provas suficientes de serem úteis no ensino dos conteúdos e por haver preconceito no uso de jogos para a formação, não foi ainda dedicado tempo suficiente para o desenvolvimento da área.

Pode, assim, afirmar-se que não houve ainda uma oportunidade de se desenvolverem ferramentas que auxiliem a sua criação, talvez por ainda não haver muitos jogos de sucesso neste nicho de mercado.

---

<sup>16</sup> <http://www.leangames.co.uk/games.php>

Por não haver uma plataforma evidente pelas suas vantagens, a opção das ferramentas de desenvolvimento passa por avaliar as especificações técnicas a implementar e verificar qual o motor de jogo, ou outra ferramenta, que melhor responde às necessidades.

### 2.5.1. Escolha da tecnologia a usar para a implementação

Na procura das possíveis ferramentas a utilizar, ponderou-se o uso de motores de jogo gráficos, como o *Thinking Worlds*<sup>17</sup>, muito usado para criar simulações, e o *Unity*<sup>18</sup>, um motor de jogo gráfico conhecido pela sua compatibilidade entre plataformas como Mac, PC, vários *browsers* e plataformas móveis, como *iPhone* e agora *Android*.

De referir que existem também empresas que criam Jogos Sérios, usando motores gráficos e motores de jogo populares no mercado, como o motor gráfico do *Half-Life 2*<sup>19</sup>, *The Sims*<sup>20</sup> e *Second Life*<sup>21</sup>.

Apesar de a procura não ter revelado Jogos Sérios que tenham sido desenvolvidos com o recurso a motores gráficos específicos para a criação de outro tipo de jogos, ponderou-se o uso de motores gráficos utilizados na criação de aventuras gráficas, para a realização deste projecto.

O motivo que levou a ser ponderado o uso destes motores, deveu-se ao facto de haver algumas semelhanças entre os dois tipos de jogos, no que diz respeito à interactividade entre o utilizador, o cenário e as tarefas passíveis de serem realizadas. Apesar de não ser uma solução ideal, foi ponderada essa hipótese, antes de ser tomada qualquer decisão.

Fez-se uma breve pesquisa pelos motores de aventuras gráficas mais populares disponíveis no mercado, comparou-se as vantagens e desvantagens que estes trazem, de modo a definir qual seria o mais indicado para se utilizar.

O factor a que se deu mais importância foi a usabilidade, uma vez que todos os motores gráficos cumprem a maior parte dos requisitos necessários para a criação de aventuras gráficas.

Apesar de a pesquisa ter devolvido um número elevado de outras soluções passíveis de serem usadas para este fim, optou-se por apenas listar as mais populares, uma vez que são as que possuem mais documentação e ajuda comunitária, para resolução de problemas comuns que surjam durante a implantação.

De referir que, apesar de existirem motores gráficos profissionais famosos, como o SCUMM<sup>22</sup>, desenvolvido pela *LucasArt* para a criação de aventuras gráficas, estes não foram considerados, uma vez que não estão disponíveis para o público.

Segue-se a lista dos três motores gráficos mais populares [Gouveia, 2010]:

---

<sup>17</sup> <http://www.thinkingworlds.com/index.php>

<sup>18</sup> <http://unity3d.com/>

<sup>19</sup> <http://orange.half-life2.com/hl2.html>

<sup>20</sup> <http://thesims.ea.com/>

<sup>21</sup> <http://secondlife.com/>

<sup>22</sup> SCUMM - <http://www.justadventure.com/articles/Engines/SCUMM/SCUMM.shtml>

1. *Adventure Game Studio*<sup>23</sup>;
2. *Wintermute Engine*<sup>24</sup>;
3. *Visionaire Studio*<sup>25</sup>.

Avaliaram-se os pontos fortes e fracos de cada um destes motores gráficos.

### 2.5.1.1. Adventure Game Studio

Começando a avaliação dos motores gráficos populares, temos o *Adventure Game Studio*. Este motor gráfico permite a criação de jogos de aventuras gráficas do tipo *point&click*.



Ilustração 1 – Área de trabalho do *Adventure Game Studio*

#### Vantagens

1. Editor de script com *intellisense* incorporado;
2. Rápida curva de aprendizagem;
3. Expansível através de uma arquitectura baseada em *plugins*.

#### Desvantagens

1. Não suporta personagens 3D;
2. Última actualização foi há mais de um ano;
3. Disponível apenas para Windows;
4. Vocacionado para jogos 2D e jogos antigos;
5. Termos de licença ambíguos.

<sup>23</sup> <http://www.adventuregamestudio.co.uk/>

<sup>24</sup> <http://dead-code.org/home/>

<sup>25</sup> <http://www.visionaire-studio.net/>

### 2.5.1.2. *Wintermute Engine*

À semelhança do motor gráfico anterior, o *Wintermute Engine* foi desenvolvido com o objectivo de criar e executar jogo de aventuras gráficas, do tipo *point&click*. Este consegue produzir jogos em 2D, bem como jogos em que o cenário é em 2D, mas as personagens são em 3D.

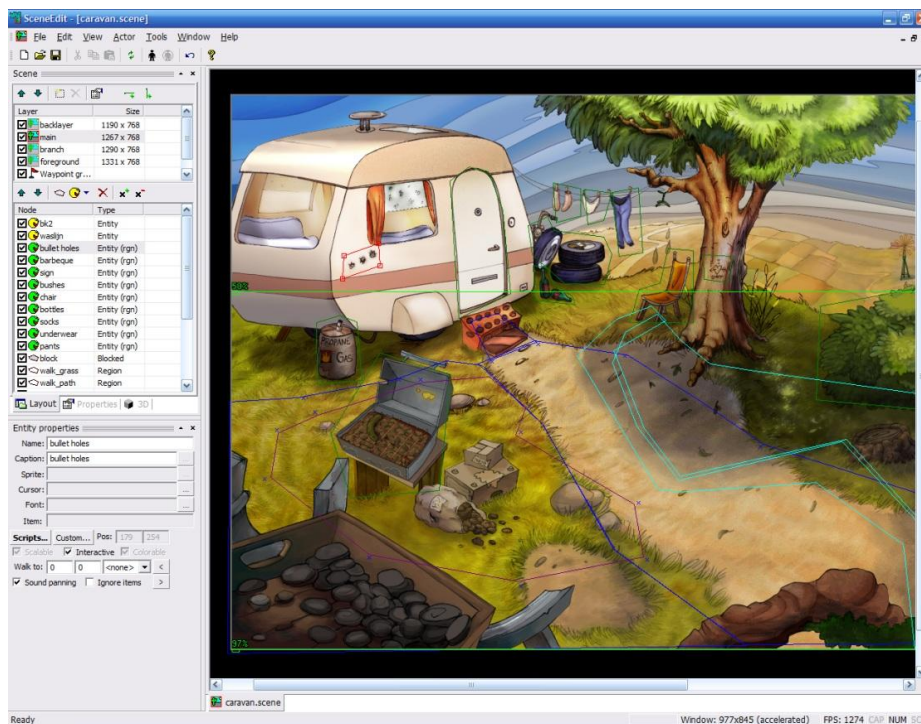


Ilustração 2 – Ambiente de trabalho do *Wintermute Engine*

#### Vantagens

1. Único motor que suporta personagens 3D;
2. Livre para uso comercial e *Open-Source*;
3. Sistema de *scripting* poderoso que permite bastante flexibilidade;
4. Expansível através de uma arquitectura baseada em *plugins*.

#### Desvantagens

1. Disponível apenas para *Windows*;
2. Mais complexo de trabalhar do que as alternativas;

### 2.5.1.3. Visionaire Studio

Concluindo este tópic, apuram-se, agora, as vantagens e desvantagens do *Visionaire Studio*.

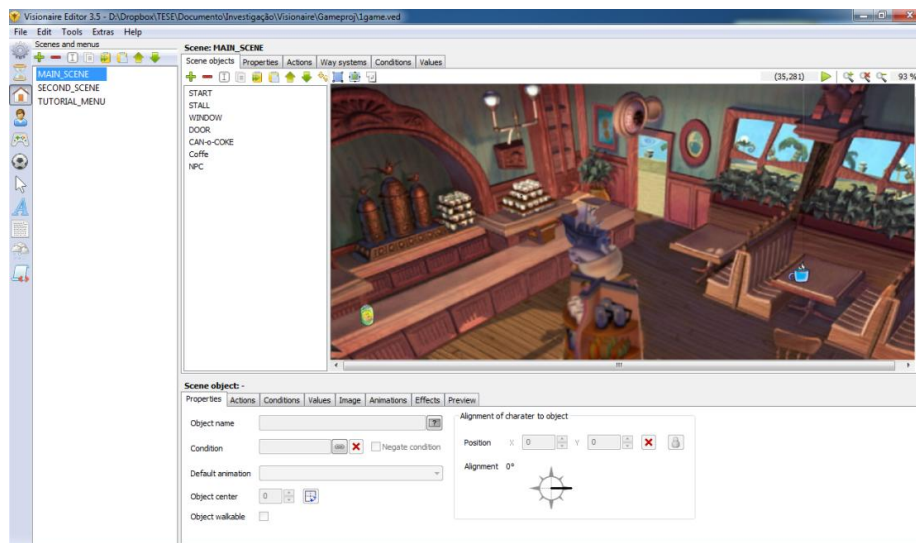


Ilustração 3 – Ambiente de trabalho do *Visionaire Studio*

#### Vantagens

1. Rápida curva de aprendizagem;
2. Simplicidade na criação de cenários, objectos e interações;
3. Criação do jogo através de menus, opções e parâmetros configuráveis;

#### Desvantagens

1. Pouca e fraca documentação disponível;
2. Gestão de eventos bastante limitada;
3. Licença bastante ambígua nos termos e condições;

Dado o *Visionaire Studio* ter-se mostrado mais adequado para a realização deste projecto, analisaram-se as suas características, para melhor se entender as especificidades e o seu funcionamento.

Tendo como finalidade averiguar a viabilidade de execução do projecto com esta ferramenta, fez-se um levantamento das desvantagens em usar este motor gráfico, dadas as funcionalidades que o jogo teria de ter:

1. Ao comprar a licença por 45 euros, o jogo só pode ser vendido no máximo por 15 euros, caso contrário, é necessário assinar um contrato com a empresa;
2. Pouca documentação disponível;
3. Fraca qualidade da documentação existente (traduções medíocres de Alemão para Inglês);

4. Dificuldade em programar eventos temporais no jogo;
5. Processamento dos eventos e acções são principalmente associados a personagens e não a objectos;
6. Limita a implementação de código personalizado pelo programador;
7. Não permite entrada de texto, só caracteres individuais;
8. Não permite criar menus de contexto;
9. Não permite fazer *tracking* da posição do cursor;
10. À medida que se adicionam cenários, objectos, acções, etc, a ferramenta de desenvolvimento fica mais lenta;
11. Há uma notória lentidão, se o jogo estiver a ser executado ao mesmo tempo que certos *browsers*;

A alternativa a usar motores gráficos é desenvolver o jogo de raiz, usando *Flash* e *ActionScript3*. Esta decisão não pode ser tomada de consciência leve, por trazer vantagens e desvantagens difíceis de avaliar. Fez-se, então, o estudo das vantagens e desvantagens desta hipótese.

#### **Vantagens**

1. Maior liberdade de implementação;
2. Não existem as possíveis limitações do motor gráfico que se escolheria;
3. Permite criar uma solução mais personalizada;
4. Evita chegar a um impasse a meio da implementação, devido a algum factor que, acidentalmente, não tenha sido considerado;
5. Existem inúmeras bibliotecas livres que ajudam a implementação;
6. Existem vários tutoriais em vários graus de dificuldade;
7. Existe um elevado número de fóruns de entajuda de programadores;

#### **Desvantagens**

1. Maior tempo de desenvolvimento;
2. Maior carga horária na execução;
3. Todas as acções e eventos de dispositivos de entrada têm de ser programados;
4. Pode gerar uma solução menos robusta (a qualidade do jogo será directamente proporcional à qualidade da programação);
5. Maior risco de insucesso.

Após este processo de selecção de ferramentas a utilizar, optou-se por fazer o jogo de raiz, usando *Flash* e *ActionScript3*, pelas vantagens enumeradas e pelas características específicas do projecto. Sabendo os riscos que se corre ao escolher esta ferramenta, pretende-se que as vantagens desta tragam mais liberdade de implementação e permitam criar a versão do jogo que se propôs.

### 3. Metodologia de Investigação e Desenvolvimento

*"The great thing in the world is not so much where we stand, as in what direction we are moving."*

**Oliver Wendell Holmes**

Neste capítulo serão abordados os temas, a metodologia seguida para criar o projecto, ferramentas utilizadas para a concepção e desenvolvimento, uma breve definição de cada etapa e que trabalho foi realizado.

#### 3.1. Processo de Desenvolvimento

A maior parte das metodologias existentes baseiam-se na premissa que o projecto vai ser implementado por vários indivíduos, que constituem um grupo com conhecimentos polivalentes. Uma vez que esta premissa não se aplica a este projecto, teve de se adaptar as metodologias conhecidas.

Seguiu-se a Estrutura do Desenvolvimento do Projecto mais tradicional, que possui as seguintes fases:

1. Iniciação;
2. Definição;
3. Especificação e Design;
4. Desenvolvimento;
5. Entrega;
6. Avaliação;

A fase de Gestão do Projecto é transversal, realizada ao longo de todo o projecto, daí não haver um limite específico, como nas fases enumeradas anteriormente.

As fases de Avaliação e Testes são fases realizadas ao longo de todo o trabalho, à semelhança da gestão do projecto. A avaliação referida no ponto 6 da lista anterior pretende avaliar o sucesso final do projecto, através dos resultados obtidos e já representa as ilações tomadas para futuros projectos.

Foram ainda utilizadas algumas características do método de *Prototipagem Rápida*:

- Entrega contínua e rápida de *software* utilizável (neste caso, acréscimo de funcionalidades);
- Pequenas entregas e pontos de situação;
- Alterações de última hora aos requisitos são aceites;
- Conversações cara-a-cara sobre que funcionalidades a adicionar;
- Simplicidade;
- Organização.

Descreve-se com detalhe as tarefas pertencentes a cada uma das fases da estrutura seguida.



Ilustração 4 – Diagrama da Estrutura de Desenvolvimento do Projecto

### 3.1.1. Iniciação

A iniciação resume-se a três pontos principais: a identificação do problema, a pesquisa bibliográfica e a selecção da equipa.

#### **Identificação do problema:**

Esta fase do processo exige que sejam estudadas outras formas de resolver o problema, como por exemplo, o recurso a outras tecnologias, ainda que não estejam relacionadas com Multimédia. Uma vez que a necessidade sentida é específica e uma vez que já existem várias ferramentas que a auxiliam, não há vantagens em procurar outras soluções. Seria fugir ao problema.

Independentemente da área de ensino, a utilização de jogos como ferramentas para auxiliar a formação e a aprendizagem é sempre importante. Os jogos, como serão abordados nos tópicos “Definição de Jogos Sérios” e “Vantagens de Jogos Sérios na aprendizagem” desta dissertação, trazem inúmeras vantagens, desde a motivação à assimilação dos conteúdos, através de uma experiência simulada.

A área dos Jogos Sérios é relativamente recente, a pesquisa revela que começaram a ser debatidos os primeiros tópicos deste género de jogos, no início do ano 2006. Por ser uma área jovem, por haver um preconceito sobre as vantagens dos jogos no ensino e por estes serem vistos pelos adultos como ferramentas de entretenimento, a pesquisa e desenvolvimento nesta área avança de forma lenta. Por estas razões, a informação disponível é limitada, mas suficiente para o projecto a que esta dissertação se propõe.

As perspectivas futuras são que esta área tenha uma expansão considerável nos próximos anos, na medida em que começa a atrair cada vez mais a atenção de entidades, quer na área de ensino, quer na área de jogos.

### **Pesquisa Bibliográfica:**

A Pesquisa Bibliográfica para esta dissertação fez-se, maioritariamente, com o recurso à maior fonte de informação livre à escala mundial, a *internet*.

Por ser tão conveniente e estar sempre disponível, fez-se uso de motores de pesquisa, como o *Google*<sup>26</sup> e o *Google Scholar*<sup>27</sup>, que fornecem boas matérias para a investigação de uma forma rápida e relevante.

Foi consultado o conteúdo resultante das pesquisas feitas nos motores já referenciados, desde livros, artigos, teses em áreas semelhantes, vídeos de palestras, quer nas áreas de criação e desenvolvimento de jogos, quer na área de *Lean Manufacturing*.

Todo o material usado para a escrita desta dissertação está referenciado no capítulo da Bibliografia. Foram utilizados, ainda, os repositórios de Dissertações do ISEP, da FEUP e da Universidade do Minho.

### **Seleccção da equipa:**

Uma vez identificado o problema, pretende-se analisar se é viável ser produzido por uma equipa da empresa ou se é necessário recorrer a outras empresas (*outsourcing*). Uma vez que se trata de uma tese de mestrado, *outsourcing* não é opção, e a equipa resume-se a quem se encontra a realizar a tese.

Esta situação cria algumas limitações, uma vez que não se possui competências a todos os níveis, para assegurar a qualidade do projecto nas diversas áreas do desenvolvimento.

Sempre que necessário, foi solicitado o parecer de colegas de trabalho com competências em áreas em que possuía limitações, principalmente no *design* dos menus do jogo.

A verificação da qualidade das entregas intermédias ao Orientador do projecto: documentos, ecrãs de jogo, texto a incluir, neste ou noutro tipo de elementos, foi assegurada por este, que aconselhou as alterações necessárias para que o trabalho correspondesse aos critérios exigidos.

### **3.1.2. Definição**

O principal objectivo desta fase é definir o que se vai construir, por quem e como. Após o levantamento de requisitos, a análise do tema, matéria a incluir e o intervalo temporal disponível, definiu-se a seguinte ordem de trabalhos:

- Definir toda a estrutura do jogo;
- Definir os quatro cenários do Jogo;
- Criar a plataforma: menus, pontuação, *site web*, entre outros;
- Implementar de acordo o tempo disponível, um dos cenários;

---

<sup>26</sup> <http://www.google.pt/>

<sup>27</sup> <http://scholar.google.pt/>

- Testar a validade desse cenário.

Existem alguns documentos que foram criados para auxiliar este tópico. Respondem às questões: A que audiência se destina o jogo, requisitos, tecnologias, conteúdos, planeamento e riscos. São de extrema importância, pois contêm a maior parte, se não toda a informação da proposta de trabalho.

A maior parte da resposta dada a estes tópicos fez-se com recurso a reuniões com o orientador, que agiu como cliente, pesquisa e escolha cuidada, com a aprovação do mesmo através de discussões sobre o melhor rumo do projecto.

O orientador serviu, ainda, como mediador para o projecto internacional LLA, na qual o ISEP participa através do centro de investigação GILT, como já foi referido, certificando-se que era dada uma resposta a todos os requisitos.

### 3.1.3. Especificação e Design

Os documentos propostos no sub-tópico anterior já incluem em detalhe toda a informação necessária, para dar resposta ao que é pedido na fase da Especificação e Design. O desenho gráfico dos cenários proposto ajuda a criar uma ideia inicial do que se vai produzir na fase de desenvolvimento.

Desta fase resulta um conjunto de documentos, que vão ajudar a implementação, de modo a manter esta focada no que é essencial. Segue-se uma lista dos documentos que têm de ser criados antes de se avançar para a implementação:

- Lista de funcionalidades;
- Lista de interacções dos componentes;
- Definição do menu de navegação;
- Lista de materiais necessários;
- Lista de requisitos;
- Lista de entregáveis (dvd, cd, etc...);
- Listas de *hardware* necessário e requisitos mínimos para o jogo;
- Documento *Proof of Concept* (entrega bastante importante);
- Lista de normas de design (tipos de letra, tamanhos, cores);
- Estilo gráfico escolhido.

Não obstante estes documentos serem de entrega obrigatória, uma vez que o projecto possui uma equipa pequena e um tempo de implementação bastante reduzido, optou-se por elaborar apenas os que são essenciais para o projecto.

Começando pela lista de funcionalidades, esta foi produzida e pode ser consultada no quarto capítulo deste documento. Uma vez que contem as funcionalidades do jogo, desde interacção com objectos e acções, ajudou a escolher a tecnologia a utilizar, de modo a permitir a implementação destes factores.

A lista de interacções dos componentes não foi necessária, uma vez que a solução encontrada, após ser definida a tecnologia a utilizar, não justifica a sua criação.

A definição do menu de navegação pode ser consultada no quarto capítulo, usando o diagrama da interacção dos ecrãs de menu e os desenhos gráficos dos cenários.

Não foi necessária a criação de um documento para o estilo gráfico escolhido, pretendendo-se que o grafismo do jogo seja uma mais-valia para a autenticidade e realismo dos cenários projectados. Para isso, é necessário o uso de fotografias para o fundo dos cenários e objectos neles contidos.

A lista de materiais necessários pode ser deduzida da explicação dos cenários no quarto capítulo. Está relacionada com os materiais de multimédia para o jogo, bem como com o material necessário para a implementação, que se resume basicamente a um computador.

A lista de requisitos, apesar de não estar especificada, tem vindo a ser preenchida ao longo deste documento. O único requisito que o jogo tem de responder é permitir a aprendizagem e permitir pôr em prática os conteúdos de *Lean Manufacturing*, mais especificamente o Método 5S. Este tem de responder a esse requisito de forma realista, apresentando cenários realistas e comuns no dia-a-dia de produção, de modo a que o jogador se identifique e altere comportamentos de trabalho.

Não é necessário criar uma lista de entregáveis, o jogo ficará *online* e, para concluir a entrega, serão dadas as autenticações do servidor e criado um CD/DVD com todo o projecto e materiais utilizados.

Uma vez que não existe uma equipa de designers e de programadores, não existe a necessidade da criação de uma lista de normas de *Design*, de modo a que todos os colaboradores realizem o seu trabalho de acordo com as regras e este seja uniforme ao logo do projecto.

Em relação á lista de *hardware* necessário e requisitos mínimos para o jogo, dado ser um jogo realizado em *Flash* é compreensível que esta não seja muito exigente no que diz respeito a *hardware*.

O documento *Proof of Concept* é uma entrega bastante importante. Devido ao projecto já ser documentado com esta dissertação, não faz sentido repetir toda a informação aqui fornecida num documento à parte. Assim sendo, este documento pretende, até certo ponto, responder a todos os requisitos do *Proof of Concept* e a outros aspectos relevantes.

Pode, assim, afirmar-se que este documento pretende concentrar toda a informação que, de outra forma, estaria dividida em vários. Neste encontra-se toda a informação necessária para se compreender as bases do que se propõe fazer e o seu desenvolvimento.

### 3.1.4.Desenvolvimento

Nesta fase, é iniciada a produção do projecto. Começa, assim, a prototipagem, teste e avaliação, à medida que se vai avançando na implementação.

A cada avaliação das funcionalidades testadas são acrescentadas ao protótipo mais funcionalidades, que por sua vez serão posteriormente testadas e avaliadas, recomeçando assim o ciclo. O protótipo é testado, para verificar se certas funcionalidades já respondem aos requisitos do sistema. Caso se verifique que responde, avança-se para outros requisitos. Se pelo contrário ainda não responde, este é trabalhado e melhorado antes de se avançar.

Usar este tipo de abordagem ao problema é o ideal, uma vez que a *prototipagem rápida* é vocacionada para a criação de sistemas e entregas complexas. O número de protótipos pode variar pelos factores: tamanho do projecto, complexidade, competências da equipa e tempo disponível para implementação. Esta metodologia permite que haja uma certa flexibilidade, caso algum requisito mude ou seja acrescentado.

Dentro desta abordagem, pode ter-se uma prototipagem: horizontal, vertical, evolucionária ou uma de “deitar fora”. A que se usou foi a prototipagem evolucionária, o que significa que o protótipo forma a base, que é evoluída nas iterações seguintes até se considerar como produto final.

No que diz respeito ao número de protótipos que vão ser produzidos, pretende-se criar um geral com os menus e ligações aos cenários e um por cada cenário.

Esta fase conclui-se mediante o sucesso dos testes de aceitação realizados ao produto final. Estes testes finais terão de ser rigorosos, para que haja uma satisfação plena na resposta que se quer dar às necessidades sentidas.

Para evitar que esta etapa falhe, ao longo do desenvolvimento, sempre que se termina a implementação de uma funcionalidade, esta é testada extensivamente para verificar se o seu funcionamento corresponde ao pretendido. Só se continua a implementação, se as novas funcionalidades passarem nesses testes.

Haverá sempre correcções. Ainda que se corra o risco de introduzir novos erros, pretende-se sempre corrigi-los, de modo a que não sejam deixados sem solução, designados por “problemas conhecidos”.

### 3.1.5.Entrega

Esta fase inclui todos os processos necessários para preparar a entrega de um projecto multimédia.

Esta só começa quando o projecto passar com sucesso nos testes. Segue-se a preparação para a entrega (que inclui arquivar o projecto e o *debriefing* da equipa) e a entrega final do produto.

Todas as tarefas pendentes têm de estar concluídas e, a partir deste ponto, não são autorizadas mais alterações. Deve-se acabar a documentação do produto e preparar a entrega. Como, neste caso, a entrega será um site onde o jogo será colocado, deve-se preparar o servidor, colocar todos os recursos necessários *online* e certificar-se que tudo está em ordem. Cópias da documentação, bem como suporte digital (CD, ou DVD) com todo o conteúdo *online*, devem ser entregues.

Segue-se a arquivagem do Projecto. Assim que a entrega esteja concluída, deve-se arquivá-lo. Este passo é importante para futuros projectos.

Deve ser feito um balanço sobre a forma como este decorreu e devem ser tiradas conclusões sobre o método de trabalho e o que se pode melhorar no processo.

Têm de ser guardadas cópias de todo o material entregue, bem como o que se usou para a sua realização.

Nesta fase, pode ser ainda debatida a necessidade de criar Acordos de Direitos Autoriais, licenciamento de patentes e suporte técnico a quem se destina o projecto.

### 3.1.6. Avaliação

Concluindo esta metodologia de trabalho, segue-se a avaliação do projecto. Este processo interactivo está vocacionado para o uso do produto que se implementou e não nas suas características e funcionalidades.

Esta fase é de extrema importância, uma vez que, se for feita atempadamente, permite detectar falhas no *design* do produto, permitindo a sua correcção e evitando custos em refazê-lo.

Devem ser avaliados os seguintes critérios:

- Usabilidade;
- Design do produto;
- Robustez do protótipo;
- Revisões realizadas por especialistas;

#### **Usabilidade:**

Avaliar a eficácia do produto e a sua facilidade de uso. Têm de ser avaliados factores, como a facilidade de aprendizagem pelo utilizador ao usar o produto, a facilidade de recuperação de erros do utilizador, se o produto é interessante e agradável de ser usado, entre outros.

#### **Design do produto:**

Este aspecto da avaliação permite verificar se o produto foi desenvolvido de acordo com as especificações delineadas nos documentos de *design* e se responde aos requisitos pedidos.

Este processo é realizado no fim de cada protótipo produzido e é realizado pela equipa de produção, pelos representantes do projecto e por um conjunto de utilizadores finais.

#### **Robustez do protótipo (produto final):**

Uma vez concluído, é necessário estudar o grau de aceitação do público a que se destina. Algumas técnicas de avaliação incluem:

- Observar os utilizadores enquanto estes interagem com o produto;
- Fazer entrevistas a grupos de utilizadores através de questionários;
- *Benchmarking* (comparação) e análise competitiva de outros produtos no mercado;
- Avaliações heurísticas (avaliação do software em relação a procedimentos de usabilidade aceites);
- Contratar empresa especializada em avaliações de produtos.

Com estes resultados, pode depois, elaborar-se uma lista de melhoramentos a serem realizados ao produto. Pode ser ponderada a hipótese de realizar uma segunda versão deste baseada na sua evolução.

Dada a dimensão do projecto, não foi sentida a necessidade de realizar uma avaliação tão extensiva como foi descrito. A avaliação escolhida para este projecto é executada através de um questionário, que vários alunos terão de preencher, após concluírem o jogo.

Alunos que frequentem o curso de Engenharia Mecânica são a população ideal para testarem o Jogo, uma vez que estudam o conteúdo deste, nas suas aulas. A opinião destes é de extrema importância, dado que delineará os melhoramentos futuros antes do lançamento oficial do jogo.

Permitirá, ainda, saber se o jogo responde à necessidade sentida na aprendizagem destes conteúdos, se é relevante, motivador e se cumpre todos os requisitos necessários, através do grau de satisfação dos utilizadores.

#### **Revisões realizadas por especialistas:**

Este processo envolve uma avaliação realizada por peritos (internos e externos) em sistemas multimédia e em conteúdos. Estes darão *feedback* das melhorias que devem ser implementadas, para melhorar a qualidade final do produto.

##### **3.1.6.1. Questionário de avaliação do jogo**

Existem vários tipos de questionários, sendo mais comum um que possui várias afirmações, em que os alunos assinalam, numa escala de 0 (mínimo) e 5 (máximo), o quanto concordam com elas. Este documento é de preenchimento obrigatório, pelos alunos que o irão avaliar.

## 4. Conceção

“Games lubricate the body and the mind...”

**Benjamin Franklin**

Neste capítulo, debate-se a concepção do projecto, desde a ideia às ferramentas de implementação, passando pelos conceitos do jogo, objectivos, *design* gráfico dos cenários e outros tópicos.

### 4.1. Ideia do jogo

O objectivo do jogo é aumentar a produtividade do colaborador, através do ensinamento de conteúdos de *Lean Manufacturing*, mais especificamente o Método 5S. Para a empresa, resulta num aumento da sua competitividade e redução dos desperdícios, uma vez que está a otimizar os recursos disponíveis.

Dado que simular fisicamente postos de trabalho, de modo a que os colaboradores pratiquem e aprendam a aplicar as regras dos 5S é dispendioso em tempo e trabalho, surge a necessidade de criação de um Jogo Sério, que simule um ambiente de trabalho e ajude na aprendizagem destes temas.

Propõe-se, então, um Jogo Sério que dê resposta a esta necessidade, permitindo pôr em prática o Método 5S, servindo ainda como ferramenta motivadora, para que o colaborador aprofunde o seu conhecimento na matéria.

A ideia prática do jogo é confrontar o utilizador com uma tarefa a ser por ele executada. Para a realização, são necessárias ferramentas e matérias-primas. É fornecida ao utilizador uma lista com todos os itens, de onde o jogador terá de escolher os que precisa (*sort*). Atribui-se uma identificação, *tags* (vermelhas, amarelas e verdes) aos itens, consoante a sua importância na realização da tarefa.

O utilizador terá de escolher os sítios onde se podem guardar os itens até à altura de serem necessários. Deverá ser escolhido o sítio mais conveniente para cada (*stabilize*).

Ao visualizar o ambiente de trabalho, deve avaliar se é necessário fazer algum tipo de limpeza e, em caso afirmativo, executá-la (*shine*).

Através de vários cenários, o utilizador terá uma ideia da aplicabilidade de cada um destes procedimentos. Poderá criar e sugerir zonas destes cenários para aplicação de outros procedimentos, também relevantes (*standardize*).

No último tópico (*sustain*), pretende-se que o utilizador seja capaz de manter estes procedimentos.

Para a validação deste jogo, deverá ser feita a avaliação da performance do utilizador nos diversos cenários, no início e no final dos melhoramentos por ele propostos.

Os quatro cenários que se propõem são:

1. Armazém;
2. Posto de trabalho fabril (ou bancada partilhada por mais de um funcionário);
3. Posto de trabalho num escritório de uma empresa;
4. Ambiente de um computador (pastas, documentos, reciclagem, atalhos, ambiente de trabalho).

Todos os cenários pretendem transmitir as mesmas competências, de modo a que a assimilação deste seja executada através de comparações entre eles.

A partir deste jogo, o utilizador poderá concluir que este conteúdo se pode aplicar a praticamente todos os aspectos relacionados com produtividade na área em que exerce as funções.

Resumindo, pretende-se que o jogador organize vários cenários de acordo com o Método 5S de modo a aumentar a sua produtividade, aprendendo conceitos de *Lean*.

### 4.1.1. Concepção dos cenários da aplicação

#### 4.1.1.1. Cenário 1 – Armazém

Cena 1.1 – Armazém desorganizado

**Localização:**

Armazém desorganizado.

**Objectivos:**

Encontrar vários objectos de uma lista;

Encontrar e preencher o formulário de requisição de objectos;

Organizar uma parte específica do armazém.

**Objectivos da aprendizagem:**

Pretende-se que o utilizador aprenda e aplique os conceitos do Método 5S para um Armazém.

**Lugares:**

Entrada do armazém onde se encontra a secretária;

O jogador pode atravessar o armazém do início ao fim, visualizando todas as prateleiras.

**Tema ou material:**

Ambiente de armazém, organização e facilidade na requisição de materiais.

**Personagens:**

A personagem do jogador é a própria pessoa.

**Objectos:**

O jogador pode mover e rotular as caixas e prateleiras do armazém, bem como organizar os formulários da secretária onde irá entregar a requisição das peças que pretende levantar.

**Guião:**

O utilizador começa por ver um armazém. À entrada tem uma secretária com um computador e vários documentos desorganizados, empilhados e espalhados. Continuando a andar em direcção as prateleiras, passa por várias caixas empilhadas nos cantos do armazém, sem rótulos ou qualquer tipo de identificação. Terá acesso a corredores com prateleiras, também estas sem identificação. As prateleiras armazenam caixas nas mesmas condições das referidas anteriormente.

O utilizador, ao seleccionar a caixa, pode visualizar os objectos que esta contém e procura, assim, os que pretende levantar. Dado o estado actual do cenário, o utilizador irá perder bastante tempo a procurar e a requisitar os objectos que necessita.

Quando estiver na posse dos objectos que pretende, terá de fazer a requisição destes, preenchendo um documento que terá de procurar na pilha dos documentos da secretária, que visualizou à entrada do armazém. Após ter concluído o processo de preenchimento do formulário, deixá-lo-á pousado em cima da secretária para ser processado.

O jogador será cronometrado, para se ter um ponto de partida para a avaliação do seu desempenho em actos seguintes, com o objectivo de se poder aprovar ou não os melhoramentos realizados. De modo a complementar a avaliação dada pelo intervalo de tempo decorrido, diagramas de *spaghetti* serão usados para verificar se o percurso feito pelo jogador é o mais rápido.

Ainda no acto 1, o utilizador repetirá o nível, agora com o modo de edição ligado. Desta forma, o utilizador vai organizando as caixas, prateleiras e a secretária do armazém à medida que vai jogando. No final, poderá submeter para avaliação a sua proposta.

Os melhoramentos serão avaliados com uma terceira ronda cronometrada e pela forma como organizou o armazém, por exemplo, se colocou os itens menos usados em prateleiras mais altas, se os itens mais usados estão nas prateleiras mais perto da entrada do armazém, entre outros factores. Os diagramas de *spaghetti* serão uma ferramenta indispensável para a avaliação do

modelo proposto. Pode visualizar-se uma lista das acções que realizou para organizar o armazém (dando conhecimentos teóricos dos 5S).

#### Cena 1.2 – Armazém organizado

##### **Localização:**

Armazém organizado.

##### **Objectivos:**

Encontrar vários objectos de uma lista;

Encontrar e preencher o documento de requisição de objectos;

Organizar uma parte específica do armazém.

##### **Objectivos da aprendizagem:**

Pretende-se que o utilizador aprenda e aplique os conceitos do Método 5S para um Armazém.

Mostrar ao jogador a diferença de produtividade quando o local de trabalho está organizado.

##### **Lugares:**

Entrada do armazém onde está a secretária;

O jogador pode atravessar o armazém do princípio ao fim, visualizando todas as prateleiras.

##### **Tema ou material:**

Ambiente de armazém, organização e facilidade na requisição de materiais.

##### **Personagens:**

A personagem do jogador é a própria pessoa.

##### **Objectos:**

O jogador pode mover e rotular as caixas e prateleiras do armazém, bem como organizar os formulários da secretária, onde irá entregar a requisição das peças que pretende levantar.

##### **Guião:**

Na segunda parte do jogo, visualiza-se, agora, o mesmo armazém já organizado com o Método 5S em prática. A lista dos objectos que o utilizador tem para levantar é a mesma do acto anterior, por exemplo A23:Chave de fendas, A4:Chave estrela, B34:Módulo de ignição, A35:Chave correspondente.

Desta vez não haverá caixas sem rótulos no chão do armazém, todas as prateleiras estão ordenadas alfabeticamente ou da forma mais apropriada. As caixas serão numeradas de 20 em 20 ou de acordo com o inventário em vigor.

O utilizador deve aperceber-se que reduziu consideravelmente o tempo que levou a procurar todos os itens que necessita. Após ter encontrado tudo o que precisa, terá, à semelhança do acto 1, de fazer o preenchimento do formulário de requisição do material. Desta vez, a secretária encontra-se organizada: Existem sobre a mesa arquivadores para colocar os formulários e outros documentos. Os formulários estão num canto da mesa. Preencher e depositar numa caixa para estes documentos é simples. Poderá submeter o formulário digitalmente, no computador, se preferir.

Todo este processo é temporizado, permitindo visualizar o aumento da performance do utilizador. Mais uma vez, irá ser usado diagrama de *spaghetti* como auxiliar na avaliação da performance do jogador.

### Cena 1.3 – Armazém por medida

#### **Localização:**

Armazém organizado passível de ser reestruturado.

#### **Objectivos:**

Encontrar vários objectos de uma lista;

Encontrar e preencher o formulário de requisição de objectos;

Organizar uma parte específica do armazém.

#### **Objectivos da aprendizagem:**

Pretende-se que o utilizador aprenda e aplique os conceitos do Método 5S para um Armazém e obtenha ideias para melhorar o método de organização proposto, permitindo organizá-lo de acordo com a sua experiência no ramo.

#### **Lugares:**

Entrada do armazém onde está a secretária;

O jogador pode atravessar o armazém do início ao fim, visualizando todas as prateleiras.

#### **Tema ou material:**

Ambiente de armazém, organização, facilidade na requisição de materiais e reestruturação do mesmo.

**Personagens:**

A personagem do jogador é a própria pessoa.

**Objectos:**

O jogador pode mover e rotular as caixas e prateleiras do armazém, bem como organizar os formulários da secretária, onde irá entregar a requisição das peças que pretende levantar.

**Guião:**

O Acto 3 destina-se a obter *feedback* do utilizador. Caso se trate de um utilizador já com anos de experiência no negócio, registar e implementar as suas ideias é de extrema importância para a actualização do jogo.

Pretende-se registar as ideias do utilizador em texto, permitir alterações nos cenários de jogo (prateleiras, caixas, secretária) e, numa segunda fase, verificar os melhoramentos implementados.

#### 4.1.1.2. Cenário 2 - Posto de trabalho fabril

Cena 1.1 – Posto de trabalho fabril

**Localização:**

Posto de trabalho fabril.

**Objectivos:**

Organização do espaço de trabalho;

Criação de novas áreas para guardar ferramentas e materiais;

Desenhar as silhuetas das ferramentas nos sítios da parede onde se vão guardar;

Separar as ferramentas por frequência de utilização;

Procurar e encaixar as ferramentas no lugar correcto;

Processos de limpeza, standardização, selecção (quase todos os 5S).

**Objectivos da aprendizagem:**

O objectivo deste cenário é mostrar como pode ser aplicado o Método 5S nos postos de trabalho fabril. Este é o principal cenário, para o qual este método foi elaborado.

**Lugares:**

Local de trabalho onde está localizada a caixa de ferramentas;

Parede de ferramentas que terá de criar.

**Tema ou material:**

Ambiente fabril de produção, organização, facilidade em encontrar e usar as ferramentas que se pretende.

**Personagens:**

A personagem do jogador é a própria pessoa.

**Objectos:**

Várias ferramentas de trabalho;

Caixa de ferramentas e objectos para a criação da parede de ferramentas.

**Guião:**

Pretende-se, neste cenário, aplicar as regras dos 5S num posto de trabalho fabril.

No início, existe uma caixa de ferramentas desorganizada e todas as ferramentas estão dentro das gavetas ou pousadas no seu topo. Esta possui ainda material desnecessário e raramente utilizado. A caixa<sup>28</sup> a utilizar no jogo é o tipo de caixa mais usado em fábricas e oficinas, possui várias prateleiras e rodas para tornar fácil a sua mobilidade.

À semelhança do cenário anterior, será atribuída uma tarefa. Para a realizar, o jogador terá de usar as ferramentas que estão na caixa.

O jogador terá de organizar a caixa e criar um espaço numa parede onde irá colocar as ferramentas menos utilizadas ou que não têm utilidade para aquela tarefa. Apesar de ser necessário deslocar-se ao local onde estão as ferramentas, este percurso será percorrido menos vezes.

Para tornar este processo mais criativo, o jogador terá de desenhar a silhueta das ferramentas no local onde estas irão ser colocadas.

A melhor maneira de avaliar a aplicação do Método 5S pelo jogador, é cronometrar o tempo que este demora a encontrar uma determinada ferramenta. Pode usar-se o diagrama de *spaghetti*, para visualizar os movimentos do jogador e melhor se avaliar a sua performance.

#### 4.1.1.3. Cenário 3 - Posto de trabalho num escritório de uma empresa

Cena 1.1 – Posto de trabalho escritório

**Localização:**

Posto de trabalho num escritório.

---

<sup>28</sup>Imagens de caixas podem ser visualizadas em [www.beta-tools.com](http://www.beta-tools.com)

**Objectivos:**

Aplicar os 5S ao ambiente do escritório;  
Organizar documentos (separar, descartar e ordenar);  
Encontrar um documento em específico.

**Objectivos da aprendizagem:**

O objectivo deste cenário é mostrar como pode ser aplicado o Método 5S num escritório. Permitir que o jogador aplique o Método 5S na organização da sua secretária de trabalho e entenda o efeito desta na sua produtividade.

**Lugares:**

Secretária de escritório;  
Canto do escritório com estantes para armazenar arquivadores.

**Tema ou material:**

Ambiente de escritório, organização e facilidade na pesquisa de documentos.

**Personagens:**

A personagem do jogador é a própria pessoa.

**Objectos:**

Documentos, arquivadores de vários tipos, capas, caixote do lixo, etiquetas e outro material de escritório que auxilie a organização.

**Guião:**

O terceiro cenário é reconhecido por quem trabalha num escritório. Manter o posto de trabalho organizado é uma tarefa metódica, que exige algum tempo e dedicação.

As vantagens de ter uma secretária organizada são inúmeras, principalmente no aumento da produtividade, por se despendem menos tempo à procura de um determinado documento, que se perdeu de vista.

O que se propõe com este cenário é que o utilizador organize uma secretária (do mesmo modo que organizou a secretária no armazém) e um conjunto de estantes com documentos e arquivadores.

À primeira vista, temos uma secretária muito desorganizada, há papéis amarrotados que necessitam de ir para o lixo, documentos importantes empilhados, facturas por arquivar, cartas de correio por abrir e um sem número de documentos que são expectáveis numa secretária de um indivíduo que realize qualquer trabalho burocrático.

O utilizador pode, então, puxar e empurrar a pilha de documentos, de modo a poder colocar separadores, arquivadores ou caixas de documentos em cima da mesa. Pode atribuir rótulos a estes auxiliares de organização, colocar e remover arquivadores da estante, de modo a melhorar a arrumação do espaço de trabalho.

O utilizador pode seleccionar objectos, ver o seu conteúdo e, assim, decidir como pretende organizá-los. Existe um cesto do lixo, para onde os documentos que não forem necessários podem ser arrastados. Na parte superior do ecrã, existe ainda uma barra onde poderá ir buscar materiais como arquivadores, prateleiras, etiquetas, entre outros, que podem ser arrastados para área de trabalho. Existe interacção entre os documentos com os itens da barra superior. Com isto, o utilizador pode começar a arrastar os documentos para os respectivos sítios.

No final, o jogador deve responder a uma pergunta, como por exemplo, “Mostre a factura da electricidade do mês Agosto de 2010”. O jogador terá de procurar no local onde arquivou o documento pedido e entregá-lo.

Cronometrando o tempo entre esta tarefa e uma idêntica, executada antes do processo de arrumação, permite avaliar a diferença de tempo demorado a encontrar o documento pretendido.

#### 4.1.1.4. Cenário 4 - Ambiente do computador

Cena 1.1 – Ambiente do computador

**Localização:**

Computador pessoal.

**Objectivos:**

Sugerir um sistema de ficheiros comum a todos;

Ordenar o ambiente de trabalho de um computador.

**Objectivos da aprendizagem:**

O objectivo deste cenário é mostrar como pode ser aplicado o Método 5S num sistema digital, como um computador. Permitir que o jogador aplique o Método 5S na organização do seu computador de trabalho.

**Lugares:**

Num ambiente de trabalho de um computador *Windows*;

Na raiz do disco rígido do sistema.

**Tema ou material:**

Ambiente de computação, organização digital, facilidade na orientação em sistemas digitais.

**Personagens:**

A personagem do jogador é a própria pessoa.

**Objectos:**

Pastas e ícones espectáveis num computador de trabalho.

**Guião:**

Porque será que é tão difícil navegar pelo sistema de ficheiros no computador de outras pessoas? É realmente vantajoso, ou produtivo, ter todos os documentos, imagens ou outro tipo de ficheiros espalhados pelo ambiente de trabalho?

O que se propõe com este jogo é que o utilizador aplique o método 5S no PC de uma empresa de contabilidade, por exemplo.

Existem dois possíveis cenários: um em que o utilizador cria uma árvore de pastas para guardar ficheiros de um dado negócio (contabilidade) e outro em que organiza um ambiente de trabalho de um computador.

Para a organização do ambiente de trabalho, simula-se, de forma realista, o ambiente usado na plataforma *Windows*, com ícones: Meu computador, reciclagem, ficheiros e pastas.

O objectivo chave é a organização de informação digital. O método de avaliação de performance do jogador será realizado com recurso a diagramas de *spaghetti*, criados através da monitorização da posição do cursor no ecrã do computador. Para a criação da raiz de pastas, a avaliação será feita verificando e comparando a raiz, proposta pelo jogador, a uma raiz ideal, proposta por especialistas na área de negócio.

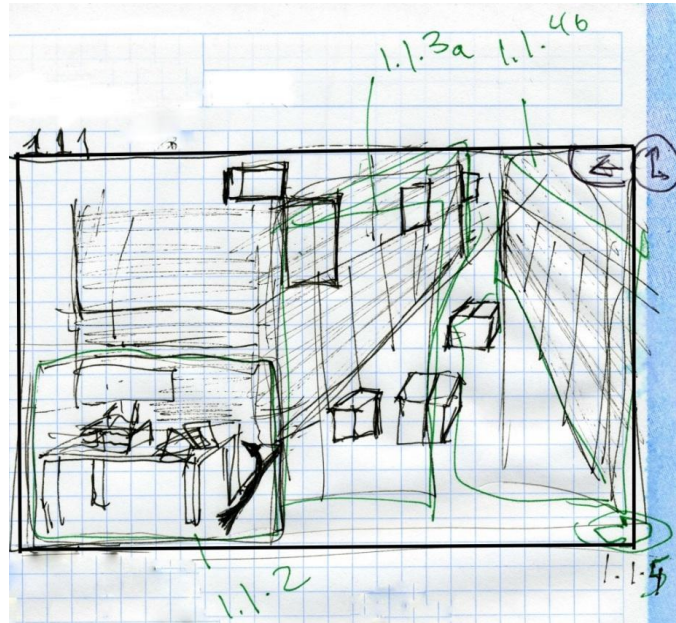
## 4.2. Desenho gráfico dos cenários

Este tópico é uma peça fundamental para a compreensão do jogo que se pretende desenvolver. São dados exemplos de ecrãs de jogo, através de esboços desenhados à mão, e ilustrações encontradas na *internet*, para ajudar a criar a ideia base para os cenários.

Com este documento, pretende-se criar uma referência visual dos conceitos do jogo, elaborados no tópico anterior.

#### 4.2.1. Cenário 1

Para o primeiro cenário, é necessário criar três cenas: Armazém desorganizado (com e sem modo de edição), Armazém organizado e cena com vista aérea para reordenar o Armazém.



**Ilustração 5 – Ecrã com a vista do Armazém desorganizado**

Como se pode observar na Ilustração 5, começa-se por ver um Armazém desorganizado, constituído por uma área com a secretária (1.1.2), onde se encontram os formulários para requisição dos materiais, entre outros documentos, e áreas de prateleiras organizadas em corredores (1.1.3.a e 1.1.3.b).

É possível clicar em cada uma das áreas (a verde), para ampliar e aceder a essa cena com mais especificidade e poder interagir com os objectos nela contidos.

Existe um botão seguinte (1.1.5) com a visualização de mais corredores com prateleiras (ver Ilustração 5).

No canto superior direito, existe um botão para voltar ao menu principal do jogo e um relógio, que se destina a cronometrar as tarefas do jogador. Estes dois elementos são estáticos, no que diz respeito ao posicionamento no ecrã, e independentes dos cenários.

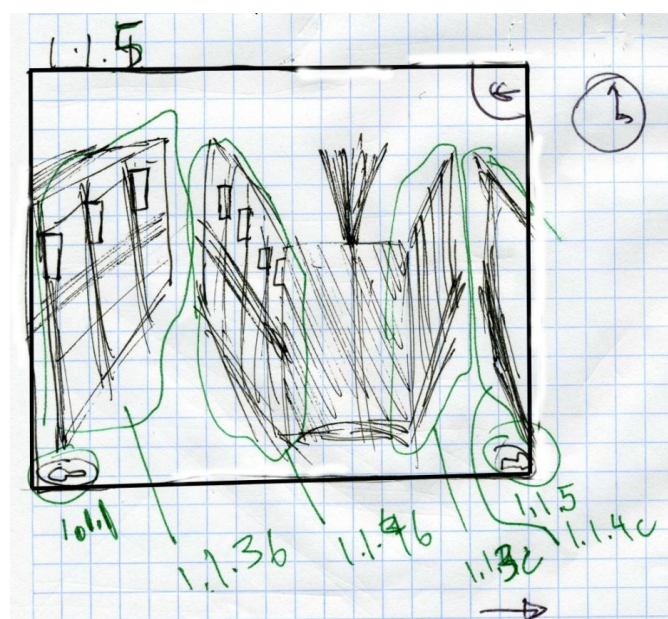
Se o jogador clicar na secção da secretária (área 1.1.2), poderá visualizar uma de duas cenas: uma secretária desorganizada ou uma secretária organizada, consoante o cenário em que o jogador se encontra.



**Ilustração 6 – Cena da secretária organizada/desorganizada**

Na Ilustração 6, pode visualizar-se uma parte do cenário do Armazém, constituído pela secretária desorganizada (1.2.1) e pela secretária organizada (1.2.2), que será o objectivo e estado final que o jogador terá de alcançar.

É permitido ao utilizador, no cenário em que o Armazém se encontra desorganizado, que este o organize, à medida que joga. Para organizar a secretária, o utilizador terá uma barra com as ferramentas de organização disponíveis, semelhante à existente no do escritório, cenário 3.

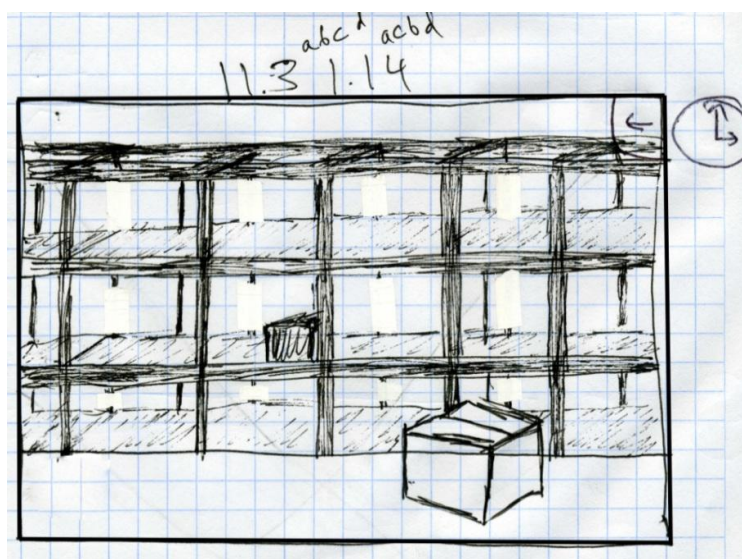


**Ilustração 7 – Cena com mais corredores do Armazém**

Com a Ilustração 7, pretende-se mostrar a cena que é visualizada, quando o jogador clica nas setas de navegação.

Caso esteja na fila que pretende, pode clicar então nas áreas a verde, escolhendo as prateleiras à esquerda ou a direita. Se, pelo contrário, ainda não está no corredor pretendido, pode avançar ou recuar, através das setas de navegação em baixo.

Para tornar o jogo mais realista, o jogador perderá tempo a percorrer o corredor, antes de mudar para o seguinte.



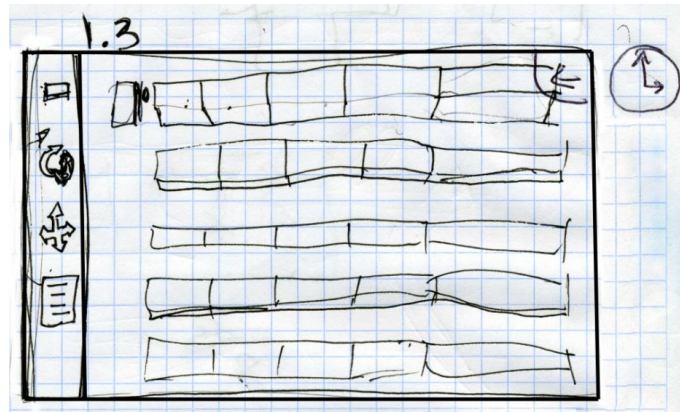
**Ilustração 8 – Cena com a prateleira seleccionada**

Independentemente de ter escolhido a fila da esquerda ou direita, a vista da prateleira é sempre igual (Ilustração 8), só mudam as posições dos caixotes pousados. Como se trata da cena do armazém desorganizado, pode reparar-se que algumas filas têm caixotes pousados fora das prateleiras ou pousados em sítios não adequados.

O jogador pode arrastar os caixotes e seleccionar, para ver o seu conteúdo. Caso o modo de edição esteja ligado, pode acrescentar, remover e nomear caixotes.

O botão de recuar faz o jogador voltar à cena dos corredores de prateleiras, na Ilustração 7.

Para concluir, no cenário do Armazém, a última cena permite que o jogador altere a organização estrutural deste.



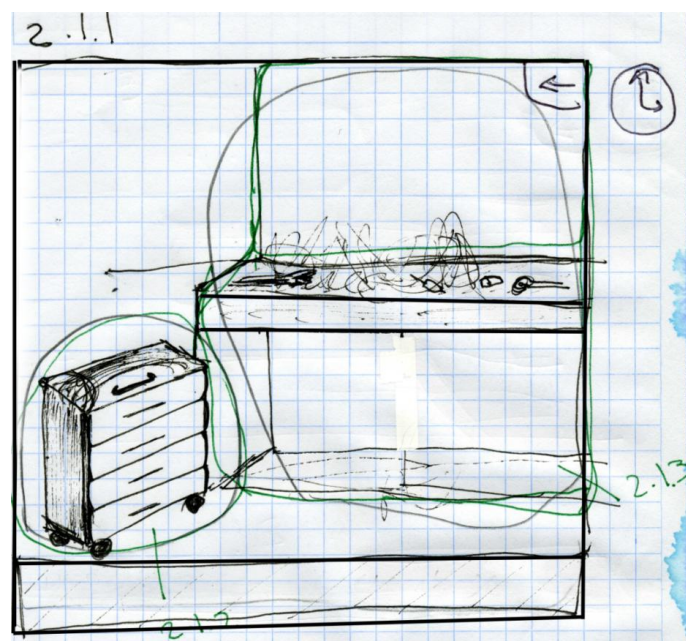
**Ilustração 9 – Vista aérea para organização estrutural do Armazém**

Na Ilustração 9, visualiza-se uma perspectiva aérea do Armazém. Esta perspectiva ajuda o jogador a melhor organizá-lo, caso não concorde com a disposição actual das prateleiras. Tem ao seu dispor uma barra com ferramentas, para deixar as suas sugestões, rodar prateleiras, adicionar rótulos, entre outras funcionalidades.

#### 4.2.2. Cenário 2

O cenário 2 é o posto de trabalho fabril. Neste cenário, existem três cenas: a bancada de trabalho, a parede em frente, onde se vai desenhar e colocar as ferramentas que são menos usadas ou volumosas, e a caixa das ferramentas.

O jogador terá de organizar a caixa de ferramentas e criar um espaço numa parede, onde irá colocar as ferramentas menos utilizadas ou que não têm utilidade para aquela tarefa.



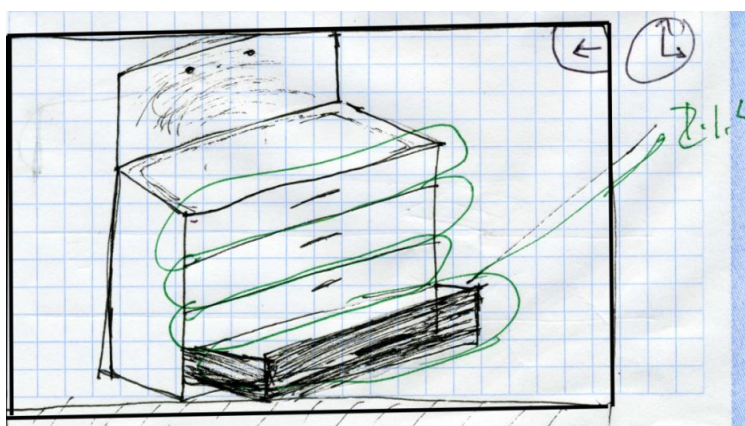
**Ilustração 10 – Cena com a caixa de ferramentas, bancada e parede**

Na Ilustração 10, pode visualizar-se o espaço de trabalho onde ocorre o jogo. Como nos cenários anteriores, ao clicar nas áreas marcadas a verde (2.1.2 e 2.1.3), estas são ampliadas, permitindo ver mais pormenores e interagir com os objectos.



**Ilustração 11 – Exemplo de uma caixa de ferramentas e bancada de trabalho**

Os botões de voltar, nas ilustrações seguintes, fazem o jogador regressar à cena da Ilustração 12 e, por sua vez, o botão de voltar desta, faz o jogador regressar ao menu principal.



**Ilustração 12 – Cena com a caixa de ferramentas e gavetas**

Ao seleccionar a área 2.1.2 da Ilustração 10, é visualizada a cena da Ilustração 12 com a caixa das ferramentas e respectivas gavetas. Este ecrã permite que o jogador escolha qual a gaveta que pretende abrir.

A Ilustração 13 pretende mostrar o conteúdo da gaveta que o jogador escolheu abrir. O conteúdo, inicialmente, estará desorganizado.

O jogador pode, enquanto organiza o cenário, tirar as ferramentas da gaveta, arrastando-as para a barra posicionada no fundo da cena.

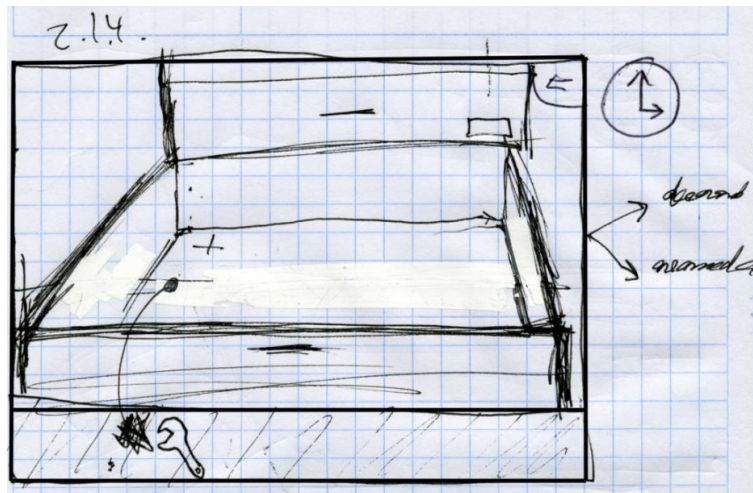


Ilustração 13 – Cena com uma das gavetas aberta, da caixa de ferramentas

De seguida, o utilizador pode pintar a silhueta da ferramenta seleccionada na parede em frente à bancada de trabalho. A barra do fundo possui a área de desenho, o botão em forma de pincel que activa o modo de pintura, botão para posicionar o desenho na parede, como se pode ver na Ilustração 14.

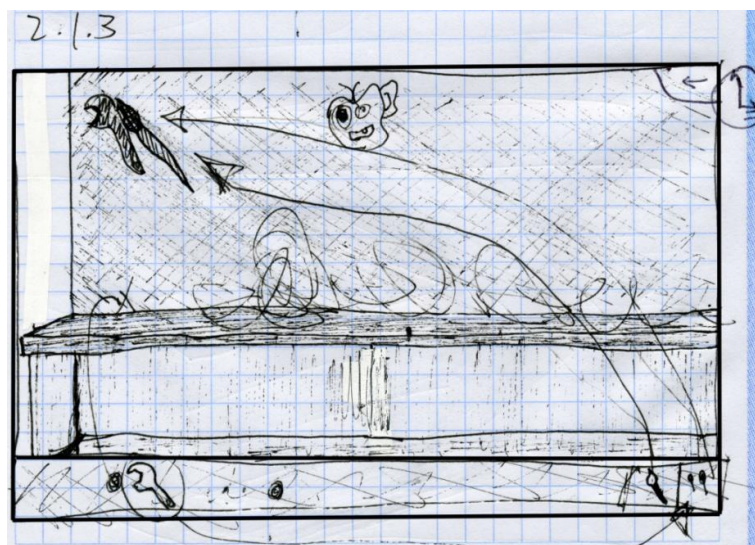


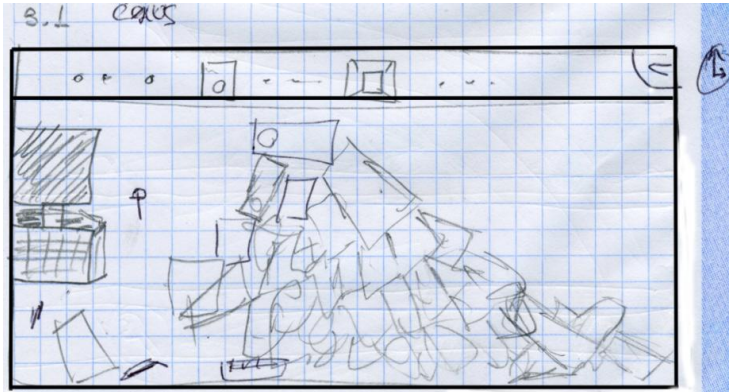
Ilustração 14 – Cena da bancada de trabalho e da parede de ferramentas

A avaliação do jogador consiste em verificar se este encontra uma determinada ferramenta pedida.

Uma imagem desta ferramenta surgirá no fundo do ecrã e o jogador deverá procurar e arrastar a ferramenta para cima da mesma, simbolizando que encontrou o que era pedido.

### 4.2.3. Cenário 3

O tema do cenário 3 é o posto de trabalho num escritório. O objectivo é organizar um posto de trabalho de um escritório de acordo com o Método 5S.



**Ilustração 15 – Secretária desorganizada**

A cena inicial consiste numa secretária desorganizada, como a da Ilustração 15. O jogador deve organizar este posto de trabalho, movendo os objectos que se encontram em cima da mesa.



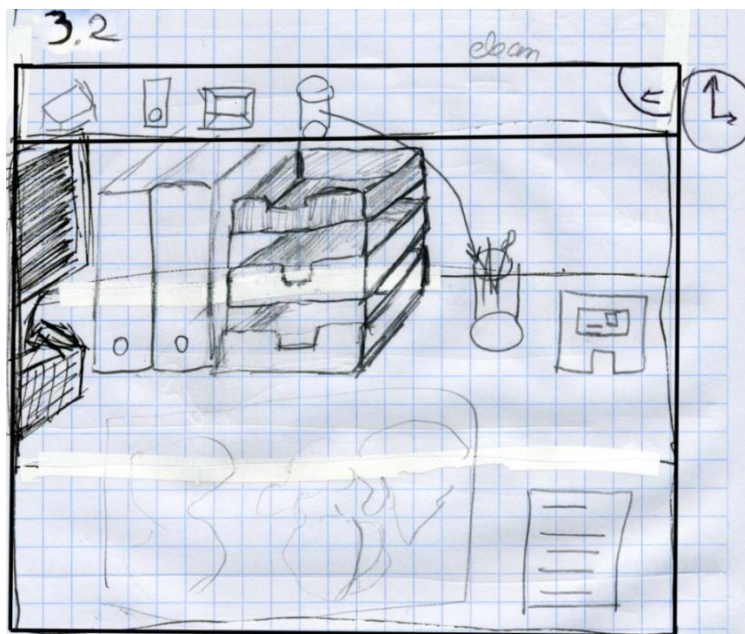
**Ilustração 16 – Exemplo de postos de escritório desorganizados**

Deve usar as ferramentas de organização, como capas, pastas, etc, disponíveis na barra superior.



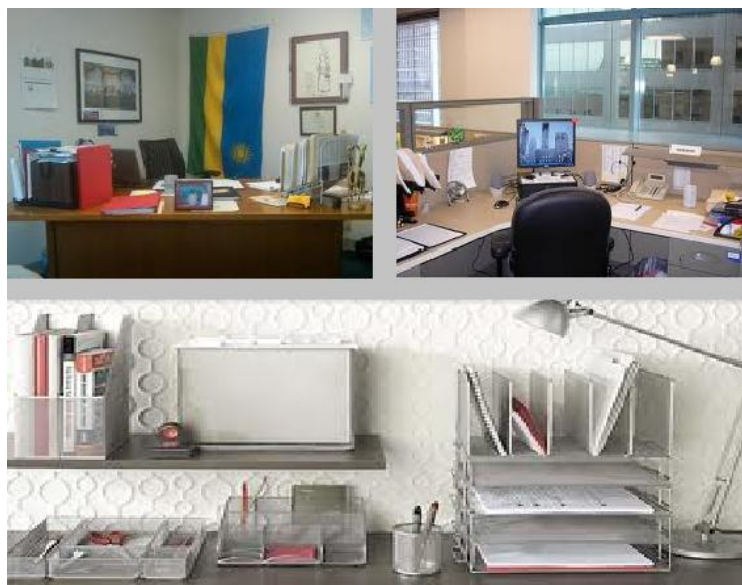
**Ilustração 17 – Exemplos de ferramentas de organização.**

A Ilustração 18 mostra um exemplo de uma situação ideal possível, que se pretende obter no final deste cenário. O jogador terá ainda um armário, onde pode colocar as capas com os documentos que arquivou.



**Ilustração 18 – Secretária organizada**

O canto inferior direito será o espaço de visualização do documento que o jogador terá de encontrar, ao realizar a sua avaliação de performance.



**Ilustração 19 – Exemplos de postos de escritório organizados**

#### 4.2.4. Cenário 4

O cenário 4 pretende simular um ambiente de trabalho de um computador. Tem como finalidade criar um cenário autêntico. Será realizado com recurso a imagens capturadas de ecrãs de computador da plataforma *Windows 7*.

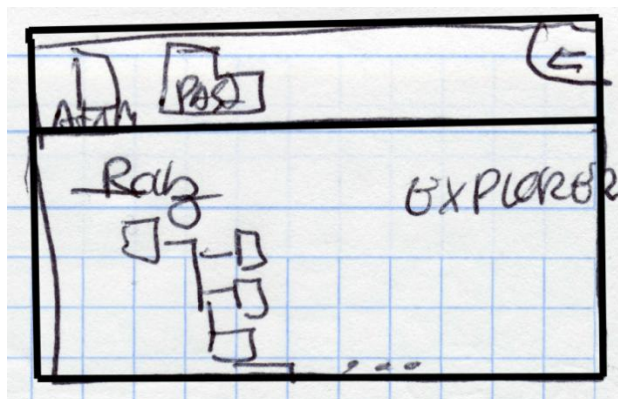


Ilustração 20 – Cena para a criação da raiz de ficheiros

Para a tarefa em que o jogador deve criar uma raiz de ficheiros, para o negócio onde o computador se insere, pretende-se que o ambiente de jogo seja idêntico ao “Explorador do *Windows*”, como se mostra na Ilustração 20.

Serão permitidas ações, como botão direito do rato, deslocar por arrasto pastas e ficheiros. O jogador pode usar as ferramentas dispostas na barra colocada na parte superior do ecrã.

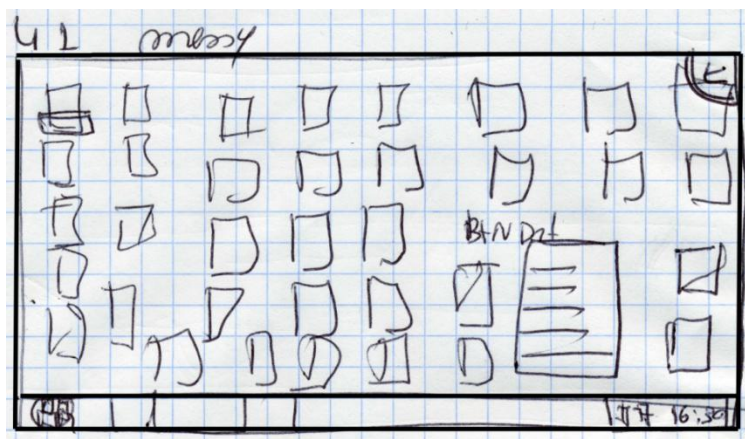
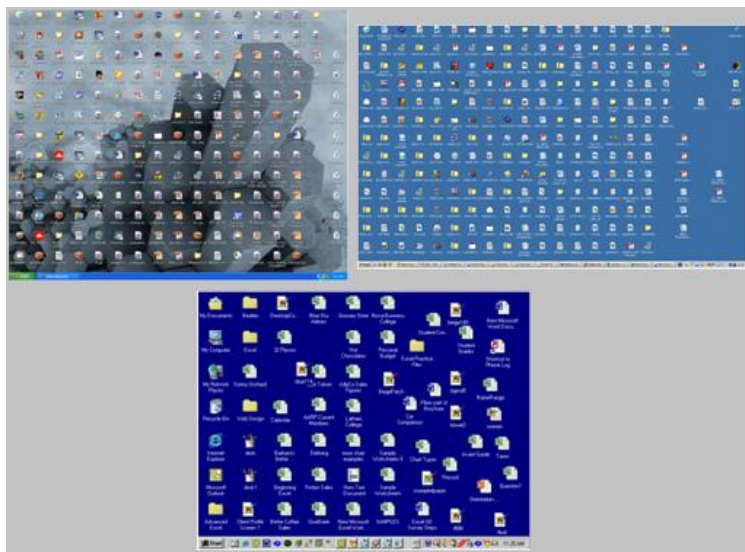


Ilustração 21 – Ambiente de trabalho antes de ser organizado

Para a segunda tarefa deste cenário, a organização do ambiente de trabalho do computador, serão usadas imagens de fundo e de objectos capturadas de computadores que possuem o *Windows 7* como sistema operativo.

Inicialmente, será mostrada uma cena idêntica à da Ilustração 21: um ambiente de trabalho desorganizado, com documentos, atalhos e pastas espalhados pelo ecrã.



**Ilustração 22 – Exemplos de ambientes de trabalho desorganizados**

No final, pretende-se um ambiente de trabalho organizado, idêntico ao da Ilustração 23. Um ambiente limpo e fácil de utilizar.



**Ilustração 23 – Exemplo de um ambiente de trabalho organizado**

### 4.3. Desenho da estrutura do jogo e cenário escolhido

Neste tópico, faz-se uma referência das especificações técnicas para o cenário escolhido, para o protótipo do Jogo Sério.

Serão abordados temas como a estrutura do jogo, menus, funcionalidades que o jogo terá de permitir, eventos, lista de materiais necessários, entre outros tópicos.

Optou-se por delinear toda a estrutura do jogo, desde os menus iniciais à pontuação, área para exposição de conteúdos informativos, créditos e um cenário completo dos quatro projectados. Assim, é igualmente possível obter *feedback* dos jogadores, através de um cenário jogável, analisando se é justificável investir nos restantes ou mudar a abordagem.

Uma vez que se trata de uma área em que ainda existe pouca actividade, não se consegue prever, à partida, se o jogo terá sucesso e se será relevante nesta. Esta abordagem concede alguma segurança, permitindo que não se desperdice tempo e dinheiro num projecto que não forneça resultados favoráveis.

#### 4.3.1. Diagrama da estrutura do jogo

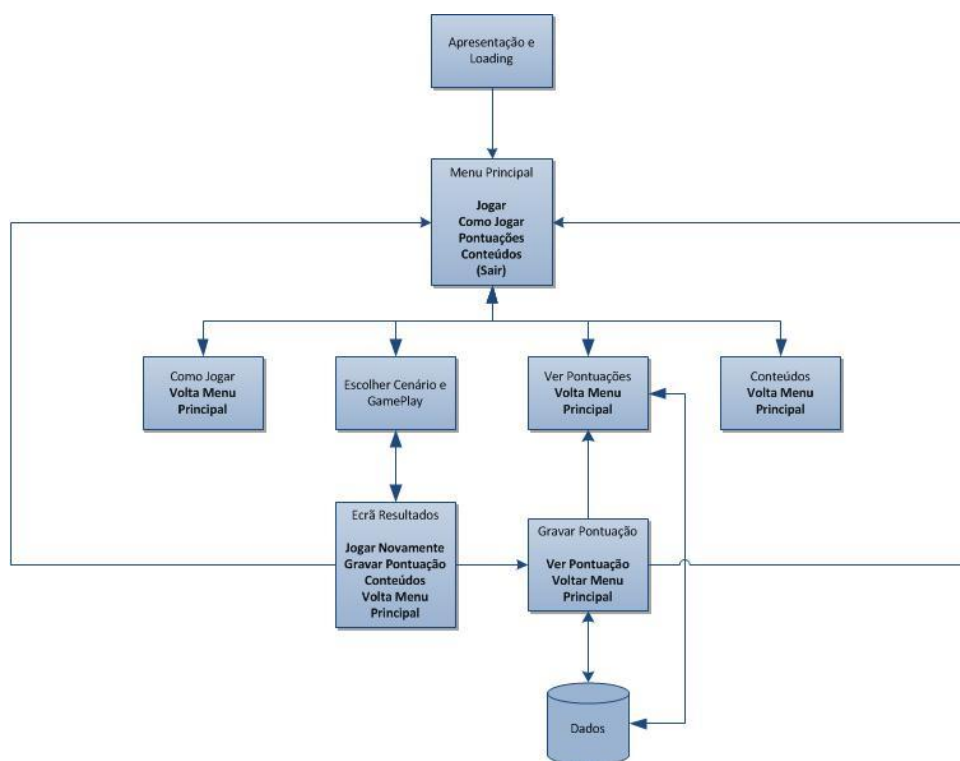


Ilustração 24 – Wireframe dos ecrãs do jogo

#### 4.3.2. Núcleo de funcionalidades

##### 4.3.2.1. Interações

###### Botão esquerdo do rato:

- Clique simples selecciona objecto;
- Clique contínuo:

- Objecto - Permite arrastar objecto (*Drag&Drop*);
- Modo pintar – Permite pintar a silhueta de uma ferramenta;
- Duplo clique:
  - Abre, permitindo a visualização do objecto com mais detalhe (*Zoom In*);
  - Editar nome da etiqueta.

#### **Botão direito do rato:**

- Clique simples abre o submenu.

#### **Input do teclado:**

- Utilização dos caracteres do teclado e teclas especiais como o *enter*.
- Clique simples abre menu de contexto com opções.

#### **4.3.3.Eventos e variáveis**

- Posicionamento pelas coordenadas do cursor do rato;
- *Tracking* das posições do rato para os diagramas de *spaghetti*;
- Sistema de cronometração de tarefas;
- Variáveis de controlo aliadas a cada objecto (alterado de posição, entre de outro objecto);

#### **4.3.4.Interface com o utilizador**

- Menu;
- Área de jogo;
- Objectos;
- Objectos interagem com outros objectos;
- Botões para navegação das diversas áreas de jogo;
- Relógio para a cronometração de tarefas;
- Sistema de apontamentos pessoais (*Post-Its* ou editor de texto);
- Permitir adição de rótulos nos objectos com texto introduzido pelo utilizador.

#### **4.3.5.Lista de materiais**

- **Menus**

Menus para todos os ecrãs: Inicial, Menu principal, Dados do jogador, Escolher cenário, Resultados, Gravar pontuações, Ver Pontuações, Ver conteúdos, Como jogar, Sair.

- **Cenários**

Desenhar cenários do Armazém e do Computador (Explorador do *Windows* e Ambiente de trabalho - ver Desenho Gráfico dos Cenários).

- **Objectos**

Desenhar vários objectos passíveis de serem encontrados nos cenários já descritos, (documentos, computador, prateleiras, caixas, material de escritório, ícones de documentos digitais, Meu computador, Reciclagem, ficheiros e pastas, ferramentas, peças, etc).

- **Botões**

Botões de navegação;

Setas de navegação;

Botões para customizar o cenário;

- **Barras**

Barras superior para o relógio e os auxiliares de organização;

Barra inferior semelhante a superior.

- **Sons**

Sons de fundo para a introdução, *loading* e menu principal;

Ao clicar num documento, criar pasta, largar documento, finalizar cenário, alarmes do cronómetro.



## 5. Implementação do Trabalho

“Choose a job you love, and you will never have to work a day in your life.”

**Confucius**

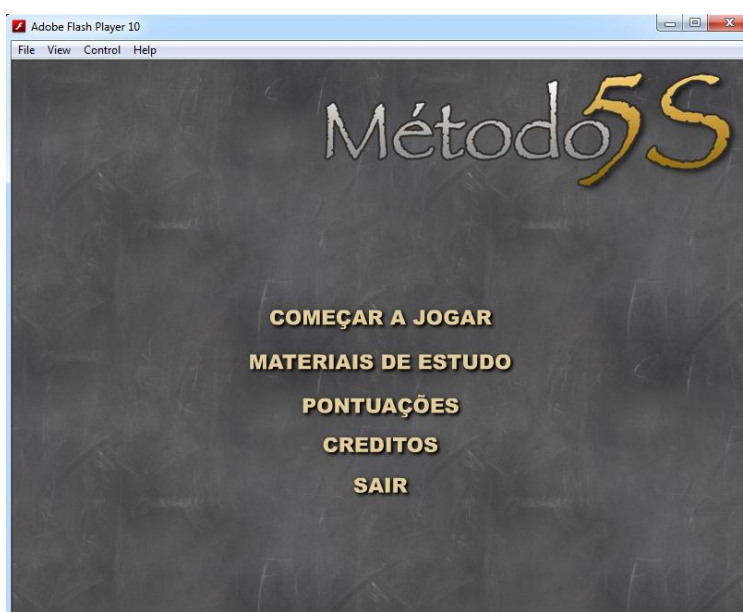
Pretende-se, com este capítulo da dissertação, apresentar o protótipo do jogo implementado, à medida que se vai explicando o funcionamento deste e das suas funcionalidades.

Este jogo é constituído por duas partes distintas, o menu e os cenários individuais, que representam *swfs* (ficheiros que contém aplicações *Flash*) separadas.

### 5.1. Explicação do menu do jogo

A comunicação entre o menu e os cenários é feita através de um processo de carregamento de conteúdos do *Flash*.

O primeiro ecrã do jogo é menu principal, na Ilustração 25. Este contém todas as opções do jogo.



**Ilustração 25 – Primeiro ecrã do jogo**

Uma vez que se trata do menu principal do jogo, este ecrã permite ao utilizador escolher a opção da tarefa que pretende realizar, seja começar o jogo, rever conteúdos para a aprendizagem da matéria teórica a que o jogo faz referência, verificar a sua posição em relação a outros jogadores na tabela de pontuações, ver os créditos do jogo ou sair.

Começa-se a explicação das opções por ordem ascendente, de modo a simplificar o processo.

A última opção é a de sair do jogo. Uma vez que o jogo será jogado numa página *web*, o botão sair apenas redirecciona o jogador para um motor de pesquisa.

O motivo pelo qual se redirecciona o jogador para um motor de pesquisa baseia-se no incentivo à aprendizagem. O jogador pode, assim que acaba o seu jogo, pesquisar algum conceito, que sinta que não entendeu completamente, ou aprofundar mais o seu conhecimento nalgum tópico que tenha despertado o seu interesse.

A escolha do motor de pesquisa baseia-se em dois factores: número de respostas dadas e popularidade.

Continuando a subir nas opções do menu geral, seguem-se os Créditos do Jogo.

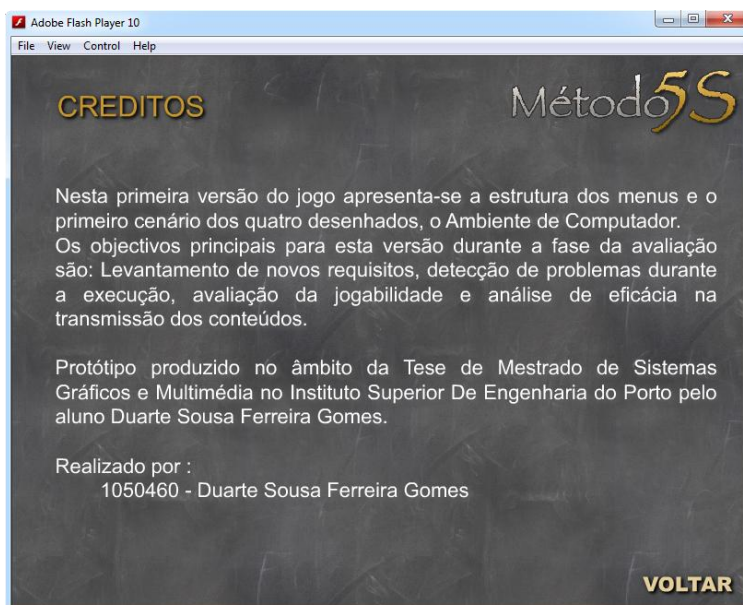


Ilustração 26 – Ecrã de Créditos

No ecrã da Ilustração 26, pretende-se fazer um enquadramento do trabalho, transmitir ao jogador os objectivos que se pretendem alcançar. Assim, identifica-se o autor do trabalho, o âmbito para o qual este foi desenvolvido, permitindo obter um melhor *feedback* de quem jogar.

Continuando a explicação das opções do menu, segue-se a tabela de pontuações. Na Ilustração 27, encontra-se o ecrã que contém a tabela de pontuações.

NomeJogador	NomeCenário	PontuaçãoOrganização	PontuaçãoTempo	TempoDemorado	PontuaçãoTotal	Data
Duarte	Ambiente de Computador	200	100	400	600	2011-06-17
David	Ambiente de Computador	100	200	100	200	2011-06-15

**Ilustração 27 – Tabela de pontuações do jogo**

Nesta tabela, Ilustração 27, pode ser consultada a seguinte informação: Nome do jogador, cenário que jogou, pontuação atribuída à sua organização do cenário, pontuação atribuída ao tempo que demorou a realizar a tarefa, pontuação total e a data em que jogou. Mais à frente, neste documento, será explicado o processo de atribuição da classificação obtida.

A opção seguinte fornecida é a de “Materiais de Estudo”, Ilustração 28. Esse ecrã tem como principal objectivo permitir ao jogador rever os conteúdos abordados pelo jogo. Aqui pode ler toda a informação teórica sobre o Método 5S.

Fica aqui criada a área e as bases para futuramente se incluir num esquema de “aulas” com outros conteúdos teóricos, de modo a dinamizar o ensino de conteúdos de *Lean Manufacturing*.

**MATERIAIS DE ESTUDO** Método 5S

5S garante que existe um lugar para tudo e que tudo está no seu lugar – limpo, arrumado e pronto a ser usado

5S representa um posto de trabalho seguro e livre de riscos.

1. Sort/Seleccionar – classificar as coisas e eliminar as desnecessárias;
2. Stabilize (set in order)/Sistematizar – organizar as coisas de uma forma sistemática;
3. Shine/Polir – limpar o posto de trabalho;
4. Standardize/Standardizar – standardizar as actividades normais de trabalho e desenvolver instruções de trabalho;
5. Sustain/Sustentar – manter 5S como cultura de trabalho.

**Ilustração 28 – Área de consulta dos materiais de estudo**

Para concluir a explicação do menu principal do jogo, resta definir o comportamento da primeira opção do menu. Seleccionando “Começar a jogar”, o menu avança para o ecrã de escolha do cenário, que o jogador pretende jogar.

O menu de escolha de cenário, Ilustração 29, é semelhante ao menu principal da aplicação. Tem-se quatro opções que correspondem aos quatro cenários desenhados para o jogo, que, por sua vez, representam os quatro ambientes de produção escolhidos: O Armazém, o Posto de Trabalho Fabril, o Escritório e o Ambiente de Trabalho de um Computador na área da contabilidade.



Ilustração 29 – Menu de escolha de cenário a jogar

Da perspectiva de programação, apesar das semelhanças com o menu anterior, este já oferece outro desafio, que é carregar *swfs* individuais, em que cada um possui um cenário diferente.

O motivo pelo qual se optou por implementar cada cenário num *swf* diferente deve-se ao facto de ser mais prática a organização de código e gestão de conteúdos destes.

Foram criados três *swfs* temporários, que representam os cenários que se decidiu deixar por implementar nesta primeira fase.

O ideal será criar uma pasta para cada cenário.

Como apenas o cenário 4, Ambiente de Computador, foi produzido, só este se encontra numa pasta separado dos outros.

Seguem-se as linhas de código responsáveis pelo carregamento dos *swfs*.

```

//adicionar evento dos botões
bt_voltar.addEventListener(MouseEvent.CLICK,btnClick);
COMPUTADOR.addEventListener(MouseEvent.CLICK, btnClick);
ARMAZEM.addEventListener(MouseEvent.CLICK, btnClick);
ESCRITORIO.addEventListener(MouseEvent.CLICK, btnClick);
FABRIL.addEventListener(MouseEvent.CLICK, btnClick);

//Callback dos botões
function btnClick(event:MouseEvent):void
{
    if(event.target.name=='bt_voltar')
    {
        gotoAndStop("Splash");
    }
    else
    {
        //usa o loader para chamar o jogo
        var newSWFRequest:URLRequest;
        if(event.target.name == "COMPUTADOR")
        {
            newSWFRequest = new
URLRequest("swfs/cenario4/cenario4.swf");
        }
        else
        {
            newSWFRequest = new URLRequest("swfs/" +
event.target.name + ".swf");
        }
        loader.x = 0;
        loader.y = 0;
        loader.load(newSWFRequest);
        stage.addChild(loader);
    }
}

```

**Código 1 – Função que carrega os diferentes *swfs* dos cenários**

Como se pode verificar, Código 1, todos os botões chamam a mesma função, que vai ser responsável por carregar os *swfs*. Esta recebe, por parâmetro, o evento *MouseEvent* que, por sua vez, possui informações do objecto que o despoletou. Avaliando essa informação, pode saber-se o nome do botão em que o jogador carregou e abrir o *swf* pedido.

Caso o jogador tenha seleccionado o botão de um dos cenários que não tenham sido desenvolvidos, pode voltar para o menu da escolha destes, pressionando o botão esquerdo do rato em qualquer parte do ecrã.

De modo a que o jogador não fique preso, caso entre nos cenários não desenvolvidos acidentalmente, estes foram equipados com um evento que faz o jogador voltar ao menu de escolha de cenários, se pressionar o botão esquerdo do rato em qualquer parte do ecrã.

Este processo ajuda a que, quando os novos *swfs* com os cenários estiverem prontos, seja, apenas, necessário atribuir uma pasta e um nome específico a estes para poderem ser carregados.

## 5.2. Explicação do quarto cenário – Ambiente de Computador

Passando agora para o último cenário da lista, o quarto – Ambiente de Computador, o processo de carregamento deste é em tudo semelhante aos cenários anteriores.

Uma vez que este se encontra completo, apenas se pode voltar ao menu principal, quando se terminar o jogo e a sua classificação seja registada.

### 5.2.1. Explicação dos ecrãs iniciais do cenário

Este cenário pretende emular o Sistema Operativo *Windows 7*, por ser o SO mais usado no mercado empresarial e no mercado industrial.

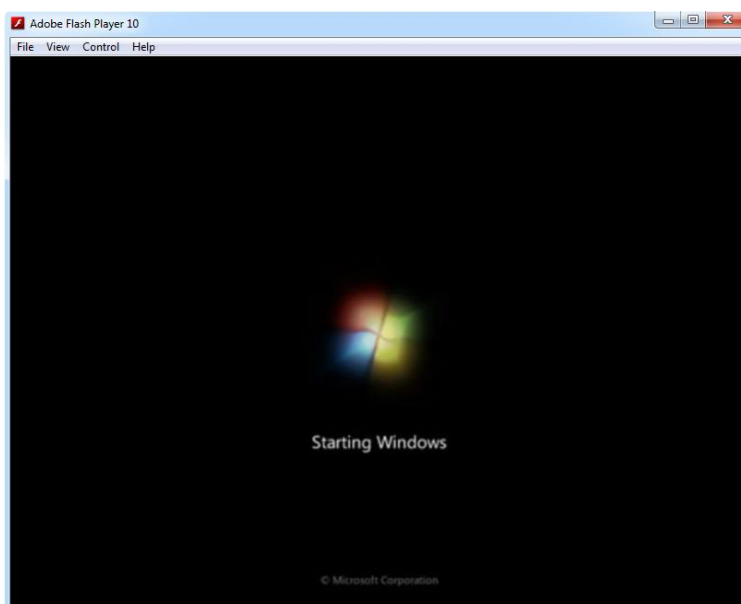


Ilustração 30 – Primeiro ecrã: Ecrã de carregamento do cenário

O primeiro ecrã, Ilustração 30, pretende, para além de tornar mais realista o cenário através da simulação de ligar o computador onde irá decorrer o jogo, carregar todo o material necessário.

Está ainda associado um evento temporal, que faz avançar para a fase de autenticação, passado um segundo e meio após ter sido aberto.

Classe associada: *menuLoading.as\**

Esta classe possui um *Movieclip* associado, que corresponde ao ecrã de *loading* do jogo e que antecede o ecrã de introdução do nome do jogador.

Durante os breves instantes em que este ecrã é visível, são carregadas informações necessárias para se poder jogar o cenário. Estas informações são lidas de ficheiros XML, que possuem o nome e outros dados de cada elemento e a sua posição no ecrã.

Segue-se o ecrã de autenticação, Ilustração 31.

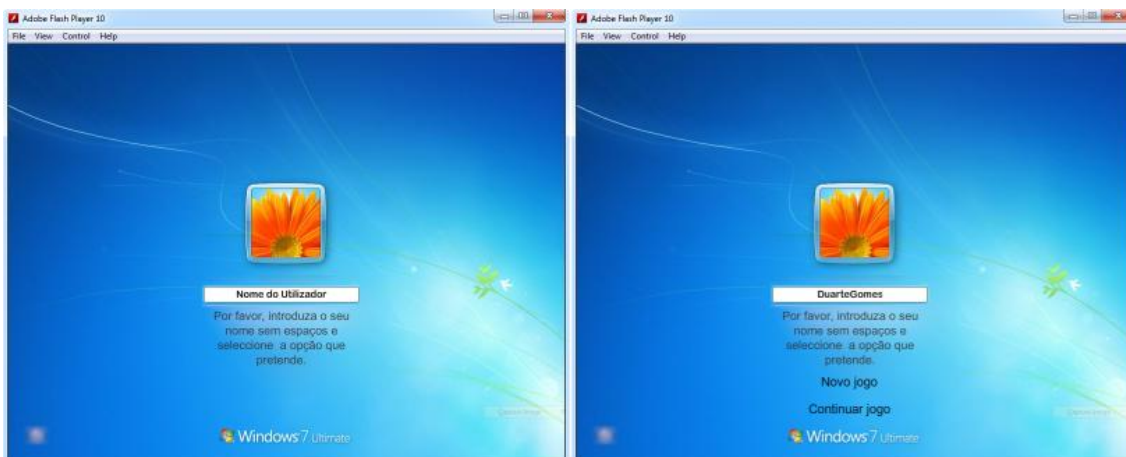


Ilustração 31 – Ecrã de autenticação (esquerda por preencher, direita preenchido)

Neste ecrã, o jogador terá de introduzir o seu primeiro e último nomes sem espaços. O motivo pelo qual é feito este pedido deve-se ao facto de se criar um *SharedObject* com o nome do jogador. Um *SharedObject* é um ficheiro que guarda o estado do jogo, semelhante aos *cookies* num *browser*. O jogador pode introduzir espaços, mas serão removidos automaticamente, uma vez que o *Flash* não autoriza que o nome do ficheiro de um *SharedObject* possua espaços.

O processo de criação/leitura de um *SharedObject* não apresenta complexidade que justifique uma explicação do seu funcionamento.

O jogador deve seleccionar a caixa de introdução do nome e preenchê-la. Para submeter, deve pressionar a tecla *enter*, como faria num formulário.

Uma vez preenchida, serão dadas duas opções ao utilizador: Começar um jogo novo ou continuar um jogo que tenha previamente salvado. A opção de continuar o jogo só será mostrada, se o jogador tiver salvado o seu jogo anteriormente.

Um novo jogo terá de ser começado, caso o jogador nunca tenha salvado o jogo ou tenha concluído (com ou sem sucesso) o cenário na vez anterior. Se o jogador escolher continuar o jogo, serão lidas todas as informações do seu *SharedObject* e carregadas, para que possa continuar no ponto onde salvou. Se por outro lado começar um jogo novo, caso nunca tenha jogado, será criado um novo *SharedObject* para o jogador ou rescrito o actual.

Classe associada: *menuEntrada2.as\**

Esta classe é responsável pelo ecrã de autenticação no jogo. Possui as funções responsáveis por adicionar os eventos à caixa de texto, onde o jogador irá introduzir o seu nome. Para fazer isto, é necessário remover os eventos do *stage*, que estão à escuta que certas teclas sejam pressionadas, como a tecla M, de modo a evitar abrir o menu, quando o jogador estiver a escrever o nome.

Criar e salvar um *SharedObject*

Os *SharedObjects* servem para armazenar dados, uma vez que este jogo permite que jogador possa interrompê-lo e continuar a jogar, quando for mais oportuno.

Estes objectos têm uma importância acrescida. São usados, ainda, para guardar as melhores organizações propostas pelos jogadores, armazenando todos os elementos presentes no cenário e todas as informações deste, como posição, nome, conteúdo, etc.

Posto isto, a função de introdução de nome verifica a existência de um *SharedObject* com o nome para o jogador e, consoante se existe ou não um, torna visível a opção de continuar um jogo ou recomeça-lo. Caso exista, é na mesma mostrada a opção de jogar um novo jogo.

Assim que seleccione a opção, o jogo pode então começar.

```

//Abrir um SharedObject (pode criar se não houver)
var so:SharedObject = SharedObject.getLocal("cenario4_" +
Main.novoJogo.nomeJogador);
if (so.data.xml == undefined) //Porque se gravou o XML
{
    //Realiza a acção pretendida
}
so.close();

//Gravar o SharedObject
var so:SharedObject = SharedObject.getLocal("cenario4_" +
this.nomeJogador);
//Definir a estrutura de dados
so.data.xml = Main.xml;
so.data.zonas = Main.zonas;
Main.pastas.sortOn("naPasta", Array.NUMERIC);
so.data.pastas = Main.pastas;
so.data.documentos = Main.documentos;
so.data.reciclagens = Main.reciclagens;
so.data.trackArray = Main.trackArray;
so.data.duploClickArray = Main.duploClickArray;
so.data.logArray = Main.logArray;
so.data.tempo = this.tempoRest;
so.data.tempoDemorado = this.tempoDemorado;
//Gravar e fechar
so.flush();
so.close();

//Limpar um SharedObject (não remove o ficheiro)
var so:SharedObject = SharedObject.getLocal("cenario4_" +
this.nomeJogador);
so.clear();
so.close();

```

**Código 2 – Exemplo de como manipular um *SharedObject***

No Código 2, pode observar-se como se manipula um *SharedObject*.

É importante fechar a ligação do *SharedObject*, quando esta já não for necessária, através da função *close*. Caso não se feche esta ligação, pode resultar em perda dos dados.

De referir, ainda, que, apesar de não ser possível guardar classes não serializadas em *SharedObjects*, se acidentalmente acontecer, não resulta num erro de compilação. A execução normal da aplicação só revela um problema de acesso, por não encontrar o objecto em questão ou a errada leitura deste, quando o encontra. Este facto leva o programador a procurar o problema num local errado, uma vez que este se encontra, quando se guarda e não quando se lê o *SharedObject*.

## Objectos e classes guardados em SharedObject

Explica-se, agora, o processo de criação de uma classe que gera objectos, que podem ser guardados em *SharedObjects*.

```
//Na Classe Main
registerClassAlias("ZonaDef", ZonaDef);

//ZonaDef.as
package classes
{
    public class ZonaDef
    {
        public var nome:String;
        public var posX:int;
        public var posY:int;
        public var w:int;
        public var h:int;
        public var estado:Boolean;

        public function ZonaDef(nome:String = "", x1:int = 0,
y1:int = 0, w1:int = 0, h1:int = 0, estado:Boolean = false):void
        {
            this.nome = nome;
            this.posX = x1;
            this.posY = y1;
            this.w = w1;
            this.h = h1;
            this.estado = estado;
        }
    }
}
```

Código 3 – Exemplo de uma classe cujo objecto pode ser guardado em *SharedObject*

No Código 3, encontra-se a classe *ZonaDef* responsável por converter um elemento *Zona* (que irá ser definida mais à frente neste tópico), num elemento que permita ser guardado em *SharedObject*. A classe *ZonaDef* apenas guarda as variáveis essenciais, para que o jogador possa resumir o seu jogo noutra ocasião.

A função responsável por tornar o objecto da classe *ZonaDef* compatível com o *SharedObject* é a função *RegisterClassAlias*.

Esta função permite preservar a classe (tipo) de um objecto, quando este é codificado em AMF. Evita que o objecto seja guardado sem o seu *alias* (nome, ou identidade), permitindo que, quando for feita a descodificação do objecto guardado, a classe seja recuperada.

Sempre que o *Flash* descodifica um objecto de uma classe cujo, *alias* destes não tenha sido guardado, este torna-se num objecto anónimo.

O principal objectivo desta etapa é determinar quanto tempo este demora a encontrar um determinado documento, de modo a avaliar o estado de organização do cenário.

Classe associada: *menuObjectivos.as\**

Antes de se começar o jogo e logo após a introdução do nome, o jogador irá visualizar um ecrã com os objectivos deste.

Apesar de não ser necessária a criação de uma classe para este ecrã, optou-se por fazê-lo, de modo a facilitar a introdução de novas funcionalidades, caso sejam necessárias futuramente. De momento, esta classe apenas possui o evento do botão “continuar”, para permitir começar o jogo.

Segue-se uma breve explicação do procedimento que o jogador terá de seguir para “submeter” o documento quando o encontrar. No canto inferior direito do ecrã estará colocada uma seta que, ao clicar, abre uma área de submissão do documento. Nesta serão visualizados o tipo e nome do documento a procurar. Caso o documento se encontre no ambiente de trabalho, terá de arrastar e largar em cima da área de submissão. Se, por outro lado, este se encontrar dentro de uma pasta, um duplo *click* no documento chega para o submeter.

A escolha do documento a entregar é aleatória. Um modo de conseguir isto é adicionando os documentos, presentes no cenário, a um *array* (conjunto de elementos) à medida que se lêem. De seguida, usa-se uma função que devolve um número aleatório entre um e o número máximo de elementos do *array* em questão. Resta associar o documento do *array* ao número saído.

```
//Devolve um número aleatório
public function randomNumber(low:int=0, high:int=1):int
{
    return int(Math.floor(Math.random() * (1+high-low)) + low);
}
```

**Código 4 – Função que gera um número aleatório**

No Código 4, pode-se visualizar a função usada para criar números aleatórios.

### **5.2.2. Explicação técnica do funcionamento do jogo**

Uma vez que se utilizou o *FlashDevelop* para ajudar a escrever as classes deste cenário, foi necessário reencaminhar o ficheiro *Main.as* para a pasta do projecto que escolhemos, de modo ao *Flash* ignorar o *Main.as* que criou, quando se começou o projecto.

Uma vez configuradas as propriedades do projecto, pode, então, começar-se a desenvolvê-lo, criando as classes que forem necessárias, usando a ferramenta de desenvolvimento preferida.

Name	Date modified	Type	Size
Main.as	03-10-2011 21:28	AS File	53 KB
caixaInfo.as	03-10-2011 20:03	AS File	2 KB
Reciclagem.as	03-10-2011 12:53	AS File	29 KB
Jogo.as	03-10-2011 02:10	AS File	10 KB
Pasta.as	30-09-2011 18:40	AS File	43 KB
Icon.as	30-09-2011 17:54	AS File	14 KB
Documento.as	30-09-2011 17:53	AS File	19 KB
IconDef.as	28-09-2011 19:03	AS File	1 KB
tipoElemento.as	28-09-2011 19:02	AS File	3 KB
Fundo.as	24-09-2011 18:57	AS File	8 KB
rightClick.as	24-09-2011 18:23	AS File	1 KB
Zona.as	22-09-2011 16:15	AS File	10 KB
menuSair2.as	20-09-2011 22:25	AS File	4 KB
ReciclagemDef.as	20-09-2011 22:07	AS File	1 KB
Cronometro.as	22-08-2011 23:06	AS File	7 KB
menuEntrada2.as	22-08-2011 22:29	AS File	4 KB
procuraDocumento.as	22-08-2011 17:27	AS File	3 KB
PastaDef.as	19-08-2011 16:21	AS File	2 KB
firmNivel2.as	19-08-2011 15:30	AS File	2 KB
firmNivel.as	19-08-2011 15:29	AS File	4 KB
menuObjectivos3.as	19-08-2011 15:21	AS File	2 KB
botaoSave.as	19-08-2011 12:17	AS File	1 KB
nomePastaMenu.as	16-08-2011 19:00	AS File	1 KB
DocumentoDef.as	18-07-2011 18:43	AS File	1 KB
nomeMenu.as	21-06-2011 11:11	AS File	1 KB
ZonaDef.as	14-06-2011 16:27	AS File	1 KB
menuLoading.as	26-05-2011 23:03	AS File	1 KB
menuObjectivos2.as	25-05-2011 12:03	AS File	1 KB
menuObjectivos.as	23-05-2011 15:13	AS File	1 KB
Selector.as	14-05-2011 21:01	AS File	1 KB
adobe	27-09-2011 15:12	File folder	

**Ilustração 32 – Listagem das classes criadas para o quarto cenário**

Na Ilustração 32, está presente a listagem das classes que foram criadas para responder ao estilo de jogo, objectivos, requisitos e forma de jogar.

A diferença entre o projecto do quarto cenário e o projecto do menu do jogo baseia-se em que, tirando os objectos da biblioteca, este não possui elementos no seu *stage* que tenham sido adicionados antes de o projecto ser compilado.

Quer isto dizer que todos os elementos são colocados e removidos do *stage*, através do código das classes, o leva a que tenha de haver um maior controlo destes mesmos objectos, para que sejam adicionados ou removidos nos tempos certos. Assim, evita-se que não haja acesso a objectos que não se encontram disponíveis.

### 5.2.2.1. Adicionar e remover elementos do stage

Para adicionar e remover elementos do *stage*, nomeadamente ecrãs, foi usado o seguinte método.

```
//Classe Main
//Ecrãs iniciais de jogo
private var loading:menuLoading;

//Primeiro Ecrã de Loading
loading = new menuLoading();
loading.x = 0;
loading.y = 0;
loading.name "loading";
loading.addEventListener(menuLoading.FIM_MENU_LOADING,
menuEntradaLoading, false, 0, true);
stage.addChild(loading);

//Menu inicial de introdução do nome do jogador e tipo de jogo
private function menuEntradaLoading(e:Event):void
{
    stage.removeChild(loading);
    //Código para carregar ecrã seguinte
    (...)
}

//Classe Loading
public static var FIM_MENU_LOADING:String = "FIM_MENU_LOADING";

public function menuLoading():void
{
    //Carregar todos os ficheiros necessários
    (...)
    //Quando estiver concluído chama função avançar
    (...)
}

public function avançar():void
{
    dispatchEvent(new Event(FIM_MENU_LOADING));
}
```

Código 5 – Adicionar e remover elementos ao stage

No Código 5, pode observar-se como é adicionado e removido o ecrã de *loading*. No fundo, este ecrã é um objecto da classe *menuLoading*. É criado, chamando o construtor da sua classe, e é adicionado ao *stage* através da função *addChild*.

De seguida, atribui-se-lhe um evento de escuta, que aguarda que o objecto sinalize o fim das suas tarefas e que está pronto para a sua remoção. Assim que este evento se realizar, a função responsável remove o objecto através da função *removeChild*, carregando o ecrã seguinte.

Classe associada: *Main.as*

A classe Main, por ser a classe principal da aplicação (daí o seu nome), é a mais extensa do cenário. Já que se referenciou esta classe, como sendo a classe principal do projecto, terá de realizar todas as tarefas, de modo a que seja possível jogar o cenário.

É nesta função que se realizam as seguintes operações:

- Definir o fluxo dos ecrãs que o jogador vai vendo durante o jogo;
- Carregar os ficheiros com todos os elementos do cenário, textos, possíveis configurações e resultados;
- Definir variáveis de jogo (lista de: documentos, pastas, zonas, registos, pontuações, respostas, textos, organizações possíveis, resultados, *track* das posições do rato, variáveis de fim de jogo, etc.);
- Gerir o espaço disponível para os elementos no cenário;
- Carregar todos os ecrãs já referidos (*loading*, ecrã de nome, objectivos, pontuações, cenários de jogo, etc.);
- Guardar e actualizar os objectos de todas as outras classes;
- Adicionar e remover objectos (incluindo ecrãs) ao *stage*;
- Desenhar o diagrama de *spaghetti*;
- Criar registo para o relatório;
- Consultar o relatório de cada etapa do cenário;
- Definir as classes que podem ser guardadas no *SharedObject*;
- Adicionar eventos gerais para o teclado e rato ao *stage*;
- Entre outros.

Até certo ponto, esta classe vai adicionando e removendo os ecrãs e objectos do *stage*, à medida que o jogador os vai visualizando, com a excepção de que, durante o *loading* do cenário, são carregados todos os ficheiros, que serão necessários ao jogo.

As restantes tarefas são realizadas, sequencialmente, durante a interacção do jogador com o cenário. Uma vez que, por serem operações simples, não ocupam muitos recursos e desde que o sistema onde esteja a ser executado o jogo não se encontre sobrecarregado, nunca levam muito tempo a serem executadas.

Após a visualização do ecrã dos objectivos o jogador terá de seleccionar opção “continuar”.

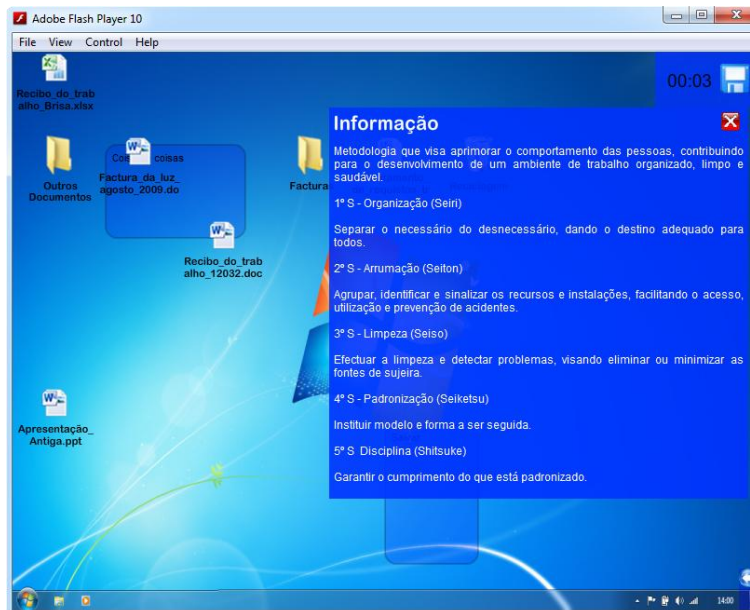


Ilustração 33 – Ecrã de área do jogo

Na Ilustração 33, pode visualizar-se a área de jogo. É constituída por quatro áreas principais. A área principal é o fundo onde são colocados os documentos e pastas de forma aleatória. Subtraindo no espaço disponível e por ordem descendente na posição no ecrã, tem-se: a área do cronómetro/botão de salvar o jogo; a área de notificações (pode ser fechada) e a área de entrega de documentos (pode ser maximizada e minimizada).

Classe associada: **Fundo.as**

A classe Fundo, como o nome indica, é responsável por adicionar o “fundo do ecrã”, mais conhecido como “papel de parede” ou *wallpaper*, ao *stage* do jogo. Uma vez que, para este fim, não era necessário criar uma classe específica, esta tem uma importância acrescida em relação às restantes, por realizar as seguintes operações:

- Escutar eventos de teclado como a tecla M (abrir submenu);
- Fazer o *tracking* da posição do cursor do rato;
- Desenhar o gráfico das posições resultantes do processo de *tracking*;
- Registar eventos, como quais as opções que foram seleccionadas no submenu;
- Guardar variáveis de controlo sobre o jogo.

A classe Fundo, em conjunto com as classes Main e Jogo é das classes mais importantes. Todos os elementos são adicionados ao objecto Fundo, que a sua classe adiciona ao *stage*.

### 5.2.3. Explicação das áreas do cenário – Cronómetro e botão salvar

Inicia-se a explicação das áreas do cenário.

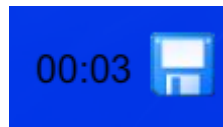


Ilustração 34 – Área do cronómetro e botão salvar o jogo

Na Ilustração 34, pode visualizar-se a área que possui o cronómetro e botão para salvar o jogo.

O cronómetro consiste num fundo azul com uma *textbox* (caixa de texto) dinâmica. Esta *textbox* servirá para mostrar ao jogador, numa primeira fase do jogo, quanto tempo demorou até encontrar o documento ou, numa segunda instância do jogo, o tempo restante até acabar a ronda.

Tem de haver um processo que se encarregue de cronometrar o tempo e actualizar o texto da *textbox*. Este facto é válido para qualquer objecto que esteja no cenário, independentemente da linguagem em que se esteja a escrever essa classe. O *ActionScript3* não é excepção à regra.

Com este conceito em mente, pode avançar-se para a análise da função que controla o cronómetro.

```
function timerUpdate(TimerEvent):void
{
    t_msecs++;
    if (t_msecs == 10)
    {
        t_msecs = 0;
        t_secs++;
    }
    if (t_secs == 60)
    {
        t_secs = 0;
        t_mins++;
    }
    if (t_mins == 60)
    {
        t_mins = 0;
        t_hrs++;
    }
    updateTime();

    //Verificar condições de final de jogo
    (...)
}
//Função que mantém actualizada a TextBox
function updateTime():void
{
    texto.text = "";
    texto.appendText((t_mins > 9) ? t_mins.toString() : ("0" + t_mins.toString()));
    texto.appendText(":");
    texto.appendText((t_secs > 9) ? t_secs.toString() : ("0" + t_secs.toString()));
}
```

Código 6 – Código para cronometração do tempo

Para a contagem crescente do tempo despendido, no Código 6, é invocada uma função com o nome *TimerUpdate*. Esta chamada está associada a um evento temporal e é feita a cada segundo que passa. Deste modo, pode incrementar-se as variáveis dos segundos, minutos ou horas. Este processo, por si só, não actualiza a *textbox*, onde se pode consultar o tempo decorrido. Sendo assim, é necessário chamar, a cada segundo que passa, uma função (*updateTimer*) responsável para esse efeito.

Apesar da diferença da operação inversa, uma contagem decrescente, o processo é semelhante ao já definido, com a excepção de que a função subtrai ao valor inicial e, assim que chega ao fim, dá-se por concluída a ronda, executando funções próprias para esse propósito.

Classe associada: *Cronometro.as*

Esta classe, como o nome indica, pretende cronometrar o tempo que o jogador demora a procurar o documento pedido na primeira etapa do jogo e a fazer uma contagem decrescente do tempo disponível, para voltar a encontrar um documento aleatório, já com o cenário organizado à sua maneira.

Esta classe tem ainda a função de, a cada segundo que passa, verificar se as condições de fim do jogo já foram alcançadas. Na primeira etapa, só existe a condição do jogador encontrar o documento. Na segunda etapa, o jogador deve organizar o cenário. Não existindo condição de fim de jogo, esta acaba quando o jogador seleccionar salvar, significando que está pronto para a próxima etapa. Concluindo, as condições de fim do jogo para a terceira e última etapa são: o jogador encontrou o documento pedido ou esgotou-se o tempo que tinha disponível para procurar o documento em questão.

```
//Verificar condições de final de jogo
public function verificaFinalJogo():Boolean
{
    if (encontrouDoc || acabouTempo)
    {
        acabouJogo = true;
        criarRegisto("Acabou a etapa do jogo.");
        return true;
    }
    return false;
}
```

Código 7 – Verificar as condições finais do jogo

No Código 7, pode ver-se com detalhe esta função. Uma vez que lhe serão feitas muitas chamadas (uma por segundo), é exigido uma certa simplicidade, para que não haja uma sobrecarga no jogo e, por consequência, se note alguma lentidão.

Passando para a explicação do botão de salvar o jogo, este processo é bastante simples e consiste em abrir o *SharedObject* em utilização e reescrevê-lo, de modo a que actualize as suas variáveis.

```
//Colocar o botão de salvar o jogo no fundo
guardar = new botaoSave();
guardar.x = 760;
guardar.y = 13;
guardar.addEventListener(botaoSave.SALVAR, salvar, false, 0, true);
fundoJogo.addChild(guardar);

//Função que responde ao evento
public function salvar(e:Event):void
{
    //Chama a função de salvar o jogo
    novoJogo.salvarJogo();
    if (novoJogo.etapaActual == 2)
    {
        //Abrir os resultados antes da última etapa
        abrirFimNivel2();
    }

    //Adicionar ao Registo do jogo
    //Gravar registo de que salvou o jogo
    criarRegisto("O jogador salvou o jogo.");
}

//Função para salvar o jogo (Classe Jogo)
public function salvarJogo():void
{
    var so:SharedObject = SharedObject.getLocal("cenario4_" +
this.nomeJogador);
    so.data.xml = Main.xml;
    so.data.zonas = Main.zonas;
    Main.pastas.sortOn("naPasta", Array.NUMERIC);
    so.data.pastas = Main.pastas;
    so.data.documentos = Main.documentos;
    so.data.reciclagens = Main.reciclagens;
    so.data.trackArray = Main.trackArray;
    so.data.duploClickArray = Main.duploClickArray;
    so.data.logArray = Main.logArray;
    so.data.tempo = this.tempoRest;
    so.data.tempoDemorado = this.tempoDemorado;

    so.flush();
    so.close();
}
```

**Código 8 – Código para salvar o jogo**

No excerto de Código 8, pode consultar-se o evento e as funções que este evoca, para que seja possível guardar toda a informação necessária, para o jogador poder retomar o jogo noutra ocasião.

O processo começou por arranjar um ícone que representasse o processo de gravação. Uma vez que os jogadores associam o ícone de uma disquete a este processo de armazenamento de dados, este parece ser ideal para o pretendido.

Criado o *Movieclip* com a imagem da disquete, é associado a este um *EventListener* (executa uma função assim que um determinado evento designado ocorrer). Esta função, por sua vez, chama outra, reservada para guardar os dados, abre ecrãs de resultados, caso seja necessário, e cria o registo de que o jogador salvou o jogo.

São guardadas informações, como o nome do jogador, o tempo que já demorou a procurar o documento, qual o documento que deve encontrar, *arrays* com os documentos, pastas, zonas, registos, entre outros dados.

Classe associada: *botaoSave.as\**

Esta classe é responsável pelo botão de salvar. É nesta que se define o comportamento do botão, como os eventos responsáveis por mudar o estilo do cursor, quando este entra na sua área e é esta, que lança o evento para o Main pedir à classe Jogo para salvar os dados do jogador.

#### 5.2.4. Explicação das áreas do cenário – Área de entrega do documento

A terceira área do ecrã está reservada a outro elemento, que já foi referido neste capítulo: a área de entrega de documentos.

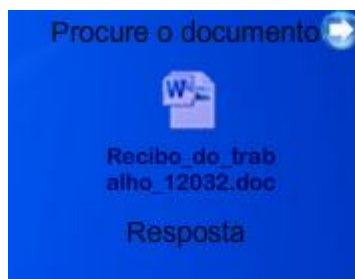


Ilustração 35 – Área de entrega de um documento

Na Ilustração 35, está visível a área para entrega do documento que é pedido ao jogador que encontre. Uma vez que já foi dada a explicação para esta área, faz-se um breve resumo. O jogador deve arrastar e largar o documento que é pedido. Caso acerte, a palavra “Resposta” será substituída por “Resposta certa” e a ronda terminará; caso contrário, será substituída por “Resposta errada, tente novamente”. Para qualquer das opções, é atribuída uma pontuação consoante a sua resposta.

Classe associada: *procurarDocumento.as\**

Esta classe é responsável pelas funções que controlam a caixa de entrega do documento. Possui funções para as animações, para mostrar e esconder a área de entrega, funções para adicionar o documento escolhido à caixa, entre outras. As suas variáveis ajudam a saber qual o nome e número de identificação do documento escolhido.

### 5.2.5. Explicação das áreas do cenário – Janela de informações

A janela de informações surge sempre na mesma área do ecrã, de modo a não confundir o jogador, e pode ser fechada através do botão vermelho com uma cruz branca, situada na parte superior direita da janela.

Esta janela pretende fornecer ao jogador informações relevantes sobre os conteúdos que este está a pôr em prática.

Classe associada: *caixaInfo.as\**

O objectivo da classe caixaInfo baseia-se em preencher o ecrã informativo que o jogador visualiza, antes de começar a jogar. Apesar de ser possível, e mais rápido, criar um *Movieclip* estático com o texto que se pretende mostrar a cada etapa do jogo, este, ao ser dinâmico, facilita futuramente a tarefa de evoluir para um sistema de notificações mais avançado.

De momento, a forma de preencher a sua caixa de texto é através da leitura de um dos campos de um documento XML.

### 5.2.6. Explicação das áreas do cenário - Submenu

Existem algumas operações que podem ser realizadas, mas exigem que o jogador conheça o ambiente onde está a jogar. Dado que se está a simular um ambiente de computador *Windows* no ambiente de trabalho (ou em qualquer outra parte), pode aceder-se a um submenu, que varia com a zona do *Windows* onde nos encontramos. Este menu é acedido com o recurso ao botão direito do rato.

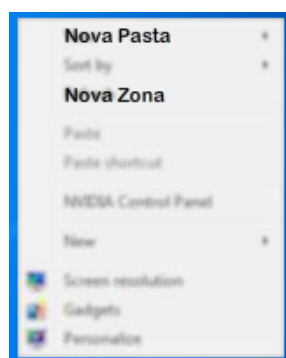


Ilustração 36 – Submenu aberto com o botão direito do rato

Na Ilustração 36, pode visualizar-se o submenu aberto com o botão direito do rato. Sempre que o jogador pretender abrir o submenu, terá de pressionar a tecla M (Menu) do teclado do seu computador ou pressionar o botão direito do rato.

Antes de se avançar para uma explicação mais detalhada de como funciona este submenu, existem dois aspectos muito importantes, que se devem referir.

Uma vez que se trata de um protótipo, as opções dispostas no submenu são sempre as mesmas, independentemente da área onde se decide abrir o submenu. O outro aspecto, bem mais importante que o primeiro, é o facto de o evento associado ao botão direito do rato não ser permitido em aplicações *Flash*.

O motivo pelo qual a *Adobe* não permite que o *Flash* receba eventos deste tipo é, ainda hoje, desconhecido. Existem disponíveis na *internet* várias explicações, contudo, nenhuma é oficial e passam por especulações. O que se pode afirmar com certeza é que este é um factor limitador do crescimento desta grande ferramenta para criação de animações, jogos e *websites*.

Apesar desta última limitação, é possível chamar o submenu, através do botão direito do rato. Uma vez que se usou um processo que envolve a utilização de métodos de outras tecnologias, como o *javascript*, a variabilidade do funcionamento deste depende do *browser*, sistema operativo e da versão de *Flash* onde corre a aplicação.

O jogador pode, então, carregar no botão direito do rato e, em vez de surgir o menu típico de uma aplicação *Flash*, surgirá o submenu que se criou.

Resumindo, é possível chamar o submenu de duas formas: através da tecla M do teclado e, se for suportado pelo computador onde corre o jogo, através do botão direito do rato.

```
this.stage.addEventListener(KeyboardEvent.KEY_DOWN, displayKey, false,
0, true);

function displayKey(keyEvent:KeyboardEvent)
{
    switch (keyEvent.keyCode)
    {
        (...)
        case Keyboard.M:
            if (keyEvent.keyCode)
            {
                menu = true;
                var menuAux:menuBotaoDireito = new menuBotaoDireito();
                menuAux.x = mouseX;
                menuAux.y = mouseY;
                menuAux.addEventListener(MouseEvent.ROLL_OUT, outClip2,
false, 0, true);
                menuAux.name = "menuAux";
                if (Main.fundoJogo != null)
                {
                    Main.fundoJogo.addChild(menuAux);
                }

                //Opções do submenu
                (...)
            }
            break;
            (...)
    }
}
```

Código 9 – Código necessário para abrir o submenu

No Código 9, podem consultar-se as linhas de código responsáveis por abrir e preencher o submenu. Começa por adicionar-se um evento ao *stage* que escuta quais as teclas que são pressionadas.

A função que é chamada verifica se o código da tecla que recebeu faz parte de uma ou várias teclas, cuja acção tenha sido previamente definida. Se pertencer a esse grupo, são desencadeadas as acções programadas. Caso contrário, é ignorado esse evento e continua a aguardar por um que corresponda ao que se delineou.

Assim que o jogador pressionar a tecla M (ou botão direito do rato), a função, que verifica se teclas foram pressionadas, cria e adiciona ao *stage* o *Movieclip*, que contém o submenu. Este submenu é, depois, preenchido com as opções. Existem duas opções capazes de ser executadas no submenu: criar Nova Pasta e criar uma Nova Zona. A cada uma destas opções é adicionado um evento, que chama a mesma função e que realiza as acções correspondentes.

Mais a frente, nesta dissertação, explica-se o que é uma Zona e como funciona.

```
private function clickOpcao (e:MouseEvent):void
{
    if (e.currentTarget.name == "Zona_txt")
    {
        //Adicionar nova Zona
        var zona:Zona = new Zona;
        zona.desenharZona ();
        criarRegisto("Foi criada uma nova Zona.");
        fecharMenu ();
    }
    if (e.currentTarget.name == "Pasta_txt")
    {
        //Adicionar nova Pasta
        var x:int = stage.mouseX - 50;
        var y:int = stage.mouseY - 20;
        var novaPasta:Pasta = new Pasta("Nova Pasta",
"imagens/icons/pasta.png");
        novaPasta.criarNovaPasta(this, x , y);
        criarRegisto("Foi criada uma nova Pasta.");
        fecharMenu ();
    }
}
//Funções para modificar o cursor do rato
(...)
function fecharMenu():void
{
    var menu1:menuBotaoDireito = this.getChildByName("menuAux") as
menuBotaoDireito;
    if (menu1 != null)
    {
        this.removeChild(menu1);
        menu = false;
    }
}
```

Código 10 – Funções chamadas pelo código referente ao submenu

No Código 10, pode consultar-se a função que responde, quando se selecciona as opções do submenu, a função que o fecha, quando o cursor sai da sua área ou quando uma opção é seleccionada, e as funções que mudam o cursor dependente do objecto situado por baixo.

A função, que responde à selecção de opções, verifica se a pressionada foi para criar Nova Pasta ou para criar Nova Zona, chamando as funções presentes na classe Pasta e Zona, respectivamente.

Classe associada: *Fundo.as*

### 5.2.7. Explicação dos objectos do cenário - Zonas

Explicam-se, de seguida, quais os restantes elementos com que o jogador interage mais: Pastas, Documentos, Zonas e Atalhos.

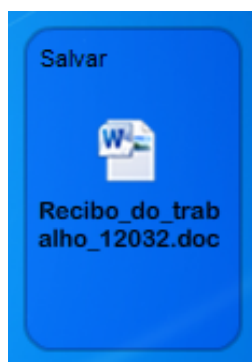


Ilustração 37 – Exemplo de uma Zona chamada Salvar

Na Ilustração 37, pode visualizar-se uma Zona. Uma Zona é uma área do ecrã criada pelo jogador, com o intuito de marcar uma parte do cenário. Depois de criar a Zona, o jogador pode colocar, sobre esta, vários documentos/pastas/atalhos relacionados entre si. A Zona possui um nome que permite identificar melhor o elo que liga os documentos/pastas entre eles.

De um ponto de vista mais técnico, o documento, ou pasta, não fica associado a uma determinada Zona. Apesar de ser fácil a implementação desta ideia, a Zona pretende servir como uma referência visual e como um auxílio à organização.

Para criar uma Nova Zona, o jogador terá de abrir o submenu, pressionando a tecla M (ou botão direito do rato) e escolhendo a opção “Nova Zona”. Seguidamente, deve deslocar o cursor para o local onde a nova Zona vai ficar. Com o botão esquerdo premido, deve mover o cursor para o sítio onde ficará o vértice inferior direito/esquerdo, consoante moveu o rato para a direita ou para a esquerda.

Assim que concluir a estrutura da Zona, resta atribuir-lhe um nome.



Ilustração 38 – Processo de criação de Zona (Criar Forma)

Na Ilustração 38 (lado esquerdo), pode visualizar-se o jogador a criar uma Zona nova. Nesta fase, o jogador está a delinear a área que a Zona cobrirá. Recapitulando, o jogador deverá:

1. Abrir o menu, pressionando a tecla M ou botão direito do rato;
2. Escolher a opção “Nova Zona”;
3. Delinear a área que a Zona cobrirá, clicando no canto superior direito e, com o botão pressionado, deslocar o cursor até ao local do ecrã onde, na diagonal, ficará o vértice oposto.

Segue-se a fase de atribuição de um nome à Zona, Ilustração 38 (lado direito). O jogador deve seleccionar a caixa de texto que, de momento, expõe uma nota que o ajuda a melhor perceber este processo.

O nome da Zona deve ajudar o jogador a perceber a ligação entre os documentos que depois colocará naquela área do ecrã.

Por exemplo, se chamar “Facturas” à Zona, o jogador sabe que, naquela área já organizada, apenas se encontram facturas e que, se procurar recibos, não os encontrará aí.

Após seleccionar a caixa de texto, a nota de ajuda desaparecerá e o jogador poderá, então, escrever o nome que pretende atribuir àquela Zona. Para este exemplo, o nome é “Zona de Pastas”. Para concluir este processo, o jogador deverá pressionar a tecla *enter*.

É possível, ainda, remover Zonas já existentes ou criadas pelo jogador. Para isso, este deverá arrastá-las para a reciclagem, como faria normalmente no ambiente de trabalho do seu computador.

Classe associada: ***Zona.as***

Apesar do elemento associado a esta classe ser simples e ter poucas variáveis que o definam: posição X e Y no ecrã, largura, altura, nome e estado, esta classe torna-se mais extensa devido a varias funções necessárias para o jogador a desenhar e atribuir-lhe um nome.

O elemento associado a esta classe é um *Movieclip* constituído por um quadrado arredondado nas pontas, com preenchimento azul-escuro, com alguma transparência e uma caixa de texto no canto superior esquerdo deste.

O processo de criação de uma nova Zona já foi explicado. Abordam-se, agora, as funções que o tornam possível. Associado ao *click* do botão esquerdo está uma função que regista a posição inicial. Há, ainda, uma função que é chamada sempre que há uma movimentação do rato com o botão pressionado. É responsável por desenhar temporariamente, usando a posição actual do rato como ponto final. Esta função ajuda o jogador a ter uma percepção do tamanho final da zona.

Assim que o jogador largar o botão esquerdo do rato, é chamada uma função que desenha definitivamente a Zona, atribuindo-lhe a cor e transparência final. A função associada, ao largar do botão que desenha a Zona, tem ainda o código, que adiciona uma caixa de texto (com as linhas pintadas a preto nas extremidades) com as directrizes que indicam ao jogador como deve proceder.

Uma outra função é chamada, assim que o jogador seleccionar a caixa o texto. O texto que estava na caixa desaparece, de modo a que o jogador introduza o nome da Zona. Assim que este pressionar a tecla *Enter*, uma outra função atribui o nome escolhido à Zona. Para referenciar a conclusão do processo, são alteradas as propriedades da caixa de texto (tirando as linhas das extremidades e mudando o tipo de caixa para estático), de modo a que só dê para visualizar o nome escolhido.

Existem ainda funções para desenhar as Zonas que estiverem definidas previamente no ficheiro XML de objectos.

Classe associada: ***ZonaDef.as***

Esta classe é responsável por converter as informações sobre as Zonas em dados passíveis de serem armazenados em *SharedObject*.

Uma explicação detalhada, desta classe e das razões que levaram à sua criação, pode ser consultada de seguida na classe *DocumentoDef*.

## Criar uma Zona

Explica-se, agora, o processo de criação de uma nova Zona.

```
//Adicionar nova Zona
var zona:Zona = new Zona;
zona.desenharZona ();

public function desenharZona ():void
{
    adicionarZona = true;
    numZonas++;
    Main.fundoJogo.addEventListener(MouseEvent.CLICK, mdown,
false, 0, true);
}

private function mdown(e:MouseEvent):void
{
    startPoint = new Point(e.stageX, e.stageY);
    drawLayer = new MovieClip();
    drawLayer.x = e.stageX;
    drawLayer.y = e.stageY;
    Main.fundoJogo.addChild(drawLayer);
    Main.fundoJogo.addEventListener(MouseEvent.CLICK, mmove,
false, 0, true);
    Main.fundoJogo.addEventListener(MouseEvent.CLICK, mup, false,
0, true);
}

private function mmove(e:MouseEvent):void
{
    this.name = "NovaZona_" + numZonas;
    this.estado = true;

    //desenhar quadrado
    drawLayer.graphics.clear();
    drawLayer.graphics.lineStyle(2, 0x000000);
    drawLayer.alpha = 0.40;
    var newx:int = e.stageX;
    var newy:int = e.stageY;
    this.w = Math.abs(newx - startPoint.x);
    this.h = Math.abs(newy - startPoint.y);
    drawLayer.graphics.drawRoundRect(0, 0, w, h, 25, 25);
    drawLayer.width = w + 3;
    drawLayer.height = h + 3;
}
}
```

Código 11 – Criar uma nova zona 1ª Parte

No Código 11 e Código 12, encontram-se as funções associadas aos seguintes eventos do rato:

- Botão esquerdo (pressionar, manter e largar);
- Movimento do cursor do rato.

```

private function mup(e:MouseEvent):void
{
    this.posX = startPoint.x;
    this.posY = startPoint.y;
    Main.fundoJogo.removeEventListener(MouseEvent.MOUSE_DOWN,
mousedown);
    Main.fundoJogo.removeEventListener(MouseEvent.MOUSE_MOVE,
mmove);
    Main.fundoJogo.removeEventListener(MouseEvent.MOUSE_UP, mup);

    var quadrado:Sprite = new Sprite();
    quadrado.graphics.beginFill(0x0000FF);
    quadrado.graphics.lineStyle(2, 0x000000);
    quadrado.graphics.drawRoundRect(0, 0, w, h, 25, 25);
    quadrado.graphics.endFill();
    quadrado.alpha = 0.20;
    drawLayer.graphics.clear();
    this.mouseChildren = true;
    this.addChild(quadrado);

    //Criar textbox
    texto = new TextField();
    //Definir propriedades da textbox
    (...)
    texto.addEventListener(KeyboardEvent.KEY_DOWN, handler, false,
0, true);

    //Evento para apagar o que foi escrito e remover eventos do
stage
    (...)
    this.addChild(texto);
}

private function handler(e:KeyboardEvent):void
{
    // if the key is ENTER
    if (e.charCode == 13)
    {
        if (Main.fundoJogo != null)
        {
            var zona:Zona =
Main.fundoJogo.getChildByName("NovaZona_" + numZonas) as Zona;
            if (zona != null)
            {
                //Código para gravar o nome
                (...)
                //Gravar toda a informação na var. geral de
zonas
                (...)
            }
        }
        adicionarZona = false;
    }
}

```

Código 12 – Criar uma nova zona 2ª Parte

A função associada ao botão esquerdo é responsável por marcar a posição inicial da Zona e atribuir os restantes eventos, tornando possível a criação dinâmica de Zonas, anteriormente explicada.

Se o jogador mover o rato, com o botão pressionado, vai ser evocada uma função, que desenha temporariamente a Zona, tendo, no ponto final desta, a posição actual do cursor.

Assim que o jogador largar o botão esquerdo, a função correspondente a este evento marca a posição final da Zona, remove os eventos anteriores desnecessários, adiciona a *textbox*, onde o jogador preencherá o nome da Zona, e remove os eventos do *stage*, para que seja possível o preenchimento desta.

Uma vez que, a *textbox* tem associado um evento que recebe todos os caracteres pressionados, assim que o jogador pressionar a tecla *enter*, sinalizando que acabou de introduzir o nome, uma função grava o nome numa variável temporária. Seguidamente, esta grava a Zona já com o seu nome e as suas propriedades, na variável global de Zonas do jogo.

Por último e logo após a função receber a tecla *enter*, são repostos os eventos que estavam associados ao *stage* do jogo.

#### 5.2.8. Explicação dos objectos do cenário – Documentos

Continua-se a explicação dos objectos presentes no cenário, focando, agora, a atenção nos documentos, uma vez que estes são os elementos mais simples dos cenários. O motivo pelo qual são os objectos mais simples deve-se ao facto de que, ao contrário das Pastas e da Reciclagem, não permitem que lhes sejam adicionados outros elementos.

Seguem-se, na Ilustração 39, dois tipos de documentos que se podem encontrar no cenário.

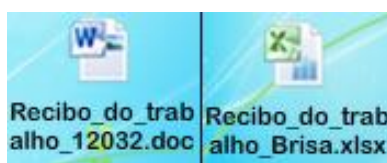


Ilustração 39 – Exemplos de documentos que se podem adicionar ao cenário

Desde que haja um *icon* em ponto pequeno e outro em grande, é fácil adicionar outro tipo de documentos ao cenário. Mais à frente e após concluir-se a explicação dos outros elementos passíveis de serem encontrados neste cenário, como as Pastas, os Atalhos e a Reciclagem, será explicado como é que se adiciona mais tipos de documentos ao cenário e como é que estes são adicionados depois à área disponível.

À semelhança do sistema *Windows* que se pretende simular, os documentos, bem como as pastas e as reciclagens, são construídos da mesma forma.

Estes elementos são constituídos por uma imagem (*icon*) que representa o tipo ou género de elemento, seguido de um nome com a extensão deste elemento.

Do ponto de vista de programação, estes elementos são constituídos por um *movieclip*, que contém outro *movieclip* vazio, onde se carrega a imagem, através de um *loader*, e uma *textbox* dinâmica que é preenchida com o nome do elemento que se está a criar.

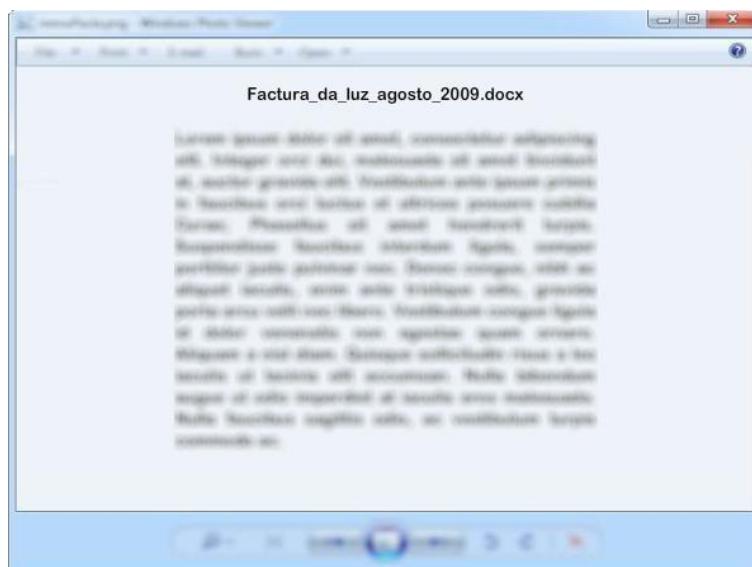


Ilustração 40 – Visualização do conteúdo de um documento

É permitido ao jogador abrir o documento, de modo a que possa verificar se o nome atribuído ao ficheiro é o mais correcto (Ilustração 40).

Classe associada: ***Documento.as***

Esta classe pertence ao grupo das mais importantes do cenário, juntamente com o Main, Fundo, Jogo, Pasta, Reciclagem e Icon.

Como o nome indica, é responsável pelos documentos presentes no cenário. Possui funções para as seguintes actividades:

- Criar o elemento documento;
- Atribuir os eventos como: *click*, *duplo-click*, *drag&drop*, entre outros;
- Definir as acções para os respectivos eventos;
- Alterar nomes e outras propriedades dos documentos;
- Verificar se é possível adicionar o documento a outros elementos;
- Abrir e visualizar o conteúdo do documento;
- Entre outras.

Classe associada: *DocumentoDef.as*

Esta classe e todas as classes que possuem “Def” no nome servem para criar objectos passíveis de serem guardados dentro de *SharedObjects*.

Uma vez que os *SharedObjects* são guardados no formato AMF (*Action Message Format*), quando existe a necessidade de guardar um objecto de uma classe específica, este guarda como um objecto normal, sem guardar informações sobre a classe que o criou. Para ser possível gravar informações sobre a classe, é necessário “serializar” a classe.

Serialização é o processo de conversão, num formato passível de ser armazenado em ficheiros ou *buffers* de memória, de um objecto ou de uma estrutura de dados. Este processo permite que estes dados sejam depois recriados na mesma ou noutras plataformas.

Por questões de simplicidade, decidiu-se criar classes diferentes para o objecto *Movieclip* e para o objecto com as mesmas informações para ser guardado. Assim, sempre que se leia do *SharedObject*, é possível recriar o *Movieclip* com os dados do documento em questão.

Desta forma, evita-se ainda a serialização de uma classe, que está directamente associada a um *Movieclip*, que é fortemente desaconselhado em *ActionScript3*.

Sendo assim, esta classe converte os dados do objecto Documento, para que seja possível o seu armazenamento no *SharedObject*.

#### Mudar nome a um elemento

O processo de mudança de nome a um elemento é semelhante, quer este seja Documento, Atalho, Pasta ou outro.

Neste processo é adicionado, a cada elemento um evento de *click*.

A função que responde a este evento chama-se *mudarNome*. Contudo, a sua chamada só é realizada, se for pressionada a tecla E (há um evento associado ao *stage* para esse fim), assinalando que o cenário está em modo de edição de nomes; caso contrário, ocorre um seleccionar de objecto normal.

Caso a função *mudarNome* seja chamada, é criada uma *Textfield* adicional, para que torne possível a edição do seu conteúdo. À *Textfield* é ainda adicionado um evento para preparar a introdução de texto.

O Código 13 pretende mostrar a preparação do *stage* para a introdução de texto. Uma vez que este tem associado um evento, que espera que certas teclas sejam pressionadas, é necessário removê-lo, para evitar que seja accionado sem intenção.

Seguidamente, o texto que se encontrava na *Textfield* é seleccionado, para que o jogador o possa apagar de uma vez, usando, a tecla *backspace* do teclado.

```

private function mouseClickedTextto (e:MouseEvent):void
{
    stage.removeEventListener(KeyboardEvent.KEY_DOWN,
Main.fundoJogo.displayKey);
    if (docOrigem)
    {
        var texto:TextField = docOrigem.getChildByName("novoNome")
as TextField;
        if (texto != null)
        {
            texto.removeEventListener(MouseEvent.CLICK,
mouseClickedTextto);
            texto.alwaysShowSelection = true;
            texto.selectable = true;
            texto.setSelection(0, texto.text.length);
        }
    }
}

```

Código 13 – Mudar o nome a um elemento (documento) 2ª Parte

Na terceira parte deste processo, será explicada a função responsável por receber o texto, guardá-lo e voltar a adicionar os eventos que foram removidos durante este processo.

```

private function handler (e:KeyboardEvent):void
{
    // if the key is ENTER
    if (e.charCode == 13)
    {
        if (Main.fundoJogo != null)
        {
            var texto:TextField =
docOrigem.getChildByName("novoNome") as TextField;
            if (texto)
            {
                novoNome = texto.text;
                docOrigem.nome.text = novoNome;
                //Remover a TextField Auxiliar
                docOrigem.removeChild(texto);
                //Código para gravar o novo nome na variável
geral dos documentos
                (...)
                //Voltar a adicionar eventos do stage/objectos
                Main.fundoJogo.editarNomes = false;
            }
        }
    }
}

```

Código 14 – Mudar o nome a um elemento (documento) 3ª Parte

A função do Código 14 é responsável por escutar todas as teclas que o jogador pressionar. Assim que este pressiona a tecla *enter*, sinalizando que acabou de escrever o nome do elemento, esta:

- Lê o que foi escrito na *Textfield*;
- Atribui o texto a uma variável temporária;
- Remove a *Textfield*;
- Guarda o novo nome do elemento;
- Atribui os eventos ao elemento e ao *stage* que foram removidos durante o processo.

### 5.2.9. Explicação dos objectos do cenário – Atalho

Continuando a explicação dos elementos do cenário, será agora abordado o objecto Atalho, elemento que, a nível técnico, mais se assemelha a um Documento.



Ilustração 41 – Exemplo de um Atalho

O Atalho, Ilustração 41, é outro elemento que se pode encontrar no ambiente de trabalho de um computador. Este, como o nome indica, é um elemento que apenas contém o caminho (ou localização) de outro ficheiro no computador.

Podem existir atalhos para os mais variados elementos, como: músicas, imagens, ou aplicações.

Este elemento apenas se encontra no cenário, para lhe acrescentar realismo. A interacção com este está limitada à alteração do seu nome, à sua posição no cenário e à sua posição dentro de outras pastas.

Este elemento pode, ainda, ser eliminado, caso o jogador entenda que não é necessário.

Classe associada: ***Icon.as***

Como o próprio nome indica, esta classe é responsável por adicionar e controlar o elemento Icon.

Esta é semelhante à classe dos Documentos, tendo como diferenças os seguintes tópicos:

- Não permite ver o conteúdo de um Atalho, uma vez que este não possui essa característica;
- Não é possível submeter um Atalho na caixa de entrega de Documentos.

Classe associada: *IconDef.as*

Esta classe é responsável por converter as informações do elemento Atalho, em dados passíveis de serem armazenados em *SharedObject*.

Remete-se a explicação desta para a classe DocumentoDef, onde já foi fornecida uma explicação detalhada deste processo e das razões que levaram à criação deste tipo de classes.

### 5.2.10. Explicação dos objectos do cenário – Pastas

Passando para as pastas, estas têm como único objectivo armazenar outras pastas, atalhos e documentos, permitindo, assim, uma maior organização dos ficheiros.

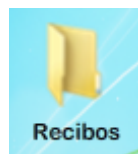


Ilustração 42 – Exemplos de pastas que se podem adicionar ao cenário

Na Ilustração 42, pode visualizar-se o aspecto gráfico das pastas.

Do ponto de vista gráfico, são idênticos aos Documentos, os elementos gráficos que o constituem são os mesmos que os primeiros referidos.

Sobre o aspecto da programação, estes elementos são muito semelhantes e possuem processos parecidos, como mudar o nome da pasta, adicionar ou remover outros elementos (desde que sejam outras pastas, ou a reciclagem). Quando se abre uma pasta, visualizam-se os elementos que esta contém.

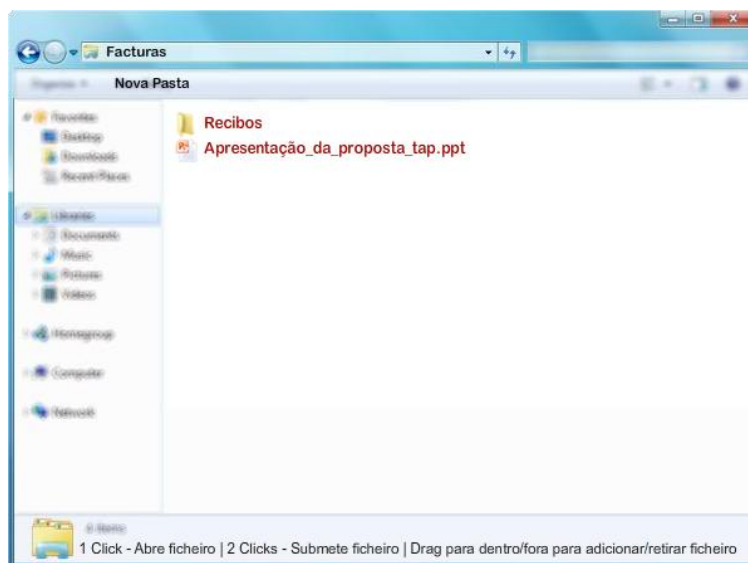


Ilustração 43 – Visualização do conteúdo de uma pasta

Na Ilustração 43, encontra-se a pasta aberta para consulta do seu conteúdo. O jogador pode adicionar documentos, atalhos ou pastas, arrastando estes de fora (raiz do cenário) para dentro da janela desta. Este processo é ainda válido para a raiz do cenário: se o jogador arrastar um documento/atalho/pasta para cima de outra pasta, este é adicionado. O mesmo se passa com a Reciclagem, para apagar documentos/atalhos/pastas.

O processo de remoção dos elementos da pasta é semelhante ao anterior. Deve seleccionar-se o elemento e, com o botão esquerdo do rato premido, arrastar para fora da janela, largando-o, para remover.

Caso pressione uma vez, o elemento é aberto, para poder visualizar o seu conteúdo. Para submeter o documento pedido, se este se encontrar dentro da pasta, deve pressionar duas vezes. Mais à frente serão fornecidas explicações técnicas destes processos.

Para concluir, existem duas formas de criar uma nova pasta no cenário, dependendo do contexto em que o jogador se encontra. Este pode fazê-lo através do submenu, caso queira criar na raiz do cenário. Em alternativa, pode seleccionar a opção “Nova Pasta”, posicionada abaixo do nome na janela, caso pretenda criar uma nova pasta dentro de outra cujo conteúdo esteja a visualizar.

Classe associada: *Pasta.as*

Esta classe é bastante importante para o cenário, dado fazer referência ao elemento Pasta. Esta tem as mesmas características que um Documento, contudo permite mais funcionalidades. Uma pasta é um elemento que serve de contentor, armazena outros elementos, incluindo Documentos, Atalhos e outras pastas.

Como será espectável, muitas das suas funções para as seguintes actividades são comuns aos Documentos. A lista seguinte pretende discriminar algumas funções e os seus objectivos:

- Criar o elemento Pasta;
- Atribuir os eventos como: *click*, *duplo-click*, *drag&drop*, entre outros;
- Definir as acções para os respectivos eventos;
- Alterar nomes e outras propriedades das pastas;
- Verificar se é possível adicionar a pasta a outros elementos;
- Associar um documento ou pasta à pasta destino;
- Abrir e visualizar o conteúdo da Pasta e seus elementos;
- Actualização de conteúdos;
- Abrir conteúdos dos elementos de uma Pasta;
- Criar pastas dentro de pastas;
- Entre outras.

Classe associada: *PastaDef.as*

Esta classe é responsável por converter as informações do elemento Pasta em dados passíveis de serem armazenados em *SharedObject*.

Remete-se a explicação desta para a classe DocumentoDef, onde já foi fornecida uma explicação detalhada deste processo e das razões que levaram à criação de classes deste tipo.

Criar uma pasta

Como já foi explicado, o jogador pode adicionar Novas Pastas. Se estiver posicionado na raiz do cenário, pode fazê-lo através do submenu, seleccionando “Nova Pasta”. Caso esteja dentro de uma pasta, pode seleccionar a opção “Nova Pasta”, colocada abaixo do nome.

```
//Classe Fundo
//Adicionar nova pasta (pelo menu)
var x:int = stage.mouseX - 50;
var y:int = stage.mouseY - 20;
var novaPasta:Pasta = new Pasta("Nova Pasta",
"imagens/icons/pasta.png");
novaPasta.criarNovaPasta(this, x , y);

//Classe Pasta
public function criarNovaPasta(fund:Fundo, x:int, y:int):void
{
    //Adicionar a pasta
    this.id = numPastas;
    this.texto.text = "Nova Pasta";
    this.name = this.texto.text;
    this._nome = this.texto.text;
    this.imagem = "imagens/icons/pasta.png";
    this.miniatura = "imagens/icons/icons_pequenos/pasta.png";
    this.detalhe = "Pasta criada pelo utilizador";
    this.posicao = x + " " + y;
    this.x = x;
    this.y = y;
    Main.fundoJogo.addChild(this);

    //Adicionar ao Array de pastas
    var teste:PastaDef = this.createDef();
    Main.pastas.push(teste);

    var registo14:String = "Nova pasta criada com o nome: " +
this._nome + ".";
    criarRegisto(registo14);
}
```

Código 15 – Criar uma pasta na raiz do cenário

No Código 15, encontra-se um excerto de uma das formas de criar uma nova pasta. Neste caso, cria-se uma pasta no ambiente de trabalho, a raiz do cenário.

Este processo não inclui a alteração do nome da pasta após esta ser criada, uma vez que, quando se cria uma pasta no sistema operativo *Windows*, é necessário seleccioná-la para mudar o nome. Se assim não se fizer, a pasta ficará com o nome “Nova Pasta”.

A pasta vai ser criada na posição do cursor do rato, para que fique logo na sua posição final. São atribuídas algumas variáveis, para definir as imagens para a miniatura, posição, nome, entre outros. Para concluir o processo, resta actualizar o *array* geral, que guarda todas as pastas.

```
//Classe Pasta
//Adicionar nova pasta pela janela de conteúdo de outra

var menu1:menuPasta = new menuPasta(); //janela com o conteúdo
menu1.TituloPasta.text = nomePasta;
menu1.novaPasta1.addEventListener(MouseEvent.CLICK,
abrirMenuNomePasta, false, 0, true);

private function abrirMenuNomePasta(e:MouseEvent):void
{
    caixaNomePasta = new nomePastaMenu();
    caixaNomePasta.name = "nomePasta";
    caixaNomePasta.x = 275;
    caixaNomePasta.y = 245;
    Main.fundoJogo.addChild(caixaNomePasta);
    caixaNomePasta.ok_btn.addEventListener(MouseEvent.CLICK,
darNomePasta, false, 0, true);
}

function darNomePasta(e:MouseEvent):void
{
    var nome:String;
    if (caixaNomePasta.nomePasta.text == "")
    {
        nome = "Nova Pasta";
    }
    else
    {
        nome = caixaNomePasta.nomePasta.text;
    }
    Main.fundoJogo.removeChild(caixaNomePasta);

    //Adicionar a pasta
    var novaPasta:Pasta = new Pasta(nome,
"imagens/icons/pasta.png");
    novaPasta._nome = nome;
    novaPasta.name = nome;
    novaPasta.imagem = "imagens/icons/pasta.png";
    novaPasta.miniatura = "imagens/icons/icons_pequenos/pasta.png";
    novaPasta.detalhe = "Pasta criada pelo utilizador";
    novaPasta.naPasta = pastaMenu.id;

    //Guardar a pasta no Array Principal
    (...)

    var registo15:String = "Nova pasta criada com o nome: " + nome +
".";
    criarRegisto(registo15);
}
```

Código 16 – Criar uma pasta dentro de outra

No Código 16, pode visualizar-se a segunda e última forma de criar uma Nova Pasta. Desta forma, caso o jogador esteja a visualizar o conteúdo de uma pasta e decidir que é necessário adicionar-lhe uma nova, pode fazê-lo directamente.

O processo é um pouco diferente do anterior. A criação da pasta é feita através de um botão posicionado na janela de visualização do conteúdo da pasta destino. Ao pressionar este botão, surgirá uma nova janela, para introduzir o nome da pasta. Assim que pressionar o botão verde, esta pasta é criada e adicionada à pasta destino. Para concluir este processo, resta actualizar o *array* geral, que guarda todas as pastas.

### 5.2.11. Explicação dos objectos do cenário – Reciclagem

O elemento seguinte que se analisa é a Reciclagem.



Ilustração 44 – Reciclagem quando cheia (à esquerda) e vazia (à direita)

O elemento Reciclagem (Ilustração 44) comporta-se como uma pasta, exceptuando algumas ressalvas. Podem na mesma ser enviados e retirados elementos desta, mas, como se trata da Reciclagem, estes elementos, se lá estiverem, não podem ser consultados.

Existem ainda dois *icons* passíveis de se aplicar à reciclagem: um é o *icon* que representa a reciclagem cheia e se assemelha a um caixote do lixo com papéis enrugados dentro, significando que existe um ou mais elementos na Reciclagem. Por contraposição, existe ainda o *icon* que representa a Reciclagem vazia. Este assemelha-se, da mesma forma, ao caixote do lixo do escritório, mas agora vazio, significando que não há elementos dentro da Reciclagem.

O jogador pode adicionar documentos, atalhos ou pastas, arrastando estes de fora (raiz do cenário) para dentro da janela da Reciclagem. Este processo é ainda válido para a raiz do cenário, se o jogador arrastar um documento/atalho/pasta para cima da Reciclagem, este é adicionado.

O processo de remoção dos elementos da Reciclagem é semelhante ao anterior. Deve seleccionar o elemento e, com o botão esquerdo do rato premido, arrastar para fora da janela largando-o, para remover.

Para submeter o documento pedido, deve pressionar duas vezes sobre o mesmo, caso se encontre dentro da Reciclagem.

Mais à frente, nesta dissertação, serão explicados os processos mais importantes. Alguns dos quais são comuns entre elementos, como mudar o nome, adicionar/remover elementos, entre outros.

Classe associada: ***Reciclagem.as***

Esta classe faz referência ao objecto Reciclagem. Apesar de serem elementos diferentes, existem muitas semelhanças entre este e uma Pasta. A Reciclagem também armazena outros elementos, como Pastas, Documentos, Atalhos e Zonas. Também pode ver o seu nome alterado. Tem a mesma interacção com o jogador que o Documento, Atalhos e as Pastas, entre outros aspectos.

Assim sendo, podem ser encontradas nesta classe funções para as seguintes actividades:

- Criar o elemento Reciclagem;
- Atribuir os eventos como: *click*, *duplo-click*, *drag&drop*, entre outros;
- Definir as acções para os respectivos eventos;
- Alterar nomes e outras propriedades da Reciclagem;
- Abrir e visualizar o conteúdo da Reciclagem;
- Funções que alteram a imagem da Reciclagem, consoante esta está cheia ou vazia;
- Verificar as combinações com outros elementos;
- Função para eliminar outros elementos;
- Entre outras.

Não existem na classe funções para abrir o conteúdo de cada elemento eliminado, uma vez que a própria plataforma *Windows* também não o permite.

Classe associada: ***ReciclagemDef.as***

Esta classe, à semelhança das classes *PastaDef*, *ZonaDef*, *AtalhoDef* e *DocumentoDef*, é responsável por converter as informações do objecto Reciclagem em dados passíveis de serem armazenados em *SharedObject*.

Mais uma vez, remete-se a explicação desta classe para a classe *DocumentoDef*, onde já foi fornecida uma explicação detalhada deste processo e das razões que levaram à criação deste tipo de classes.

Adicionar um documento à reciclagem (semelhante para pastas)

Explica-se, agora, o processo de adição de um elemento à Reciclagem. Este processo é semelhante à adição de um elemento a uma pasta.

```
//Classe Reciclagem - Adicionar elemento à reciclagem
public function eliminarFicheiro(recicla:Reciclagem, pasta:Pasta,
doc:Documento):void
{
    if (_listaReciclagem.length < recicla._numMaxFicheiro)
    {
        if (doc != null)
        {
            var guardarDoc:DocumentoDef = doc.createDef();
            _listaReciclagem.push(guardarDoc);

            //tirar o doc do stage e adicionar à reciclagem
            if (Main.fundoJogo)
            {
                Main.fundoJogo.removeChild(doc);
            }

            //Verificar a verificar a resposta
            (...)

            for each (var value:DocumentoDef in Main.documentos)
            {
                if (value.id == doc.id)
                {
                    if (doc.posCorrecta)
                    {
                        value.posCorrecta = true;
                    }
                    value.posCorrecta = false;
                    value.numaPasta = true;
                    value.naPasta = -1;
                }
            }

            _cheia = true;
            alterarIcon();
        }
        if(pasta != null)
        {
            //Processo semelhante para pastas
            (...)
        }
    }
    else
    {
        trace("Reciclagem já se encontra cheia");
    }
}
}
```

Código 17 – Adicionar um elemento à reciclagem

No Código 17, pode consultar-se a função responsável por adicionar elementos à Reciclagem. Como se pode observar, esta função é válida para Pastas e para Documentos, consoante os parâmetros que lhe são passados, quando é chamada. Existe, ainda, uma função semelhante à anterior, para remover Atalhos.

A primeira operação é verificar se existe espaço na Reciclagem, para adicionar o elemento em questão. Caso seja possível, remove-se o elemento da raiz do cenário, adicionando-o à Reciclagem e actualizando as suas propriedades, para que fique registado que o elemento foi eliminado.

Retoma-se a explicação dos restantes ecrãs do jogo.



Ilustração 45 – Ecrã de resultados da primeira etapa

Na Ilustração 45, encontra-se o ecrã de resultados, que jogador visualiza, depois de ter encontrado o documento pedido na primeira etapa do jogo.

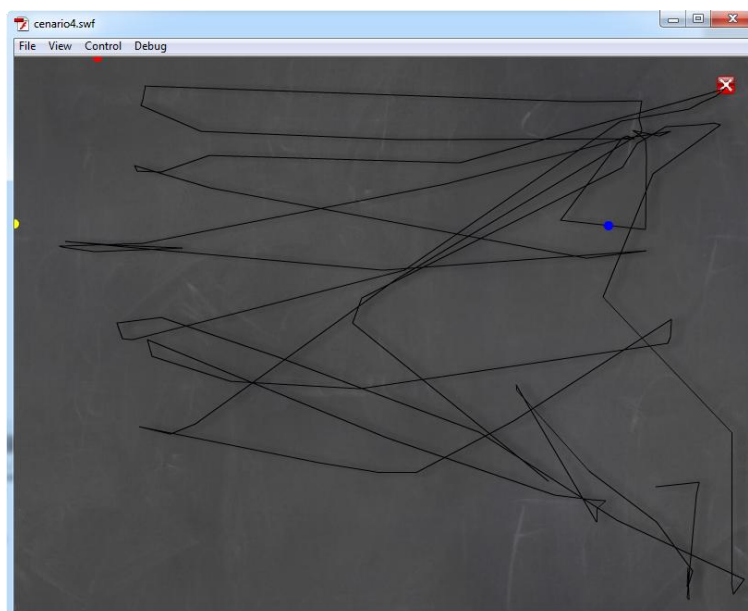
Neste ecrã, são dadas informações como: a pontuação por ter encontrado o documento pedido, a pontuação atribuída à organização inicial, o tempo despendido a encontrar o documento e que pontuação se atribuiu a esse tempo e a pontuação total.

Existem, ainda, dois botões que abrem o diagrama de *spaghetti* e o registo das acções do jogador, para consulta de informações adicionais.

Classe associada: *fimNivel.as\**

Esta classe é responsável pelo primeiro ecrã de fim de nível. Este ecrã surge, quando o jogador encontra pela primeira vez o documento pedido, antes de organizar o cenário pelo Método 5S.

Nesta classe, pode encontrar-se as funções para mostrar o registo das acções do jogador (*logs*) e mostrar o diagrama de *spaghetti* desenhado pelo movimento do rato, ao longo do cenário. Possui, ainda, funções para contabilizar os pontos atribuídos ao jogador pelos seus comportamentos no jogo e funções para preencher os campos que lhe permitem visualizar as suas pontuações iniciais.



**Ilustração 46 – Ecrã com o diagrama de *spaghetti***

Na Ilustração 46, pode ser consultado o diagrama de *spaghetti* construído a partir das posições do rato, registadas durante a procura.

Este diagrama pode indicar informações, como a densidade e concentração de documentos num determinado ponto do ecrã, como é que o jogador procurou o documento, de informações, entre outras.

#### Diagrama de *spaghetti*

Faz-se agora uma interrupção da explicação dos ecrãs do jogo, para explicar a importância do Diagrama de *spaghetti* na avaliação do desempenho do jogador.

Um diagrama de *spaghetti*, Ilustração 47<sup>29</sup>, é uma representação visual do processo que, através de uma linha contínua, traça o caminho entre um elemento/estação ou actividade. Esta linha contínua permite que as equipas de gestão de processos identifiquem redundâncias no fluxo de trabalho, permitindo assim removê-las, agilizando o processo [Bialek, L. Duffy, & W. Moran, 2009].

---

<sup>29</sup> [http://www.ehow.com/how\\_4803373\\_spaghetti-diagram-lean-process.html](http://www.ehow.com/how_4803373_spaghetti-diagram-lean-process.html)

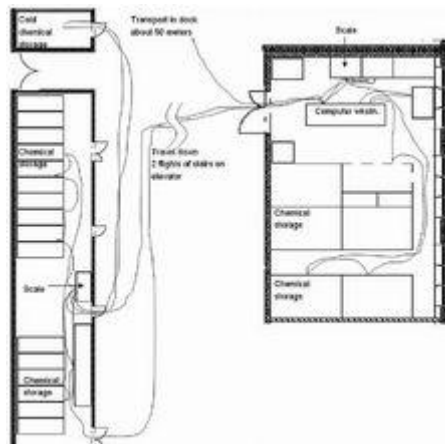


Ilustração 47 – Exemplo de um diagrama de *spaghetti*

Estes diagramas são, ainda, uma preciosa ajuda para visualizar o fluxo de materiais numa linha de produção ou processo, uma vez que, pela ilustração deste, são visíveis os transportes desnecessários, permitindo assim a sua optimização.

Para o cenário, usou-se o Diagrama de *spaghetti*, de modo a avaliar os movimentos que o jogador fazia com o cursor do rato. A linha contínua, característica deste diagrama, é desenhada através de uma função que armazena, desenha e une os pontos do diagrama.

Através desta informação, pode tentar-se extrapolar, uma vez que é impossível ter a certeza do método que o jogador usou para procurar o documento, dado que a tendência dos utilizadores de computador é mover o cursor do rato para o local do ecrã onde estão a olhar.

#### *Tracking* das posições do cursor do rato

Segue-se, a função cujo objectivo é guardar, para um *array* de pontos, as posições do cursor durante o jogo.

```
//Guardar o ponto da posição do rato no Array de Posições
private var timer:Timer = new Timer(230);
timer.addEventListener(TimerEvent.TIMER, mark, false, 0, true);
timer.start();

public function mark(e:TimerEvent):void
{
    pos = mouseX + "|" + mouseY;
    Main.trackArray.push(pos);
}
```

Código 18 – Gravar uma posição do cursor no *array* das posições

No Código 18, pode visualizar-se a função chamada periodicamente, para guardar a posição X e Y do rato. Esta função adiciona o ponto com as coordenadas da posição do rato a um *array*, constituído por elementos de texto, também conhecidos por *strings*.

Cada elemento é constituído por uma posição X, um separador e a posição Y. Escolheu-se o carácter “|”, por ser fácil de filtrar no processo seguinte.

Estes pontos serão usados para desenhar o diagrama de *spaghetti*, que se explica de seguida.

Desenho do gráfico de spaghetti

Segue-se a explicação da função responsável por desenhar o diagrama de *spaghetti*.

```
//Leitura do Array de Posições - Desenhar o diagrama de Spaghetti
private function desenharDiagrama (e:Event):void
{
    //Criar diagrama
    diagrama = new blackboard();
    (...)
    fundoJogo.addChild(diagrama);

    //Propriedades da linha e filtros
    (...)
    var oldx:uint;
    var oldy:uint;
    var primeiraVez:Boolean = true;

    //Desenhar Rectas
    for each (var ponto:String in trackArray)
    {
        var trackArrayPonto:Array = ponto.split("|");
        //move to the correct starting position for drawing
        if (primeiraVez)
        {
            oldx = trackArrayPonto[0];
            oldy = trackArrayPonto[1];
            primeiraVez = false;
        }
        //Desenhar as linhas
        (...)
        oldx = trackArrayPonto[0];
        oldy = trackArrayPonto[1];
    }
    //Código para colocar pontos
    //Pontos coloridos para abrir documentos, pastas, reciclagem,
etc
    (...)
}
}
```

Código 19 – Criar o diagrama de *spaghetti*

No Código 19, está explicitada a forma de criação do diagrama de *spaghetti*. É criado um *movieclip* vazio, para usar como base do diagrama. Depois, é percorrido o *array* geral, que guarda os pontos.

Para cada elemento é feita uma separação dos dois valores pelo carácter “|” e, à medida que se vai avançando no *array*, é conectado o ponto anterior com o actual. É necessário proteger

este processo, para o caso do primeiro ponto a ser colocado, uma vez que não tem nenhum ponto anterior ao qual possa ser ligado.

Continua-se, agora, a explicação dos ecrãs do jogo.

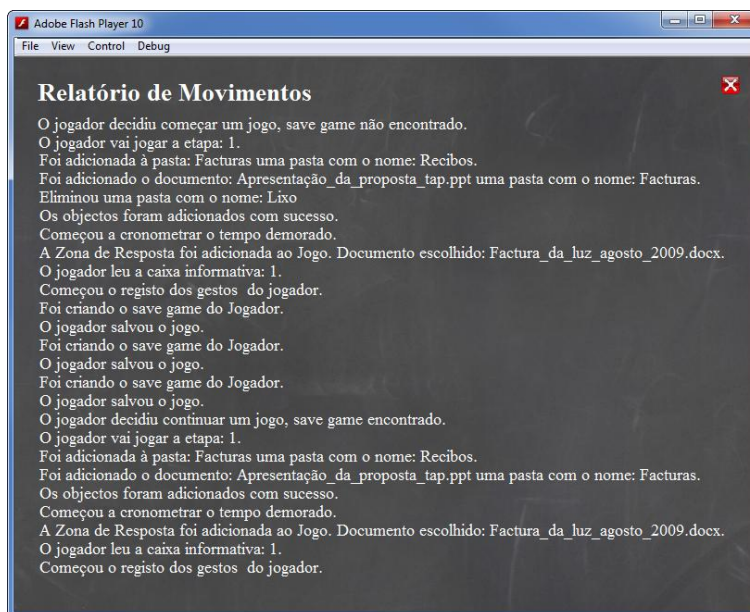


Ilustração 48 – Ecrã com o registo das acções do jogador

Se o jogador pressionar o botão ao lado direito do gráfico, surgirá o ecrã da Ilustração 48. Neste ecrã, o jogador pode consultar as suas acções.

Apesar de esta informação ser visível pelo jogador, serve, também, para avaliar, numa segunda etapa, as acções efectuadas pelos jogadores. Assim, pode-se tentar descobrir certos padrões comuns no comportamento dos jogadores e melhorar a sua performance.

Este ecrã conclui a primeira etapa do cenário.

Só na segunda etapa é que se sugere ao jogador que crie novas pastas e novas zonas, uma vez que, só agora, necessitará delas para organizar o cenário.

Este facto define, então, o objectivo para a próxima etapa do jogo. Uma vez que procurar um determinado documento num ambiente desorganizado se provou uma tarefa bastante difícil, cabe ao jogador propôr um sistema de organização para o ambiente de trabalho.

Classe associada: *menuObjectivos2.as\**

Esta classe representa o ecrã com os objectivos para a segunda etapa do jogo. Nesta etapa, o jogador deve organizar o cenário segundo os princípios do Método 5S. Este ecrã transmite, ainda, ao jogador a informação de como se acede ao submenu.

Muito à semelhança da classe anterior, esta só possui o evento do botão “continuar”, que inicia a segunda etapa do jogo.

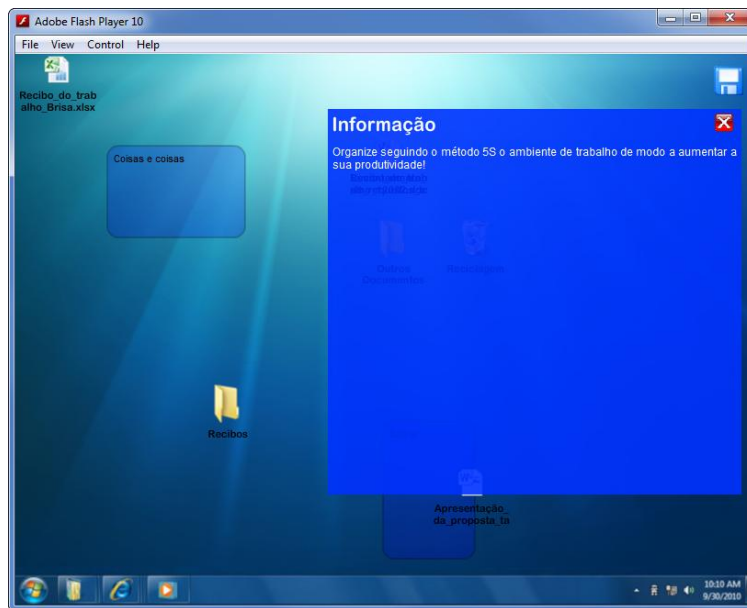


Ilustração 49 – Ecrã do cenário (2ª etapa)

Na Ilustração 49, pode visualizar-se a nova área de jogo a ser organizada. Optou-se por mudar o fundo do ecrã do jogo, para reforçar a ideia ao jogador, que já progrediu e cujo objectivo mudou.

À semelhança da outra área de jogo, mostra-se mais uma informação relativa ao objectivo deste. Estas mensagens podem, futuramente, ser modificadas para conter mais informações sobre tópicos diferentes ou criar um sistema de notificações mais complexo.

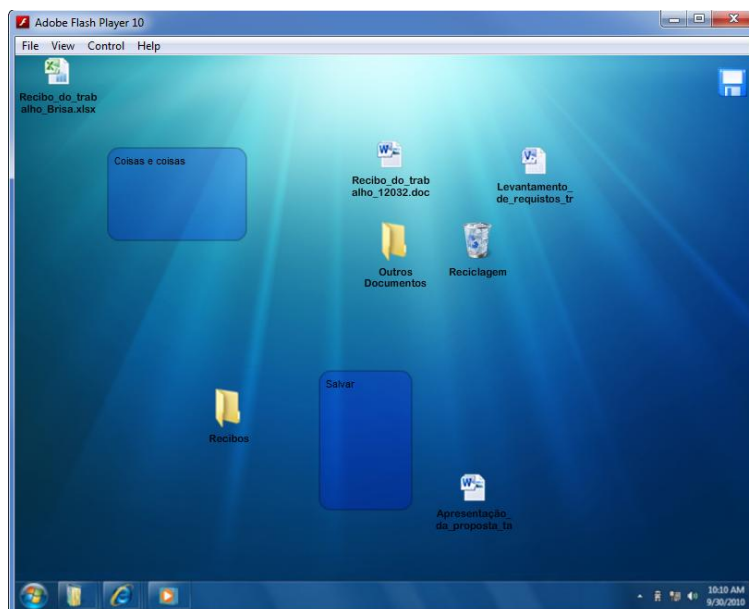


Ilustração 50 – Ecrã de organização (2ª etapa) sem janela de informação

Na Ilustração 50, visualiza-se a mesma área de jogo para a segunda etapa, agora com a caixa de informações fechada.

A grande diferença em relação aos objectos desta aérea para a outra, que não é visível pela observação da ilustração, é que já se introduziu alguma organização, de modo ajudar o jogador.

Nesta etapa, não é necessário haver um cronómetro, pois não se pretende apressar o jogador e fazer com que este tome decisões precipitadas. Dado que o único objectivo para esta etapa é propôr uma organização, para avançar, o jogador só terá de pressionar o botão, para salvar o jogo.



Ilustração 51 – Ecrã que confirma que foi salvo o método de organização

Assim que o jogador pressionar o botão, para salvar a organização que propôs, surgirá o ecrã, que pode ser visualizado na Ilustração 51. Este possui a informação de que a organização foi guardada com sucesso.

Classe associada: *fimNivel2.as\**

Esta classe é muito semelhante à classe anterior. Contudo, os eventos que esta invoca são bastante diferentes dos anteriores.

Associada ao ecrã de jogo que possui a informação de que a organização proposta pelo jogador, para este cenário, foi guardada com sucesso, a classe é responsável por chamar a função que guarda todos os dados relevantes do jogo. Este processo tem de ser realizado antes de se iniciar a terceira fase, onde o jogador terá de encontrar outro documento com a sua organização em vigor.

Continuando, seleccionando a opção seguinte, passa-se para o terceiro ecrã, presente na Ilustração 52, que possui os objectivos da terceira e última etapa do cenário.

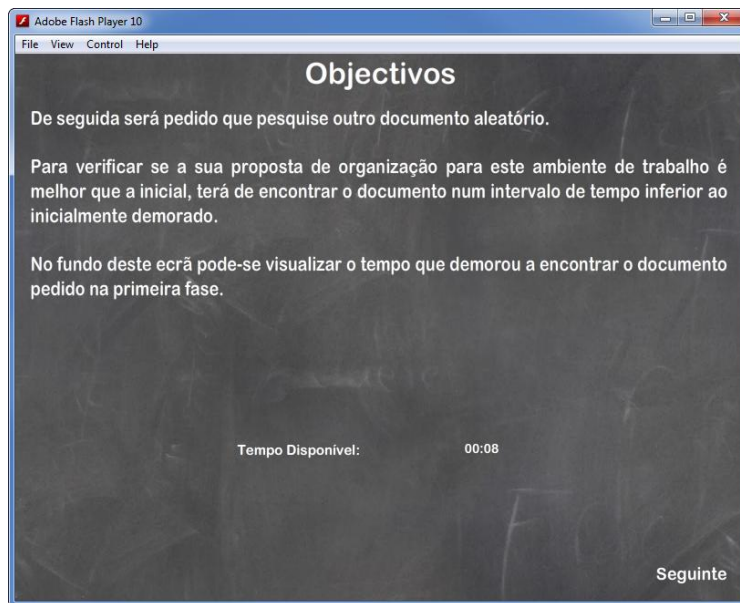


Ilustração 52 – Terceiro ecrã de objectivos

Nesta etapa, é pedido ao jogador que pesquise outro documento seleccionado aleatoriamente. Desta vez, o jogador terá de o encontrar dentro do tempo disponível, que pode ser consultado no ecrã anterior.

Caso o jogador consiga encontrar o documento em menos tempo do que o inicialmente demorado, pode concluir-se, discutivelmente, que o método, por ele proposto de organização do cenário, é mais indicado do que o inicialmente aplicado.

Classe associada: *menuObjectivos3.as\**

Esta classe representa o ecrã que o jogador visualiza, assim que submeter a sua organização para o cenário actual.

Este ecrã, para além de mostrar o objectivo da terceira e última fase, pretende transmitir ao jogador o tempo que este terá para a completar.

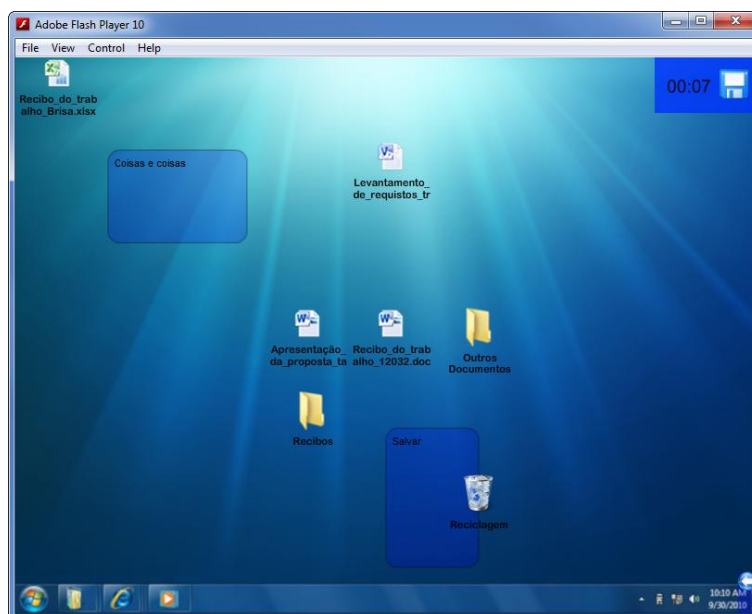
Na terceira etapa, o jogador terá de voltar a encontrar outro documento, de modo a validar a organização por si proposta.

Para além do evento do botão “continuar”, que inicia a terceira etapa do jogo, esta classe tem ainda uma função que dá a conhecer ao jogador o tempo disponível, preenchendo uma caixa de texto colocada no ecrã com este propósito.

Nesta nova etapa, conclui-se que não houve alterações ao cenário, nem aos seus elementos. Optou-se, também, por não mudar o fundo do ecrã entre etapas, como vinha a ser hábito, para não dar a falsa percepção de que houve modificações no cenário.

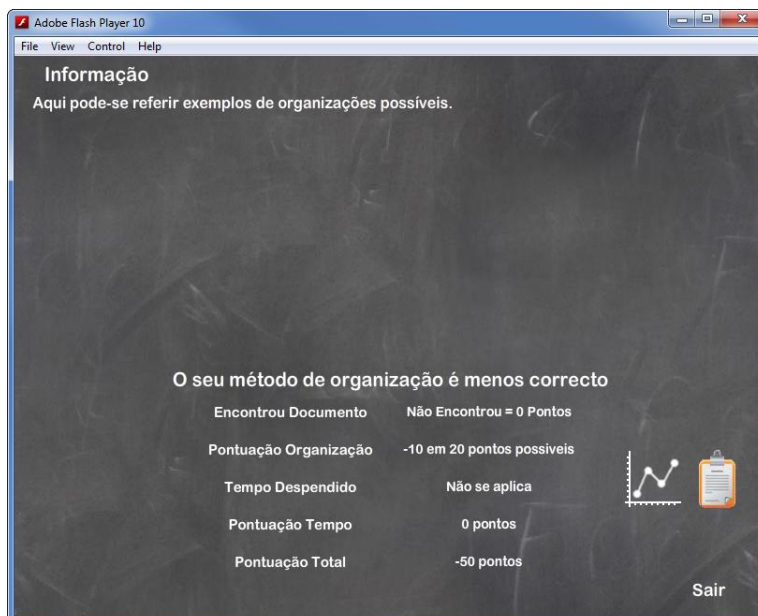
As únicas diferenças do cenário anterior para este são o Cronómetro, que foi adicionado e, agora, está em contagem decrescente, e a área de entrega do documento, que voltou a ser adicionada.

Apesar de o jogador estar em contra-relógio, nesta tarefa, é possível ser-lhe mostrada uma janela com informações relevantes.



**Ilustração 53 – Ecrã do cenário do jogo (3ª etapa) sem janela de informação**

A Ilustração 53 pretende apresentar o ecrã de jogo sem a janela de informações aberta. Como tem vindo a ser referido, para concluir esta etapa o jogador deve submeter o novo documento seleccionado aleatoriamente.



**Ilustração 54 – Terceiro e último ecrã de resultados finais**

Para concluir esta parte da explicação dos ecrãs do cenário, visualiza-se, na Ilustração 54, o último ecrã do jogo (antes de regressar ao menu).

Este ecrã é idêntico ao já analisado na Ilustração 45. Neste pode ser consultada a pontuação actualizada do jogador. São estas pontuações que serão enviadas para a base de dados e poderão ser consultadas na tabela de pontuações do jogo.

Podem ser consultadas informações como: avaliar se este método de organização é melhor ou pior que o inicial; se o jogador encontrou o documento pedido; pontuações da organização (se colocou os documentos e pastas nas suas posições correctas); pontuações atribuídas ao tempo com que completou a tarefa e a pontuação total.

Classe associada: *menuSair2.as\**

Esta classe pertence ao último ecrã, que o jogador visualizará antes de voltar ao local de escolha dos cenários do menu.

Esta função é muito semelhante à função *fimNivel*, já definida neste tópico e apresenta as mesmas estatísticas sobre o jogador, uma vez que possui funções semelhantes para os seguintes pontos:

- Registo de *logs*;
- Criar o gráfico de *spaghetti*;
- Contabilizar os pontos atribuídos ao jogador pelos seus comportamentos no jogo;
- Preencher os campos que lhe permitem visualizar as suas pontuações finais.

A estes acrescentam-se, ainda, informações sobre se o seu método de organização é, ou não, melhor que o actual, caso encontre o documento antes do tempo.

Uma vez que este é o último ecrã do cenário, nesta classe encontram-se ainda as funções que submetem a pontuação do jogador para a base de dados e as funções que removem todos os elementos do ecrã.

Desta vez, o botão de continuar foi substituído pelo botão de sair, que direcciona o jogador para o ecrã de escolha de cenários do menu principal.

#### Gravar pontuação

Explica-se, agora, a função que grava a pontuação final do jogador na base de dados.

No Código 20, encontra-se a função responsável por enviar a pontuação do jogador para uma página em PHP, que, por sua vez, guarda na base de dados, criada para armazenar as pontuações dos jogadores.

```

//Gravar pontuação na base de dados
public function gravarPontuacao():void
{
    var request:URLRequest = new
URLRequest("http://localhost/www/cenario4/php/pontuacaoGravar.php?com
ndo=gravar_pontuacao");
    var requestVars:URLVariables = new URLVariables();

    requestVars.nome_jogador = this.nomeJogador;
    requestVars.id_cenario = this.cenarioActual;
    if (this.cenarioActual == 4)
    {
        requestVars.nome_cenario = "Ambiente de Computador";
    }
    requestVars.pontuacao_total = this.pontuacao;
    requestVars.pontuacao_organizacao = this.pontuacaoOrganizacao;
    requestVars.tempo_demorado = this.tempoDemorado;
    requestVars.pontuacao_tempo = this.pontuacaoTempo;

    request.data = requestVars;
    request.method = URLRequestMethod.POST;

    var urlLoader:URLLoader = new URLLoader();
    urlLoader = new URLLoader();
    urlLoader.dataFormat = URLLoaderDataFormat.TEXT;
    for (var prop:String in requestVars)
    {
        //trace("Sent: " + prop + " is: " + requestVars[prop]);
    }
    try
    {
        urlLoader.load(request);
        criarRegisto("Pontuação do jogador gravada com sucesso.");
        Main.concluido = true;
    }
    catch (e:Error)
    {
        trace(e);
    }
}

```

**Código 20 – Guardar a pontuação na base de dados**

A base de dados só possui uma tabela com os vários campos para armazenar a informação do cenário e do jogador. Contudo, não há nenhuma funcionalidade com complexidade que justifique um tópico só para esta.

### 5.2.12. Explicações teóricas da programação das classes do cenário

Este tópico aborda as classes que ainda não foram referidas neste relatório, ajudando a perceber a estrutura geral do jogo, e fornece mais detalhes sobre cada uma das classes utilizadas. Não se vai entrar em pormenores de como foi escrita cada função de cada classe, porque, desta forma, não se obteria um conhecimento global do funcionamento do cenário.

Na Ilustração 32, encontra-se uma listagem das classes criadas durante o desenvolvimento deste cenário. Uma vez que possuem nomes, à primeira vista confusos, será dada, de seguida, uma explicação da finalidade de cada classe.

Com este conhecimento em mente, numa segunda fase é fácil para o utilizador ler o código das classes e perceber o objectivo de cada função, dado que esta possui comentários que o ajudam nessa tarefa e foi feito um esforço para que o nome da classe, função ou variável fosse intuitivo.

O significado do asterisco, depois do nome da classe, serve para indicar que esta pode, futuramente, servir outros propósitos. Dada a sua simplicidade, é discutível, se estas devem ser unidas a outras, de modo a simplificar a estrutura do jogo e tornar este mais rápido. Optou-se por separar, para facilitar a leitura de código e para deixar em aberto que novas funcionalidades sejam acrescentadas futuramente.

Geralmente, este tipo de classes estão associadas a *Movieclips*, como ecrãs de objectivo, menus de fim de jogo, caixas informativas, zonas de entrega de documentos ou outros, e permite um maior controlo deste tipo de objectos.

Segue-se uma lista com as classes que, até agora, careciam de uma explicação detalhada, ordenada alfabeticamente:

### ***Jogo.as***

Segue-se a classe Jogo. Como foi referido, esta faz parte de um conjunto de classes bastante importantes para o cenário.

Ao contrário das outras classes, a classe Jogo não se materializa num objecto no cenário, apesar do objecto que esta cria ser adicionado ao fundo, de modo a poder ser acedido durante todo o jogo.

Como o nome indica, a classe jogo armazena informação sobre o estado do jogo actual. São armazenados e actualizados cerca de quinze parâmetros como:

- Pontuações parciais e totais;
- Variáveis *booleanas* (tomam apenas o valor de verdadeiro ou falso), que indicam fim de jogo, se encontrou o documento, se viu a caixa de informação, se pode continuar a jogar, entre outras;
- Variáveis sobre que cenário e que etapa está a ser jogada;
- Variáveis para cronometrar tempo decorrido e o tempo restante;
- Variáveis com o nome do documento escolhido e se já foi encontrado;
- Variáveis para verificar o final do jogo;
- Entre outras.

Esta classe possui ainda as funções que verificam o final do jogo (chamada pela classe Cronómetro), que permitem salvá-lo, reiniciá-lo e gravar pontuações na base de dados.

#### ***nomeMenu.as\****

A escolha do nome desta classe não foi a mais indicada. Esta está associada a um *Movieclip*, que é adicionado ao elemento sempre que o jogador o selecciona e que tem por objectivo mostrar numa caixa de texto o nome completo deste. Este *Movieclip* é essencial, uma vez que qualquer elemento pode ter um nome bastante extenso, não sendo possível visualizá-lo na sua totalidade, quando este está no ambiente de trabalho.

Esta classe possui a função que lê o nome do elemento e o adiciona a essa caixa de texto, tornando-o completamente visível.

Esta caixa será adicionada/removida sempre que o jogador selecciona ou não um elemento.

#### ***nomePastaMenu.as\****

Esta classe está associada a um *Movieclip* especial, criado para que seja permitido ao jogador introduzir o nome da pasta, caso pretenda adicionar outra, enquanto consulta o conteúdo da primeira.

O que este *Movieclip* permite ao jogador, após seleccionar o botão “Nova Pasta”, é escrever o nome da pasta que acabou de criar.

Clicando no botão verde, termina a operação da criação de uma Nova Pasta.

#### ***Selector.as\****

Esta classe está associada a um *Movieclip*, que contém um quadrado azul com transparência, que surge quando o jogador selecciona um elemento do cenário (Documentos, Atalhos, Pastas ou Reciclagem). Ao surgir este *Movieclip*, o jogador é informado que o elemento que seleccionou está activo, sendo uma referência visual que o sistema entendeu o pedido e respondeu de acordo com o que é esperado. Pretende-se, assim, tornar o sistema mais interactivo e que a simulação do ambiente gráfico seja mais realista.

#### ***tipoElemento.as\****

Esta classe não tem nenhum objecto gráfico associado. O único objectivo desta é, fornecendo um *Movieclip*, retornar o seu tipo, permitindo assim, identificar a sua natureza.

Esta permite identificar todos os *Movieclip* correspondentes a todos os elementos que têm vindo a ser descritos neste capítulo, como Documentos, Atalhos, Pastas, Reciclagens, Zonas, entre outros.

### ***adobe/BitmapSaver.as***

O objectivo desta classe é fazer *PrintScreens* do ecrã de jogo. Escrita pela *Adobe* e livre para uso, esta classe foi usada para implementar uma funcionalidade que permite gravar remotamente todos os ecrãs que o jogador viu ao longo do cenário, gravando visualmente a organização do cenário e as pontuações do jogador.

Esta funcionalidade não está activa na versão final do jogo.

### ***adobe/Serializer.as***

Esta classe é usada pela classe anterior. O seu objectivo é transmitir e receber dados complexos do *Flash* para um servidor remoto em PHP, usando as suas funções nativas, como o *serialize* e *unserialize*.

É uma classe que permite a comunicação entre as diversas plataformas.

### ***adobe/images/JPGEncoder.as***

Esta é também uma classe livre de uso, desenvolvida pela *Adobe* e auxilia a *BitmapSaver* a criar o ficheiro JPEG com a imagem.

Não se vai entrar em pormenores de como esta classe cria e estrutura o ficheiro, de modo a este se tornar numa imagem, por não fazer parte do âmbito do trabalho.

### ***adobe/images/PNGEncoder.as***

Esta classe é livre de uso, desenvolvida pela *Adobe* e auxilia a *BitmapSaver* a criar o ficheiro PNG com a imagem. À semelhança da classe *JPGEncoder*, não se vai entrar em pormenores de como esta consegue criar o ficheiro que constitui a imagem.

### ***Outras funcionalidades***

Existem, ainda, outras funcionalidades que não se encaixam numa categoria específica:

- Funções que registam a actividade do jogador (presentes em todas as classes);
- Funções para armazenar os diversos elementos;
- Funções para ler de ficheiros ou de *SharedObjects*;
- Funções para devolver números aleatórios, remover espaços de frases, etc.
- Entre outras.



## 6. Avaliação do Trabalho

*"Quality means doing it right when no one is looking."*

**Henry Ford**

Neste capítulo, será feita uma avaliação do trabalho, através da análise e interpretação de resultados obtidos.

Serão abordados os seguintes temas: definição dos instrumentos de recolha de dados, processo de angariação de avaliações, tratamento de dados, reflexão sobre os dados recolhidos, entre outros.

### 6.1. Instrumentos de recolha de dados

A avaliação foi realizada por um grupo de alunos, que frequentam o Mestrado de Engenharia Mecânica no ISEP, uma vez que faz parte do seu curso a aprendizagem de conteúdos *Lean*.

Esta sessão de avaliação decorreu no dia 4 de Outubro de 2011, na sala F225, numa aula de Mestrado leccionada pelo Eng. Pereira Lopes, no Departamento de Engenharia Mecânica.

Foi feita uma apresentação aos alunos, cujos slides podem ser consultados nos anexos. Aos que chegaram depois de a aula já ter começado, foi-lhes explicado, individualmente, os objectivos e funcionalidades do jogo, antes de começarem a jogar.

No final, foram recolhidos quinze (15) formulários preenchidos, tendo sido todos validados e aceites. Segue-se o método de avaliação utilizado.

#### 6.1.1. Método de avaliação utilizado

Para além da avaliação constante ao longo da implementação, característica do estilo de desenvolvimentos de prototipagem, realizou-se ainda uma avaliação através do preenchimento de um questionário, numa sessão de avaliação do jogo.

Antes de se continuar a desenvolver este capítulo, deve referir-se que, na noite anterior à avaliação, a *Adobe* lançou uma actualização para o *Flash*, que passou da versão 10.3 para a versão 11. Versão esta que já se encontrava instalada, no dia seguinte, nos computadores do Departamento de Engenharia Mecânica do ISEP.

### 6.2. Questionário para a avaliação do Jogo

O questionário de avaliação consiste numa tabela com afirmações, que pode ser consultada nos Anexos desta dissertação. Os alunos responderam a cada uma destas questões, assinalando, numa escala de 0 (mínimo) a 5 (máximo), o quanto concordavam com as afirmações.

Apresenta-se o significado de cada um dos níveis possíveis de serem escolhidos.

- 0 – Discordo Completamente;
- 1 – Discordo;
- 3 – Indiferente;
- 4 – Concordo;
- 5 – Concordo Plenamente.

### 6.3. Parâmetros a avaliar

Segundo Tsukumo [Tsukumo, 1997], os Métodos de Avaliação de *Software* mais comuns baseiam-se nos seguintes parâmetros: Funcionalidade, Usabilidade *Confiabilidade*, Eficiência, *Manutenibilidade*, Portabilidade.

Alguns destes parâmetros, que vão ser debatidos no tópico de Discussão de Resultados, apenas foram passíveis de avaliar através da observação do grupo, enquanto interagiu com o jogo. Apesar de esta observação não poder ser quantificada, nem dar certezas nos resultados, já fornece uma ideia clara da aceitação do jogo e dos aspectos que devem ser melhorados.

### 6.4. Apresentação dos resultados

Segue-se, agora, a tabela resultante do processamento dos quinze questionários preenchidos.

Tabela 3 – Respostas ao questionário

Avaliação							
Número	Discordo completa/	Discordo	Indiferente	Concordo	Concordo Completa/	Não resp.	
Perguntas	1		6, 13, 14,		2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 15.	1	
	2		5,	8, 9,	2, 3, 4, 6, 10, 11, 12, 13, 14, 15.	1, 7,	
	3			3, 5, 6, 9, 11, 13, 15.	1, 4, 8, 10, 12, 14,	2, 7,	
	4		5, 6,	4, 11, 14,	2, 7, 8, 9, 10, 13, 15.	1, 3, 12,	
	5			14,	1, 4, 5, 6, 8, 10, 11, 13, 15.	2, 3, 7, 9, 12,	
	6			4, 5, 6, 13,	1, 2, 3, 8, 9, 10, 11, 12, 14, 15.	7,	
	7		6, 9,	5, 8, 11, 12, 13, 15.	1, 2, 3, 4, 10, 14,		7
	8	3, 9, 10, 14,	1, 2, 4, 5, 8, 13,	11, 12, 15.		6, 7,	
	9	3, 5, 9, 10, 14,	1, 2, 7, 8, 13,	4, 11, 12, 15.		6,	
	10		1,	12,	2, 3, 4, 5, 7, 8, 10, 13, 14, 15.	6, 9, 11,	
	11	9, 10,	1, 2, 6, 7, 12,	4, 8, 11, 13, 14, 15.	5,	3	
	12				4, 5, 8, 9, 10, 11, 12, 13, 14, 15.	1, 2, 3, 6, 7,	
	13	1, 2, 10, 11, 15.	3, 4, 5, 7, 9, 12, 13,	8, 14,	6,		

Na Tabela 3, podem ser consultadas as pontuações atribuídas a cada afirmação. Numeraram-se os questionários, foi criada a tabela anterior e anotou-se, para cada afirmação, a pontuação atribuída por todos os jogadores.

Para que se percebam melhor os resultados, criou-se a tabela seguinte, onde se contabilizaram as pontuações totais obtidas para cada afirmação do questionário.

**Tabela 4 – Respostas ao questionário (pontuações finais)**

<b>Avaliação</b>							
Número		Discordo completa/	Discordo	Indiferente	Concordo	Concordo Completa/	Não resp.
<b>Perguntas</b>	<b>1</b>	0	3	0	11	1	0
	<b>2</b>	0	1	2	10	2	0
	<b>3</b>	0	0	7	6	2	0
	<b>4</b>	0	2	3	7	3	0
	<b>5</b>	0	0	1	9	5	0
	<b>6</b>	0	0	4	10	1	0
	<b>7</b>	0	2	6	6	0	<b>1</b>
	<b>8</b>	4	6	3	0	2	0
	<b>9</b>	5	5	4	0	1	0
	<b>10</b>	0	1	1	10	3	0
	<b>11</b>	1	5	6	1	1	0
	<b>12</b>	0	0	0	10	5	0
	<b>13</b>	5	7	2	1	0	0
<b>TOTAL</b>		<b>15</b>	<b>32</b>	<b>39</b>	<b>81</b>	<b>26</b>	<b>1</b>

Na Tabela 4, podem ser consultadas as pontuações totais para cada afirmação.

No tópico seguinte, Discussão de Resultados, será analisado ao pormenor o valor individual obtido para cada afirmação.

## 6.5. Discussão dos resultados obtidos

Neste tópico, serão apresentados gráficos, bem como uma explicação do resultado obtido para cada uma das afirmações do questionário.

### 1. O jogo é intuitivo.

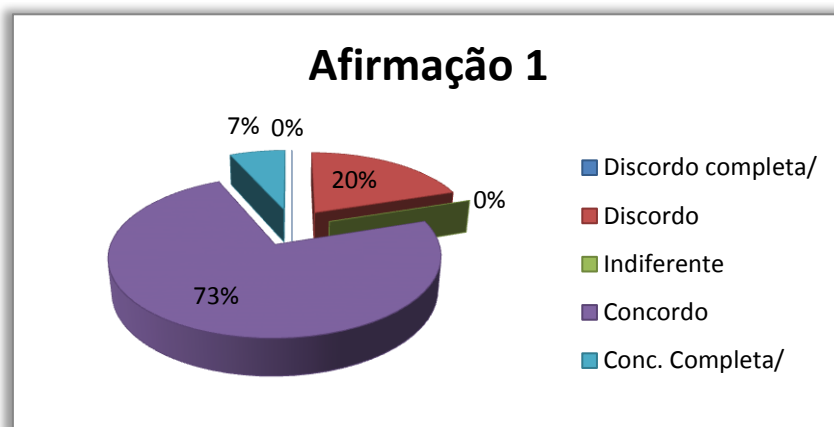


Gráfico 1 – Gráfico circular para a 1ª afirmação

Para a primeira afirmação, oitenta por cento (80%) dos jogadores concordaram que este é intuitivo. É um ótimo resultado, significando que a simulação do ambiente de trabalho, familiar a todos estes, está realista, correspondendo ao objectivo pretendido.

### 2. O cenário proposto é realista.

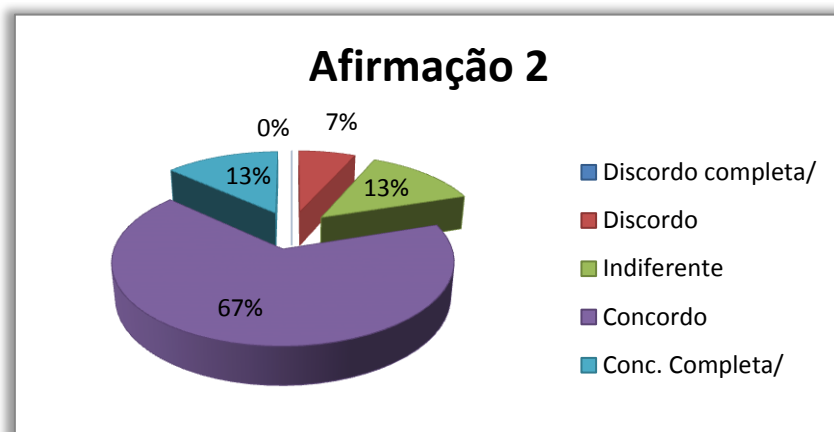


Gráfico 2 – Gráfico circular para a 2ª afirmação

Apesar das dificuldades sentidas em criar elementos espectáveis de serem encontrados no ambiente de trabalho de um computador, na área de contabilidade, foi possível, de acordo com estes resultados, criar um cenário realista.

Este resultado significa, ainda, que o nível de complexidade do cenário transmite ao jogador a ideia de que o computador faz parte de um ambiente de trabalho empresarial, mas não o impede de jogar, por este não possuir conhecimentos na área.

### 3. *O jogo permitiu-me compreender melhor o Método 5S.*

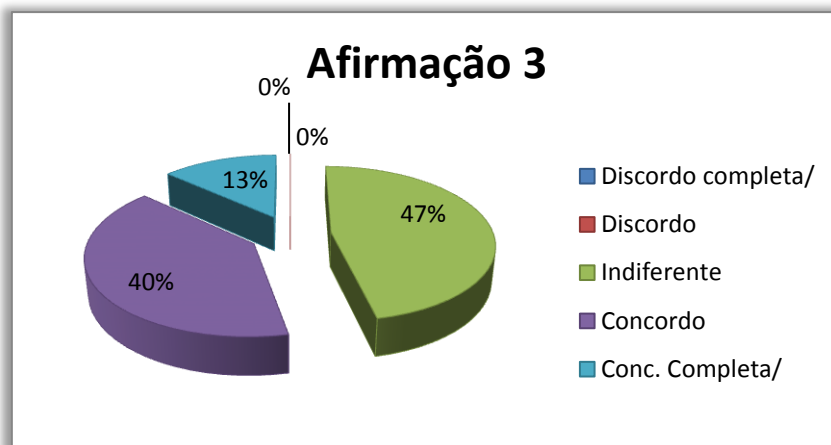


Gráfico 3 – Gráfico circular para a 3ª afirmação

Já a afirmação 3 dividiu o grupo. Apesar de cinquenta e três por cento (53%) concordarem que o jogo lhes permitiu compreender melhor o Método 5S, quarenta e sete por cento (47%) mostraram-se indiferentes. Isto pode ser devido ao jogo não entrar em pormenores sobre a matéria, explicando-se o conceito geral e os objectivos de cada etapa do processo.

Outra observação que deve ser feita é que o grupo que avaliou o jogo é constituído por alunos que estão a frequentar o Mestrado em Engenharia Mecânica e que já possuem experiência nesta área. Para estes o jogo teria de aprofundar a sua matéria, para que conseguisse cativá-los, o que já não faz parte do seu âmbito.

### 4. *Este jogo motiva a aprendizagem.*

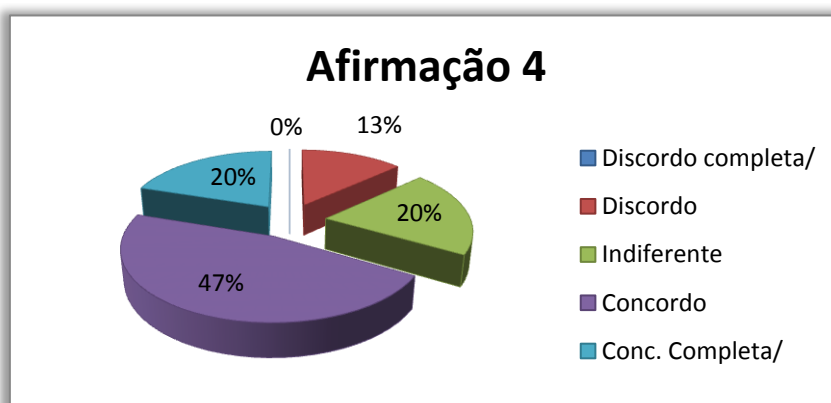


Gráfico 4 – Gráfico circular para a 4ª afirmação

Esta afirmação obteve um resultado muito positivo: sessenta e sete por cento (67%) dos alunos concordam que o jogo motiva aprendizagem dos conteúdos. Sendo esse um dos principais requisitos, pode afirmar-se que foi cumprido com sucesso.

**5. Este tipo de abordagem deve ser aplicado a outros tópicos do Lean.**

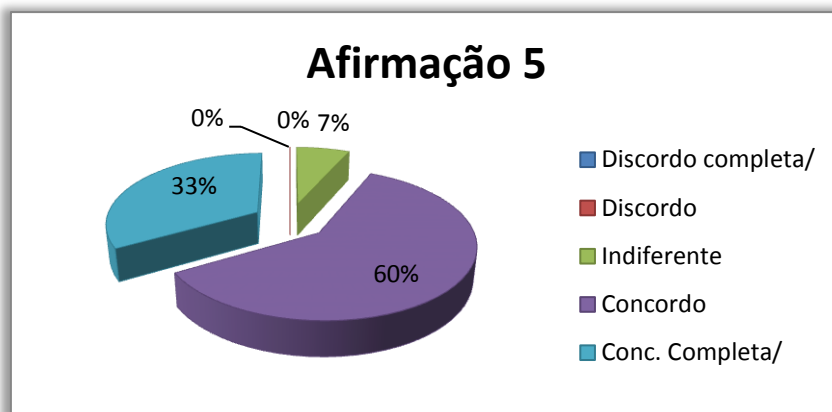


Gráfico 5 – Gráfico circular para a 5ª afirmação

Esta afirmação vem confirmar que existe uma necessidade por responder de Jogos Sérios que dinamizem e auxiliem a aprendizagem, de uma forma prática, de conteúdos de *Lean Manufacturing*. Existem, também, outros tópicos de *Lean*, que podem ser ensinados através deste tipo de jogos.

**6. A informação fornecida no jogo tem qualidade.**

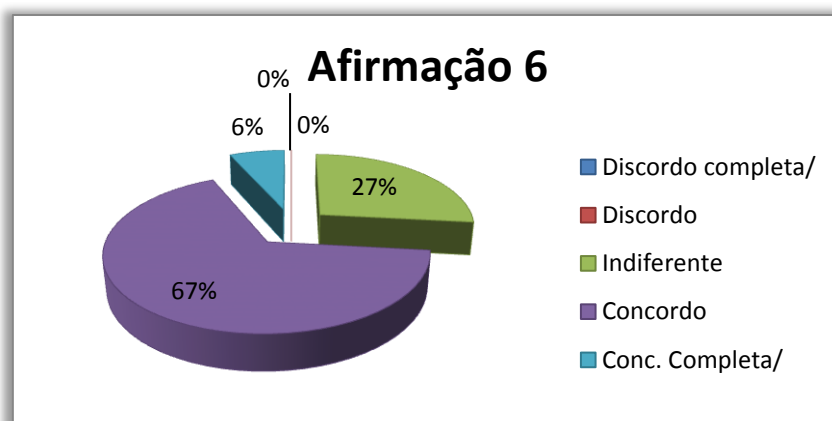


Gráfico 6 – Gráfico circular para a 6ª afirmação

Uma vez que a informação utilizada no jogo resulta da junção de uma pesquisa realizada e de uma apresentação sobre o Método 5S, feita para a *LLA*, era previsível que esta tivesse qualidade. Restava saber se está ao nível de alunos de Mestrado, que já possuem alguma experiência prática. Pelo gráfico anterior, pode concluir-se que a qualidade da informação apresentada está a esse nível.

**7. A informação fornecida no jogo tem a profundidade adequada.**

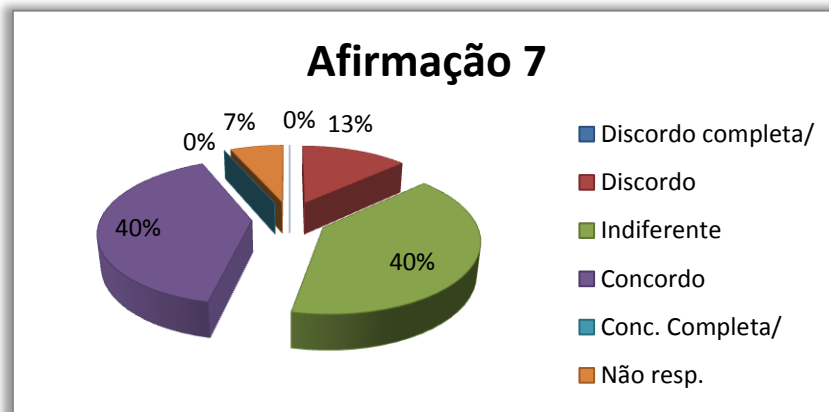


Gráfico 7 – Gráfico circular para a 7ª afirmação

Em relação ao detalhe da informação fornecida, metade dos alunos que preencheram os questionários são da opinião que não tem profundidade que chegue. Esta afirmação pode ser explicada pelos mesmos motivos da afirmação número três. Os alunos, que avaliaram este jogo, já possuem experiência prática, bem como uma licenciatura em Engenharia Mecânica, logo já estão familiarizados com os conteúdos.

Em conversa, depois da avaliação, estes afirmaram que, se a aplicação for direccionada para alunos que começam agora a aprender estes conteúdos, a profundidade é a mais indicada.

**8. Tenho conhecimento da existência de outro tipo de jogos para o Método 5S.**

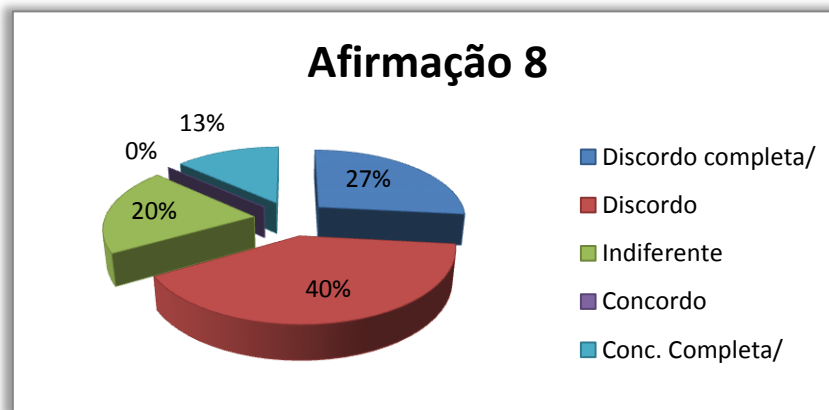


Gráfico 8 – Gráfico circular para a 8ª afirmação

As pontuações atribuídas a esta afirmação permitiram concluir que a maior parte dos alunos não conhece nenhum jogo que tenha como conteúdo o Método 5S.

Aos alunos que responderam que conhecem jogos que possuem o Método 5S como conteúdo, afirmaram, depois da avaliação, que só conhecem jogos reais com este tema e que para a plataforma digital é a primeira vez que experimentaram um.

**9. Já joguei outro tipo de jogos sobre Lean.**

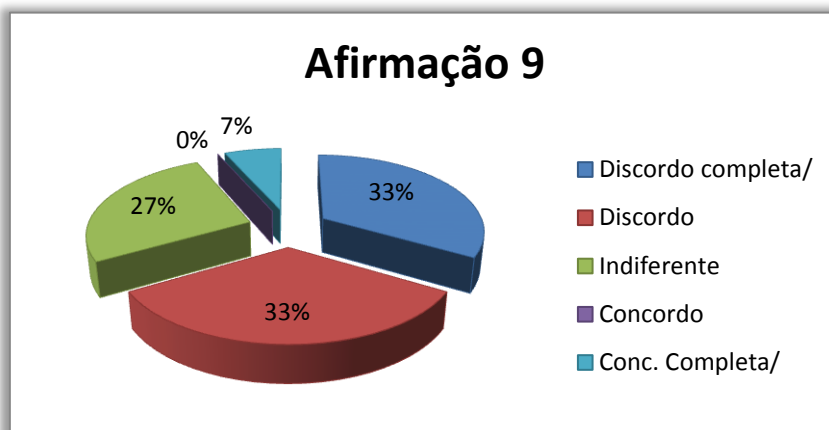


Gráfico 9 – Gráfico circular para a 9ª afirmação

Esta afirmação espelha a realidade da situação actual desta área de ensino. Já se concluiu que existem pouquíssimos jogos digitais e bastantes jogos reais de *Lean* que, pela amostra do gráfico anterior, não estão a ser jogados.

**10. Este jogo deve ser melhorado.**

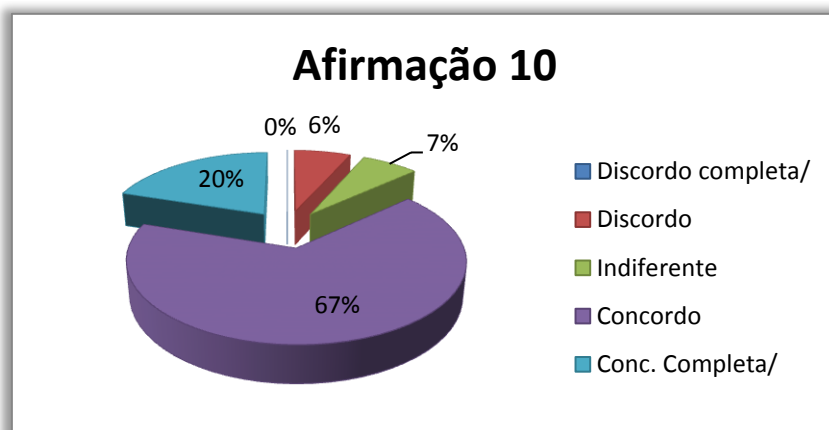


Gráfico 10 – Gráfico circular para a 10ª afirmação

A afirmação 10 é bastante relevante, pois permite avaliar se existe qualidade na base do jogo, de modo a dar continuidade à implementação deste. Da totalidade dos alunos, oitenta e sete por cento (87%) afirmou que este jogo deve ser melhorado, no sentido de implementar os cenários em falta.

A próxima afirmação não foi bem conseguida na forma como foi escrita.

De modo a evitar duplas interpretações na altura do preenchimento do questionário, foi explicado para todos os alunos o que se pretendia dizer.

**11. A aprendizagem do Método 5S deve ser feita num formato diferente.**

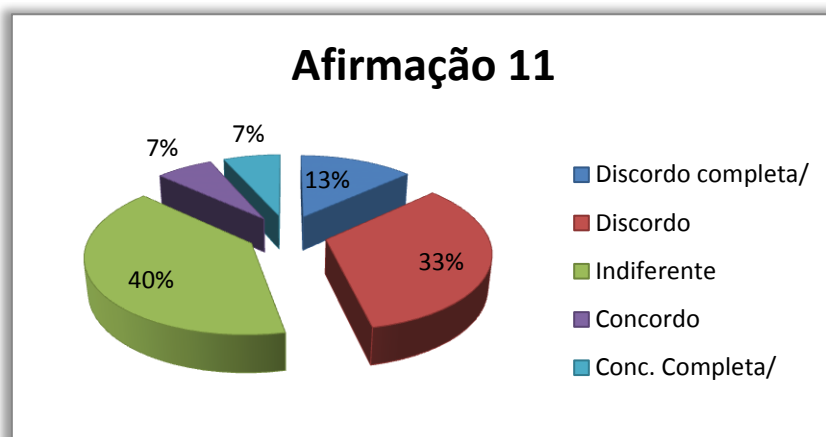


Gráfico 11 – Gráfico circular para a 11ª afirmação

Deve entender-se por “formato diferente” se a aprendizagem destes conteúdos deve ser feita de uma forma diferente da que experimentaram.

Esta afirmação parece tê-los dividido. Apesar da diferença entre aqueles que têm uma forte opinião sobre esta (catorze por cento (14%) concordam e quarenta e seis por cento (46%) discordam), houve quarenta por cento (40%) dos alunos que se mostraram indiferentes perante este ponto. A razão de tanta indiferença pode estar centrada no facto de os alunos já terem conhecimentos na matéria leccionada. A pergunta seguinte pretende fazer o despiste para esse caso. Assim sendo, pode concluir-se que quarenta e seis por cento (46%) das pessoas inquiridas são da opinião que este formato é bom para o ensino destes conteúdos.

**12. Este tipo de ferramentas é uma mais-valia para a formação.**

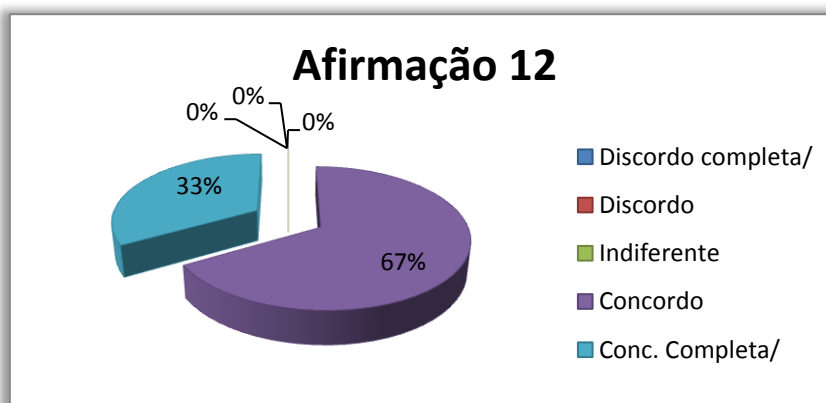


Gráfico 12 – Gráfico circular para a 12ª afirmação

Para esta afirmação não houve qualquer tipo de dúvidas. Como se tem vindo a afirmar, jogos como este são uma mais-valia no auxílio da formação. Existe um consenso generalizado de que

são necessárias mais ferramentas deste tipo para dinamizar a aprendizagem e treino de conteúdos de *Lean*.

### 13. *Este jogo é muito complexo.*

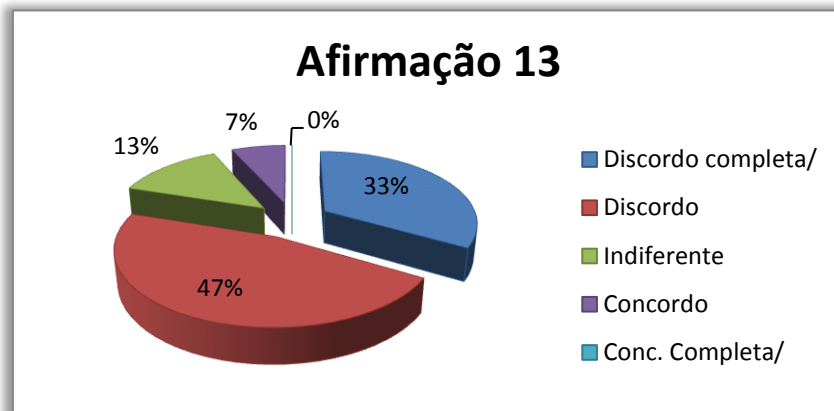


Gráfico 13 – Gráfico circular para a 13ª afirmação

Concluindo o questionário de avaliação, a afirmação 13 servia de controlo para avaliar se é necessário complicar ou simplificar o cenário inicial proposto. Caso este fosse muito complexo e fosse necessário o jogador estar familiarizado com conceitos de contabilidade, teria de ser simplificado.

Como se pode concluir pelo gráfico, quarenta e sete por cento (47%) dos alunos não concordam com a afirmação e trinta e três por cento (33%) não concordam completamente. Para satisfazer este segundo grupo de alunos, pode optar-se por torná-lo mais complexo. Contudo, a maioria dos alunos afirma que o cenário tem a complexidade ideal.

#### 6.5.1. Comentários feitos verbalmente

Enquanto decorriam as avaliações, os alunos, que iam acabando e entregando a sua avaliação, sugeriam verbalmente algumas ideias. Um aluno sugeriu que em vez das três fases que o jogo tem, usar cinco, uma para cada fase do Método 5S, de modo a que quem jogar interiorize melhor o conceito deste.

Esta ideia já tinha sido debatida na fase de concepção. Como foi difícil encontrar um objectivo do jogo para certos processos do Método 5S, de modo a que este fosse coeso, optou-se por usar apenas três. Contudo, aprecia-se o facto de os alunos terem revelado interesse em ajudar a melhorá-lo.

Vários alunos comentaram, ainda, que acharam o jogo bom, que tem uma ideia interessante e bem conseguida.

### 6.5.2. Apontamentos escritos nas folhas do questionário

Um aluno sugeriu que se use mais imagens, para explicar o Método 5S na área de “Materiais de Estudo” do jogo. Este é um ponto válido, que será tido em conta no trabalho futuro deste projecto.

Um outro aluno apontou pequenos problemas que encontrou no jogo, como: um documento não tinha nome e uma pasta não dava para apagar. Este facto pode dever-se, como foi referido, às actualizações recentes do *Flash*.

Por explicar ficou o facto de que mais nenhum aluno, que tenha experimentado o jogo, se tenha deparado com estes problemas. Feita uma análise ao jogo guardado deste aluno, conclui-se que chegou ao fim, jogou mais de uma vez e que estes pequenos problemas apontados não influenciaram a sua jogabilidade.

### 6.5.3. Avaliação final

No geral, houve um bom ambiente na sala durante a avaliação, quatro ou mais jogadores jogaram o cenário mais de uma vez. Todos os jogadores acabaram o jogo. Não houve nenhum incidente a registar.

Resumindo, os resultados obtidos superaram as expectativas, havendo uma grande aceitação do jogo pelo grupo que o avaliou.

Como previsto, a maior parte dos alunos inquiridos não conhecem outros jogos digitais sobre *Lean Manufacturing*, o que justifica que a pesquisa feita antes da implementação do jogo só tenha encontrado um jogo (o jogo das bicicletas) com estas características.



## 7. Conclusões

*“Enough research will tend to support your conclusions.”*

**Arthur Bloch**

Neste capítulo, serão abordados alguns tópicos, como a discussão sobre o sucesso do projecto, revisão dos objectivos, auto-avaliação do trabalho executado, dificuldades sentidas, resultados, desenvolvimentos futuros, considerações finais e encerramento, realizando, assim, a sua conclusão.

### 7.1. Retrospectiva do trabalho desenvolvido

Esta dissertação propôs um Jogo Sérico que permite pôr em prática o Método 5S, servindo ainda como ferramenta motivadora, para que quem o jogue aprofunde o seu conhecimento na matéria.

Foi feita uma avaliação do estado actual dos Jogos Séricos para esta área. Tendo em conta as necessidades e requisitos obrigatórios, foi elaborado e implementado um jogo para dar resposta a esta situação.

Após a avaliação realizada ao cenário implementado, pôde concluir-se que esta é uma ferramenta motivadora para a aprendizagem dos conteúdos propostos. Face a este resultado, implementando os restantes cenários, ter-se-á uma solução completa com probabilidade de sucesso no mercado.

### 7.2. Revisão do Objectivos e Contribuições da Dissertação

Esta dissertação tinha como objectivos específicos os seguintes:

- Fazer um levantamento dos pontos fortes dos Jogos Séricos com mais sucesso no mercado. É prioritário analisar os jogos existentes, qual a sua função, a melhor forma de transmitir informação, como avaliar o desempenho do jogador, entre outros aspectos;
- Identificar as necessidades actuais, para as quais o *software* existente não dá resposta;
- Criar e validar uma ferramenta de treino em condições seguras que incentive o aluno a melhorar numa situação simulada de produção, formando-o nos conceitos *Lean* e preparando-o para realizar melhor as suas funções.

A pesquisa realizada permitiu obter um panorama da situação actual dos Jogos Séricos direccionados para os conteúdos de *Lean Manufacturing*. Esta evidenciou que seria difícil o levantamento dos pontos fortes e fracos de cada Jogo Sérico encontrado, uma vez que apenas se

encontrou um jogo. Os alunos, que realizaram a avaliação, também não conheciam outros jogos na área, o que confirmou a ausência sentida e explicada no capítulo da introdução.

Foram identificadas as necessidades actuais e, em resposta, criou-se uma ferramenta que incentiva a aprendizagem do Método 5S, dinamizando, assim, o ensino de conteúdos de *Lean Manufacturing*.

### 7.3. Auto-avaliação do trabalho realizado

Após muito trabalho e dedicação ao projecto, pode dizer-se que, apesar de existirem melhoramentos a serem feitos, como em qualquer outro projecto, se obteve um bom jogo que servirá para quem quiser aprender e adquirir conhecimentos sobre os conteúdos que este aborda.

Foram tomadas decisões difíceis que levaram o seu tempo a ser ponderadas, correram-se riscos ao escolher as plataformas do projecto, mas, no final, obteve-se um jogo e uma boa base para a criação dos restantes cenários projectados.

A pesquisa para a realização desta dissertação tornou-se uma mais-valia para a actualização e enriquecimento dos meus conhecimentos nesta área profissional.

### 7.4. Dificuldades Sentidas

Abordam-se, agora, as dificuldades sentidas durante este projecto. Um aspecto que o influenciou foi não existirem, no mercado, jogos do género que fornecessem uma referência do que existe e das suas qualidades e limitações.

Havendo jogos de referência, é simples analisar as suas qualidades, sendo estas requisitos para o jogo que se pretende desenvolver. Estudando as limitações destes mesmos jogos de referência, podem-se tentar implementar funcionalidades, para que o jogo a desenvolver não possua essas limitações. Assim, adquire-se uma vantagem no mercado, apresentado um jogo superior.

Outra dificuldade sentida surgiu no momento de escolha da ferramenta de implementação do jogo. Como já foi referido, não havendo Jogos Sérios para ensinar conteúdos de *Lean*, como é espectável, também não existem ferramentas que permitam a implementação destes. Não havendo uma solução clara para este problema, teve de haver uma ponderação séria.

### 7.5. Resultados

Deste projecto resultou um Jogo Sérico que ensina o Método 5S, através de um cenário que simula o ambiente de um computador.

Outro benefício deste projecto é a informação que se reuniu para esta dissertação. Poderá inspirar ou tirar dúvidas a outras pessoas, que pretendam criar jogos ou investir nesta área.

Este trabalho pode ser, ainda, mais uma contribuição para acabar com o preconceito de que os jogos distraem o sujeito da aprendizagem e apenas servem para diversão.

A potencialidade dos jogos vai para além do entretenimento, são ótimas ferramentas motivadoras para o progresso e crescimentos de aptidões nos jogadores.

## 7.6. Desenvolvimentos futuros

No que diz respeito a trabalhos futuros, agora que se obteve um bom *feedback* e a confirmação que a ferramenta proposta responde às necessidades sentidas, resta implementar os restantes cenários, já delineados, completando assim o jogo.

## 7.7. Considerações finais

O jogo está *online* através do *site* [duartegomes.com/metodo5S](http://duartegomes.com/metodo5S).

Podem dar opiniões e sugestões para o *correio electrónico* [metodo5s@yahoo.com](mailto:metodo5s@yahoo.com).

## 7.8. Enceramento

Com este trabalho, pretende-se colaborar na melhoria do desempenho dos profissionais, tendo como objectivo um aumento na produtividade das empresas, através da utilização de Jogos Sérios na formação dos seus colaboradores.

A execução desta dissertação, para além de ter proporcionado um conhecimento mais aprofundado nas áreas da Informática e Gestão, promoveu uma maior capacidade de realização para futuros projectos.

O precioso contributo do Prof. Doutor Vaz de Carvalho e do Eng. Pereira Lopes foi uma mais-valia para a realização deste trabalho e para o meu enriquecimento profissional e pessoal.



## 8. Referencias Bibliográficas

Este capítulo está reservado para as referências bibliográficas consultadas para a realização do projecto e para a escrita desta dissertação.

### Bibliografia

- Lean manufacturing*. (12 de Setembro de 2011). Obtido em 18 de Setembro de 2011, de wikipedia: [http://en.wikipedia.org/wiki/Lean\\_manufacturing](http://en.wikipedia.org/wiki/Lean_manufacturing)
- Serious game*. (12 de Setembro de 2011). Obtido em 19 de Setembro de 2011, de wikipedia: [http://en.wikipedia.org/wiki/Serious\\_game](http://en.wikipedia.org/wiki/Serious_game)
- ThroughPut Solutions*. (18 de Fevereiro de 2011). Obtido em 4 de Outubro de 2011, de Lean Manufacturing Definitio - As We See It: <http://tpslean.com/leanwaystoimp/leanmanufacturing.htm>
- Bialek, R., L. Duffy, G., & W. Moran, J. (2009). *Process Analysis Tools - Spaghetti Diagram*. Obtido em 16 de Fevereiro de 2011, de asq.org: <http://asq.org/learn-about-quality/process-analysis-tools/overview/spaghetti-diagram.html> 5/10/2010
- Chapman, N., & Chapman, J. (2001). *Digital Media Tools*. John Wiley and Sons Ltd.
- Chapman, N., Chapman, J., & Chapman, J. (2000). *Digital Multimedia*. John Wiley and Sons Ltd.
- Chen, S., & Michael, D. (19 de Outubro de 2005). *Proof of Learning: Assessment in Serious Games*. CMP Media Inc.
- Derryberry, A. (2007). *Serious games: online games for learning*. San Jose, CA, USA: Adobe.
- Djaouti, D., Alvarez, J., & Jessel, J.-P. (2010). *Can "Gaming 2.0" Help Design "Serious Games"? A Comparative Study*. Toulouse, France: IRIT, University of Toulouse.
- Eck, R. V. (2006). *Digital Game-Based Learning: It's Not Just the Digital Natives...* *EDUCAUSE Review*, vol. 41, no. 2 (March/April).
- England, E., & Finney, A. (2001). *Managing Multimedia: Project Management for Web and Convergent Media: Book 1. People and Processes (3rd Edition)*. Benjamin-Cummings Publishing Company.
- Gouveia, D. (2010). *An Analysis and Comparison of Some Existing Graphic Adventure Game Engines*.
- Gunter, G., Kenny, R., & Vick, E. (2006). *A Case for a Formal Design Paradigm for Serious Games*. Orlando, FL, USA: University of Central Florida College of Education, CREATE - Center for Research in Education, Arts, Technology, and Entertainment.
- Manzano, C. (5 de Maio de 2007). *Serious Game*. Obtido em 17 de Fevereiro de 2011, de <http://www.seriousgame.com.br/>: <http://www.seriousgame.com.br/>

- Martens, I., Colpaert, J., De Boeck, L., Vaz de Carvalho, C., Ávila, P., Pereira Lopes, M., & Bastos, J. (2010). *Lean Learning Academy (LLA)*. Porto, Portugal.
- McGonigal, J. (2010, Março 17). *Gaming can make a better world*. Retrieved Outubro 4, 2011, from TED.com:  
[http://www.ted.com/talks/jane\\_mcgonigal\\_gaming\\_can\\_make\\_a\\_better\\_world.html](http://www.ted.com/talks/jane_mcgonigal_gaming_can_make_a_better_world.html)
- Mcmeans, C. (16 de Fevereiro de 2011). *Advantages Of Using 5s Simulation Games*. Obtido em 4 de Outubro de 2011, de articlesnatch.com:  
<http://www.articlesnatch.com/Article/Advantages-Of-Using-5s-Simulation-Games/1984525>
- Mcmeans, C. (16 de Fevereiro de 2011). *Advantages Of Using 5s Simulation Games*. Obtido em 4 de Outubro de 2011, de articlesnatch.com:  
<http://www.articlesnatch.com/Article/Advantages-Of-Using-5s-Simulation-Games/1984525>
- Mcmeans, C. (16 de Fevereiro de 2011). *The Benefits of Using Lean Simulations*. Obtido em 4 de Outubro de 2011, de <http://ezinearticles.com>: <http://ezinearticles.com/?The-Benefits-of-Using-Lean-Simulations&id=5983494>
- Mcmeans, C. (16 de Fevereiro de 3022). *Effective Ways of Teaching 5S*. Obtido em 4 de Outubro de 2011, de [ezinearticles.com](http://ezinearticles.com): <http://ezinearticles.com/?Effective-Ways-of-Teaching-5S&id=5943501>
- MSU - *Serious Game Design*. (s.d.). Obtido em 19 de Setembro de 2011, de <http://seriousgames.msu.edu/>
- Pather, S., & Wang, H. (2009). *The Use of Simulation Games to Enhance the Understanding of Manufacturing Processes*. Toowoomba, Australia: University of Southern Queensland.
- Peterson, J., & Roland, S. (1998). *The 5S Pocket Guide, Productivity*. Portland, USA: Productivity Press.
- Prensky, M. (2003, Outubro). Digital Game-Based Learning. In *ACM Computers in Entertainment, Vol. 1, No. 1*. New York, NY, USA: Games2train.
- Serious Game Initiative*. (s.d.). Obtido em 19 de Setembro de 2011, de <http://www.seriousgames.org/>
- Serious Games classification*. (s.d.). Obtido em 19 de Setembro de 2011, de [serious.gameclassification](http://serious.gameclassification.com/): <http://serious.gameclassification.com/>
- Serious Games Interactive*. (s.d.). Obtido em 22 de Setembro de 2011, de <http://www.seriousgames.dk/node/511>
- SIGI - *Serious Game Institute*. (s.d.). Obtido em 19 de Setembro de 2011, de [seriousgamesinstitute](http://www.seriousgamesinstitute.co.uk/): <http://www.seriousgamesinstitute.co.uk/>

- Strategos.inc. (2008). Origins & History of Lean Manufacturing. In Strategos.inc, *Just in Time, Toyota Productions & Lean Manufacturing*. Wyandotte, Kansas City, Missouri, USA: Strategos.inc.
- The Serious Games Blog* . (s.d.). Obtido em 19 de Setembro de 2011, de seriousgames.ning.com  
- The Intersection Of Serious Games, Virtual Worlds, Mobile Tech, AI, CAD/CAM, The Real-Time Web, Conversational Marketing and Cloud Computing:  
<http://seriousgamesblog.blogspot.com/>
- The Serious Games Showcase & Challenge*. (s.d.). Obtido em 19 de Setembro de 2011, de  
<http://www.sgschallenge.com/challenge.shtml>
- Tsukumo, A. e. (1997). Qualidade de Software: Visões de Produto e Processo de Software. *VIII CITS - CONFERÊNCIA INTERNACIONAL DE TECNOLOGIA DE SOFTWARE: QUALIDADE DE SOFTWARE*. Curitiba, Paraná, Brasil: Anais.
- Vaughan, T. (2001). *Multimedia: Making It Work: A Comprehensive Guide to Creating Stunning Multimedia from Start to Finish*. Osborne McGraw-Hill.
- Wang, L. P. (2005). *Computer-Simulation-Assisted Lean Manufacturing Training*. Provo, Utah, USA: Brigham Young University.



## 9. Anexos

Nos anexos podem ser encontrados os documentos que auxiliam a compreensão desta dissertação.

### 9.1. Anexos A - Diagramas Use-Case dos cenários

Cenário 1

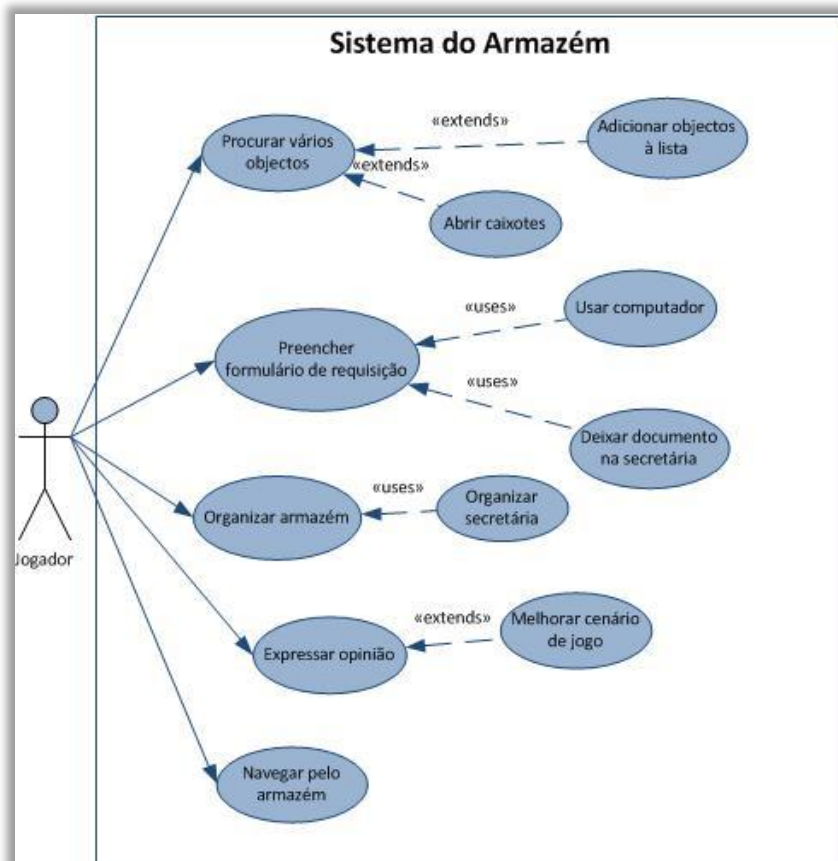


Ilustração 55 – Diagrama Use-Case do cenário 1

Cenário 2

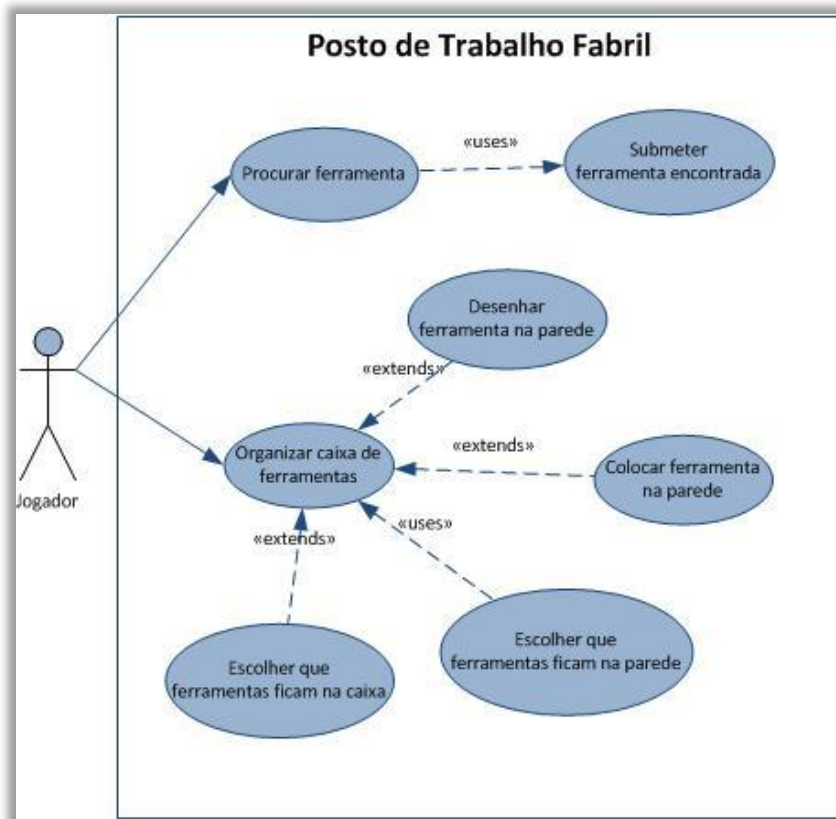


Ilustração 56 – Diagrama Use-Case do cenário 2

Cenário 3

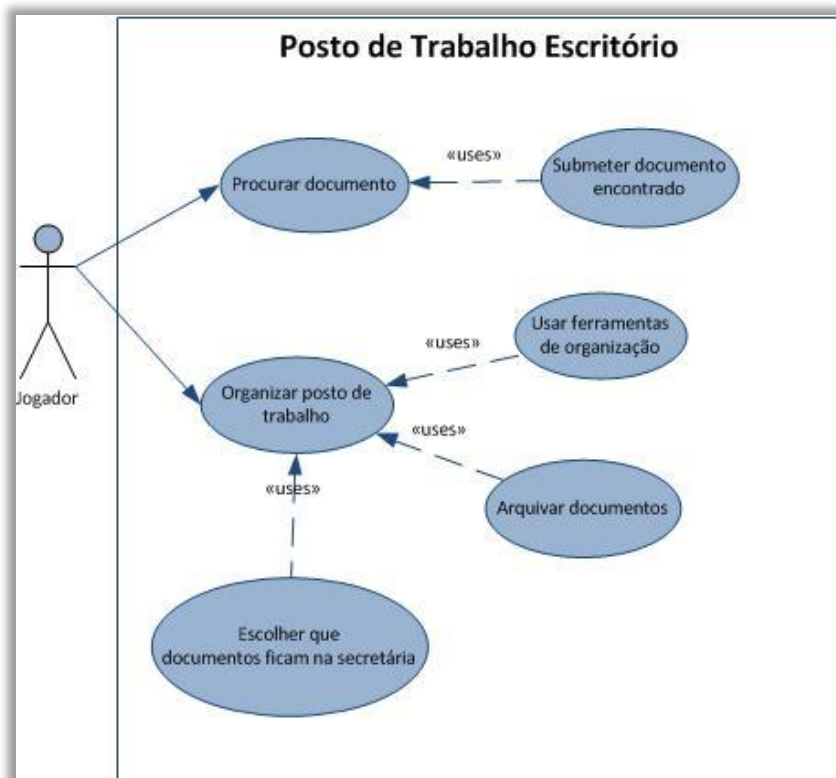


Ilustração 57 – Diagrama Use-Case do cenário 3

Cenário 4

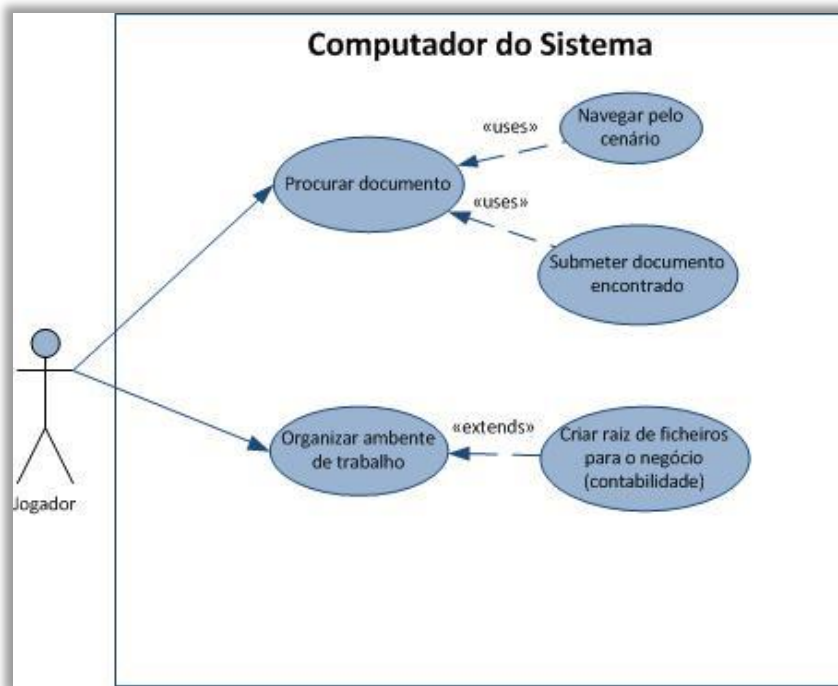


Ilustração 58 – Diagrama Use-Case do cenário 4

## 9.2. Anexos B – Questionário de Avaliação

*Jogos Sérios na Gestão – LLA – Método 5S*

Questionário de Avaliação					
Afirmações \ Respostas	Discordo Completamente	Discordo	Indiferente	Concordo	Concordo Completamente
1 O jogo é intuitivo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2 O cenário proposto é realista.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3 O jogo permitiu-me compreender melhor o Método 5S.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4 Este jogo motiva a aprendizagem.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5 Este tipo de abordagem deve ser aplicado a outros tópicos do <i>Lean</i> .	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6 A informação fornecida no jogo tem qualidade.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7 A informação fornecida no jogo tem a profundidade adequada.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8 Tenho conhecimento da existência de outro tipo de jogos para o Método 5S.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9 Já joguei outro tipo de jogos sobre <i>Lean</i> .	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10 Este jogo deve ser melhorado.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11 A aprendizagem do Método 5S deve ser feita num formato diferente.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
12 Este tipo de ferramentas é uma mais-valia para a formação.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
13 Este jogo é muito complexo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Ilustração 59 – Questionário de avaliação preenchido pelos alunos

Questionário para avaliação do Jogo Método 5S realizado no âmbito da tese de Mestrado de Sistemas Gráficos e Multimédia no ISEP do aluno Duarte Sousa Ferreira Gomes número 1050460.

### 9.3. Anexos C – Slides da apresentação da avaliação do jogo

<h2>AVALIAÇÃO DO JOGO SÉRIO “MÉTODO 5S”</h2> <hr/> <p>Tese de Mestrado em Sistemas Gráficos e Multimédia Duarte Sousa Ferreira Gomes</p>	<h3>Agenda</h3> <ul style="list-style-type: none"><li>• Introdução;</li><li>• Algumas considerações;</li><li>• Link do Jogo;</li><li>• Ecrãs de Jogo;</li><li>• Resumo dos Objectivos;</li><li>• Avaliação;</li><li>• Sugestões;</li><li>• Questões;</li><li>• Jogar;</li><li>• Agradecimento.</li></ul>
<h3>Introdução</h3> <ul style="list-style-type: none"><li>• A sessão de hoje pretende avaliar o protótipo criado para o quarto cenário do <i>Jogo Sério</i>, desenvolvido no âmbito da Tese de Mestrado em Sistemas Gráficos e Multimédia.</li><li>• <i>Jogo Sério</i> cujo objectivo principal é dinamizar a aprendizagem de conteúdos <i>Lean Manufacturing</i>, mais propriamente o <i>Método 5S</i>, usando uma abordagem prática.</li><li>• O tema do cenário que se vai avaliar é o ambiente de trabalho de um computador e a sua organização.</li></ul>	<h3>Algumas considerações</h3> <ul style="list-style-type: none"><li>• Este cenário tem apenas 5 meses de implementação (em conjunto com a escrita do relatório).</li><li>• Problemas ou falhas podem surgir durante o jogo. Caso aconteça, por favor, informe qual operação estava a tentar realizar no momento da falha e reabra o site.</li><li>• Dependendo dos serviços da Internet do ISEP, pode haver alguma anomalia não prevista.</li></ul>
<h3>Link para o Jogo</h3> <ul style="list-style-type: none"><li>• <a href="http://duartegomes.com/metodo5S/">http://duartegomes.com/metodo5S/</a></li></ul> 	<h3>Jogo: Menu Principal / Sair</h3> 

Ilustração 60 – Apresentação para avaliação do jogo (1/5)

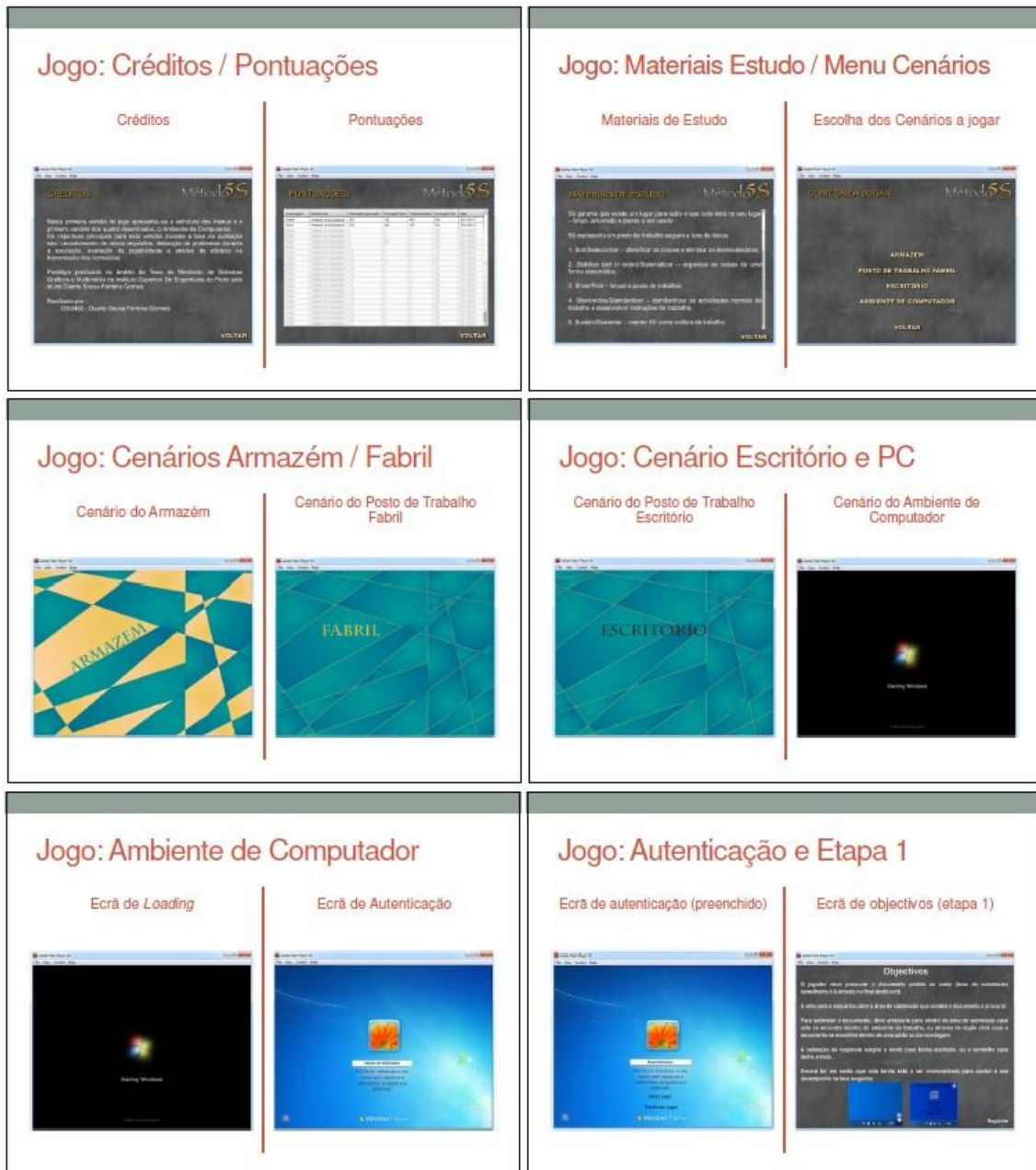


Ilustração 61 – Apresentação para avaliação do jogo (2/5)

<h3>Jogo: Etapa 1 Procurar Documento</h3> <p><b>Ecrã de Jogo (etapa 1)</b></p>  <p><b>Caixa de Informações</b></p> 	<h3>Jogo: Elementos</h3> <p><b>Elementos do cenário</b></p>  <p><b>Informações</b></p> <ul style="list-style-type: none"> <li>• Elementos presentes no cenário: Atalhos, Reciclagem, Documentos, Pastas e Zonas;</li> <li>• Podem ser abertos para consulta do seu conteúdo (excepção dos atalhos);</li> <li>• Só podem ser submetidos Documentos;</li> </ul>
<h3>Jogo: Funcionalidades</h3> <p><b>Abrir Documentos</b></p>  <p><b>Abrir Pastas</b></p> 	<h3>Jogo: Funcionalidades</h3> <p><b>Abrir Reciclagem</b></p>  <p><b>Abrir Submenu (Tecla M ou Botão Direito do Rato)</b></p> 
<h3>Jogo: Funcionalidades</h3> <p><b>Criar Pastas</b></p>  <p><b>Criar Zonas (processo)</b></p> 	<h3>Jogo: Resultados / Gráfico</h3> <p><b>Ecrã de resultados da etapa 1</b></p>  <p><b>Diagrama de spaghetti</b></p> 

Ilustração 62 – Apresentação para avaliação do jogo (3/5)









<h3>Jogo: Logs / Objectivos Etapa 2</h3> <p>Ecrã de leitura dos registos</p>  <p>Ecrã de objectivos (etapa 2)</p> 	<h3>Jogo: Organizar e Submeter</h3> <p>Ecrã de jogo (etapa 2)</p>  <p>Ecrã que confirma que foi guardada a organização proposta</p> 
<h3>Jogo: Etapa 3 Procurar Documento</h3> <p>Ecrã de objectivos (etapa 3)</p>  <p>Ecrã de jogo (etapa 3)</p> 	<h3>Jogo: Resultados Finais / Menu</h3> <p>Ecrã de Resultados Finais</p>  <p>Voltar ao menu de escolha dos cenários</p> 
<h3>Jogo: Resumo dos Objectivos</h3> <ul style="list-style-type: none"> <li>• Resumo do objectivo de cada etapa do jogo:</li> <li>• Etapa 1 – Encontrar o Documento pedido (pontos serão atribuídos, se for organizado);</li> <li>• Etapa 2 – Organizar o cenário apresentado;</li> <li>• Etapa 3 – Encontrar o Documento pedido agora no cenário organizado.</li> </ul>	<h3>Avaliação</h3> <ul style="list-style-type: none"> <li>• Preenchimento de um questionário.</li> <li>• Terão de responder a cada das afirmações, assinalando, numa escala de 0 (mínimo) a 5 (máximo), o quanto concordam com estas.</li> <li>• Opções disponíveis: <ul style="list-style-type: none"> <li>• 0 – Discordo Completamente;</li> <li>• 1 – Discordo;</li> <li>• 3 – Indiferente;</li> <li>• 4 – Concordo;</li> <li>• 5 – Concordo Plenamente.</li> </ul> </li> </ul>

Ilustração 63 – Apresentação para avaliação do jogo (4/5)

<h3>Sugestões</h3> <ul style="list-style-type: none"><li>• O vosso feedback é bastante importante para mim, uma vez que possuem experiência de trabalho neste ramo.</li><li>• Peço, a quem estiver disposto a ajudar, que usem a parte de trás do questionário para deixar sugestões.</li><li>• Caso pretendam obter informações adicionais sobre o projecto, podem entrar em contacto através de: <a href="mailto:duartegomes@msn.com">duartegomes@msn.com</a></li></ul>	<h3>Questões</h3> <ul style="list-style-type: none"><li>• Dúvidas e questões antes de começar a avaliação?</li></ul>
<h3>Jogar</h3> <ul style="list-style-type: none"><li>• Podem começar a jogar! Divirtam-se!</li></ul>	<h3>Agradecimento</h3> <ul style="list-style-type: none"><li>• Muito obrigado pela vossa participação! ☺</li><li>• <a href="mailto:DuarteGomes@msn.com">DuarteGomes@msn.com</a></li></ul>

Ilustração 64 – Apresentação para avaliação do jogo (5/5)