FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Ludificação de um sistema de gestão documental e de processos

André Filipe Barbosa Morais

VERSÃO FINAL



Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: André Monteiro de Oliveira Restivo

Co-orientador: Telma Cristina Marques Salgueiro

21 de Julho de 2018

© André Filipe Barbosa Morais, 2018

Resumo

Esta dissertação consistiu na elaboração e implementação de uma solução ludificada para um sistema de gestão documental e de processos, mais concretamente, o *iPortalDoc*. Este sistema tem como destinatários um vasto conjunto de empresas, de diferentes áreas, permitindo a gestão dos seus documentos e tarefas, com recurso a *workflows*. Cada documento está associado a um *workflow* que pode conter várias ações a serem realizadas pelos utilizadores.

Inicialmente foram estudados os conceitos teóricos relacionados com o tema *ludificação*, nomeadamente a motivação, os tipos de jogador e os elementos do jogo. Esse estudo envolveu também o processo de design baseado no jogador, que foi a base do fluxo seguido por esta dissertação, e ainda a estrutura Octalysis, que permite analisar e construir estratégias em torno das diversas características de um jogo. Nesta revisão bibliográfica foram ainda descritos os conceitos gerais sobre sistemas de gestão documental e de processos, terminando esta fase inicial com a apresentação de alguns trabalhos relacionados.

Foi também analisada a proposta de solução anteriormente apresentada para o *iPortalDoc*, que visava, essencialmente, o auxílio aos utilizadores em caso de dúvidas. De facto, a complexidade do sistema e a falta de *feedback* é um dos problemas identificados ao sistema, tendo sido um fator a ter em conta na projeção do sistema *ludificado*. Após uma análise ao sistema, concluiu-se também que a página de estatísticas necessitava de uma reformulação, tendo esta sido um dos focos desta dissertação. Para além disso, a solução foi projetada com o objetivo de aumentar a produtividade das respetivas empresas.

A solução proposta contemplou três área distintas: um conjunto de *feedbacks*; uma área de jogo; e uma nova página de estatísticas. O conjunto de *feedbacks* permite ajudar o utilizador em diversas situações, com a apresentação de informação na interface do sistema. A área de jogo consiste numa interface do *iPortalDoc*, em que cada utilizador poderá consultar a sua pontuação, o seu nível, os emblemas conquistados, bem como a atividade dos seus colegas. A aquisição de pontos e de emblemas acontece após a realização de determinadas ações ou *workflows*. A solução engloba também uma interface de configuração associada à área de jogo, permitindo a configuração dos vários níveis e da distribuição de pontos por parte do administrador. Por fim, a nova página de estatísticas permite apresentar os dados com vários níveis de detalhe, de uma forma mais fluída, e através de uma interface elegante e responsiva.

A implementação da solução não foi totalmente realizada, não tendo sido possível validar as alterações efetuadas num sistema de maiores dimensões e num contexto real. Apesar disso, foram apresentados os dados a serem recolhidos, antes e após a implementação da solução, bem como um inquérito a ser respondido pelos utilizadores, de forma a averiguar se de facto as alterações efetuadas terão o impacto desejado.

Abstract

This dissertation consisted of the development and implementation of a *gamification* solution for a document and process management system, in particular, the *iPortalDoc*. This system is intended for a wide range of companies from different areas, allowing the management of their documents and tasks, based on workflows. Each document is associated with a workflow that can contain several actions to be performed by the users.

Initially, the theoretical concepts related to the *gamification* were studied, namely the motivation, the player types and game elements. This study also involved the player-based design process, which was the basis of the flow followed by this dissertation, and also the Octalysis structure, which allows us to analyze and build strategies around the various characteristics of a game. In this state of the art review, the general concepts about document and process management systems were described. This initial phase ended with the presentation of some related works.

The previously proposed solution for *iPortalDoc* aimed at helping users in case of doubts. The complexity of the system and the lack of feedback were some of the problems identified in the system, being a factor to be taken into account in the development of the *gamification* solution. After an analysis of the system, it was also concluded that the statistics page needed a reformulation, and this was one of the focuses of this dissertation. In addition, the solution was designed to increase the productivity of the companies.

The proposed solution has three distinct areas: a set of feedbacks; a game area; and a new statistics page. The set of feedbacks helps the user in various situations with the presentation of information in the system interface. The game area consists of an *iPortalDoc* interface where each user can check his score, his level, the badges earned, as well as the activity of his colleagues. The acquisition of points and badges takes place after the performance of certain actions or workflows. The solution also includes a configuration interface associated with the game area, allowing the configuration of the various levels and the distribution of points by the administrator. Finally, the new statistics page allows you to present data in multiple levels of detail, in a more fluid way, and through an elegant and responsive interface.

The implementation of the solution was not fully implemented, and it was not possible to validate the changes made in a larger system and in a real context. Nevertheless, the data to be collected before and after the implementation of the solution was presented, as well as a questionnaire to be answered by the users in order to verify if the changes made will have the desired impact.

Agradecimentos

Em primeiro lugar, quero agradecer aos meus pais pelo carinho e dedicação que tiveram ao longo destes anos, e que permitiram que me tornasse na pessoa que sou hoje. Toda a minha família merece o meu agradecimento, mas em particular a minha avó, que infelizmente já não se encontra entre nós, mas que, tal como os meus pais, sempre se preocupou em dar-me as melhores condições possíveis para a construção do meu futuro. Estarei-lhes eternamente grato por isso.

Um agradecimento particular ao professor André Restivo, por aceitar me orientar ao longo desta dissertação, mas acima de tudo, por ter sido a pessoa responsável pelo meu interesse na área das aplicações Web. Não podia faltar um agradecimento à IPBRICK SA, pela oportunidade de desenvolver esta dissertação, bem como a todos os seus colaboradores, que sempre me auxiliaram em todas as minhas questões. Um agradecimento especial para o João Castro, que é, sem dúvida, a melhor pessoa que alguém poderá algum dia ter como colega do lado. Quero agradecer também a todos os meus colegas e amigos, que tornaram estes últimos cinco anos inesquecíveis, estando sempre presentes, nos bons e nos maus momentos.

Por fim, mas sempre com um lugar especial no meu coração, um agradecimento à minha namorada, Juliana Ferreira, pelo seu carinho e atenção durante esta fase importante da minha vida, e pela simbiose criada entre ambos, levando a que me sinta completo e um felizardo todos os dias que passo ao seu lado.

André Morais

"The further you get into technology, the further you go into gaming. That's the general rule."

Nick Johnson

Conteúdo

1 Introdução **1**
- 1.1 Contexto . . . 1
- 1.2 Motivação . . . 2
- 1.3 Objetivos . . . 2
- 1.4 Estrutura do documento . . . 2

2 Revisão Bibliográfica **5**
- 2.1 Ludificação . . . 5
  - 2.1.1 Motivação . . . 6
  - 2.1.2 Tipos de jogador . . . 8
  - 2.1.3 Elementos de jogo . . . 9
  - 2.1.4 Processo de design baseado no jogador . . . 13
  - 2.1.5 Octalysis . . . 18
- 2.2 Sistemas de gestão documental e de processos . . . 21
  - 2.2.1 iPortalDoc . . . 23
- 2.3 Trabalho relacionado . . . 26
  - 2.3.1 Microsoft's Communicate Hope . . . 27
  - 2.3.2 Kaplan University . . . 27
  - 2.3.3 Bluewolf . . . 27
  - 2.3.4 LiveOps . . . 28
- 2.4 Sumário . . . 28

3 Problema **29**
- 3.1 Definição do problema . . . 29
- 3.2 Descrição da solução *ludificada* anteriormente proposta . . . 33
- 3.3 Sumário . . . 36

4 Solução **37**
- 4.1 Processo de design baseado no jogador . . . 37
  - 4.1.1 Resultados do negócio e métricas de sucesso . . . 37
  - 4.1.2 Público alvo . . . 38
  - 4.1.3 Objetivos do jogador . . . 39
  - 4.1.4 Modelo de envolvimento do jogador . . . 40
  - 4.1.5 Espaço de jogo e planeamento da jornada . . . 41
  - 4.1.6 Economia do jogo . . . 41
  - 4.1.7 Teste e iteração . . . 41
- 4.2 Análise à solução *ludificada* anteriormente proposta . . . 42
- 4.3 Proposta de solução . . . 42

4.3.1 Conjunto de *feedbacks* . . . 43
4.3.2 Área de jogo . . . 43
4.3.3 Página de estatísticas . . . 48
4.4 Comparação entre as soluções propostas . . . 52
4.5 Sumário . . . 53

5 Implementação e Resultados **55**
5.1 Conjunto de *feedbacks* . . . 56
5.2 Área de jogo . . . 58
5.3 Página de estatísticas . . . 70
5.4 Sumário . . . 77

6 Validações **79**
6.1 Métodos de *user experience research* . . . 79
6.2 Validações a efetuar à solução implementada . . . 82
6.2.1 Testes de usabilidade . . . 82
6.2.2 Inquérito . . . 83
6.3 Sumário . . . 84

7 Conclusões e Trabalho Futuro **85**
7.1 Satisfação dos Objetivos . . . 85
7.2 Trabalho Futuro . . . 86
7.2.1 Conjunto de *feedbacks* . . . 86
7.2.2 Área de jogo . . . 86
7.2.3 Página de estatísticas . . . 87

Referências **89**

Lista de Figuras

2.1 Enquadramento do conceito de *gamification* [6] 6
2.2 Contínuo da autodeterminação [15] 7
2.3 Tipos de jogadores segundo Bartle [10] 8
2.4 Hierarquia dos elementos de jogo [23] 11
2.5 Processo de design da experiência do jogador [3] 13
2.6 Objetivos comuns ao negócio e ao jogador/utilizador (adaptada de [3]) 14
2.7 Exemplo de um modelo de envolvimento do jogador [3] 15
2.8 Jornada do jogador [3] 16
2.9 Economia do jogo [3] 17
2.10 Estrutura de *ludificação* Octalysis [4] 19
2.11 Estrutura de *ludificação* Octalysis para os diferentes tipos de jogadores [4] 21
2.12 Exemplo de um *workflow*: aprovação de um documento [7] 23
2.13 Exemplo da interface geral do *iPortalDoc* 24
2.14 Exemplo da interface de pesquisa do *iPortalDoc* 25
2.15 Exemplo da interface do histórico do *iPortalDoc* 26
2.16 Exemplo da interface de email do *iPortalDoc* 26

3.1 Exemplo da interface de estatísticas de documentos do *iPortalDoc* 30
3.2 Exemplo do relatório gerado relativo às estatísticas de documentos do *iPortalDoc* 31
3.3 Exemplo da interface de estatísticas de workflows do *iPortalDoc* 32
3.4 Exemplo do relatório gerado relativo às estatísticas de workflows do *iPortalDoc* . 33
3.5 Exemplo da interface do centro de ajuda [11] 34
3.6 Exemplo da interface do painel de cooperação [11] 35

4.1 Diagrama de casos de uso para a área de jogo 44
4.2 *Mockup* da interface da área de jogo 47
4.3 *Mockup* da interface de configuração de níveis da área de jogo 47
4.4 *Mockup* da interface de configuração de pontos da área de jogo 48
4.5 *Mockup* da interface de configuração de pontos da área de jogo relativos às ações 49
4.6 Hierarquia de níveis para as estatísticas de *workflows* 50
4.7 *Mockup* da interface de estatísticas relativa a *workflows* 51

5.1 Estrutura de ficheiros do *iPortalDoc* 55
5.2 Exemplo de um passo de um *tour* implementado no *iPortalDoc* 57
5.3 Diagrama de classes para a primeira fase de implementação da área de jogo . . . 59
5.4 Interface da área de jogo após a primeira fase de implementação 59
5.5 *Popup* da interface da área de jogo para alteração do avatar 60
5.6 *Popup* da interface da área de jogo para alteração do *nickname* 61
5.7 Interface de configuração dos níveis da área de jogo 61

5.8 Diagrama de classes para a segunda fase de implementação da área de jogo . . . 64
5.9 Interface de configuração de pontos da área de jogo 66
5.10 Interface de configuração de pontos da área de jogo relativos às ações 67
5.11 Interface da área de jogo com conquistas visíveis 69
5.12 Fase inicial da interface da nova página de estatísticas do *iPortalDoc* 71
5.13 Interface da nova página de estatísticas durante a atualização de um gráfico . . . 75
5.14 Interface final da nova página de estatísticas 76

6.1 Enquadramento dos vários métodos de *user experience research* em função do tipo de dados (adaptada de [13]) 81

Lista de Tabelas

4.1 Comparação de características entre as soluções propostas 52

Abreviaturas e Símbolos

AJAX	*Asynchronous Javascript and XML*
B2B	*Business-to-Business*
CSS	*Cascading Style Sheets*
DMS	*Document Management System*
HTML	*Hypertext Markup Language*
HTTPS	*Hyper Text Transfer Protocol Secure*
IBM	*International Business Machines*
IDE	*Integrated Development Environment*
JSON	*JavaScript Object Notation*
MUD	*Multi-User Dungeon*
PBLs	*Points, Badges, and Leaderboards*
PDF	*Portable Document Format*
PL/pgSQL	*Procedural Language/PostgreSQL*
RGB	*Red, Green and Blue*
SDT	*Self-Determination Theory*
SMB	*Server Message Block*
SMTP	*Simple Mail Transfer Protocol*
SUS	*System Usability Scale*
UML	*Unified Modeling Language*
XML	*Extensible Markup Language*

Capítulo 1

Introdução

1.1 Contexto

O mundo desenvolveu-se no sentido em que as pessoas têm ao seu dispor uma vasta gama de alternativas para qualquer situação que se possa imaginar. Com tantas distrações, tornou-se fundamental captar a atenção das pessoas, para assim, envolvê-las num contexto específico. Foi com este objetivo que o termo *"gamification"* começou a surgir.

Ludificação ou *gamification* pode ser definida como *"the use of game design elements in non-game contexts to motivate and increase user activity and retention"* [6], que consiste na aplicação de elementos de jogo em diferentes contextos, para motivar e aumentar a atividade e a retenção de utilizadores. Esta estratégia pode ser aplicada em diversas áreas, como por exemplo na educação, em que o objetivo é aumentar o aproveitamento escolar do aluno; na saúde, em que se pretende que um paciente adote melhores hábitos de vida; no trabalho, para melhorar o rendimento dos empregados; e até no quotidiano das pessoas. Nesta dissertação, o foco recairá na *ludificação* de *frameworks* para um sistema de gestão documental e de processos.

A *ludificação* de um sistema deve ter em conta diversos aspetos, e seguir um fluxo de trabalho desde a planificação até ao desenvolvimento do design. É importante estar bem definido o que se vai *ludificar*, o porquê e quem são os utilizadores alvo. Muitas das soluções baseadas em *ludificação* falham, devido ao facto de não haver um processo claro e formal na sua elaboração [12]. Neste documento serão abordadas metodologias e estruturas que permitirão tomar as melhores decisões relativamente à estratégia de *ludificação* a implementar.

Um sistema de gestão documental e de processos é usado para gerir, armazenar e rastrear documentos eletrónicos, e para computar todos os processos de uma organização. Quase todas as empresas têm ou precisam destes sistemas, portanto, a aplicação de técnicas de *ludificação* nas suas interfaces irá motivar o trabalhador, e aumentar a sua produtividade e, por consequente, melhorar a competitividade da empresa.

O sistema de gestão documental e de processos que será estudado nesta dissertação é o *iPortalDoc*. É um software desenvolvido pela empresa IPBRICK SA e funciona como um serviço de

valor acrescentado para a Intranet, estando o mesmo integrado de forma nativa no Sistema Operativo IPBRICK. É um sistema com uma estratégia *business-to-business* (B2B), que tem como destinatários, um vasto grupo de empresas, de diferentes ramos. O *iPortalDoc* permite o registo, classificação e tratamento de todas as comunicações (emails, conversações de *Instant Messaging*, chamadas telefónicas e faxes), bem como o tratamento de todos os processos de uma organização.

1.2 Motivação

Tendo em conta a interação e dependência da aplicação com o utilizador final, pretende-se disponibilizar as melhores técnicas de *ludificação* às interfaces e funcionalidades mais utilizadas do *iPortalDoc*, de forma a motivar e facilitar o dia a dia destes utilizadores. Apesar da *ludificação* não ser algo inovador, no que diz respeito a sistemas de gestão documental e de processos, é algo não muito habitual e que, para os seus utilizadores, será uma mudança significativa relativamente aquilo que podem esperar do sistema. Esta mudança de paradigma implicará um maior envolvimento com um sistema, incutindo nas pessoas que o usam a ideia de que este evolui com o passar do tempo, e que se adapta às necessidades dos utilizadores.

As melhorias, por mínimas que sejam, da experiência do utilizador são um fator motivador para esta dissertação, que procura vivamente satisfazer quer os objetivos da empresa, quer os dos seus clientes.

1.3 Objetivos

Com a elaboração desta dissertação, pretende-se que sejam cumpridos os seguintes objetivos:

- Estudar e reanalisar técnicas de *ludificação* previamente identificadas para o *iPortalDoc*;
- Desenvolver novas interfaces para o *iPortalDoc*, ou adaptar as mais utilizadas (ações pendentes, lista de documentos, pesquisas, arquivo de emails e gestão de emails), de acordo com o decidido no ponto anterior;
- Validar se, de facto, as técnicas aplicadas contribuíram para a eficiência dos utilizadores.

1.4 Estrutura do documento

Esta dissertação está organizada em sete capítulos. No primeiro capítulo é realizada a introdução ao projeto, com a sua contextualização, apresentação da motivação e dos respetivos objetivos. No Capítulo 2 é apresentada a revisão bibliográfica, com os conceitos teóricos relativos ao tema em questão, nomeadamente à *ludificação* e a sistemas de gestão documental e de processos, e ainda descrições sucintas sobre trabalhos relacionados. No Capítulo 3 é descrito o problema, com a apresentação de algumas lacunas do *iPortalDoc*. Este capítulo, contempla também uma secção

com a solução anteriormente proposta para o sistema. No capítulo seguinte é apresentada a respetiva solução ao problema identificado, onde é descrito, inicialmente, o processo de design baseado no jogador, adotado para o caso do *iPortalDoc*. De seguida é realizada uma análise à solução anteriormente proposta, sendo também descrita, detalhadamente, a proposta de solução para o sistema. O capítulo encerra com a comparação entre a proposta anterior e a proposta apresentada nesta dissertação. No capítulo 5 encontra-se a descrição da implementação, com os procedimentos que foram adotados e explicações sobre algumas opções tomadas. São ainda apresentadas as interfaces resultantes das implementações, para cada uma das áreas da proposta de solução. O Capítulo 6 apresenta as validações a serem realizadas aquando da conclusão da implementação da solução. Por fim, o último capítulo consiste na apresentação das principais conclusões do projeto, bem como, sugestões e melhorias a desenvolver num trabalho futuro.

Capítulo 2

Revisão Bibliográfica

Este capítulo apresenta os conceitos teóricos relativos ao tema em estudo. Inicialmente será abordada a definição de motivação no contexto da *ludificação*. Serão também apresentados os diversos elementos de jogo bem como, os diferentes tipos de jogador. Antes de se abordar os sistemas de gestão documental e de processos, será descrita uma metodologia de *ludificação* centrada no jogador, e ainda a estrutura Octalysis. Por fim, serão apresentados exemplos de trabalhos relacionados com o assunto em causa.

2.1 Ludificação

Ludificação ou *gamification* tem sido definida de diversas formas ao longo dos anos mas, recentemente, surgiu uma definição mais generalizada. Segundo Seaborn e Fels [19], ludificação consiste em *"the intentional use of game elements for a gameful experience of non-game tasks and contexts"*, ou seja, no uso intencional de elementos de jogo para uma experiência jogável em tarefas e contextos que não um jogo. Contudo, deve haver uma separação clara entre *gamification* e *game*. Contrariamente à *ludificação*, definir o conceito de *game* é substancialmente mais difícil. Um jogo ou *game* pode ser visto como um sistema em que os jogadores se envolvem num conflito artificial, caracterizado por diversos critérios, cujo resultado é quantificável. Regras, estrutura, voluntariedade, resultados incertos, conflito, representação e resolução são alguns desses critérios que, combinados em diferentes proporções, provocam uma experiência única de jogo a cada participante.

Na Figura 2.1 é possível observar o posicionamento do conceito de *gamification* em função das dimensões *Gaming*/ *Playing* e *Parts*/*Whole*. *Playing* ou *play* denota uma combinação mais livre, expressiva e improvisada de comportamentos e significados. É apenas realizado por prazer ou entretenimento. Em contraste, *gaming* ou *game*, é caracterizado por sistemas de regras explícitas e pela competição ou conflito de participantes nesses sistemas, tendo em vista determinados objetivos ou resultados. A dimensão *Parts*/*Whole* é a que permite distinguir os conceitos de *gamification* e *game*, anteriormente referidos. Enquanto que um *game* é desenvolvido para ser um

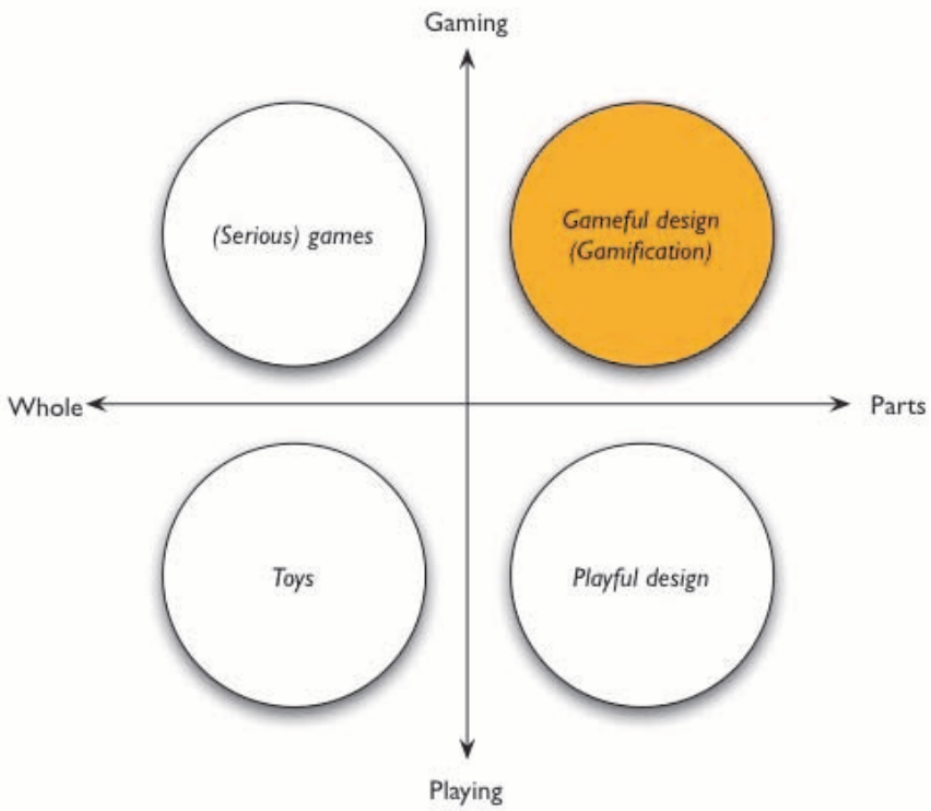


Figura 2.1: Enquadramento do conceito de *gamification* [6]

sistema de entretenimento completo, na *gamification*, apenas parte do sistema é que assume essas características.

2.1.1 Motivação

Um fator importante a destacar no que diz respeito à *ludificação* são os motivos pelos quais um jogador continua interessado e envolvido no contexto do jogo. Nesse sentido, e antes de serem apresentadas as características de um jogo, irá ser abordada a *Self-Determination Theory* (SDT) desenvolvida por Ryan e Deci [16]. Esta teoria tem o foco na motivação humana e na personalidade, e afirma que as pessoas têm três necessidades psicológicas inatas, que são consideradas como necessidades universais: a competência, a relação e a autonomia.

- **Competência** — Significa o desejo do ser humano, de controlar e dominar o meio ambiente que o rodeia e os resultados das suas ações;
- **Relação** — Diz respeito à necessidade do ser humano em interagir com as outras pessoas e de ter uma relação afetuosa com elas;
- **Autonomia** — Consiste no desejo do ser humano em controlar o rumo da sua própria vida.

A SDT também apresenta uma divisão no conceito de motivação em dois géneros: intrínseca e extrínseca.

- **Motivação intrínseca** — É definida como *"the doing of an activity for its inherent satisfactions rather than for some separable consequence"*, ou seja, realizar uma atividade pelas suas satisfações inerentes ao invés de algumas consequências separáveis [15]. De uma forma mais simplificada, uma pessoa é intrinsecamente motivada a fazer algo quando gosta de a fazer;
- **Motivação extrínseca** — Ryan e Deci [15] definem motivação extrínseca como *"a construct that pertains whenever an activity is done in order to attain some separable outcome"*, isto é, que uma atividade é realizada com o intuito de se alcançar algum resultado separável (alguma recompensa);
- **Não motivação** — É o estado de não se ter intenção de agir. Quando uma pessoa não está motivada, o seu comportamento carece de intencionalidade e de um sentimento de causalidade pessoal. A não motivação resulta de não se valorizar uma atividade, não se sentir competente para a fazer ou de não se acreditar que dará o resultado desejado.

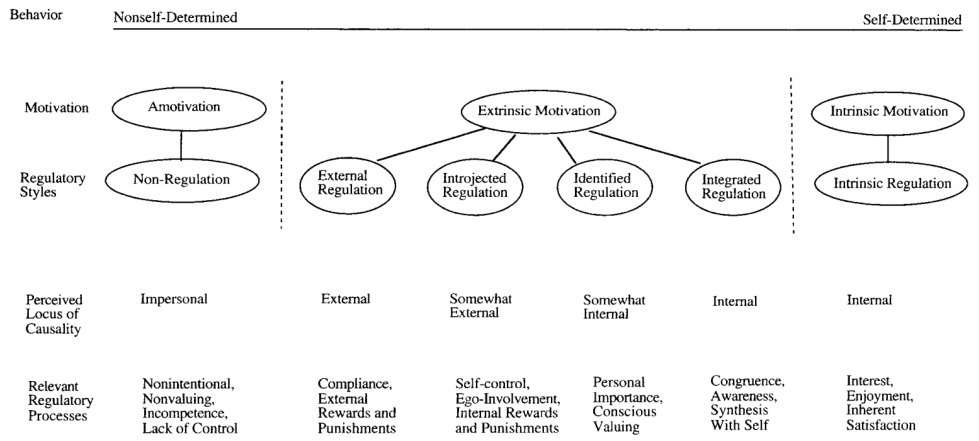


Figura 2.2: Contínuo da autodeterminação [15]

Na Figura 2.2 é possível observar o posicionamento destes três tipos de motivação em função do comportamento do ser humano, desde a não autodeterminação até à autodeterminação. Aos diversos tipos de motivação, estão ainda associados estilos regulatórios. A motivação extrínseca está dividida em quatro estilos: regulação externa, regulação interiorizada, regulação identificada e regulação integrada. Na regulação externa, o comportamento do ser humano é uma função de contingências externas de recompensa ou punição. A regulação interiorizada é uma forma de regulação relativamente controlada em que os comportamentos são realizados para evitar culpa ou ansiedade ou, para obter melhorias do ego, como por exemplo, o orgulho. A regulação identificada reflete uma valorização consciente de uma meta ou regulação comportamental, de modo que

a ação seja aceite ou adquirida como importante, pessoalmente. Por fim, a regulação integrada, é a mais auto-determinada das quatro apresentadas, e ocorre quando os regulamentos identificados são totalmente assimilados pela pessoa. As ações caracterizadas pela motivação integrada compartilham muitas qualidades com a motivação intrínseca, mas ainda são consideradas extrínsecas pelo facto de serem realizadas para obter resultados separáveis e não pelo prazer inerente.

2.1.2 Tipos de jogador

Os jogadores, com base nas suas motivações, personalidades e da forma como se relacionam com o jogo, podem ser agrupados em diferentes categorias. Bartle [1], após a análise de um conjunto de jogadores de um jogo *multi-user dungeon* (MUD), definiu quatro tipos de jogadores: exploradores, empreendedores, assassinos e sociais. Na Figura 2.3 é possível observar esses quatro grupos distintos relativamente às dimensões *world/players* e *acting/interacting*. Essas dimensões representam a fonte de interesse dos jogadores num jogo.



Figura 2.3: Tipos de jogadores segundo Bartle [10]

- **Exploradores** — Este tipo de jogadores tenta descobrir o máximo possível sobre o mundo virtual. Numa primeira fase procuram conhecer toda a narrativa do jogo, e depois, numa fase mais avançada, exploram em profundidade cada capítulo. Os exploradores só se preocupam em ganhar pontos caso estes sejam necessários para a progressão do jogo. Estas pessoas são curiosas e persistentes no sentido de encontrar a melhor solução para os desafios que enfrentam, e interagem com os outros jogadores de forma a adquirirem novas ideias.

- **Empreendedores** — Os jogadores empreendedores definem metas pessoais relacionadas com o jogo e procuram, de uma forma incansável, atingi-las. Adquirir pontos e progredir níveis costumam ser os objetivos principais deste tipo de jogadores. A exploração é apenas necessária para encontrar novos itens ou para incrementar a pontuação. Estes socializam com o intuito de obter informações sobre meios para adquirir mais pontos ou, para exibirem as suas conquistas perante as outras pessoas. Os empreendedores matam quando é necessário eliminar rivais ou quando alguém se atravessa no seu caminho.

- **Assassinos** — Estes jogadores utilizam as ferramentas fornecidas pelo jogo para causar sofrimento nos outros (isto no sentido virtual, é claro). Os assassinos assemelham-se aos empreendedores na forma de ansiarem pela vitória e pela conquista de pontos. Os que os distingue é o facto de quererem ver os outros perder. São altamente competitivos e querem ser os melhores no jogo, o que, na cabeça deles, implica a eliminação de todos os outros jogadores. A procura de pontos e aquisição de itens por parte destes jogadores é justificada pelo facto de estes os poderem tornar mais fortes, e assim, terem uma vantagem competitiva perante os outros.

- **Sociais** — Os jogadores sociais utilizam o contexto do jogo e os seus meios de comunicação de forma a interagirem com os outros. Estes jogadores também pretendem ganhar, mas de uma forma mais cooperativa e de entreajuda. São o tipo de pessoas que gostam de observar outra pessoa a jogar.

Um sistema *ludificado* pode envolver qualquer combinação destes quatro tipos. Como será apresentado mais à frente, uma pesquisa cuidadosa dos jogadores irá ajudar a entender quais os tipos de jogadores que dominam o meio ambiente e, dessa forma, ajudar a definir quais as necessidades de cada um. Um facto a ter em conta é que, apesar de haver esta distinção, pode haver jogadores que não se enquadram apenas num único tipo, apresentando características comuns a vários grupos.

2.1.3 Elementos de jogo

O que caracteriza a experiência do jogador durante um jogo, são os elementos que nele estão inseridos. Isto porque são eles que, com as suas características, provocam diferentes motivações nos jogadores. As técnicas de *ludificação* procuram captar a atenção ou aumentar a motivação de um público alvo e, por isso, fazem uso desses mesmos elementos.

Quando se pensa em *gamification* surge logo à cabeça a ideia de pontos, emblemas, e tabelas de classificação (PBLs). De facto, o uso de PBLs é um bom ponto de partida para a implementação de uma estratégia de *ludificação* [23].

- **Pontos** — Os pontos são frequentemente usados para levar o utilizador a realizar determinada tarefa de forma a obtê-los, mas não só. Também lhe transmite o estado da sua progressão e podem ser usados para determinar a condição de vencedor. Os pontos criam

a conexão entre o progresso do jogo e as recompensas extrínsecas. Do ponto de vista da pessoa que desenvolveu o jogo (ou o sistema *ludificado*), os pontos podem ser vistos como uma métrica importante na análise do sistema;

- **Emblemas** — Os emblemas vêm muitas vezes associados aos pontos. São uma representação visual de alguma conquista. Apresentam cinco características de motivação:

  1. Os emblemas podem estabelecer um objetivo para os utilizadores se esforçarem, o que permite resultados positivos na motivação;
  2. Fornecem uma orientação do que é possível fazer. Isto permite que o utilizador esteja envolvido no sistema;
  3. São um sinal daquilo com que o utilizador se preocupa e daquilo que ele já realizou. São marcadores visuais de reputação e são adquiridos com o intuito de mostrarem aos outros do que são capazes;
  4. Operam como símbolos virtuais do estado da jornada pessoal do utilizador no sistema;
  5. Os emblemas funcionam como marcadores tribais. Um utilizador que tenha alguns dos mesmos emblemas que outros utilizadores, vai sentir uma sensação de identidade com esse grupo.

- **Tabelas de classificação** — As tabelas de classificação são, talvez, o elemento mais problemático. Uma tabela de classificação dá um contexto à progressão de uma forma que os pontos ou emblemas não podem. Se o desempenho for importante, a tabela torna esse desempenho público para todos verem. Em situações concretas, os quadros de classificações podem ser motivadores poderosos. Mas por outro lado, podem ser ser bastantes desmotivadores. Se um utilizador verificar que está longe dos melhores, pode ser influenciado a deixar de tentar. Isto reduz o jogo a uma luta que gira, apenas, em torno de algumas pessoas, que estão no topo da classificação.

Contudo, Werbach e Hunter [23] entram em mais detalhe e dividem os elementos de jogo em três grandes categorias: dinâmicas, mecânicas e componentes. Como podemos observar na Figura 2.4, as categorias estão organizadas por ordem decrescente de abstração. Cada elemento está associado a uma ou mais mecânicas e cada mecânica está associada a uma ou mais dinâmicas.

- **Dinâmicas** — As dinâmicas são o grande aspeto de um sistema *ludificado*, que necessitam de considerações e de gestão, mas que não entram diretamente no jogo. São estas que suportam a "história" e a forma como o utilizador interage com o sistema. As dinâmicas de jogo mais importantes são:

  1. **Restrições** — São as limitações ou *trade-offs* que são necessários no processo de design. Nunca será possível fazer tudo ao projetar um jogo. É importante examinar o equilíbrio das componentes na *ludificação*. Os *trade-offs* são uma parte inerente do processo de design;

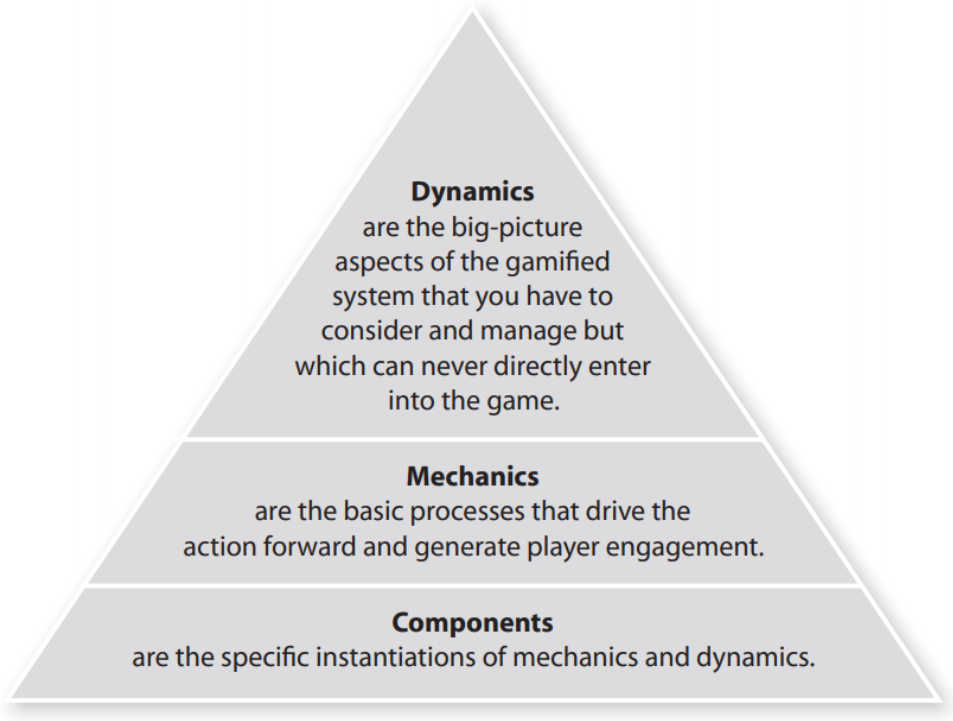


Figura 2.4: Hierarquia dos elementos de jogo [23]

2. **Emoções** — As emoções são uma parte fundamental na *gamification*. Estas conduzem o desenrolar do jogo e é importante perceber que pessoas é que estão a interagir com o sistema. O que pode frustrar um jogador, pode ser o que excita o espírito competitivo de outro;
3. **Narrativa** — Refere-se ao enredo que se desenrola ao longo de um jogo. A maior parte dos jogos têm uma história que define a interação do jogador com o jogo. A narrativa pode capturar o interesse e promover o envolvimento;
4. **Progressão** — É o crescimento e o desenvolvimento de um jogador ao nível que vai progredindo no jogo. Com o passar do tempo, a compreensão e habilidades vão aumentando, e isso, pode ser um fator motivador que aumenta o envolvimento do jogador;
5. **Relacionamentos** — Neste elemento são consideradas as interações sociais que ocorrem quando os jogos são jogados. Essas interações geram, geralmente, sentimentos de camaradagem, altruísmo e de estatuto.

- **Mecânicas** — São os processos básicos que impulsionam a ação e geram o envolvimento do jogador. Cada mecânica permite alcançar uma ou mais dinâmicas descritas. Um evento aleatório, como um prémio, pode estimular a sensação de diversão e curiosidade dos jogadores. Também pode ser uma maneira de obter novos participantes, ou manter jogadores experientes envolvidos. Os dez mecanismos mais importantes de jogo são:

1. **Desafios** — Quebra-cabeças ou outras tarefas que exigem esforço para as resolver;
2. **Chance** — Elementos aleatórios;
3. **Competição** — Um jogador ou grupo ganha, e o outro perde;
4. **Cooperação** — Os jogadores devem trabalhar juntos para alcançar um objetivo comum;
5. **Comentários** — Informações sobre as opções tomadas pelos jogadores;
6. **Aquisição de recursos** — Obtenção de itens úteis ou colecionáveis. O processo de aquisição desses itens é, muitas vezes, tão valorizado quanto os próprios itens.
7. **Recompensas** — Benefícios por alguma ação ou conquista;
8. **Transações** — Negociação entre jogadores, diretamente ou através de intermediários;
9. **Rondas** — Participação sequencial de jogadores alternados.;
10. **Estados de vencedor** — Condições que fazem um jogador ou grupo o vencedor (é também possível ter estados de empate ou derrota).

- **Componentes** — Os componentes compõem o maior grupo de elementos de um jogo e são mais específicos do que as dinâmicas ou as mecânicas. Estes elementos conduzem a ferramentas reais que podem ser utilizadas no início do processo de *ludificação* [14]. Os componentes de jogo mais importantes são:

  1. **Metas** — Objetivos definidos;
  2. **Avatares** — Representações visuais da personagem de um jogador;
  3. **Emblemas** — Representações visuais de conquistas;
  4. ***Boss Fights*** — São, geralmente, desafios difíceis no final de cada nível nível;
  5. **Coleções** — Acumulação de conjuntos de itens ou emblemas;
  6. **Combate** — Uma batalha definida, tipicamente de curta duração;
  7. **Desbloqueio de conteúdo** — Aspetos disponíveis apenas quando os jogadores alcançam determinados objetivos;
  8. ***Gifting*** — Oportunidade para compartilhar recursos com outros utilizadores;
  9. **Tabelas de classificação** — Visualização da progressão e realização do jogador;
  10. **Níveis** — Etapas definidas no progresso do jogador;
  11. **Pontos** — Representações numéricas da progressão do jogador;
  12. **Missões** — Desafios predefinidos com objetivos e recompensas;
  13. **Gráficos sociais** — Representação da rede social dos jogadores dentro do jogo;
  14. **Equipas** — Grupos definidos de jogadores que trabalham juntos para um objetivo comum;
  15. **Mercadorias virtuais** — Ativos do jogo com valor fictício ou real.

2.1.4 Processo de design baseado no jogador

Já foi apresentada uma grande quantidade de informação relativamente aos elementos de um jogo, aos tipos de jogadores e à motivação que os pode manter nesse mundo virtual. Contudo, para desenvolver um sistema *ludificado* não basta introduzir todos os elementos analisados.

Burke [3] apresenta uma estratégia de *ludificação* baseada nos objetivos do negócio onde o sistema está inserido, e em que o design é centrado no jogador/utilizador. A experiência do jogador é projetada como uma jornada e ocorre num espaço de jogo que pode abranger os mundos físico e virtual. Burke diz que *"the challenge of gamification is designing the player experience, not the technology"*, ou seja, que a dificuldade da *ludificação* é projetar a experiência do jogador, não a tecnologia. Chega mesmo a separar os conceitos de *user interface design* e de *user experience design*. O *user interface design* foca-se na interface homem-computador enquanto que o *user experience design* baseia-se nas disciplinas de *design thinking*, ciência comportamental e sistemas emergentes, e procura entregar ao jogador/utilizador a melhor experiência possível.

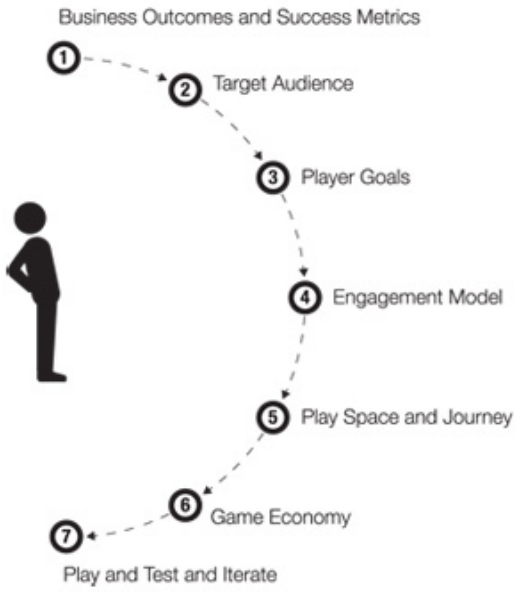


Figura 2.5: Processo de design da experiência do jogador [3]

O processo de design da experiência do jogador descrito por Burke (Figura 2.5) permite reduzir o tempo e os riscos ao projetar uma solução *ludificada*. Contempla sete passos:

1. **Definir os resultados do negócio e as métricas de sucesso**

   Muitas empresas pretendem adotar estratégias de *ludificação* por uma questão de tendência ou porque acham divertido, e não por quererem atingir um objetivo empresarial. Este pensamento pode levar ao fracasso da organização. O pensamento lógico que resulta deste ideia é: como é que uma empresa pode alcançar o sucesso sem que primeiro tenha definido qual é o sucesso? O primeiro passo no design de uma solução *ludificada* consiste em ter bem

definido qual é o objetivo do negócio, quais são os resultados alvo que se pretende atingir e quais as métricas de sucesso. Os resultados que se pretende alcançar têm de ser claros e objetivos, por exemplo, aumentar o tráfego do site em X por cento em Y meses ou, lançar X projetos em Y meses.

2. **Definir o público alvo**

   O público alvo para soluções *ludificadas* são, normalmente, clientes, funcionários, ou uma comunidade com um interesse em comum. Este público alvo participa nas principais trocas de valor da organização, isto é, os clientes trocam dinheiro por um produto ou serviço, os funcionários trocam o seu esforço e tempo por um salário, e as comunidades normalmente trocam informações. A *ludificação* aborda, geralmente, problemas que não estão presentes nas principais trocas de valor. Definir o público alvo tem como objetivo colocar limites em torno das pessoas que a organização envolve, permitindo restringir a quantidade de diferentes tipos de jogadores que precisam de ser abordados para a solução. Ter uma compreensão clara do público alvo permite definir uma correta solução para o sistema que envolva, de facto, as pessoas que se pretende.

   - **Explorar o público alvo** — Após a identificação do público alvo, é necessário dedicar tempo a apreender as suas principais características. Esta ação pode ser realizada através da observação do público, da recolha de dados e da identificação dos seus relacionamentos. É necessário falar com eles, saber o que gostam, ou não, como realizam as suas ações e quais as melhorias que gostavam de ver implementadas. Estas interações permitem captar sugestões de motivação que refletem os objetivos empresariais. Quanto maior for a perceção das personalidades dos jogadores/utilizadores, melhor será a definição das suas motivações e objetivos.
   - **Criar *player personas*** — Uma *persona* é um indivíduo imaginário que representa alguns traços de caráter comuns a um grupo de pessoas. Cada grupo tem objetivos e motivações diferentes, mas após a criação de personagens fictícias que os representa, torna-se mais fácil identificar os objetivos e motivações de cada grupo.

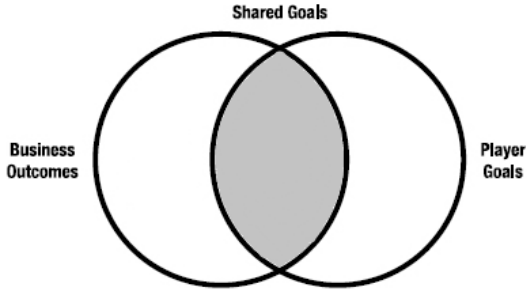


Figura 2.6: Objetivos comuns ao negócio e ao jogador/utilizador (adaptada de [3])

3. **Definir os objetivos do jogador**

   Após o conhecimento do público alvo e das suas motivações é necessário definir as metas que eles deverão atingir. Um dos grandes problemas deste passo é a falta de conhecimento do público. Isso pode ser justificado por falta de exploração no passo anterior, ou por omissão de informações relevantes, por parte das pessoas alvo.

   Nesta fase é necessário garantir que, apesar dos objetivos empresariais e do jogador não serem exatamente iguais, exista alguns que sejam comuns a ambas as partes (Figura 2.6). A solução *ludificada* será projetada com o foco nesses objetivos comuns, de forma a garantir a motivação do público alvo e, ao mesmo tempo, o sucesso da empresa.

4. **Determinar o modelo de envolvimento do jogador**

   Neste passo são abordadas as decisões sobre a estrutura da solução *ludificada*. O modelo de envolvimento do jogador descreve como é que será a interação dos jogadores com o sistema. O desafio da *gamification* é pensar nas implicações de cada um dos elementos do jogo e fazer escolhas conscientes sobre como a experiência será estruturada. Os parâmetros mais comuns no modelo de envolvimento do jogador (Figura 2.7), segundo Burke, são:

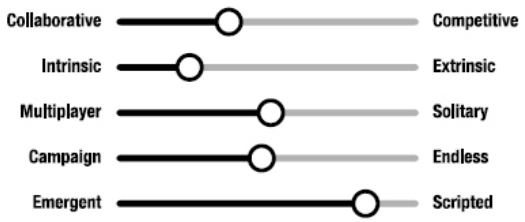


Figura 2.7: Exemplo de um modelo de envolvimento do jogador [3]

- **Colaborativo/Competitivo** — Um dos parâmetros básicos é o equilíbrio da colaboração e concorrência num sistema *ludificado*. Na maioria das soluções *ludificadas* não há um único vencedor e os jogadores são levados a ajudar os outros a terem sucesso. Desta forma, os jogos colaborativos são mais propensos a conduzir comportamentos que vão de encontro aos objetivos da organização. A maneira mais comum de combinar colaboração e competição é criar estruturas de equipas dentro do sistema. Os jogadores/utilizadores colaboram dentro da equipa, e as equipas competem entre si;
- **Intrínseco/Extrínseco** — Este parâmetro, apesar dos mesmos termos, é ligeiramente diferente do conceito de motivação intrínseca e extrínseca apresentado anteriormente. Neste contexto, intrínseco diz respeito a recompensas dentro da própria organização, como por exemplo, a motivação por estar a apreender uma nova ferramenta enquanto que extrínseco é relativo a recompensas externas, como por exemplo, prémios monetários. As soluções *ludificadas* dependem, principalmente, das recompensas intrínsecas;

- **Multijogador/Solitário** — É um parâmetro relativo ao modo de interação do jogador. Um sistema pode ser projetado para suportar interações *multiplayer* ou apenas para haver interações entre o jogador e a solução;
- **Campanha/*Endless*** — Este parâmetro diz respeito ao final da solução, isto é, se o sistema será infinito ou se terá um fim após a conclusão das metas definidas;
- **Emergente/*Scripted*** — O último parâmetro apresentado é relativo ao estilo do sistema, no sentido de ser mais aberto ou de seguir um fluxo previamente definido. Com soluções emergentes, os designers concentram-se mais no espaço de jogo. Em situações em que o resultado é conhecido (como é o caso das soluções *ludificadas*), um sistema com *scripts* é a melhor escolha. Nestes casos, os designers concentram a maior parte da sua atenção na jornada do jogador/utilizador.

5. **Definir o espaço de jogo e planear a jornada**

   Designers de sistemas *ludificados* devem definir o espaço de jogo e a jornada do jogador, tendo por base todos os fundamentos adquiridos nas fases anteriores. O espaço de jogo pode ser um lugar virtual que existe apenas dentro de uma rede de computadores, ou pode ser uma combinação de ambientes virtuais e reais. A jornada do jogador é o caminho que é definido para o jogador/utilizador seguir ao longo da sua progressão no sistema.

   - **Espaço de jogo** — Este é o espaço fornecido aos jogadores para se envolveram com o jogo e entre eles. Os jogos de vídeo normalmente possuem animações elaboradas, simulações e avatares de alta qualidade. Em contraste, nas soluções *ludificadas* o espaço de jogo costuma ser muito básico, que mostra os perfis dos jogadores, o progresso e todas as ferramentas necessárias para o jogador se envolver com o sistema.

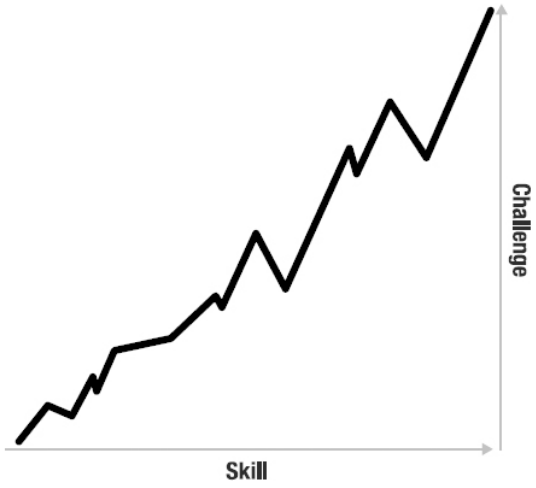


Figura 2.8: Jornada do jogador [3]

- **Jornada do jogador** — Descreve o caminho a seguir pelo jogador no sistema. Desde o embarque dos jogadores, até aos níveis mais avançados, os designers devem equilibrar cuidadosamente o desafio e a habilidade à medida que o jogador progride, a fim de manter o envolvimento (Figura 2.8). As soluções *ludificadas* são projetadas para oferecer aos jogadores uma série de ações de envolvimento que os jogadores devem dominar. Todas as tentativas para completar essas ações geram, normalmente, algum tipo de *feedback* a reconhecer o esforço do jogador. As ações devem ser desafiadores para o jogador, mas também alcançáveis, de acordo com o seu nível de habilidade. Muitas vezes, estão agrupadas em missões ou desafios que os jogadores devem completar para adquirirem um emblema e/ou passar para o próximo nível.

6. **Definir a economia do jogo**

   As soluções de *ludificação* têm uma economia no jogo, embora geralmente não seja uma economia baseada em dinheiro. A economia no jogo é composta pelos incentivos e recompensas com que os jogadores são premiados ao executar com sucesso as tarefas, completar desafios ou atingir metas. Mais uma vez, é fulcral compreender os objetivos e as motivações dos jogadores para criar uma economia de jogo que implemente um design centrado no jogador. Existem quatro "moedas" básicas que os jogadores acumulam na economia de jogo: diversão, coisas, capital social e auto-estima (Figura 2.9), e que são implementadas através de elementos de jogo.

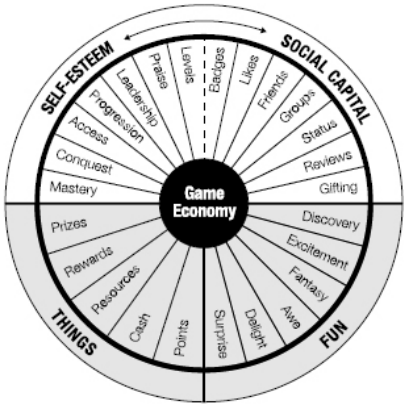


Figura 2.9: Economia do jogo [3]

- **Diversão** — Não é muito usada em soluções *ludificadas*, mas em videojogos, representa uma grande parte da economia de jogo. Apesar de não oferecerem recompensas tangíveis, efeitos de luz e som, juntamente com gráficos avançados, dão uma maior experiência ao utilizador e um maior divertimento. Em sistemas *ludificados*, o uso da

diversão é normalmente implementado através de recompensas surpresa. São recompensas inesperadas e aleatórias que provocam no jogador a sensação de incerteza do que acontecerá a seguir.

- **Coisas** — Esta "moeda" inclui os itens tangíveis que podem ser adquiridos e às vezes trocados dentro do sistema. São normalmente implementados como pontos e podem ser adquiridos com dinheiro ou através de recompensas. Se numa solução *ludificada*, quando o comportamento desejado para o utilizador não está de acordo com os seus objetivos, o uso de itens tangíveis são fundamentais para manter a sua motivação.
- **Auto-estima** — Os jogos e os sistemas *ludificados* oferecem aos jogadores/utilizador desafios, *feedback* constante sobre a progressão e reconhecem ações através do aumento de nível, acesso a serviços exclusivos ou atribuindo emblemas de conquistas. Todas estas recompensas trabalham para construir a auto-estima do jogador/utilizador e manter o seu interesse no sistema.
- **Capital social** — As pessoas ficam motivadas quando as outras pessoas, dentro dos seus círculos sociais, reconhecem as suas conquistas. Os designers utilizam o reconhecimento como uma forma de motivação. O círculo social normalmente inclui outros jogadores do jogo mas pode ser estendido para redes sociais externas (Facebook, Twitter ou LinkedIn), para aumentar o impacto na motivação do jogador.

7. **Executar o teste e iterar**

   Muitas vezes, as pessoas são levadas a pensar que uma iniciativa de *ludificação* é como criar um videojogo e que implicará um longo ciclo de desenvolvimento até um produto final ser lançado. Mas ao contrário do desenvolvimento de videojogos, a *ludificação* não é assim. Um produto, nestes casos, nunca será final. É sempre um processo de melhoria contínua, com adição de novas funcionalidades que envolvem os utilizadores de forma a eles não se desinteressarem pelo sistema. No lançamento inicial, este irá conter apenas elementos suficientes para envolver os utilizadores e atingir os objetivos empresariais.

   Embora esta metodologia permita ultrapassar as principais dificuldades na conceção de um sistema *ludificado*, ter uma boa solução não implica que irá ter sucesso. O sistema é normalmente testado por um grupo piloto de jogadores, que vão fornecendo os seus *feedbacks* durante a fase de desenvolvimento. Adquirir dados sobre todas as interações dos jogadores/utilizadores permitirá verificar quais os elementos que estão a ter o efeito desejado ou não. Esses dados permitirão melhorar a solução, iterativamente, ao longo do tempo.

2.1.5 Octalysis

Octalysis é uma estrutura de *ludificação* apresentada por Yu-Kai Chou [4], que permite analisar e construir estratégias em torno das diversas características de um jogo. Esta estrutura em forma de octógono (Figura 2.10), apresenta oito unidades principais, que segundo o autor, nos motivam a tomar certas decisões e a realizar determinadas tarefas.

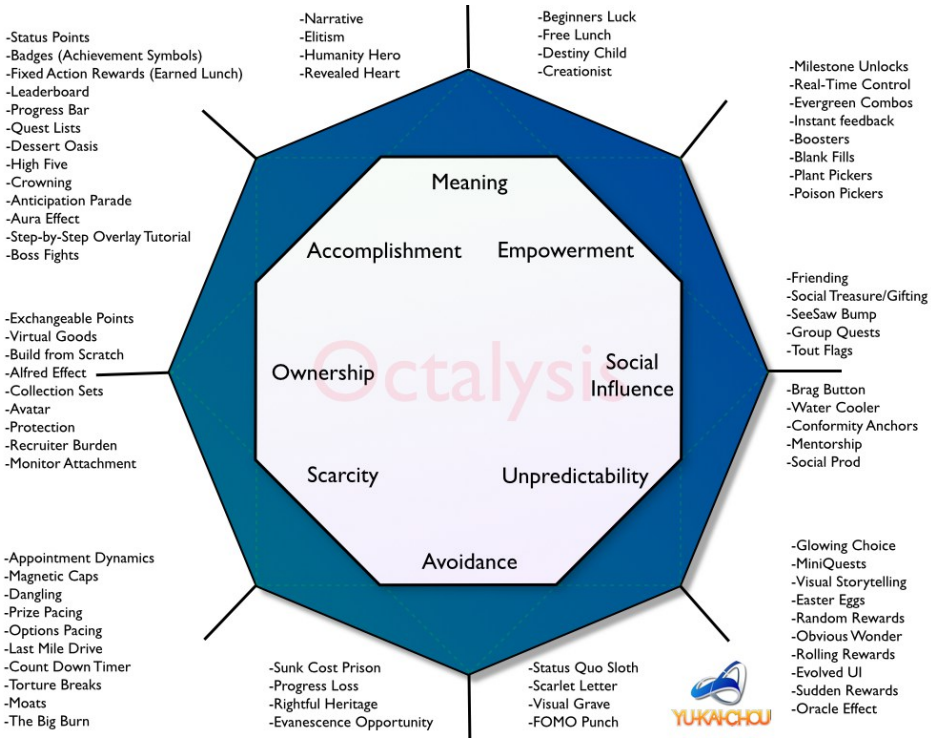


Figura 2.10: Estrutura de *ludificação* Octalysis [4]

1. **Significado épico e chamamento**

   Neste unidade principal o jogador acredita que foi "escolhido" para uma missão importante. O que o motiva é o facto de pensar que representa um papel fundamental em algo que é superior a ele próprio. Dedicar imenso tempo a manter um fórum ou a ajudar a criar coisas para uma comunidade (por exemplo, sistemas *open source*), pode ser visto como um sintoma desses pensamentos. O termo "sorte de principiante" também pode ser incorporado nesta unidade tendo em vista que o jogador pode-se considerar um felizardo por ter arrecadado um item raro no início do jogo.

2. **Desenvolvimento e concretização**

   Desenvolvimento e concretização é a unidade principal relativa aos motivos que incentivam o jogador a progredir, a desenvolver as suas habilidades e, eventualmente, a superar os seus desafios. Como já foi referido anteriormente, a superação de desafios está normalmente associado a aquisição de pontos e emblemas. Tendo este ponto de vista, esta unidade é uma das mais fáceis de projetar e onde se concentram a maioria dos PBLs.

3. **Potenciar a criatividade e o *feedback***

   São as situações em que os jogadores estão envolvidos num processo criativo e onde é necessário repetir diversos processos e tentar diferentes combinações. Para além disso, esta unidade foca-se no ciclo que envolve a ação do jogador, a observação do respetivo resultado,

o *feedback* e a sua resposta. Quando esta unidade está inserida num jogo, o designer não necessita de adicionar conteúdos extra para manter a atividade fresca e atrativa.

4. **Propriedade e posse**

   Esta unidade reflete a motivação dos jogadores por possuírem alguma coisa. Nestes casos, um proprietário sente-se tentado a adquirir mais riqueza, o que num contexto de jogo envolve obter mais produtos, como por exemplo itens especiais ou moedas virtuais. Isto leva a que o jogador esteja sempre envolvido e motivado de forma a colecionar o maior número de coisas possível.

5. **Influência e relacionamento social**

   Orientação, aceitação, respostas sociais, companheirismo, competição e inveja são os diversos elementos sociais que impulsionam as pessoas e que estão incorporados nesta unidade. Quando um jogador percebe que um amigo é capaz de realizar uma ação que achava impossível, este se sente capaz de o fazer. Outro exemplo do contexto desta unidade principal, é o caso de desenvolver produtos, que provocam a sensação de nostalgia, com o intuito de aumentar a procura.

6. **Escassez e impaciência**

   Esta unidade diz respeito ao impulso de querer algo que não se pode ter. O facto de não ter acesso provoca uma motivação na pessoa que a leva a pensar constantemente nisso. Esta ideia levou a que muitos jogos implementassem limites de tentativas ou recolha de recompensas após um determinado período de tempo.

7. **Imprevisibilidade e curiosidade**

   Caracteriza a iniciativa de se querer descobrir o que acontecerá a seguir. Quando não se sabe o que vai acontecer, as pessoas têm a tendência de pensar nisso com maior frequência. *Packs* de itens cujo conteúdo é desconhecido é um elemento constantemente presente em jogos, que exemplifica a aplicação desta unidade principal.

8. **Perda e prevenção**

   Esta unidade é baseada na prevenção de acontecimentos negativos. A sensação de possibilidade de perda de uma oportunidade, estimula as pessoas a realizar determinadas ações. Uma promoção de um produto é um exemplo de aplicação desta unidade. Quando um produto sofre uma redução considerável no preço, as pessoas procuram adquiri-lo de forma a não perderem essa oportunidade.

As unidades principais colocadas no lado direito estão mais relacionadas com a criatividade, a auto-expressão e aspetos sociais, enquanto que as do lado esquerdo são mais associadas à lógica, aos cálculos e à propriedade. Se tivermos em consideração os conceitos de motivação já descritos

anteriormente, as unidades do lado esquerdo tendem a ser baseadas na motivação extrínseca (motivação por desejar algo, quer seja um bem ou um objetivo) enquanto que as do lado direito estão mais relacionadas com motivações intrínsecas.

Chou divide ainda a jornada do jogador em quatro fases: descoberta (o porquê das pessoas quererem começar a jornada), *onboarding* (ensinamento dos jogadores sobre as regras e as ferramentas de jogo), andaimes (jornada regular de ações repetidas em direção a um objetivo) e *endgame* (como se mantém os jogadores veteranos) [4]. Na Figura 2.11 é possível observar as motivações dos diferentes tipos de jogadores, apresentados por Bartle [1], nas diferentes fases da jornada do jogadores, segundo a estrutura Octalysis.

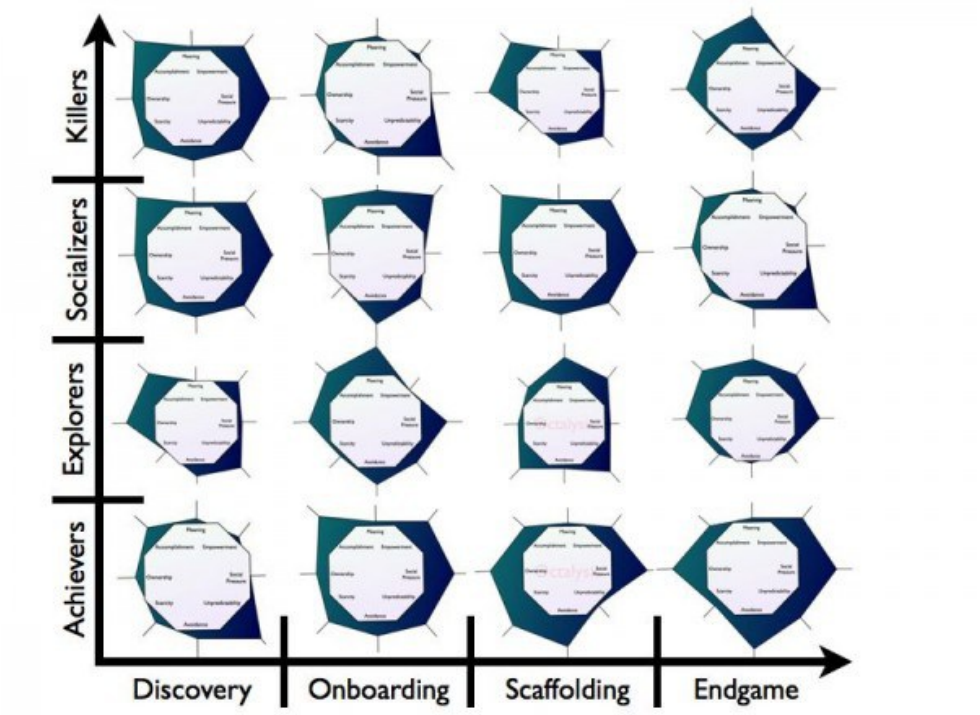


Figura 2.11: Estrutura de *ludificação* Octalysis para os diferentes tipos de jogadores [4]

2.2 Sistemas de gestão documental e de processos

Os sistemas de gestão documental (DMS) tornaram-se uma parte essencial da maioria das empresas, pois permitem gerir grandes quantidades de informação e de dados que as organizações possuem. Um DMS é usado para armazenar, pesquisar e recuperar documentos eletrónicos e/ou imagens digitalizadas de documentos em papel. Permite também acompanhar as diferentes versões dos documentos criados por diferentes utilizadores. As principais características de um sistema de gestão documental são [9]:

- **Digitalização** — A digitalização de documentos físicos é uma atividade fulcral num DMS. É importante garantir que a digitalização tem a resolução adequada de acordo com o tamanho pré-definido para o arquivo. Os documentos podem ser encriptados ou protegidos por senha como uma medida de segurança ao acesso indevido;

- **Metadados** — Os metadados são informações relativas aos ficheiros armazenados que permitem que a sua identificação e recuperação seja mais fácil. Os metadados são definidos para cada documento. Tipo de documento, data de criação e o autor são alguns dos exemplos dessas informações. Normalmente o sistema solicita a introdução desses dados ao utilizador através de interfaces amigáveis;

- **Captura** — A captura é relativa ao processo de digitalização de documentos em papel usando dispositivos como *scanners*. A interface para o *scanner* pode ser aberta a partir do sistema para que os metadados sejam introduzidos à medida que o documento esteja a ser digitalizado;

- **Integração** — Os sistemas de gestão documental têm sido integrados em aplicações ou sistemas operativos de forma a que a associação de metadados e a gestão de documentos seja mais fácil, e tudo incluído no mesmo sistema;

- **Indexação** — Permite manter o registo de identificadores únicos de documentos. Esta característica existe, principalmente, para suportar uma recuperação/consulta rápida de documentos;

- **Armazenamento** — Os documentos eletrónicos são armazenados em servidores de ficheiros ou em bases de dados;

- **Recuperação** — Recuperar os documentos eletrónicos do servidor de ficheiros ou base de dados. A recuperação simples é realizada através de chaves que identificam um único ficheiro. Uma recuperação mais completa pode envolver os metadados dos documentos;

- **Confidencialidade** — A informação não está disponível publicamente. Um DMS possui um módulo de gestão de acesso ou autenticação que permite o acesso a documentos consoante as permissões de cada utilizador;

- **Integridade** — A integridade dos documentos e dos dados é vital nos sistemas de gestão documental. Um DMS pode indicar se um documento foi adulterado ou não;

- **Disponibilidade** — Os documentos podem ser acedidos por diversos utilizadores ao mesmo tempo. Como o sistema mantém o controlo das diversas versões, estas podem ser visualizadas pelas diferentes pessoas;

- **Pesquisa** — O sistema encontra pastas e documentos através de atributos. Normalmente esses atributos são os metadados dos documentos.

O facto de um sistema de gestão documental diminuir o tempo de processamento de documentos, levando a uma maior eficiência dos trabalhadores, e também permitir um melhor envolvimento entre a organização e os elementos externos (parceiros, fornecedores e clientes), motiva a adoção destes sistemas por parte das empresas.

Aos sistemas de gestão documental vêm muitas vezes associados os sistemas de gestão de processos ou *workflows*. Um *workflow* consiste num conjunto de tarefas ou eventos que estão interligados, de forma a criar valor para uma atividade da organização. Essas tarefas são normalmente realizadas por pessoas ou grupos distintos. Um *workflow* pode aumentar os benefícios de um sistema de gestão documental, automatizando o encaminhamento dos documentos para várias pessoas, o que elimina os *bottlenecks* e permite uma maior fluidez nos processos comerciais. Os *workflows* devem notificar os utilizadores de eventos específicos relacionados com o DMS, com base nas regras criadas pelo administrador. Se um destinatário não atuar dentro de um período de tempo, o sistema deve notificá-lo ou proceder ao encaminhamento da tarefa para outro utilizador. Um componente essencial em qualquer sistema de *workflow* é a automatização de documentos. Este deve ser capaz de mover, copiar ou remover documentos automaticamente do servidor de ficheiros com base num conjunto de regras. No entanto, o sucesso de qualquer sistema de *workflow* é a capacidade de lidar com exceções à regra à medida que elas surgem [22]. Na Figura 2.12 está ilustrado um exemplo de um *workflow* que consiste na aprovação de um documento.

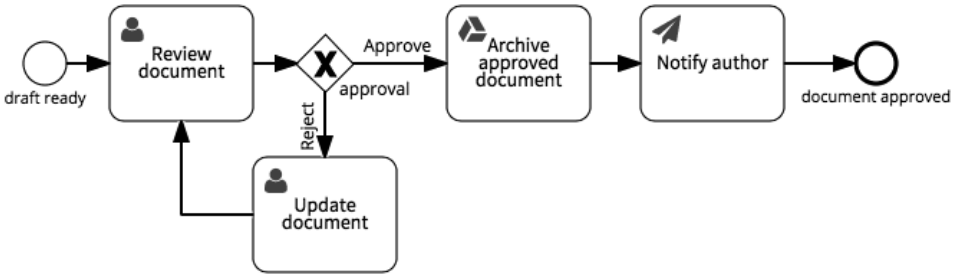


Figura 2.12: Exemplo de um *workflow*: aprovação de um documento [7]

2.2.1 iPortalDoc

O *iPortalDoc* é um sistema de gestão documental e de processos para empresas e instituições, que permite aos seus utilizadores a gestão dos fluxos dos documentos (*workflows*), bem como o seu arquivamento para gestão futura. Esta solução é vista como um substituto ao papel e não uma tecnologia adjacente.

O *iPortalDoc* apresenta uma das características referidas anteriormente, a integração. Isto porque, está inserido num outro sistema, o IPBRICK OS. Esta colaboração permite que a gestão dos utilizadores, bem como dos grupos aos quais pertencem, sejam realizadas pelo sistema operativo onde o sistema de gestão documental está inserido. De entre os benefícios que o *iPortalDoc* proporciona para a atividade da organização, destacam-se:

- facilidade de utilização, formação reduzida;
- integração no ambiente de trabalho dos utilizadores;
- segurança no acesso e ações sobre documentos;
- redução de informação em papel e controlo de processos com a utilização de *workflows*;
- facilidade de comunicação;
- integração com outros softwares (como já foi referido).

Todas as interfaces de acesso do *iPortalDoc* são assentes em interfaces de comunicação suportadas em protocolos normalizados (HTTPS, SMB, SMTP). As três interfaces que permitem a interação entre o utilizador e o sistema são:

- acesso Web através de um navegador de Internet;
- acesso SMB (partilha de rede) através de um gestor de ficheiros;
- acesso de correio eletrónico através de um leitor de emails.

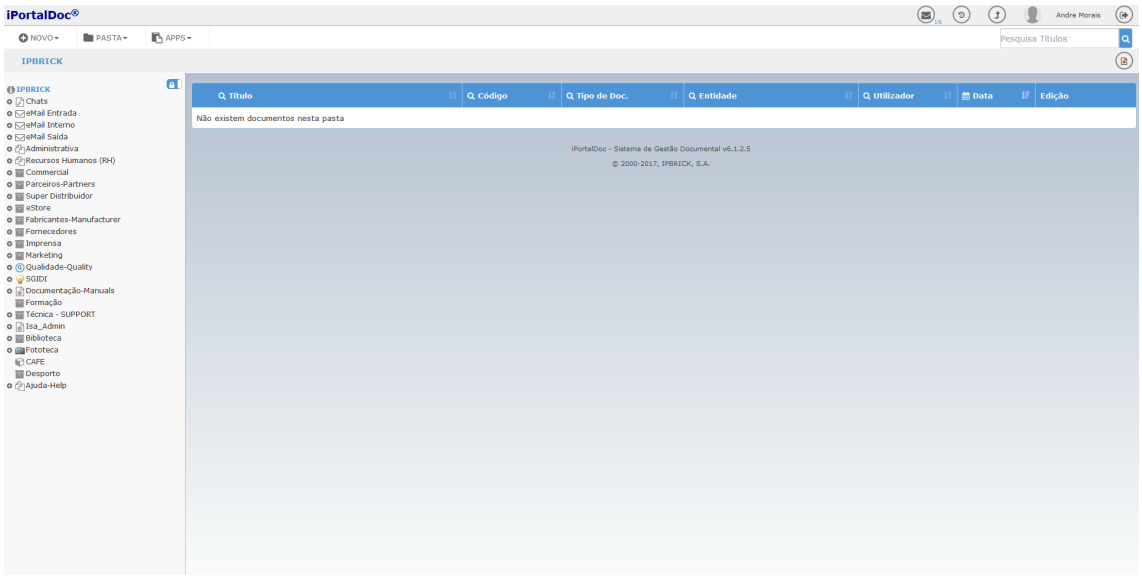


Figura 2.13: Exemplo da interface geral do *iPortalDoc*

O foco desta dissertação será direcionado para a interface de acesso Web, por ser a mais poderosa no sentido de permitir todo o tipo de operações sobre documentos, gestão de *workflows*, inserção de *templates*, entre outros. Na Figura 2.13 está apresentada a interface geral do *iPortalDoc*. No lado esquerdo é possível observar a hierarquia documental que permite o acesso as diversas pastas e documentos. Na parte central, caso uma das pastas esteja selecionada, serão apresentados, em forma de lista, os documentos presentes nesse diretório. Em cima da hierarquia

de documentos está presente uma barra de ferramentas que permite as mais diversas ações, como por exemplo adicionar documentos ou criar *workflows* (caso tenha permissões para tal). No canto superior direito encontra-se um campo de pesquisa. Esse campo de pesquisa permite procurar o ficheiro a partir de qualquer interface do sistema. A lupa ao lado desse campo abre uma nova interface que possibilita a introdução de mais parâmetros de pesquisa (Figura 2.14).

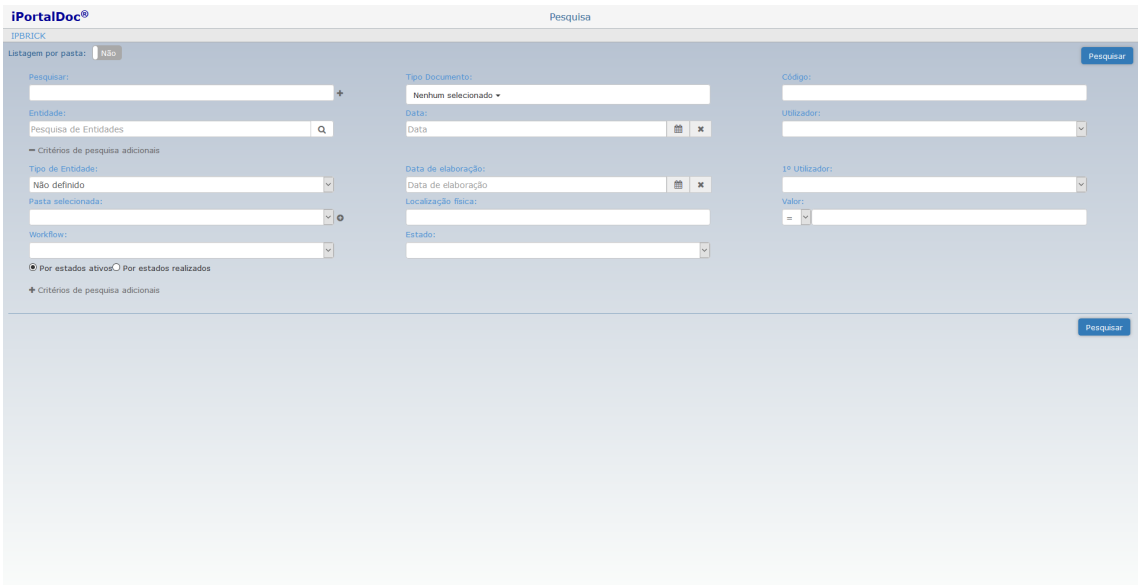


Figura 2.14: Exemplo da interface de pesquisa do *iPortalDoc*

Voltando novamente à interface geral (Figura 2.13), no canto superior direito, acima do campo de pesquisa e ao lado esquerdo da indicação do utilizador, encontram-se três ícones que permitem abrir as janelas de Emails, Histórico e Atalhos, da esquerda para a direita, respetivamente.

O conceito de *workflow* tem um papel importante no *iPortalDoc*, nomeadamente na organização do fluxo dos documentos. Os documentos podem fluir de pessoas para pessoas, desde a sua criação até ao arquivamento. Essas pessoas terão de executar diversas ações como editar, aprovar ou simplesmente ler. Quando um utilizador está associado a um documento, é-lhe incutido uma ação, que terá de realizar. Na Figura 2.15 está ilustrada uma janela exemplo do histórico do *iPortalDoc*. Esta janela não é relativa às ações que o utilizador tem pendente, mas sim, às ações que já realizou e, em maior parte dos casos, cujo *workflow* já se encontra concluído.

As ações, para além de estarem indicadas na interface respetiva, também são enviadas por email. Outra vantagem do *iPortalDoc* é permitir associar documentos com emails e enviar documentos a partir do email para a plataforma. Na Figura 2.16 está apresentada uma janela exemplo com a caixa de emails de um determinado utilizador do *iPortalDoc*.

Os documentos, para além do email, podem ser introduzidos manualmente no sistema através da interface destinada a esse procedimento; através da *shell* do sistema operativo; ou através de serviços web.

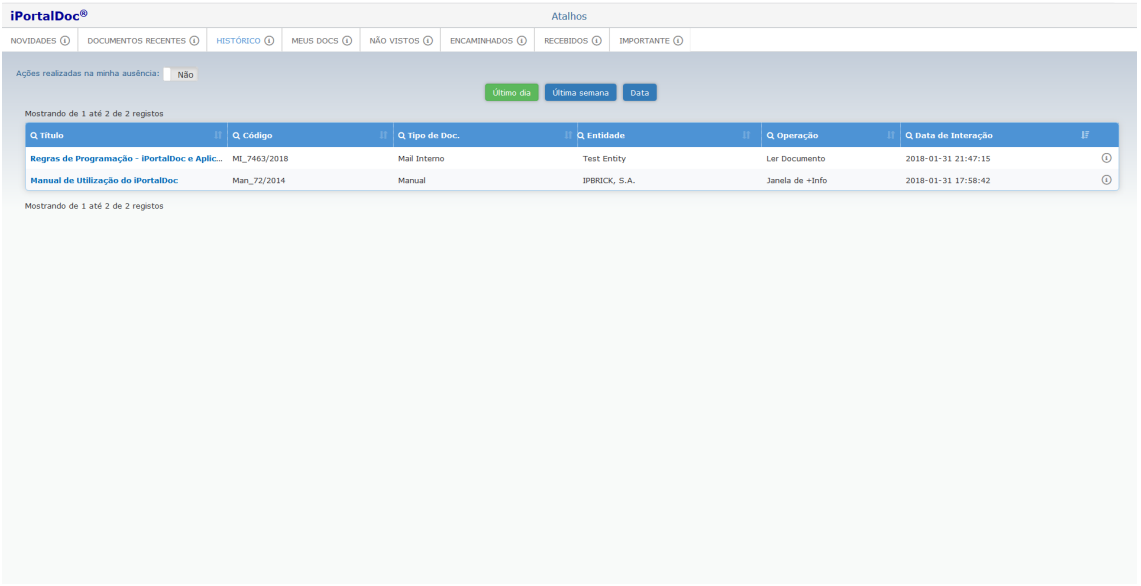


Figura 2.15: Exemplo da interface do histórico do *iPortalDoc*

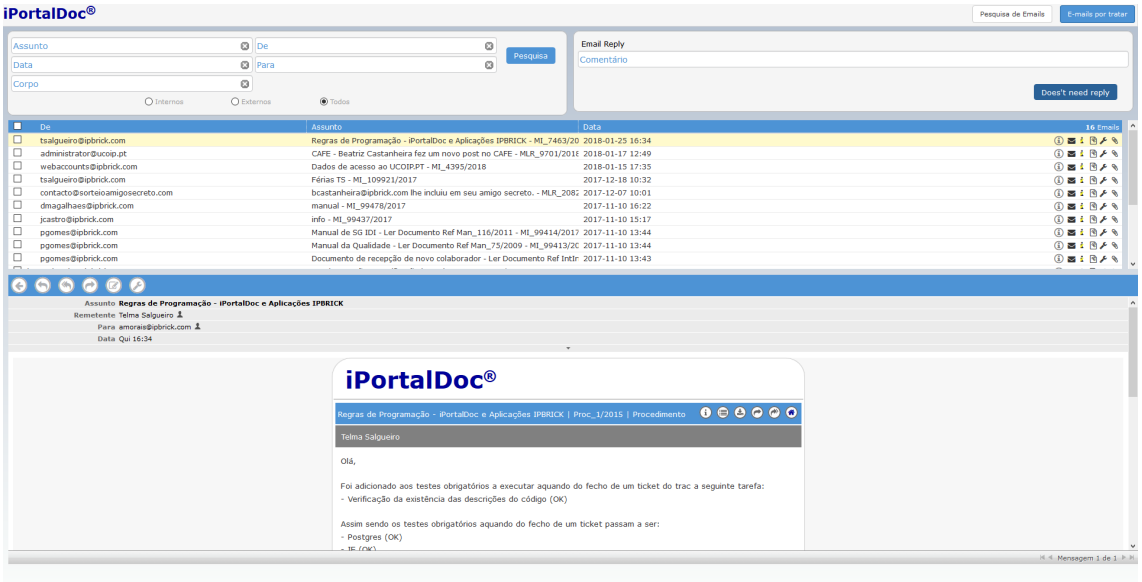


Figura 2.16: Exemplo da interface de email do *iPortalDoc*

2.3 Trabalho relacionado

As técnicas e metodologias de *ludificação* podem ser aplicadas a diferentes áreas e com diferentes propósitos. Manter a lealdade dos clientes, melhorar a aprendizagem, incutir melhores hábitos alimentares ou aumentar o envolvimento dos trabalhadores e o seu rendimento, são alguns desses propósitos. Nesta secção serão apresentados alguns exemplos de aplicação da *ludificação*, nos mais diversos contextos.

2.3.1 Microsoft's Communicate Hope

A equipa responsável por testes de garantia da qualidade na Microsoft projetou um conjunto de atividades baseadas em elementos de jogos, de forma a incentivar um elevado número de funcionários a participarem na realização de testes a novos produtos, que apesar de serem importantes, eram tediosos. No seu sistema de comunicação integrado, a Microsoft criou um programa designado por Communicate Hope, que permitia que os funcionários se juntassem a equipas que representavam instituições de caridade. Com base no número de pontos que as equipas obtiveram através da realização de testes, a empresa fez contribuições monetárias para essas instituições.

O sucesso deste sistema é comprovado pelos resultados. Mais de oito mil colaboradores participaram, ou seja, cerca de 97% do total de funcionários. É um valor considerável tendo em conta que para um programa típico, esse valor ronda entre os 50% e os 75%. Para além disso, dois terços desses funcionários, enviaram comentários e recomendações sobre o produto, que era um dos objetivos principais da empresa [8].

2.3.2 Kaplan University

A Kaplan University tinha como objetivo aumentar o envolvimento e a aprendizagem por parte dos alunos. A universidade, através da parceria com a Badgeville, implementou uma solução *ludificada*, baseada em elementos de jogo simples, os PBLs.

Os resultados iniciais do programa piloto indicaram que os alunos que participaram nesta solução, passaram até 155% do tempo mais envolvidos ativamente na sala de aula, comparativamente com os alunos que não estavam envolvidos neste sistema. Muitos dos alunos procuraram desafios adicionais de forma a ganhar mais emblemas e progredirem na classificação. Este maior envolvimento permitiu que 9% dos alunos obtivessem melhores classificações [5].

2.3.3 Bluewolf

A Bluewolf é uma empresa da IBM destinada a combinar a tecnologia da informação com os objetivos comerciais, com base no ciclo de vida do cliente. Apesar do elevado número de trabalhadores, a administração reparou que apenas uma pequena parte deles estava a espalhar a mensagem da empresa. Este facto originou uma parceria entre a Bluewolf e a Bunchball, que permitiu a introdução de um programa Going Social, que consistia em três partes: um portal, *"pack profiles"* e na peça de *ludificação*. O portal compartilhava um guia sobre como começar a usar as plataformas sociais. Os empregados podiam ganhar pontos por desenvolverem os seus *"pack profiles"*, partilhando informações sobre a Bluewolf nas plataformas sociais e fazendo *upload* de conteúdos. A peça de *ludificação*, para além dos pontos, envolvia uma loja de recompensas que permitia aos empregados trocaram os pontos por itens tangíveis, como por exemplo, peças de roupa e bilhetes para concertos.

Esta solução permitiu grandes aumentos na partilha social, aumentando o tráfego do *blog* em 81% a partir das redes sociais e do tráfego do site em 46% [8].

2.3.4 LiveOps

A LiveOps é uma empresa virtual de *call-center* que funciona sobre a metodologia *outsourcing*, ou seja, que os agentes que realizam as chamadas são trabalhadores independentes, subcontratados, e que trabalham a partir de casa. Devido ao facto de não poder gerir e controlar fisicamente os trabalhadores, a LiveOps teve de encontrar uma maneira de persuadir e envolver uma comunidade altamente distribuída, de 21 mil agentes de *call-center* independentes. Para isso, implementou um sistema *ludificado* de forma a melhorar a produtividade dos funcionários, fazendo parceria com a Bunchball, um fornecedor de plataformas *ludificadas*. A solução consistia em envolver uma comunidade competitiva, numa plataforma social motivacional, onde os colaboradores pudessem apreender novos conhecimentos, ganhar recompensas e exibir as suas conquistas.

Após uma semana, os resultados demonstraram que o tempo médio de chamada foi reduzido em cerca de 23%, enquanto que a satisfação do cliente aumentou, aproximadamente 9% [21] [8].

2.4 Sumário

Neste capítulo foram apresentados os conceitos teóricos relativos à *ludificação*. Inicialmente foram abordados os diferentes tipos de motivação, devido à sua relevância na projeção e desenvolvimento de um sistema *ludificado*. Aprofundando mais o conceito de *gamification*, foram descritos os diferentes tipos de jogador e os vários elementos que estão presentes num jogo. Relativamente ao processo de desenvolvimento de um sistema *ludificado*, foi apresentado o processo de design baseado no jogador segundo Burke e ainda a ferramenta Octalysis que, para além de poder auxiliar no desenvolvimento do sistema, permite analisar um sistema já implementado. Por fim, foram apresentados alguns casos de sistemas *ludificados*. Excluindo o caso da Kaplan University, em todos os outros, o envolvimento dos trabalhadores e o aumento do seu rendimento, faziam parte dos objetivos da organização.

Capítulo 3

Problema

Neste capítulo serão descritos os principais problemas que levam à necessidade de implementar um sistema de *ludificação* no *iPortalDoc*, bem como a apresentação detalhada da página de estatísticas, tendo em vista a sua necessidade de melhorias. Por fim, será realizada uma descrição da solução de *ludificação* anteriormente proposta, que será analisada num capítulo posterior, de forma a averiguar se de facto essa proposta é a mais indicada para o sistema.

3.1 Definição do problema

O problema desta dissertação passa por projetar e desenvolver uma solução *ludificada*, através da criação ou adaptação de interfaces já existentes, de um sistema de gestão documental e de processos, nomeadamente o *iPortalDoc*. Esta necessidade surge no sentido de ajudar a empresa a atingir determinados objetivos:

- reduzir o número de reclamações;
- aumentar o número de utilizadores da plataforma e a frequência com que a utilizam;
- atrair novas empresas ou áreas de negócio.

Contudo, estes objetivos são relativos à empresa que fornece o sistema de gestão documental e de processos, e não às empresas destinatárias da solução *ludificada*. Cada uma dessas empresas utiliza o *iPortalDoc* com os mais variados propósitos e cujos objetivos também diferem de caso para caso. Este fator impossibilita uma definição clara dos resultados do negócio e métricas de sucesso, que é o primeiro passo no processo de design baseado no jogador, descrito por Burke [3]. De facto, o problema de definição é constante ao longo das várias etapas do processo, como será constatado no capítulo seguinte. Por consequente, e pretendendo-se uma solução que possa ser adotada por qualquer empresa, independentemente da sua área de negócio e dos respetivos objetivos, o sistema *ludificado* proposto terá de ser o mais abrangente e configurável possível.

O *iPortalDoc* contém uma vasta gama de funcionalidades e interfaces, tendo sido necessário uma análise ao sistema de forma a optar pela melhor solução. Apesar de inicialmente não ter sido

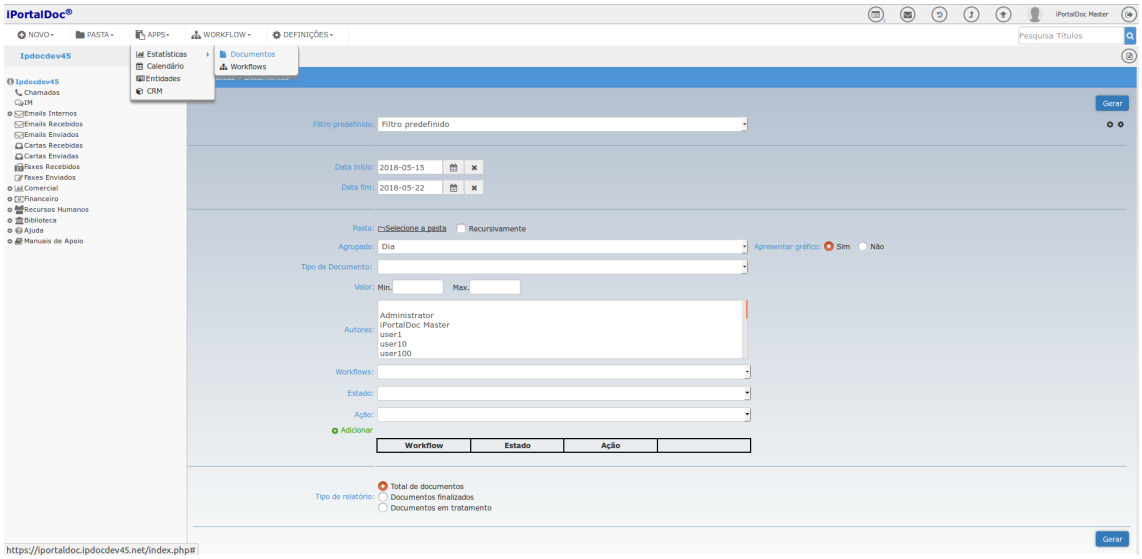


Figura 3.1: Exemplo da interface de estatísticas de documentos do *iPortalDoc*

equacionada para o processo de *ludificação*, um dos problemas detetados foi a página de estatísticas. Esta interface, como será descrito de seguida, apresenta um *layout* pouco apelativo, cujos gráficos são gerados com recurso a tecnologias já ultrapassadas (nomeadamente *PHPgraphLib*). Acrescenta-se ainda o facto de alguns dos campos já estarem desadequados face ao estado atual do *iPortalDoc*.

As estatísticas estão divididas em duas secções: documentos e workflows. A secção de documentos (ilustrada na Figura 3.1) contém diversos campos que permitem filtrar os dados que se deseja analisar. Inicialmente é possível escolher um filtro já predefinido ou, optar por editar os campos necessários, individualmente. De todos esses parâmetros destacam-se:

- **Data início/Data fim** — Permite definir o intervalo de tempo em que a data de elaboração dos documentos se insere;
- **Pasta** — Se definido, o relatório gerado é apenas referente aos documentos inseridos nesse diretório, de forma recursiva ou não, consoante a opção escolhida;
- **Agrupado/Apresentar gráfico** — Permite escolher se os documentos serão agrupados por dia ou por mês, e ainda se apresentados de forma gráfica ou não;
- **Tipo de documento** — Tal como o nome indica, restringe as estatísticas a um tipo de documento;
- **Autores** — Campo que permite selecionar quais os autores dos documentos a analisar. Podem ser selecionados múltiplos autores;
- **Workflow/Estado/Ação** — Estes campos permitem restringir os documentos a um ou mais workflows. Se os campos de *Estado* e *Ação* também estiverem definidos, o filtro é ainda

mais específico, apresentando apenas os documentos associados aquele estado/ação. Com a opção *Adicionar*, o utilizador pode incluir todos os workflows que desejar.

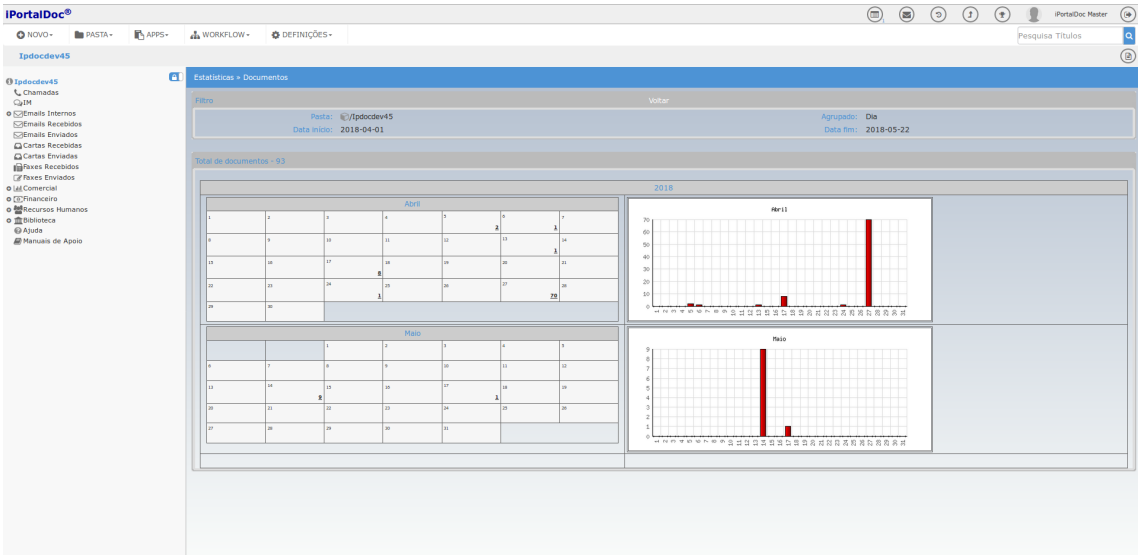


Figura 3.2: Exemplo do relatório gerado relativo às estatísticas de documentos do *iPortalDoc*

O relatório gerado pode ser relativo a todos os documentos; aos documentos já finalizados; ou apenas aos documentos em tratamento. Na Figura 3.2 é possível observar um exemplo de um relatório gerado para todos os documentos, referente a um determinado espaço de tempo. Os documentos estão agrupados por dias, e o número de documentos criados diariamente é apresentado no respetivo dia do calendário. Ao clicar nesse número, o utilizador poderá observar a lista desses documentos com a respetiva informação.

A secção de workflows (Figura 3.3) é semelhante à secção de documentos. Nesta caso, devido ao número de parâmetros ser mais reduzido, não é possível escolher um filtro predefinido. Os campos a ser preenchidos pelo utilizar são:

- **Workflow/Estados/Ações** — Estes campos definem qual o workflow que vai ser considerado para as estatísticas. Ao contrário das estatísticas de documentos, nesta situação o utilizador é obrigado a escolher um workflow e apenas um. Os campos de *Estados* e *Ações* permitem limitar ainda mais os dados a considerar, mas não são obrigatórios;

- **Utilizadores** — Campo que restringe os workflows a analisar para aqueles cujos utilizadores definidos fazem parte. Se nenhum for selecionado o sistema engloba todos os utilizadores no relatório;

- **Data início/Data fim** — Permite definir o intervalo de tempo em que o workflow se insere.

Depois de definidos os parâmetros a considerar, o sistema permite gerar diferentes estatísticas associadas ao workflow em questão, nomeadamente:

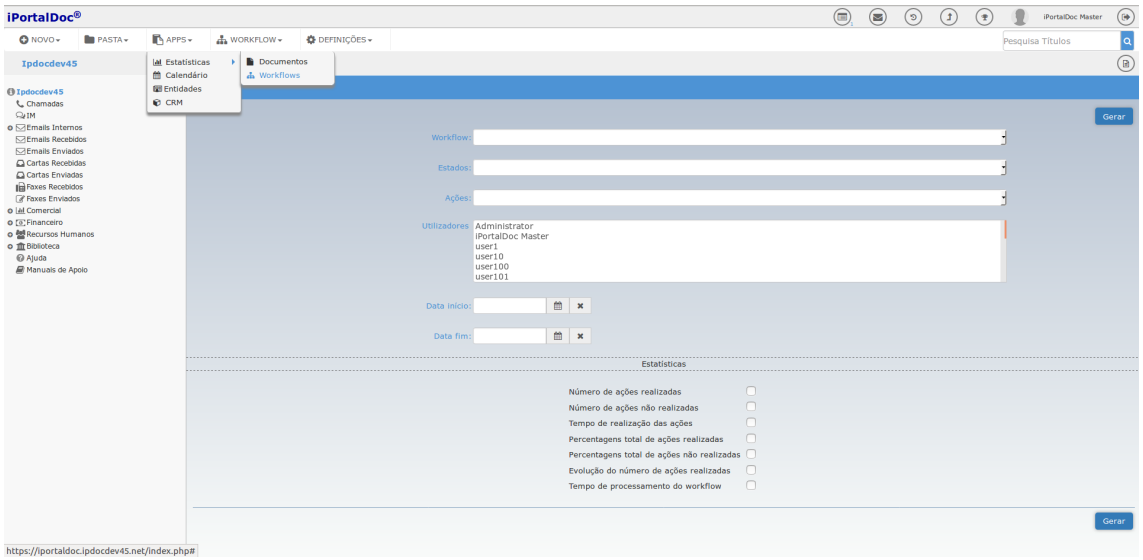


Figura 3.3: Exemplo da interface de estatísticas de workflows do *iPortalDoc*

- Número de ações realizadas;
- Número de ações não realizadas;
- Tempo de realização das ações;
- Percentagem total de ações realizadas;
- Percentagem total de ações não realizadas;
- Evolução do número de ações realizadas;
- Tempo de processamento do workflow.

Na Figura 3.4 está apresentado um exemplo de relatório que foi gerado para um *workflow* de teste. Por uma questão de melhor perceção visual, foram apenas selecionados dois dados estatísticos: o número de ações realizadas e o número de ações não realizadas. Algo a destacar é o facto de esses valores estarem apresentados em tabelas e não em gráficos como seria expectável. Desta forma, se o número de utilizadores selecionados for elevado torna-se difícil perceber, de imediato, por exemplo, quem realizou mais ou menos ações.

Posto isto, não é surpreendente que muitas empresas que utilizam o *iPortalDoc* recorram a pedidos de relatórios personalizados com dados estatísticos em vez de consultarem a pagina de estatísticas. É assim necessário adaptar a interface de estatísticas de forma a torná-la mais apelativa, fluída, interativa e, principalmente, útil para os seus utilizadores. Apesar de não ser percetível à primeira vista, a página de estatísticas tem um papel importante no conceito de *ludificação*. Isto porque permite aos administradores analisar e comparar os resultados da empresa antes e depois do sistema *ludificado* e permite também, acompanhar o impacto de pequenas alterações que possam ocorrer nesse mesmo sistema.

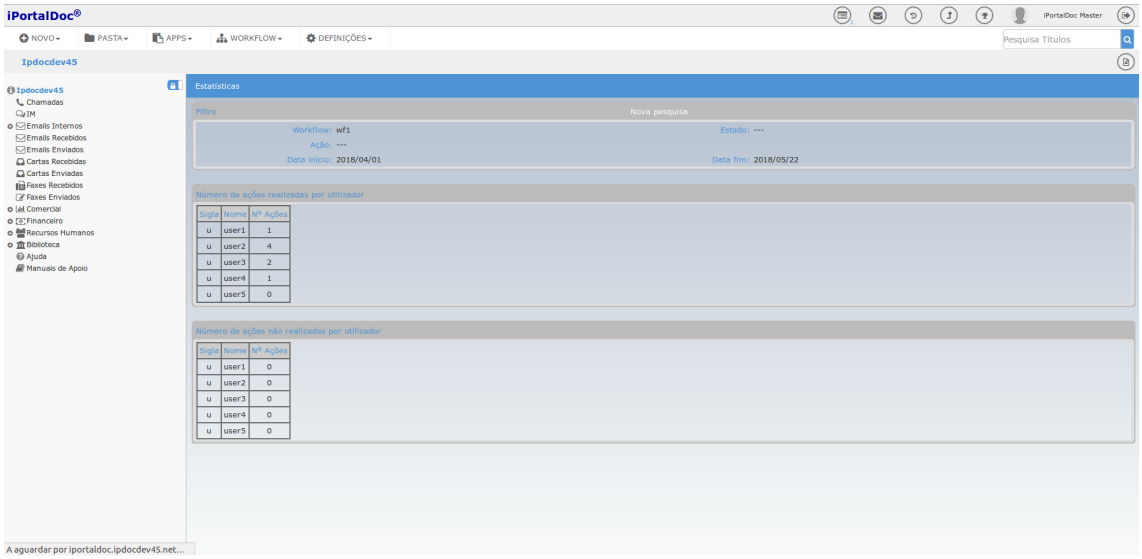


Figura 3.4: Exemplo do relatório gerado relativo às estatísticas de workflows do *iPortalDoc*

3.2 Descrição da solução *ludificada* anteriormente proposta

Como já foi referido previamente, esta dissertação é uma continuação do trabalho realizado pela Ana Rita Mendes [11]. Deste modo, um dos passos iniciais passa por analisar todo esse trabalho e a respetiva solução proposta para o sistema. A colega Ana Mendes começou por avaliar o *iPortalDoc* utilizando a estrutura de ludificação apresentada por Yu-Kai Chou [4], o Octalysis. De seguida, de forma a conhecer melhor o público alvo, realizou um conjunto de entrevistas semiestruturadas aos utilizadores do *iPortalDoc*. As questões efetuadas eram relativas ao uso pessoal do sistema, ao *feedback* dos clientes e a sugestões de melhorias. Após as entrevistas, concluiu que um dos problemas mais identificados pelos utilizadores era a complexidade do sistema, em termos do elevado número de opções e configurações nos diferentes processos, destacando a necessidade de algo que melhorasse a *user experience*. Resumindo, as principais sugestões que os utilizadores gostariam de ver implementadas eram:

- melhorar o *feedback* na ocorrência de erros;
- promover a comunicação entre equipas no que diz respeito a melhoramentos;
- possibilidade de rever a dependência de informações dos emails;
- incluir dicas de pesquisa;
- facilitar o acesso a documentos recentes.

Por fim, seguiu a metodologia desenvolvida por Werbach e Hunter [23] de forma a desenvolver uma estratégia de *ludificação*. Esta metodologia é muito semelhante ao processo de design baseado no jogador apresentada por Burke [3], que está apresentado na Revisão Bibliográfica 2, e cuja

estratégia é baseada nos objetivos do negócio onde a plataforma se insere. A solução *ludificada* proposta pela Ana Rita Mendes contempla três níveis: um conjunto de *feedbacks*; um centro de ajuda; e um painel de cooperação [11].

- **Conjunto de *feedbacks*** — Este nível inclui apenas uma ajuda não personalizada, ou seja, o *feedback* será o mesmo para qualquer utilizador, em circunstâncias idênticas. Este conjunto de *feedbacks* visam fornecer ao utilizador informações básicas sobre a funcionalidade da plataforma, informá-lo sempre que as mudanças ocorrem ao nível da interface e justificar algum erro que possa ocorrer. Este nível foi projetado para as seguintes situações:
  - o utilizador é iniciante na plataforma;
  - o software da plataforma sofreu algumas alterações que se refletem na interface com o utilizador;
  - Ocorrência de um erro de um conjunto de erros comuns.
- **Centro de ajuda** — Este nível engloba o anterior com a adição de uma área onde a ajuda é centralizada. Nesta nova secção da plataforma, o utilizador pode procurar dicas de uso ou pedir ajuda a outros utilizadores, caso não consiga encontrar a informação que pretende. A existência de um moderador era necessária de forma a haver um controlo dos *feedbacks*. Na Figura 3.5 está um exemplo de como seria a interface do centro de ajuda.

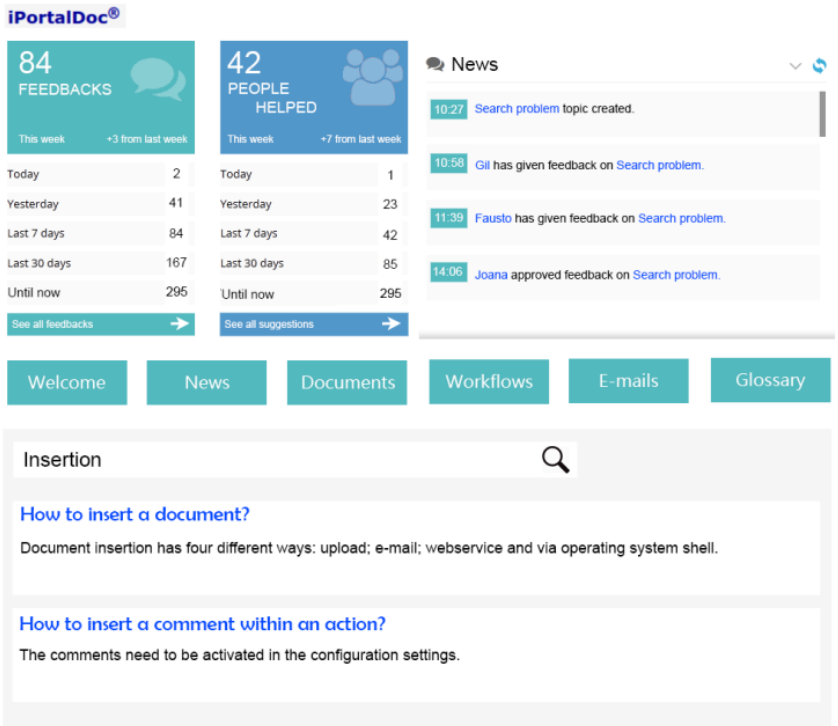


Figura 3.5: Exemplo da interface do centro de ajuda [11]

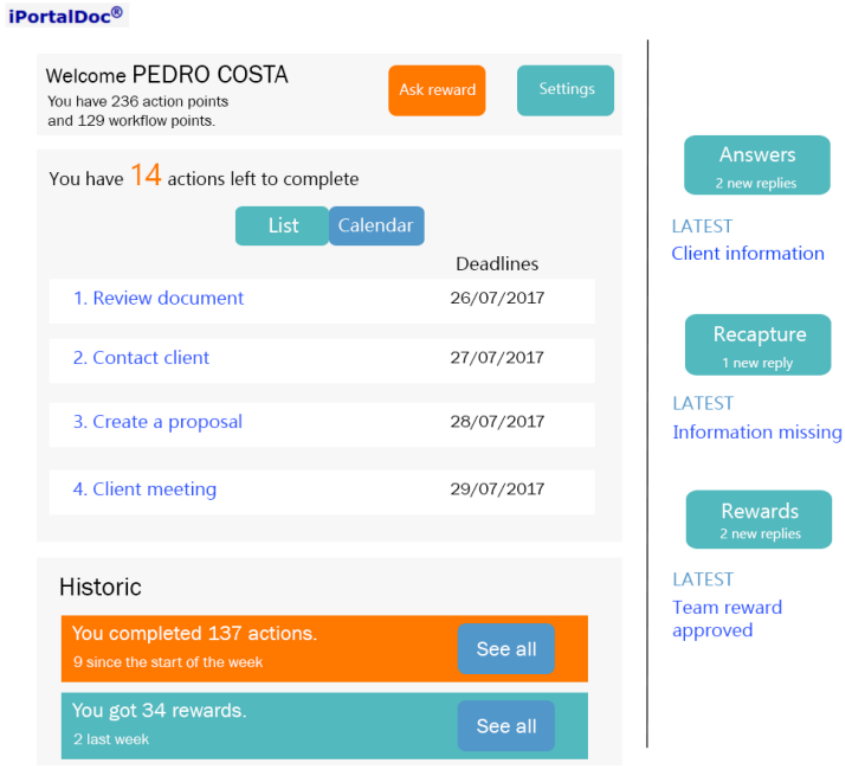


Figura 3.6: Exemplo da interface do painel de cooperação [11]

- **Painel de cooperação** — Este último nível pode ser incluído na área pessoal do sistema de gestão documental e de processos. Foi projetado de acordo com os detalhes da estrutura atual da plataforma *iPortalDoc*, que já inclui o foco para tarefas pessoais, apesar de não ter muitos componentes de promoção do espírito de equipa. Os dados estatísticos, que atualmente já se encontram na plataforma, permitem comparar o desempenho dos utilizadores em diversos parâmetros. Com base nisso, este modelo foi definido com base em alguns desses detalhes para organizar e melhorar a cooperação real entre os funcionários dentro de uma empresa.

  Neste nível existiriam três tipos de atores distintos: o elemento da equipa; o responsável do workflow; e os recursos humanos. Neste caso, designa-se por equipa o conjunto de utilizadores que têm ações num determinado workflow. O elemento da equipa pode requisitar ajuda, dar *feedback* a outros utilizadores ou realizar ações, adquirindo dessa forma pontos. Os pontos podem ser trocados por benefícios. O responsável do workflow é responsável por liderar a equipa e analisar o cumprimento de tarefas por parte dos elementos da equipa. O papel dos recursos humanos passa por tratar das burocracias relacionadas com as recompensas escolhidas pelos elementos da equipa. Na Figura 3.6 está apresentada uma interface exemplo do que seria o painel de cooperação.

3.3 Sumário

Neste capítulo foram descritos os principais problemas identificados ao *iPortalDoc* e que levaram à realização desta dissertação. A implementação de uma solução *ludificada* visa satisfazer os objetivos da empresa. Contudo, a própria projeção da solução será dificultada pelo facto de o sistema ser destinado a um vasto conjunto de empresas, com diferentes propósitos. A página de estatísticas do *iPortalDoc* foi um dos problemas detetados e que será alvo de intervenção.

Foram ainda apresentados os principais aspetos da solução *ludificada* anteriormente proposta para o sistema, que será analisada no capítulo seguinte. Este descreverá o processo que será utilizado de forma a obter uma solução para os problemas que foram identificados.

Capítulo 4

Solução

Neste capítulo será apresentada a solução *ludificada* que melhor se enquadra nos objetivos da empresa, no funcionamento do sistema, e que permitirá resolver alguns dos problemas identificados ao *iPortalDoc*. A solução será projetada seguindo o processo de design baseado no jogador. Para além da descrição das várias etapas desse processo relativamente ao *iPortalDoc*, este capítulo contempla também uma análise à solução *ludificada* anteriormente proposta, justificando eventuais opções tomadas para a solução final. Será efetuada uma descrição detalhada da proposta, com recurso a diagramas de casos de uso e a *mockups*, terminando o capítulo com a comparação entre a proposta anterior e a proposta apresentada nesta dissertação.

4.1 Processo de design baseado no jogador

4.1.1 Resultados do negócio e métricas de sucesso

Definir os resultados do negócio e as métricas de sucesso é o primeiro passo do processo de design baseado no jogador apresentado por Burke [3]. Para o caso deste sistema de gestão documental e de processos, existem duas abordagens possíveis para a projeção da solução *ludificada*. Uma em que a solução é elaborada tendo em vista um cliente (empresa) específico, o que implica um novo processo de design para cada situação. Neste caso, os encargos financeiros para o cliente seriam substancialmente maiores. A outra abordagem passa por projetar uma solução capaz de ser adotada por qualquer empresa, independente do seu ramo de negócio.

A opção desta dissertação recaiu sobre a segunda abordagem, isto porque, vai de encontro aos objetivos da empresa IPBRICK SA, permitindo captar a atenção de novos clientes e áreas de negócio, algo que a primeira abordagem levaria mais tempo a satisfazer. Contudo, ao contrário do que é especificado por Burke, a abordagem escolhida impossibilita a definição de objetivos claros para a solução. Sendo o sistema destinado ao comércio B2B, e tendo em conta as suas funcionalidades, podem ser considerados os seguintes objetivos gerais (que são comuns à maioria das empresas):

- aumentar a motivação e a envolvência dos funcionários com a empresa;

- aumentar o espírito de equipa e de colaboração;
- aumentar a produtividade.

Mas objetivos concretos como, por exemplo, reduzir o tempo médio de elaboração de um orçamento ou diminuir o número de produtos devolvidos, não podem ser definidos e considerados neste contexto. Isto implica que a projeção do sistema seja realizada de uma forma mais abrangente e capaz de ir ao encontro das necessidades das mais diversas empresas que utilizam o *iPortalDoc*.

4.1.2 Público alvo

No que diz respeito à definição do público alvo, os utilizadores do *iPortalDoc* estão classificados com vários perfis. Esses perfis que podem ir, por exemplo, de um leitor absoluto a um super utilizador, são caracterizados pelo tipo de permissões que possuem para aceder às diversas diretorias, documentos e aplicações da plataforma. Mas de uma forma mais geral, podemos classificar os utilizadores em dois tipos: os utilizadores de sistema e os utilizadores finais.

- **Utilizadores de sistema** — São normalmente pessoas da administração e responsáveis por ajustar o sistema para os processos da empresa. As tarefas destes utilizadores passam normalmente por conceber, instanciar e configurar workflows, e ainda por controlar as permissões dos restantes utilizadores do sistema;
- **Utilizadores finais** — Estes utilizadores representam a maior percentagem de funcionários da empresa, sendo os principais intervenientes no processo de criação de valor;

Os objetivos mais globais definidos no passo anterior têm como foco principal os utilizadores finais, devido à sua preponderância na produtividade da empresa. Mas, pelo facto da solução pretendida ter de ser o mais abrangente possível, os utilizadores de sistema irão assumir um papel importante no que diz respeito à configuração do próprio mecanismo de *ludificação*.

A exploração do público alvo já foi algo abordado pela Ana Mendes aquando da realização do conjunto de entrevistas já referido anteriormente. Essas entrevistas permitiram a criação de diferentes *player personas* que agrupam os utilizadores do *iPortalDoc* pelas características comuns que os representam. Os *player personas* definidos pela Ana Rita Mendes são [11]:

1. ***Player persona* 1**
    - **Nome:** Clara Rodrigues
    - **Idade:** 55
    - **Cargo:** Administradora numa Câmara Municipal
    - **Principais funções:** Procedimentos financeiros: processar faturas, consultar documentos, enviar documentos para os clientes, etc.

- **Necessidades pessoais:** Processos metódicos e repetitivos para completar as suas tarefas diárias.
- **Principais frustrações:** Clara trabalhou com um sistema de documentos em papel e tem dificuldades para se adaptar às novas tecnologias. Neste momento está habituada ao *iPortalDoc*, mas tem medo de documentos não classificados. Qualquer alteração ao seu processo de pesquisa regular ou na inserção do documento traz-lhe ansiedade, não conseguindo trabalhar de forma adequada.

2. ***Player persona* 2**

   - **Nome:** Joana Sousa
   - **Idade:** 35
   - **Cargo:** Comercial numa empresa de soluções energéticas
   - **Principais funções:** Contactar com o cliente, compartilhar documentos com ele e lidar com vários tipos de documentos provenientes de diferentes utilizadores.
   - **Necessidades pessoais:** Encontrar os documentos que passaram por ela através de palavras-chave simples.
   - **Principais frustrações:** Joana ainda não está adaptada ao *iPortalDoc*. Nem sempre tem a lista completa de parâmetros de um documento e, devido a este motivo, acha difícil encontrar um documento com o mecanismo de pesquisa atual.

3. ***Player persona* 3**

   - **Nome:** Tiago Silva
   - **Idade:** 38
   - **Cargo:** Técnico de suporte numa empresa de soluções de aquecimento
   - **Principais funções:** Tiago tem uma função técnica e geralmente entra em contacto com os clientes, pessoalmente ou através telefonemas.
   - **Necessidades pessoais:** Ele está adaptado a sistemas intuitivos.
   - **Principais frustrações:** Tiago trabalha na empresa há muito tempo, mas não necessita de usar o sistema de gestão documental e de processos regularmente. Neste momento, precisa de introduzir um novo documento para aprovação, mas não sabe como funciona esse workflow.

4.1.3 Objetivos do jogador

Neste passo é necessário estabelecer metas para os utilizadores, que envolvam as motivações pessoais e os objetivos da solução definidos pela empresa. Em termos globais foi definido, no primeiro passo, que os objetivos da solução passam por aumentar o envolvimento dos seus utilizadores com o *iPortalDoc* e, dessa forma, aumentar o rendimento e a produtividade da empresa.

Tendo estes objetivos em consideração e ainda a análise do público alvo realizado no passo anterior, é possível definir os seguintes objetivos do jogador:

- aumentar o conhecimento relativo ao *iPortalDoc*;
- aumentar a frequência de utilização;
- diminuir o tempo despendido na navegação dentro do sistema.

Estes objetivos vão de encontro às necessidades identificadas pelos utilizadores do *iPortalDoc* e, caso a solução *ludificada* permita que estas metas sejam atingidas, por consequente, os objetivos da solução serão também concretizados.

4.1.4 Modelo de envolvimento do jogador

Definir o modelo de envolvimento do jogador é a primeira fase de decisões relativas à estrutura da solução *ludificada*. Os diversos parâmetros do modelo terão de ser combinados da melhor forma, tendo em conta os objetivos do jogador, anteriormente definidos.

- **Colaborativo/Competitivo** — No *iPortalDoc* os utilizadores estão, normalmente, agrupados por grupos. A solução proposta poderá, desta forma, consistir num equilíbrio entre os dois fatores: colaboração dentro dos grupos e competição entre grupos;
- **Intrínseco/Extrínseco** — Este parâmetro irá sempre depender da empresa que está a usar o sistema. Desta forma, a melhor solução passa por um sistema cujas recompensas são especificadas por uma pessoa dentro da empresa, mais concretamente, por um utilizador de sistema, e que podem variar diariamente, semanalmente, mensalmente, etc. Para isso, a plataforma terá de ter uma secção administrativa que permita a gestão dessas mesmas recompensas;
- **Multijogador/Solitário** — Sendo o *iPortalDoc* baseado em workflows realizados por um grupo de pessoas, as interações *multiplayer* irão estar presentes na solução. Contudo, uma vertente mais solitária, com recompensas por ações individuais, permite um maior foco do utilizador pelas suas tarefas;
- **Campanha/*Endless*** — É desejado que o sistema projetado perdure nas respetivas empresas, sendo, para isso, necessário um sistema infinito. A partir do momento em que a solução for adotada, a progressão dos utilizadores terá de ser mantida;
- **Emergente/*Scripted*** — Um sistema com *scripts* é a melhor escolha para uma solução *ludificada*. Desta forma, os utilizadores são levados a seguir um comportamento estipulado e que é o pretendido para atingir os objetivos.

4.1.5 Espaço de jogo e planeamento da jornada

O espaço de jogo terá de ser algo que seja incorporado no atual sistema de gestão documental e de processos. Irá consistir numa nova secção que permita aos utilizadores consultar o seu perfil, visualizar a sua progressão e eventuais conquistas, bem como as atividades dos restantes utilizadores.

Ao contrário do espaço de jogo, o planeamento da jornada é algo difícil de especificar devido à abordagem que foi adotada para a projeção deste sistema. Desta forma, e tal como as recompensas, a jornada terá de ser algo configurável por um utilizador de sistema.

4.1.6 Economia do jogo

Definir a economia do jogo é a última etapa na projeção da solução *ludificada*. Neste passo, são definidos os elementos de jogo que serão implementados. Sendo o ponto de partida desta solução, e ainda o facto de ter de ser a mais abrangente possível, o sistema *ludificado* irá ser maioritariamente composto por elementos básicos da hierarquia dos elementos de jogo (2.4): os componentes.

- **Diversão** — Relativamente à "moeda" diversão, numa solução *ludificada* não é possível dar uma maior experiência ao utilizador com recurso a efeitos de luz e som, e com gráficos avançados, como acontece nos videojogos. Nesta situação, o divertimento será com base, por exemplo, em interfaces apelativas, com recurso a avatares e *nicknames*;

- **Coisas** — As "coisas" neste sistema englobam os pontos e as recompensas. Ambos os elementos terão de ser configurados pelos utilizadores do sistema, consoante as empresas e as jornadas pretendidas para os utilizadores;

- **Auto-estima** — O interesse no sistema por parte do utilizador pode ser mantido com recurso à construção da sua auto-estima. Essa construção baseia-se nos elementos progressão e níveis, que causam a sensação de acompanhamento por parte do sistema, mantendo os utilizadores envolvidos;

- **Capital social** — A motivação do jogador pode ser causada pelo reconhecimento das suas conquistas por parte dos outros jogadores. Nesta solução, emblemas e tabelas de classificação são elementos que fazem parte desta "moeda" da economia do jogo.

4.1.7 Teste e iteração

Este é o último passo no processo de design baseado no jogador. É de salientar que um sistema *ludificado* nunca está finalizado e requer sempre adaptações e novas funcionalidades. A solução proposta, que será descrita mais à frente neste capítulo, não envolve muitos elementos complexos, como dinâmicas e mecânicas, procurando numa primeira fase adaptar-se ao sistema, sem grandes alterações do fluxo de trabalho que se espera que o utilizador realize.

4.2 Análise à solução *ludificada* anteriormente proposta

- **Conjunto de *feedbacks***

  O conjunto de *feedbacks* é o primeiro nível da solução proposta pela Ana Rita Mendes [11]. Este nível é de facto muito importante se forem consideradas as sugestões dos diversos utilizadores do *iPortalDoc* e dos benefícios que lhes traria. Atualmente, caso um utilizador não saiba qual a funcionalidade a executar ou como fazer algo, este precisa de consultar os manuais ou questionar alguém que saiba. O facto de o utilizador estar muitas vezes "perdido" no sistema, aumenta a sua ansiedade e reduz a sua motivação, implicando um decréscimo do seu rendimento. Por estes motivos, o conjunto de *feedbacks* é algo que estará presente na solução proposta.

- **Centro de ajuda**

  Relativamente ao centro de ajuda proposto pela Ana Mendes, este nível não será considerado como fundamental para a base da solução *ludificada*, apesar das suas potencialidades para o sistema. Isto porque, com esta funcionalidade, o foco dos utilizadores recairia na participação neste centro de ajuda, e não na realização das suas ações. Por outro lado, numa primeira fase, o conjunto de *feedbacks* é a solução indicada para auxiliar os utilizadores menos habituados ao *iPortalDoc*. Caso os utilizadores não ficassem esclarecidos com o *feedback* ou com alguma questão mais particular, a IPBRICK SA tem ainda ao dispor uma wiki que, se melhor explorada, permitiria ajudar os utilizadores nessas situações.

- **Painel de cooperação**

  O painel de cooperação é o último nível da solução *ludificada* proposta pela Ana Mendes. Este nível contempla todos os anteriores, tornando assim a solução mais complexa. Apesar do centro de ajuda não ser importante nesta fase inicial, as restantes características do painel de cooperação são tidas em conta na nova proposta de solução, que será apresentada na secção seguinte. Nomeadamente, a elaboração de uma área pessoal, com os dados e o progresso do utilizador, é essencial, como em qualquer outro sistema *ludificado*. Para além disso, este nível torna-se relevante pelo facto de ter sido projetado de acordo com a estrutura atual do *iPortalDoc*, que será um dos focos desta dissertação. No entanto, alguns aspetos terão de ser revistos, como é o caso da definição das equipas e do responsável do workflow, que não é a mais adequada face à existência do conceito de grupos no sistema atual.

4.3 Proposta de solução

A solução *ludificada* proposta para o sistema de gestão documental e de processos, *iPortalDoc* surge de um longo estudo ao tema em questão, do próprio sistema e dos seus utilizadores, bem como da análise da proposta já anteriormente apresentada. Procura combater as lacunas identificadas ao sistema, mas não só. Esta solução contempla três áreas: um conjunto de *feedbacks*; uma área de jogo; e uma página de estatísticas. Enquanto que o conjunto de *feedbacks* e a página de

estatísticas surgem dos problemas identificados pelos utilizadores e da análise ao sistema, a área de jogo é proposta tendo em vista um maior envolvimento por parte dos funcionários da empresa com a plataforma.

4.3.1 Conjunto de *feedbacks*

O conjunto de *feedbacks* é proveniente da solução apresentada pela Ana Rita Mendes [11] e, tal como já foi descrito anteriormente, visa auxiliar os utilizadores nas seguintes situações:

- quando o utilizador é iniciante no sistema;
- quando existem alterações à interface do sistema devido à alteração ou implementação de novas funcionalidades;
- quando ocorre um erro de um conjunto de erros comuns.

Os *feedbacks* e a forma como são apresentados ao utilizador podem variar consoante a situação. Se pensarmos num novo utilizador, seria útil haver um *tour* entre as diversas divisões da página, com uma breve descrição para cada uma. Neste caso, esse *feedback* seria apenas apresentado na primeira vez que o utilizador acedesse à página. Caso pretendesse rever a ajuda, isso seria possível através de um atalho que iniciaria novamente esse *tour*. Este processo pode ser também adotado para o caso de existirem *updates*, apresentando sequencialmente as alterações realizadas à interface. Relativamente à ocorrência de erros, caso estes façam parte de um conjunto de erros comuns, como o erro 404 - página não encontrada, seria apresentada uma breve justificação do erro, com recurso a elementos divertidos de forma a minimizar o impacto do erro na experiência do utilizador.

4.3.2 Área de jogo

O foco desta dissertação passa pela área de jogo. A ideia principal consiste na atribuição de pontos aos utilizadores após a conclusão de determinadas ações ou workflows. Desta forma, o sistema atual mantém as mesmas características, e o fluxo normal do trabalhador não é afetado, mas sim estimulado de forma a progredir o mais rápido possível. A aquisição de pontos permite a progressão do jogador e a conquista de emblemas, bem como, possíveis recompensas. Contudo, como já foi referido, a definição dos níveis, dos pontos, das conquistas e das recompensas, tem de ser algo configurável para que a solução possa ser adotada pelos diversos clientes da IPBRICK SA. Na Figura 4.1 está apresentado o diagrama UML com os diversos casos de uso para a área de jogo.

- **Atores**
  - **Utilizador:** Este ator representa todo o conjunto de utilizadores finais do sistema de gestão documental e de processos. Tal como já foi referido, a solução *ludificada* é projetada com o foco nestes atores, que irão ser premiados pelas suas ações;



Figura 4.1: Diagrama de casos de uso para a área de jogo

– **Administrador:** O administrador envolve o conjunto restrito de utilizadores de sistema. Estes atores, para além de serem utilizadores finais, com a mesma envolvência na área de jogo, têm também permissões que lhes permitem configurar o sistema ludificado. São responsáveis por estipular o espaço de jogo e a jornada do jogador, com base nos objetivos da própria empresa;

– **Diretor:** O diretor representa o elemento com mais responsabilidades dentro do conselho administrativo de uma organização. Desta forma, é necessária a sua aprovação nas recompensas que irão ser disponibilizadas aos utilizadores;

  - **Recursos humanos:** É claro que os recursos humanos têm diversas tarefas dentro da empresa, e que utilizam o *iPortalDoc* para várias situações. Contudo, neste contexto de *ludificação*, são apenas responsáveis pelo processo de atribuição de recompensas.

- **Casos de uso**

  - **Realizar ação:** O utilizador pode realizar uma ação, de um determinado workflow, ao qual está associado um documento;
  - **Verificar estado do workflow:** Verificar se a realização da ação culmina com o estado final do workflow;
  - **Atribuir pontuação:** Os workflows e as ações podem ter pontos associados que, aquando da sua realização, são atribuídos ao utilizador. No caso de workflows, os pontos são atribuídos a todas as pessoas que estiveram envolvidas no workflow, e não apenas à pessoa que realizou a última ação;
  - **Atualizar conquistas:** Quando um utilizador realiza uma ação e, consequentemente, angaria pontos, é necessário verificar se este alcançou alguma conquista;
  - **Resgatar recompensa:** Com a aquisição de pontos e conquistas, os utilizadores vão desbloqueando recompensas que podem ser resgatadas quando o desejarem;
  - **Processar recompensa:** Algumas das recompensas podem requerer processos burocráticos, que terão de ser tratados pelos recursos humanos;
  - **Visualizar dados de jogo:** Caso de uso em que o utilizador pretende consultar informações relativas ao ambiente *ludificado*;
  - **Consultar progressão:** O utilizador pode consultar o estado da sua progressão, nomeadamente os seus pontos e o seu nível;
  - **Consultar conquistas:** O utilizador poderá consultar as suas conquistas, sempre que o desejar;
  - **Consultar classificações:** Consultar as classificações da área de jogo, que permitem comparar o estado atual do utilizador face aos restantes companheiros;
  - **Visualizar conquistas dos colegas:** Através da área de jogo, o utilizador poderá ver os últimos emblemas conquistados pelos outros utilizadores;
  - **Alterar dados pessoais:** O utilizador poderá alterar os seus dados relativos à área de jogo;
  - **Alterar avatar:** Escolher um avatar dentro de um conjunto previamente definido;
  - **Alterar *nickname*:** Caso de uso em que o utilizador altera o seu *nickname*;
  - **Definir recompensas:** O administrador poderá definir recompensas de forma a motivar ainda mais os utilizadores do sistema. Essas recompensas variam consoante a empresa e as suas possibilidades, podendo ser intrínsecas ou extrínsecas;

- **Aprovar recompensas:** As recompensas definidas pelo administrador terão de ter o aval do diretor, para serem disponibilizadas aos utilizadores;
- **Definir quais os workflows e ações a serem pontuados:** O administrador poderá escolher quais os workflows e ações que pretende incluir no processo de *ludificação*;
- **Configurar pontuações:** Caso de uso em que o administrador altera as pontuações das ações e workflows. Essas pontuações serão atribuídas ao utilizador caso este realize uma ação ou que esteja envolvido num workflow que chegou ao seu estado final;
- **Configurar emblemas:** Possibilidade de o administrador alterar, adicionar ou remover emblemas da área de jogo;
- **Alterar ícone do emblema:** Ao emblema está associado um ícone que poderá ser alterado pelo administrador;
- **Alterar detalhes:** O administrador poderá alterar informações relativas ao nível ou ao emblema, como por exemplo: nome, descrição, etc.;
- **Definir pontuação a alcançar:** Caso de uso em que o administrador define a pontuação necessária para atingir um certo nível, ou para conquistar algum emblema;
- **Configurar níveis de jogo:** O administrador poderá alterar a progressão do sistema *ludificado*, configurando os diversos níveis de jogo;
- **Alterar número de níveis:** Possibilidade do administrador adicionar ou remover níveis à progressão do sistema.

É percetível pelos diferentes casos de uso apresentados que, a área de jogo, para além da interface destinada aos utilizadores finais, terá de contemplar ainda uma interface de configuração, de forma a que os diversos parâmetros de *ludificação* possam ser configurados. Na Figura 4.2 está o esboço do que será a interface da área de jogo. Do lado esquerdo encontra-se o avatar e o *nickname* do utilizador, bem como o nível e estado da progressão. A alteração do avatar e do *nickname* poderá ser efetuada ao *clicar* em cima de cada um, respetivamente. Do lado direito estão as secções das últimas conquistas e da comunidade, que, tal como os nomes indicam, apresentarão as últimas conquistas, quer por parte do próprio utilizador, quer por parte dos seus colegas. A partir desta interface, o utilizador poderá ainda aceder à página de classificações e de recompensas. A área de jogo será apresentada numa nova janela e poderá ser acedida através de um ícone no canto superior direito, a partir de qualquer interface do *iPortalDoc*.

No que diz respeito à interface de configuração, o *iPortalDoc* já contém uma secção destinada, que poderá ser acedida a partir do menu definições. A interface que é relativa à área de jogo irá conter, numa fase inicial, duas divisões. Uma para as configurações dos níveis e outra para as configurações dos pontos.

O esboço do que será a página de configuração dos níveis encontra-se na Figura 4.3. O administrador poderá, desta forma, adicionar ou remover níveis, alterar os seus nomes e os pontos necessários para atingi-los. Para este último aspeto, e tendo em vista um elevado número de níveis, optou-se por adicionar progressões, que permitem configurar os pontos de uma forma automática.

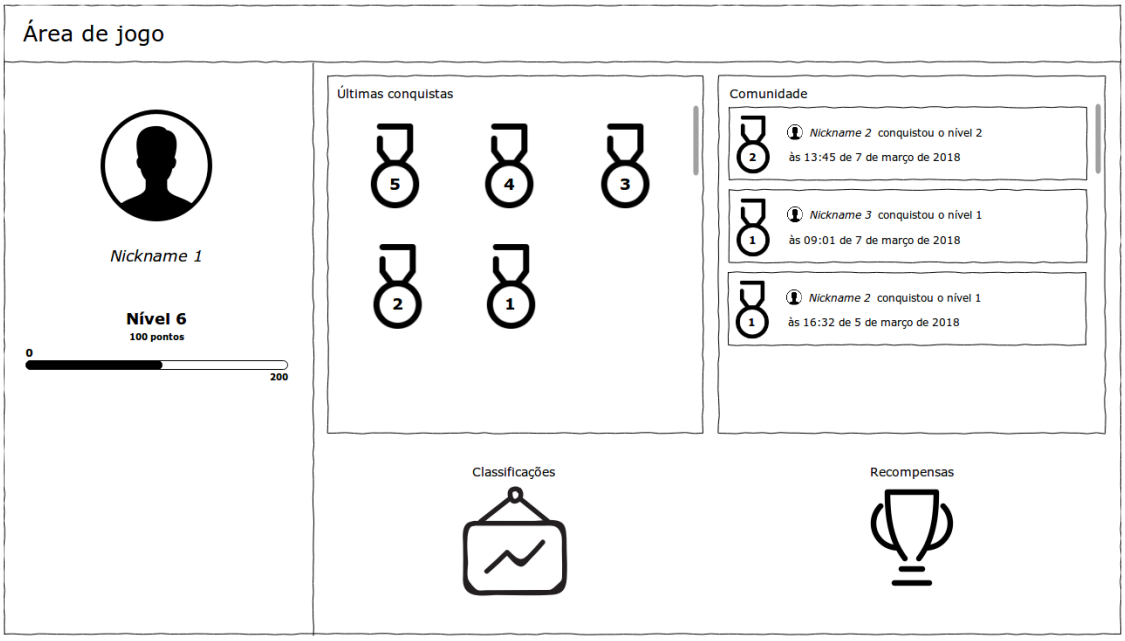


Figura 4.2: *Mockup* da interface da área de jogo

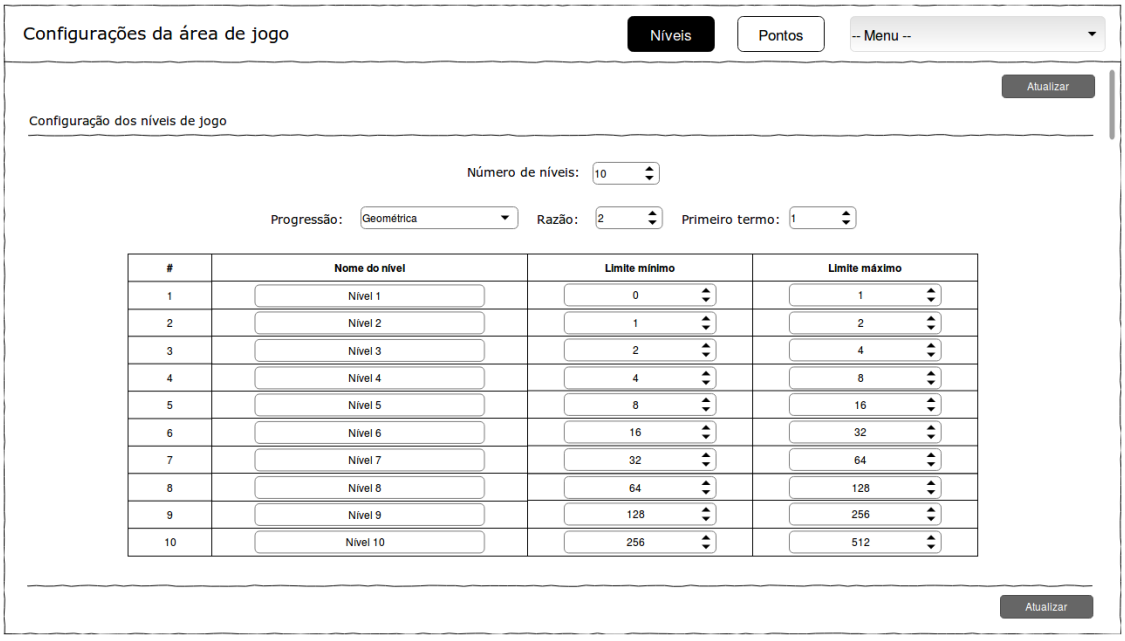


Figura 4.3: *Mockup* da interface de configuração de níveis da área de jogo

Desta forma existirão três tipos de progressões: manual, aritmética, e geométrica. A manual permitirá que o administrador altere os valores da forma que desejar. A progressão aritmética é uma sequência numérica a_n (com $n \in N$), cujo n-ésimo termo pode ser definido por:

$$a_n = a_1 + (n-1) \cdot r, \tag{4.1}$$

em que a_1 é o primeiro termo e r é a razão. Relativamente à progressão geométrica, também é uma sequência numérica b_n (com $n \in N$), sendo o n-ésimo termo definido por:

$$b_n = b_1 \cdot q^{n-1}, \tag{4.2}$$

sendo b_1 o primeiro termo e q a razão da progressão.



Figura 4.4: *Mockup* da interface de configuração de pontos da área de jogo

Para concluir a solução proposta para a área de jogo, está apresentada na Figura 4.4 o esboço da página de configuração dos pontos. O administrador poderá selecionar, através das *checkboxes*, quais os *workflows* que irão contribuir para a pontuação dos utilizadores na área de jogo. Na coluna dos pontos, poderá ser alterado o valor pontual que irá ser atribuído aos utilizadores que participaram no *workflow*, aquando da sua finalização. Caso o *workflow* contenha ações, ao *clicar* no nome do *workflow* irá abrir uma *popup* (Figura 4.5) que permitirá definir quais as ações que irão ser contabilizadas para a área de jogo, bem como a respetiva pontuação. No caso dos *workflows*/ações não selecionados, a pontuação será zero.

4.3.3 Página de estatísticas

Como foi descrito no capítulo anterior, um dos problemas detetados ao sistema *iPortalDoc* foi a página de estatísticas. A solução proposta nesta dissertação procura reformular por completo essa página, quer em termos de *layout*, quer em termos de funcionalidades. O objetivo desta

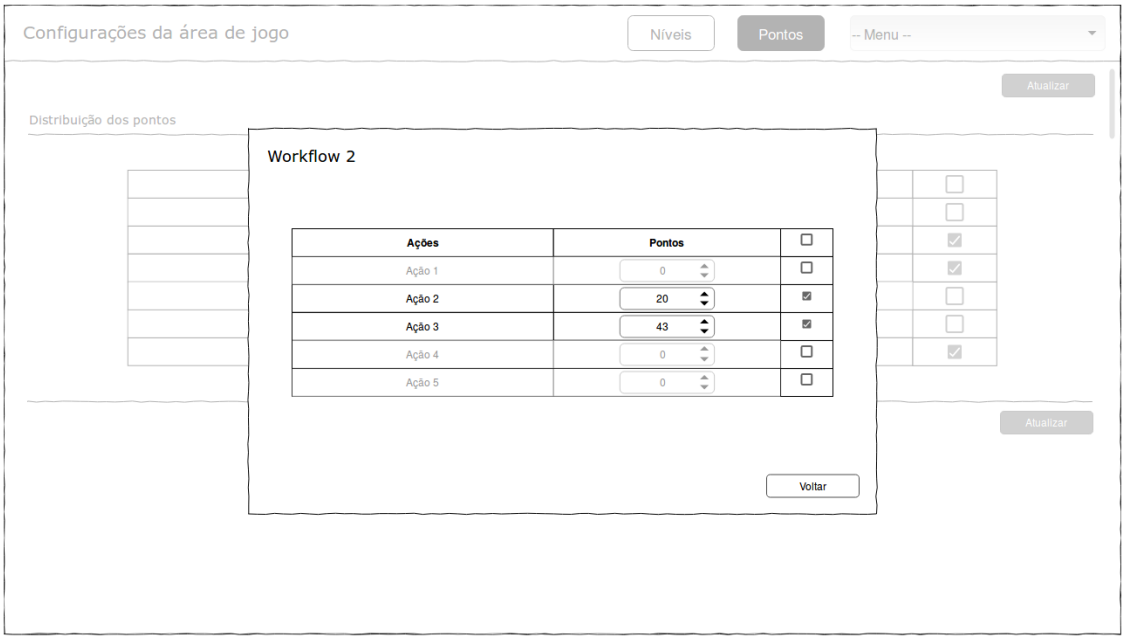


Figura 4.5: *Mockup* da interface de configuração de pontos da área de jogo relativos às ações

parte da solução não passa por analisar quais os parâmetros estatísticos que mais se adequam ao *iPortalDoc*, mas sim modificar a interface e atualizar alguns dos parâmetros que já fazem parte das estatísticas atuais, criando uma base para adição de novos parâmetros no futuro.

Existem dados estatísticos relativos aos documentos e aos *workflows* e que, no sistema atual, estão divididos em duas páginas *web*. A primeira alteração passa por agregar todas as estatísticas numa única página, com diversas divisões, que podem ser acedidas através de uma barra de navegação, no topo superior da página. Desta forma, tem-se uma divisão para os documentos e outra para os *workflows*. Nesta proposta de solução, a atenção recairá sobre as estatísticas dos *workflows*, sendo a divisão dos documentos considerada para trabalho futuro.

Ao contrário do que está implementado atualmente, pretende-se um sistema mais responsivo, de forma a que qualquer alteração dos filtros disponíveis implique uma atualização automática dos dados apresentados. Sendo assim, o conceito de gerar um relatório (ou seja, apresentar os resultados numa outra página) deixará de existir, tornando a interface mais fluída e interativa. Os dados serão apresentados, maior parte das vezes, graficamente, permitindo uma melhor análise dos dados pelos utilizadores. Outra melhoria desta solução é o facto de não restringir os dados estatísticos a um único *workflow*, recorrendo ao que ao longo deste documento será referido como hierarquia.

A hierarquia consiste num conjunto de níveis pelos quais os dados são apresentados. Imagine que se pretende saber o número de ações realizadas por um utilizador. Atualmente o *iPortalDoc* não permite fornecer este dado estatístico, isto porque os utilizadores são obrigados a indicar um *workflow*. Mas com a nova funcionalidade, será possível observar esse parâmetro quer em função de utilizadores, quer em função de *workflows*. Em termos de interação com o sistema, será

possível navegar pelos diversos níveis, com mais ou menos detalhe, através do *clique* nos diversos elementos do gráfico. Os níveis, bem como os tipos de gráfico, poderão variar consoante o dado estatístico a apresentar. Nesta dissertação irão ser projetados apenas os parâmetros: número de ações realizadas e número de ações não realizadas. Isto porque, tal como já foi referido, o objetivo é idealizar e implementar uma base para esta nova versão da interface de estatísticas, sendo que os restantes parâmetros podem ser implementados com pequenas adaptações à abordagem que será aqui detalhada.

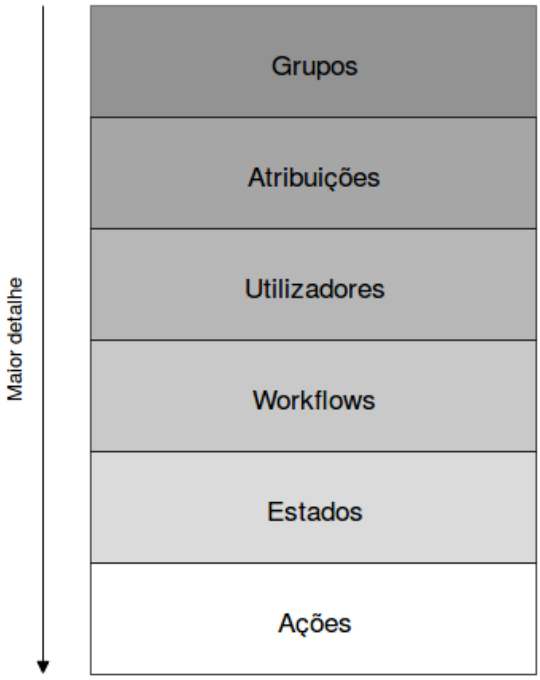


Figura 4.6: Hierarquia de níveis para as estatísticas de *workflows*

Tendo em vista a apresentação do número de ações realizadas e o número de ações não realizadas, definiu-se os seguintes níveis da hierarquia (Figura 4.6):

- **Grupos** — Este é o primeiro nível a ser apresentado ao utilizador. Caso esteja selecionado, o parâmetro estatístico surge em função dos grupos, por exemplo, *Grupo 1* com treze ações realizadas e *Grupo 5* com dez ações realizadas;

- **Atribuições** — As atribuições representam o segundo nível da hierarquia e, juntamente com os grupos, são um dos novos detalhes apresentados por esta solução. As estatísticas atuais não distinguem ações que são direcionadas a um utilizador específico, de ações que são destinadas a um grupo de pessoas. Este nível permite fazer essa distinção;

- **Utilizadores** — Neste nível os parâmetros são apresentados em função dos diversos utilizadores;

- ***Workflows*** — Neste caso, o número de ações realizadas ou não realizadas surgem em função dos workflows;
- **Estados** — Neste nível o dado estatístico é apresentado em função dos estados;
- **Ações** — O nível com mais detalhe são as ações. Com este nível, será possível observar os dados em função das ações.

Sempre que se navega em direção ao nível com maior detalhe, os dados apresentados nesse nível são referentes ao caminho escolhido. Por exemplo, se o nível atual for o nível dos utilizadores, e chegou-se até esse nível através do *clique* no *Grupo 1* e *Ações individuais*, serão apresentados o número de ações individuais realizadas pelos utilizadores do *Grupo 1*. O utilizador tem a possibilidade de definir os níveis que fazem parte da hierarquia. Se o nível não estiver escolhido, este será ignorado pela navegação, avançando para o nível seguinte, caso este exista. Se houver uma alteração de um nível com menor detalhe comparativamente ao nível atual, o gráfico será automaticamente atualizado para esse nível.

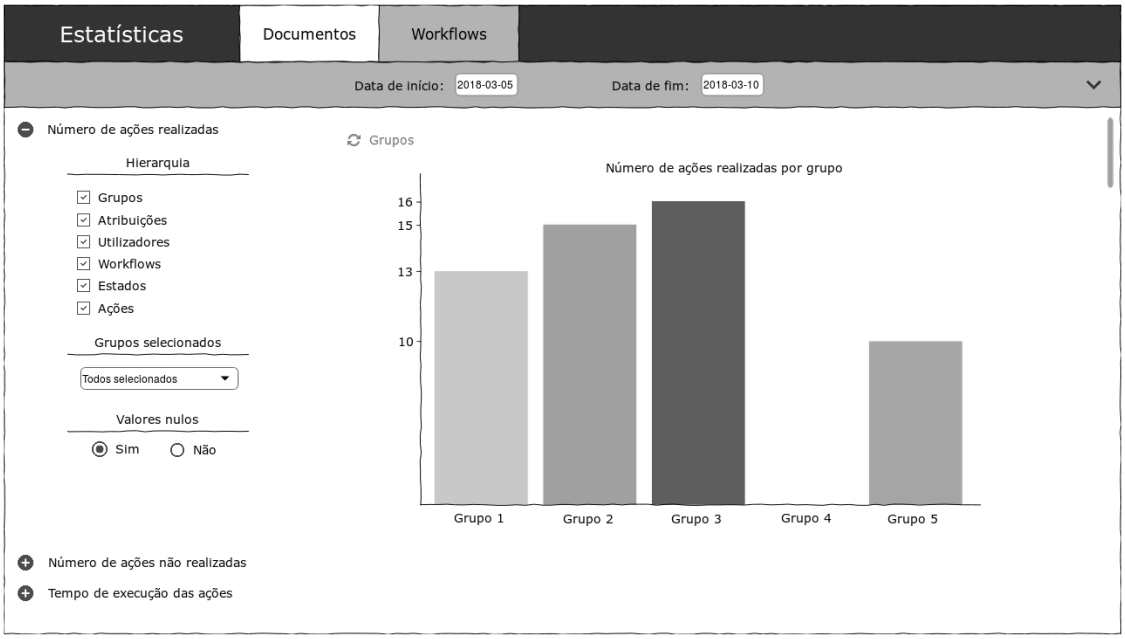


Figura 4.7: *Mockup* da interface de estatísticas relativa a *workflows*

Na Figura 4.7 esta apresentado um esboço do que será a interface de estatísticas. Como se pode observar, a seguir à barra de navegação surge uma área destinada aos filtros que serão aplicados aos diferentes gráficos. Inicialmente essa área contém os campos de data de início e de data de fim, que serão inicialmente definidos para restringir o intervalo temporal aos últimos trinta dias. Esta secção poderá ser expandida através do ícone situado no lado direito, permitindo configurar os restantes filtros que poderão existir. Os diferentes parâmetros estatísticos surgem em acordeão. Para cada parâmetro estatístico é apresentado o gráfico (lado direito), bem como uma secção de

configurações. Para cada gráfico, o utilizador poderá definir a hierarquia que pretende, escolher quais os elementos desse nível que quer visualizar no gráfico, e se deseja que os elementos com valores nulos surjam na mesma no gráfico. Estas configurações serão guardadas para cada nível de forma a que quando regresse a um nível anterior, as configurações se mantenham. Em cima de cada gráfico surge o caminho até ao nível atual, e caso não seja o nível com menor detalhe, é possível voltar a um nível anterior, *clicando* no nível que se pretende aceder.

4.4 Comparação entre as soluções propostas

Na Tabela 4.1 é realizada a comparação entre a solução anteriormente proposta, pela Ana Rita Mendes, e a solução apresentada nesta dissertação. Como já foi referido, as duas propostas assemelham-se em muitos aspetos, como por exemplo no conjunto de *feedbacks* ou na aquisição de pontos e recompensas. Contudo, a proposta anterior foca-se essencialmente num centro de ajuda. Esse centro de ajuda até pode vir a ser uma característica a ser incorporada no futuro mas, para uma primeira fase, pode ter um efeito não desejado. Isto porque os utilizadores irão ter tendência a tentar ajudar os outros, levados pela motivação extrínseca da possibilidade de conquistar alguma recompensa, retirando-os do seu fluxo normal de trabalho, reduzindo a sua produtividade.

Para além disso, a solução apresentada pela Ana Rita Mendes não faz referência a interfaces de configuração que permitem configurar níveis, emblemas e pontuações. Apesar de ser uma área destinada apenas a administradores, a interface de configuração tem um papel fulcral na solução apresentada, pois permite tornar o sistema o mais abrangente e configurável possível.

Por fim, a solução proposta nesta dissertação contempla ainda uma nova página de estatísticas. À primeira vista poderá não se relacionar muito com o sistema *ludificado* mas é de facto importante, num sistema de gestão documental e de processos, haver uma interface com dados estatísticos apelativa e de fácil perceção, de forma a avaliar o impacto das alterações na área de jogo.

Tabela 4.1: Comparação de características entre as soluções propostas

Característica	**Proposta anterior**	**Proposta atual**
Ajuda não personalizada	✓	✓
Centro de ajuda	✓	✗
Mantém fluxo normal de trabalho	✗	✓
Permite configurar sistema de jogo	✗	✓
Recompensas	✓	✓
Pontos	✓	✓
Níveis	✗	✓
Emblemas	✗	✓
Avatares e *nicknames*	✗	✓
Melhorias da página de estatísticas	✗	✓

4.5 Sumário

Na fase inicial deste capítulo foram descritos os vários passos do processo de design baseado no jogador. Foi seguida uma abordagem que permitisse projetar uma solução capaz de ser adotada por qualquer empresa. Isso implicou dificuldades na definição de alguns aspetos importantes ao longo dos diversos passos do processo, como os objetivos da empresa e os objetivos do jogador, que tiveram de ser generalizados. Os últimos passos envolveram decisões relativas à estrutura da solução *ludificada*, que permitiram definir qual o modelo de envolvimento do jogador e alguns dos elementos e características que iriam estar presentes no sistema.

De seguida efetuou-se uma análise à solução *ludificada* anteriormente proposta, permitindo identificar quais os aspetos importantes a serem incluídos na solução final. O conjunto de *feedbacks* é um desses casos, devido à sua importância face aos problemas identificados pelos utilizadores. O centro de ajuda e o painel de cooperação não foram considerados importantes para a solução final, mas alguns dos seus elementos foram tidos em consideração.

Foi também descrita, detalhadamente, a proposta de solução final, tendo por base todas as análises e estudos anteriormente realizados. A solução contempla um conjunto de *feedbacks*, uma área de jogo e uma página de estatísticas. A área de jogo foi projetada para aumentar o envolvimento do utilizador com o sistema e a sua produtividade, e vai de encontro aos objetivos gerais das empresas. Por outro lado, o conjunto de *feedbacks* e a página de estatísticas procuram minimizar os problemas identificados pelos utilizadores do *iPortalDoc*.

Por fim, foi realizada uma comparação entre a solução anteriormente apresentada para o *iPortalDoc* e a solução apresentada nesta dissertação.

Capítulo 5

Implementação e Resultados

Neste capítulo será apresentado todo o processo de implementação da solução proposta, bem como os respetivos resultados, nomeadamente as interfaces criadas e respetivas funcionalidades. Todas as implementações ocorreram numa máquina de testes da empresa, com o sistema operativo IPBRICK OS versão 6.2 baseado no Debian 7, e o *iPortalDoc* na versão 6.1.2. A base de dados usada pelo sistema foi a PostgreSQL na versão 9.1.13. Como será descrito, no desenvolvimento das várias interfaces foi utilizada a *framework Bootstrap* na versão 3.3.2.

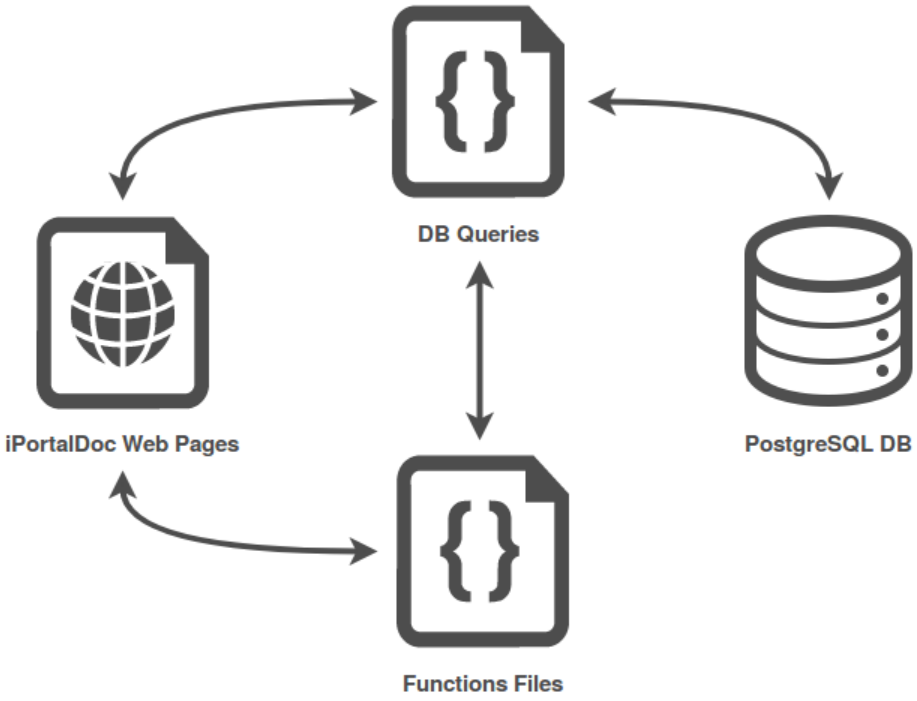


Figura 5.1: Estrutura de ficheiros do *iPortalDoc*

A estrutura de ficheiros que normalmente é utilizada no *iPortalDoc* está representada na Figura 5.1. As páginas web contemplam código em PHP, *JavaScript* e CSS. Os ficheiros, designados no

esquema por *DB Queries*, contêm as funções que executam as *queries* à base de dados. Por seu lado, os *Functions Files* constituem o conjunto de funções gerais, que podem ser utilizadas em diversos locais, e as funções que serão executadas através de pedidos AJAX. Estas funções poderão, ou não, executar outras funções da *DB Queries* de forma a obter informações da base de dados. Os *DB Queries* e *Functions Files* são ficheiros em PHP.

5.1 Conjunto de *feedbacks*

A primeira fase da implementação do conjunto de *feedbacks* passou por identificar as ferramentas que permitissem desenvolver o que foi especificado na proposta de solução, e que fossem compatíveis com o sistema. Pelo facto de o *iPortalDoc* usar o *Bootstrap* como *framework* de *front-end*, optou-se por utilizar o *plugin bootstrap-tour* em *JavaScript*. Isto porque é um *plugin* desenvolvido precisamente para o *Bootstrap*, tornando a sua inserção no sistema fácil e de rápida execução.

O *bootstrap-tour* consiste num conjunto de *popups*, que são apresentadas por uma ordem específica, e que se referem a uma determinada divisão da página HTML. Este *plugin* tem várias opções de configuração, sendo que nesta dissertação não foram muito exploradas. Estabelecer um *tour* é um processo simples que consiste em criar um objeto a partir da classe *Tour*, que contém várias propriedades, das quais se destacam (por ter sido necessário alterar os seus valores por defeito):

- **name (*string*)** — Esta propriedade define o nome do item de armazenamento onde o estado do *tour* é guardado. É possível iniciar vários *tours* na mesma página, com nomes diferentes. Por defeito vem com o valor "tour";
- **steps (*array*)** — Lista de objetos que representam os passos do *tour*;
- **storage (*object*)** — Sistema de armazenamento que se pretende usar. Podem ser os objetos *window.localStorage*, *window.sessionStorage* ou um objeto próprio. Se esta propriedade estiver a *false* o estado do *tour* não é guardado, e sempre que a página recarregar, o *tour* começa desde o início. Por defeito vem com o valor *window.localStorage*;
- **backdrop (*boolean*)** — Permite realçar o elemento associado ao passo, escurecendo os restantes. Por *default* vem a *false*.

Relativamente às propriedades dos objetos que representam os passos do *tour*, foram relevantes:

- **element (*string - seletor de jQuery*)** — Propriedade que indica qual o elemento HTML à qual a *popup* do passo faz referência;
- **placement (*string*)** — Permite definir o posicionamento da *popup*. As possibilidades são: "top", "bottom", "left", "right" (*default*) e "auto";

- **title (*string*)** — Título do *feedback*;
- **content (*string*)** — Descrição do passo a apresentar na *popup*;
- **animation (*boolean*)** — Propriedade que permite aplicar uma transição em CSS à *popup*. Por defeito está a "true".

Depois de estar definido o objeto, foi necessário inicializá-lo através do seu método *init()*. Quando se pretender iniciar o *tour* basta executar o método *start()* que, por exemplo, pode estar associado a um botão, ou ser executado sempre que é inicializada a página.

Em termos de implementação, após verificar o funcionamento do *plugin* no sistema, com a criação de um *tour*, o restante trabalho passa a ser a definição dos restantes, utilizando o mesmo processo de elaboração. Desta forma, o conjunto de *feedbacks* não foi algo muito explorado em termos de implementação, procurando apenas obter o *know-how* que permitisse, num futuro, criar os restantes *tours*. Isto porque, com o elevado número de interfaces que o *iPortalDoc* apresenta, teriam de ser criados muitos *tours*, de forma metódica, não sendo, talvez, o mais importante desta dissertação.

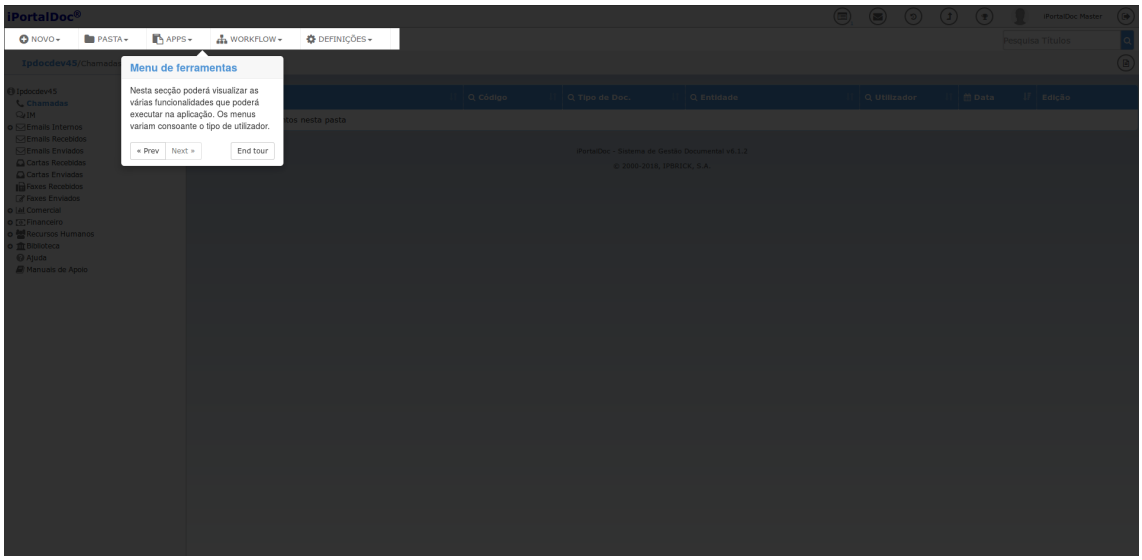


Figura 5.2: Exemplo de um passo de um *tour* implementado no *iPortalDoc*

Desta forma, foi apenas criado um *tour* que permite explicar aos novos utilizadores, as divisões da interface principal do *iPortalDoc* (Figura 2.13). Este *tour* consiste em quatro passos:

- **Hierarquia de documentos** — *"Nesta área é possível navegar ao longo da hierarquia de documentos."*;
- **Área de visualização de documentos** — *"Nesta área são apresentados os documentos que estão presentes numa dada secção, na hierarquia de documentos."*;

- **Pesquisa** — *"Área destinada à pesquisa. O utilizador pode fazer uma pesquisa rápida ou, carregar na lupa, para uma pesquisa mais detalhada."*;
- **Menu de ferramentas** — *"Nesta secção poderá visualizar as várias funcionalidades que poderá executar na aplicação. Os menus variam consoante o tipo de utilizador."* (na figura 5.2 é possível observar este passo na interface do *iPortalDoc*).

Aquando da implementação deste *tour* surgiram alguns problemas, sendo uns ultrapassados e outros não, por questões do próprio *plugin*. O primeiro consistia na sobreposição da barra superior do *iPortalDoc* sobre as *popups*, não ficando essa zona escurecida. Isto acontecia devido a uma classe *fixedTop*, que fixava essa divisão no topo da página. A solução para este problema passou por utilizar os métodos *onStart()* e *onEnd()* do *tour* criado, em que no início deste removia-se a classe e, quando o *tour* terminava, essa classe era novamente adicionada à divisão. O problema que não foi possível solucionar, deveu-se ao facto de a área de visualização de documentos ser uma *iframe*, não sendo possível criar um *tour* com passos que faziam referência a divisões de *iframes* diferentes. O *plugin* permite que possa existir passos do *tour* em diversas páginas, mas tem problemas relativamente a *iframes*, que ainda não foram ultrapassados. Por isso, a alternativa passa por referenciar a *popup* à divisão toda da *iframe* ou, caso isso não seja o mais indicado, criar um novo *tour* dentro da *iframe*, não havendo passos em diferentes *iframes*.

5.2 Área de jogo

Relativamente à área de jogo, dividiu-se a implementação em diversas fases, devido à elevada quantidade que trabalho que implicou o seu desenvolvimento.

- **1ª fase de implementação**

  Nesta primeira fase foi desenvolvida a interface da área de jogo (*mockup* na Figura 4.2) e a interface de configuração dos níveis destinada ao administrador (*mockup* na Figura 4.3). Os casos de uso implementamos nesta fase foram:

  - Alterar dados pessoais: avatar e *nickname*;
  - Consultar progressão;
  - Configurar níveis de jogo: alterar detalhes, definir pontuação a alcançar e alterar número de níveis.

  Na Figura 5.3 encontra-se o diagrama de classes para a primeira fase de implementação. Este diagrama é simples, e apresenta apenas as classes que foram necessárias para esta fase. Posteriormente foi alterado de forma a englobar todo o sistema projetado.

  - ***Game User:*** Esta classe representa o utilizador do *iPortalDoc* no contexto da área de jogo. Engloba o *nickname*, o nome do ficheiro do avatar e os pontos totais acumulados até ao momento. Em termos práticos existe uma tabela na base de dados com os



Figura 5.3: Diagrama de classes para a primeira fase de implementação da área de jogo

dados do utilizador, contudo, decidiu-se criar uma nova tabela, apenas com os parâmetros relativos à área de jogo. Desta forma, será necessário correr um *script* numa primeira instalação desta solução ludificada, de forma a que cada utilizador no sistema esteja também na tabela criada, com os parâmetros relativos a esta classe. Da mesma maneira, sempre que há um novo utilizador, este terá de ser adicionado a ambas as tabelas;

– ***Game Level:*** Classe que representa os vários níveis da área de jogo. Para além do nome do nível, contém também o número mínimo e máximo de pontos que definem o intervalo pontual para que o utilizador se enquadre naquele nível, e ainda um parâmetro que indica se esse nível está ativo ou não. Como é possível observar no diagrama, a *Game Level* apresenta uma relação de *um-para-vários* para com a classe *Game User*, ou seja, um nível poderá estar associado a vários utilizadores, mas um utilizador terá de ter, obrigatoriamente, um nível associado.

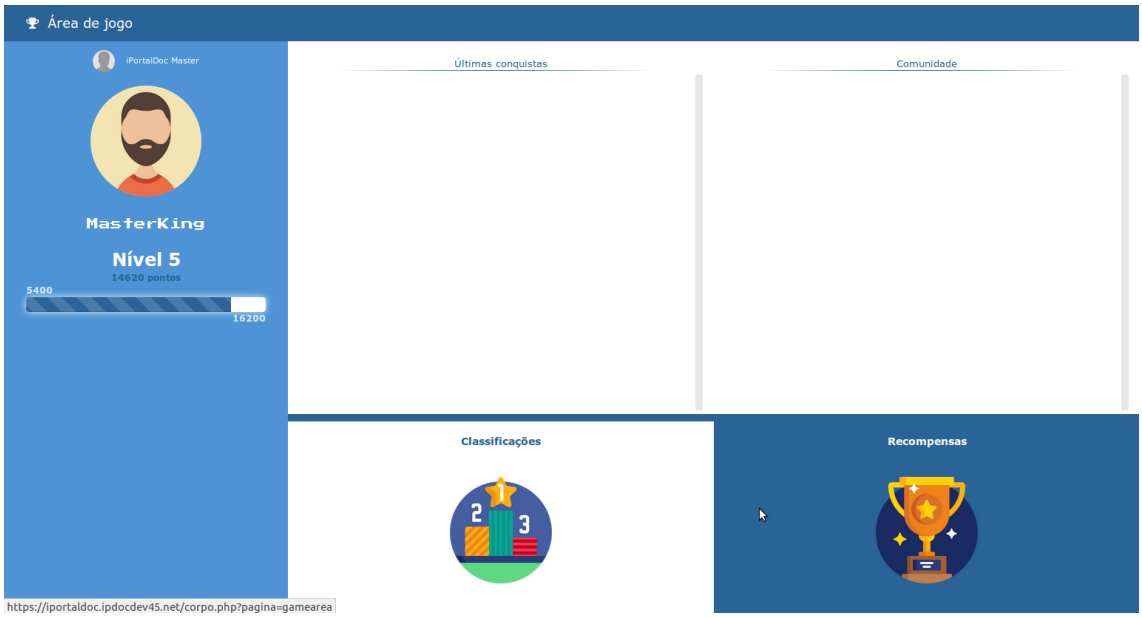


Figura 5.4: Interface da área de jogo após a primeira fase de implementação

Tendo este modelo implementado, foi possível iniciar o desenvolvimento da interface da área de jogo. A página foi desenvolvida com recurso à *framework Bootstrap*, com um design

elegante e moderno, mas que ao mesmo tempo se adequa ao restante sistema *iPortalDoc*. Na Figura 5.4 está representada essa interface da área de jogo, após a primeira fase de implementação. Como é possível constatar, nesta etapa o foco recaiu no *layout* da página, tendo sido necessária a utilização de alguns ícones, recorrendo, para isso, ao *Flat Icon*. Num futuro, estes ícones serão substituídos por outros desenvolvidos pela equipa de design da IPBRICK SA.

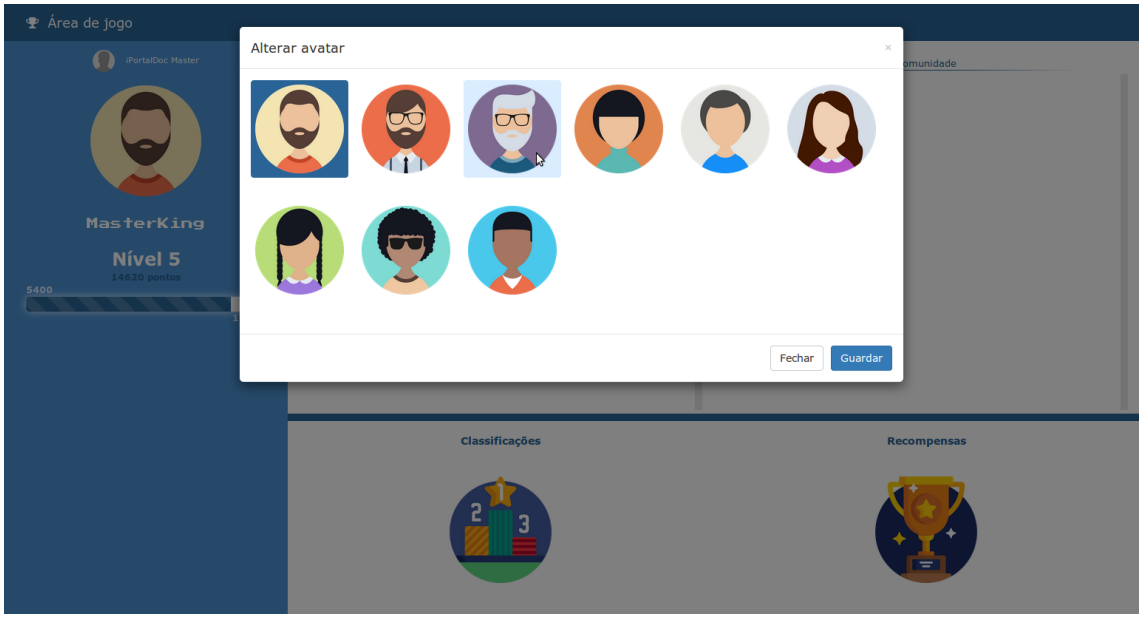


Figura 5.5: *Popup* da interface da área de jogo para alteração do avatar

A informação necessária da base de dados (dados do utilizador e do seu nível) é carregada ao iniciar a página. A funcionalidade de alterar o avatar e o *nickname* foi implementada com recurso à tecnologia AJAX, permitindo realizar pedidos ao servidor sem ter que atualizar a página. Desta forma, sempre que o utilizador *clicar* no avatar, surgirá uma *popup* (implementada com o *plugin Modal* do *Bootstrap*) que contém uma seleção predefinida de avatares, dos quais o utilizador poderá escolher um (Figura 5.5). Com o uso do *Image Picker* (um *plugin* de *jQuery*) foi possível tornar essa seleção de avatares mais agradável visualmente. Após selecionar o avatar, o utilizador poderá submeter essa alteração, sendo que, através do pedido AJAX, será efetuada a alteração na base de dados. Se o pedido for retornado com sucesso, a *popup* será fechada e será atualizado o avatar na interface (sem ser necessário um *reload* da página). O mesmo acontece ao *clicar* no *nickname*, sendo que nesta *popup* (Figura 5.6) surgirá um campo para o utilizador introduzir o nome que desejar, que será atualizado na base de dados com um procedimento semelhante ao do avatar.

A seguir ao avatar e ao *nickname* é apresentado o nome do nível, os pontos e a barra de progressão do utilizador (implementada com recurso a um dos componentes do *Bootstrap*).

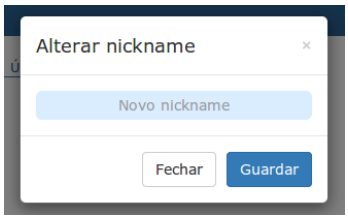


Figura 5.6: *Popup* da interface da área de jogo para alteração do *nickname*

A barra indica o estado no nível atual, em percentagem, permitindo saber se falta muito ou pouco para atingir o nível seguinte.

Relativamente à implementação da interface de configuração dos níveis da área de jogo, houve uma preocupação de manter o *layout* idêntico às restantes páginas de configuração. Contudo, foi necessária uma implementação de raiz, não havendo uma adaptação dessas páginas, pois não utilizavam a *framework* do *Bootstrap*. Na Figura 5.7 está apresentada a interface de configuração dos níveis que foi desenvolvida com base no *mockup* anteriormente apresentado (Figura 4.3). Como é possível constatar, houve uma pequena alteração nesta interface, passando a existir apenas uma coluna de pontos e não duas, com os limites dos níveis. Esta nova coluna da tabela representa o número de pontos necessários para atingir o nível em questão. Esta alterações não implicou alterações na base de dados pois é na mesma possível obter o limite mínimo e máximo dos níveis a partir desta nova coluna.

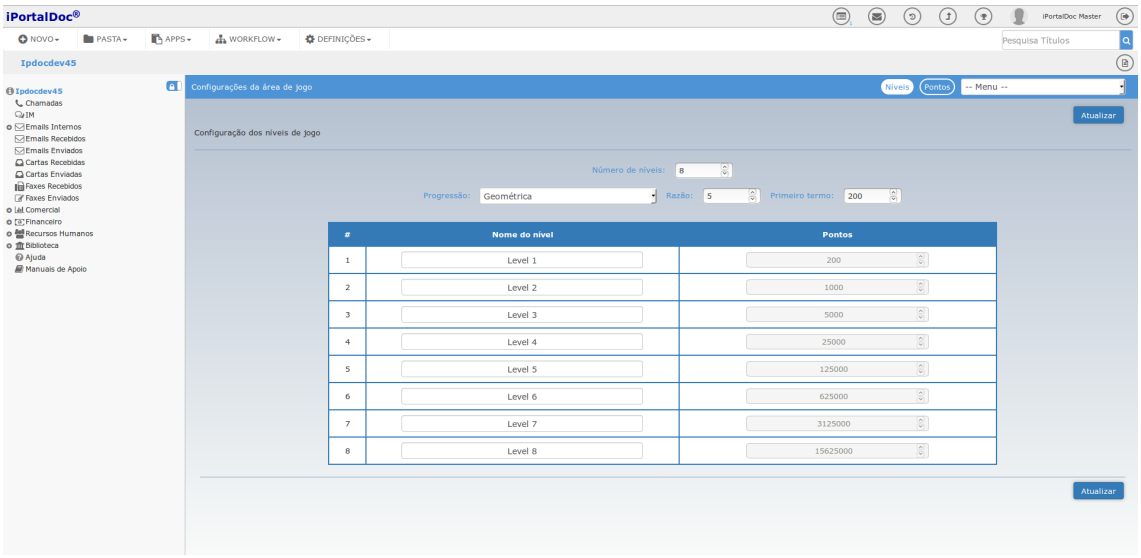


Figura 5.7: Interface de configuração dos níveis da área de jogo

Inicialmente, a tabela desta interface apresenta os níveis que estão ativos no sistema da área de jogo, com os respetivos campos preenchidos com os dados atuais da base de dados. Qualquer alteração nos campos de número de níveis, progressão, razão e primeiro termo, bem como na coluna de pontos, implicará uma alteração daquilo que é apresentado. Nesse

sentido, foram implementadas diversas funções em *JavaScript*/*jQuery* (que serão executadas em diversas situações):

- ***adjust_limits()*** — Função que define os parâmetros mínimo e máximo associados ao *input* da coluna dos pontos. Desta forma, o utilizador não poderá definir os pontos de um nível com um valor igual ou inferior ao do nível anterior, tendo o primeiro nível sempre um valor positivo;
- ***update_progression_limits()*** — Esta função atualiza a coluna de pontos com base na progressão atual e os respetivos parâmetros razão e primeiro termo. No caso da progressão manual, esta função não realiza qualquer alteração. Nos outros casos, aritmética e geométrica, o valor para cada nível é calculado a partir das expressões 4.1 e 4.2, respetivamente;
- ***check_progression()*** — Esta função bloqueia determinados campos com base na progressão escolhida. No caso da progressão ser manual, os campos razão e primeiro termo não fazem sentido, estando desta forma bloqueados. Sendo a progressão aritmética ou geométrica, todos os campos da coluna de pontos são bloqueados, pois são calculados automaticamente a partir dos campos razão e primeiro termo que, nestes casos, não estão bloqueados. Esta função executa, por fim, as duas funções apresentadas anteriormente;
- ***update_table_rows()*** — Função que adiciona ou remove linhas da tabela consoante o valor do número de níveis. No caso de se aumentar o número de níveis, é realizado um pedido AJAX de forma a obter os parâmetros gravados na base de dados, relativos aos novos níveis. Isto porque, apesar de não estarem ativos, estes níveis podem ter caraterísticas guardadas, que desta forma não serão perdidas. O pedido executado retorna o conjunto de linhas de serão adicionadas à tabela através do método *after()* da biblioteca de *JavaScript*, a *jQuery*. No caso de se pretender diminuir o número de níveis, é simplemente usado o método *remove()* nas linhas que estão a mais. Por fim, esta função executa ainda a função *check_progression()*.

Desta forma, sempre que existe uma alteração do número de níveis é executada a função *update_table_rows()*. Na situação em que a progressão é alterada, é chamada a função *check_progression()*. No caso de serem alterados os campos razão e primeiro termo, é executada a função *update_progression_limits()*. Por fim, qualquer alteração nos campos da coluna de pontos implica a execução da função *adjust_limits()*. Sempre que a página é iniciada, é executa também a função *check_progression()*, de forma a definir os limites dos diversos campos e para bloquear aqueles que não são necessários.

Tendo a configuração dos níveis de jogo realizada, o administrador poderá submeter essas alterações através de um dos botões de atualizar. Antes dessa submissão, os dados introduzidos passam ainda por um validação, de forma a detetar valores incorretos. Os níveis que

foram definidos, são atualizados na base de dados, com os novos parâmetros. Os restantes níveis passam a estar inativos.

De forma a testar as funcionalidades implementadas nesta fase, e não tendo ainda a página que permite definir qual a pontuação estipulada para um determinado *workflow* ou ação, decidiu-se que qualquer ação realizada contribuiria com um determinado valor de pontos para o utilizador que a realizou. Isto permitiu verificar o funcionamento do sistema de progressão e dos respetivos níveis.

- **2ª fase de implementação**

  Esta segunda fase de implementação englobou o desenvolvimento de uma nova interface de configuração, neste caso, relativa à configuração de pontos da área de jogo (*mockup* na Figura 4.4). Os casos de uso implementamos nesta fase foram:

  - Definir quais os *workflows* e ações a serem pontuados e configurar as respetivas pontuações;
  - Atribuir pontuação (que engloba verificar o estado do *workflow*);
  - Atualizar conquistas;

Tendo em vista o cenário de emblemas e conquistas, foi necessária uma reformulação do diagrama de classes referente à primeira fase de implementação (Figura 5.3). Numa primeira abordagem, havia uma separação do conceito de níveis e de emblemas. Contudo, depois de uma melhor análise, concluiu-se que os níveis podiam ser considerados como emblemas, sendo estes relativos a uma determinada categoria. O novo diagrama de classes está representado na Figura 5.8.

- ***Game User:*** Esta classe já estava presente no diagrama apresentado na primeira fase de implementação e representa o utilizador do *iPortalDoc*. Contudo, com a definição de emblemas e categoria de emblemas, deixa de ser necessária a existência do parâmetro pontos nesta classe. A justificação para este facto será explicada posteriormente;
- ***Badge:*** Classe relativa aos emblemas de jogo. Podem ser emblemas de níveis atingidos, de ações realizadas, de *workflow* concluídos, entre outros. Apresenta os parâmetros nome, descrição, ordem (posição na sua categoria), nome do ficheiro com a imagem do emblema, meta de pontos para garantir o emblema e ainda uma *flag* para indicar se o emblema está ativo ou não;
- ***User Achievement:*** Classe que representa a relação entre as classes *Game User* e *Badge*. Cada utilizador poderá conquistar diversos emblemas, e um mesmo emblema pode ser conquistado por vários utilizadores. Esta classe tem um papel importante neste sistema da área de jogo, registando todas as conquistas dos utilizadores, bem como o estado atual de uma categoria de emblemas. Isto é, para todas as categorias de emblemas, um utilizador terá pelo menos um emblema dessa categoria, mesmo

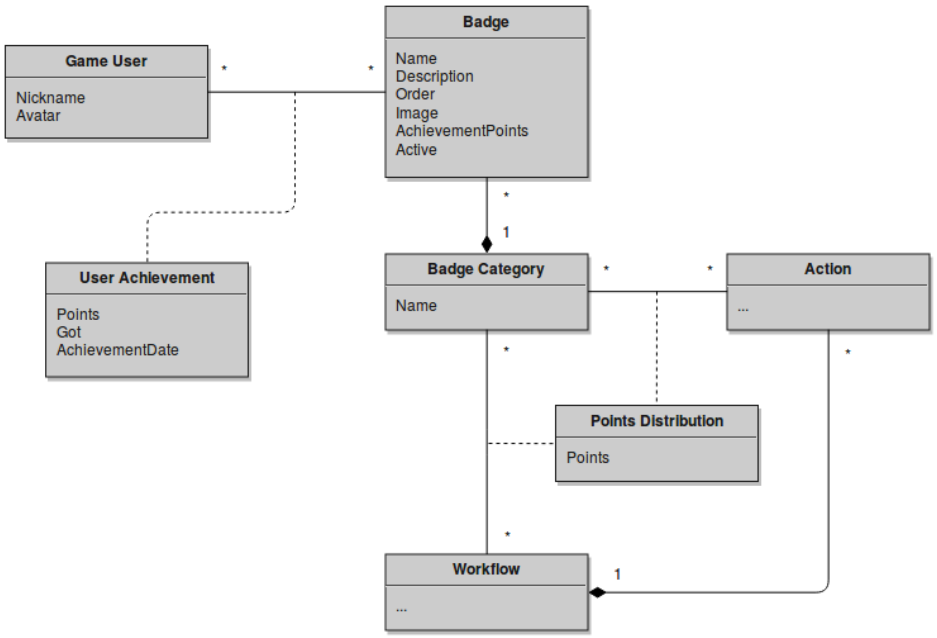


Figura 5.8: Diagrama de classes para a segunda fase de implementação da área de jogo

que não o tenha alcançado, de forma a puder registar os pontos conquistados até ao momento, relativos a essa categoria. Desta forma, esta classe apresenta os seguintes parâmetros: pontos; uma *flag* para verificar se esse emblema já foi conquistado; e ainda a data de aquisição, caso já tenha sido conquistado;

- ***Badge Category:*** Esta classe tem apenas como parâmetro um nome e representa o conjunto de emblemas de um determinado tipo. Para ilustrar melhor a relação entre o *Badge* e o *Badge Category* consideremos o seguinte exemplo. Imagine-se que é definida uma categoria de ações realizadas. A essa categoria poderão estar relacionados vários emblemas consoante o número de ações realizadas. Por exemplo, o primeiro emblema (a posição do emblema dentro da sua categoria está registada no parâmetro *order*) poderá ser alcançado com cinquenta ações realizadas, o segundo com duzentas, e assim sucessivamente. Os emblemas estão organizados por um ordem sequencial, dentro de uma categoria;

- ***Workflow:*** Esta classe representa o conjunto de ações, associadas a um documento, que um determinado grupo de pessoas terá de realizar. A conclusão de um *workflow* poderá implicar a aquisição de pontos para uma, ou mais categorias de emblemas;

- ***Action:*** Esta classe representa a tarefa de um determinado *workflow* que foi atribuída a uma ou mais pessoas. Tal como no caso do *workflow*, a realização de uma ação poderá implicar a angariação de pontos por parte da pessoa que a realizou, para um conjunto de categorias de emblemas;

- ***Points Distribution:*** Esta classe permite definir qual a pontuação a atribuir a uma determinada categoria após a conclusão de um *workflow* ou a realização de uma ação. Um *workflow* ou ação poderá contribuir para uma ou mais categorias de emblemas.

Este diagrama de classes permite então que sejam definidos vários emblemas. Contudo, esta dissertação apenas se focou nos emblemas relacionados com os níveis. Isto porque, para a criação e configuração de emblemas seria necessária a adaptação da interface de configuração, para que se pudesse alterar não só os níveis, como também, os restantes emblemas. Como inicialmente havia a distinção entre níveis e emblemas, a interface de configuração dos níveis (Figura 5.7) não foi projetada para ser possível editar a descrição, bem como o emblema de um determinado nível. O mesmo acontece com a interface de configuração de pontos da área de jogo (*mockup* na Figura 4.4). Esta foi na mesma implementada da forma que foi projetada, contudo, num futuro, terá de ser ligeiramente alterada para ser possível configurar os pontos atribuídos às restantes categorias de emblemas.

Antes de ser descrita a implementação da interface de configuração de pontos da área de jogo, será apresentado o conceito de níveis com base neste novo diagrama de classes. Existe, desta forma, uma categoria de emblemas designada por Níveis. Os níveis, que são definidos na respetiva interface de configuração (Figura 5.7), são de facto emblemas dessa categoria. Os parâmetros *Name* e *AchievementPoints* podem ser definidos nessa interface, não sendo possível, para já, alterar o parâmetro de descrição e imagem do emblema. Tal como acontecia na primeira fase de implementação, os emblemas de níveis que não foram definidos passam a estar como inativos. Neste novo diagrama, os pontos conquistados não são associados à classe *Game User* mas sim ao *User Achievement*.

Imagine-se a situação em que o utilizador não tem qualquer ponto acumulado. Desta forma, ainda não conquistou nenhum emblema, mas está associado ao primeiro emblema da categoria de níveis, com zero pontos. Quando realiza uma ação com uma determinada pontuação, esse valor é acumulado aos pontos desse *User Achievement*. De seguida é verificado se os pontos já atingiram a pontuação definida para esse nível, e se for o caso, esse *User Achievement* é definido como conquistado, sendo também definida a data de conquista. Sempre que é conquistado um emblema, é adicionado um novo *User Achievement* com o emblema seguinte dessa categoria. Os pontos desse *User Achievement* são iniciados com os pontos do último emblema conquistado, e é novamente verificado se esse emblema de nível já foi alcançado. Este processo repete-se sempre que se conquista pontos.

A interface de configuração de pontos da área de jogo, implementada a partir do *mockup* da Figura 4.4, encontra-se representada na Figura 5.9. Mais uma vez, esta página foi desenvolvida com um *layout* de acordo com as restantes interfaces de configuração. É apresentado, em forma de tabela, todos os *workflows* instanciados e que estão ativos no sistema. O *iPortalDoc* permite conceber *workflows*, com estados, ações e funções de transição. Esse *workflow* concebido é como se fosse um *template* que poderá ser instanciado, com várias configurações distintas. Essas instâncias são, de facto, os *workflows* que estão disponíveis

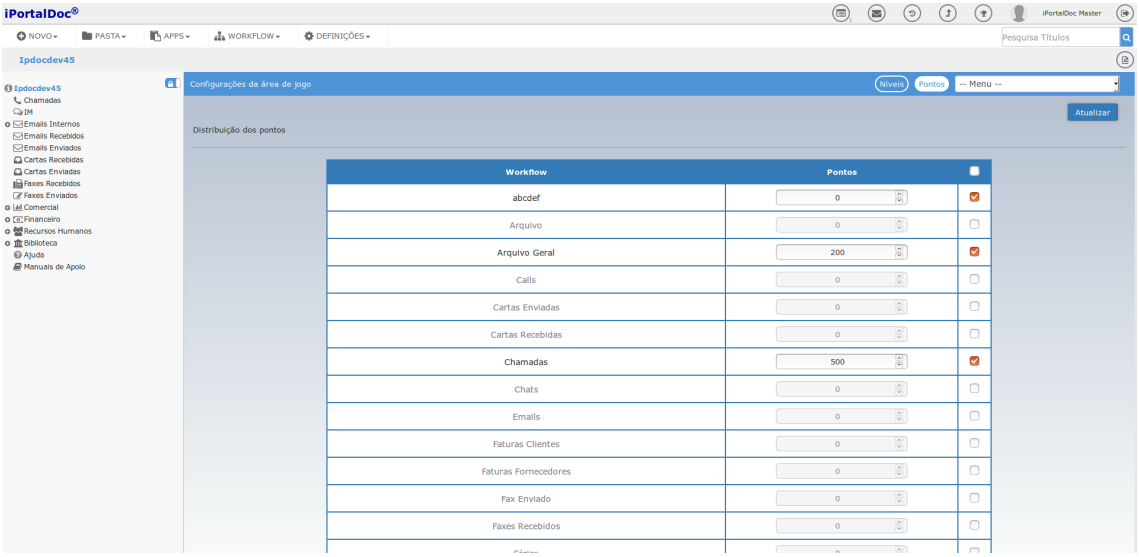


Figura 5.9: Interface de configuração de pontos da área de jogo

no sistema e que podem ser associados aos documentos introduzidos. Essas associações traduzem-se em revisões desses *workflows*.

No início da página é obtida, da base de dados, toda a informação relativa à distribuição de pontos, a partir da tabela referente à classe *Points Distribution*. Caso o *workflow* esteja nessa tabela, este aparece selecionado e com a pontuação que lhe foi atribuída. O administrador poderá então selecionar novos workflows, definindo a sua pontuação, ou remover algum que já esteja selecionado. Foram implementadas várias funções em *JavaScript/jQuery* para permitir a coerência da página, isto é, se *clicar* na *checkbox* de selecionar todos, todos os *workflows* são selecionados, caso desselecione novamente um *workflow*, a *checkbox* de selecionar todos, volta a não estar selecionada. Estas funções também permitem desabilitar os campos dos *workflows* que não estão selecionados. Mais uma vez, é importante referir que a pontuação relativa ao *workflow* é atribuída, na conclusão deste, a todos os que contribuíram para a sua realização.

A partir desta interface, é também possível selecionar as ações que irão contribuir para a aquisição de pontos. Caso um *workflow* tenha ações, ao *clicar* no respetivo nome, surgirá uma *popup* (implementada com o *plugin Modal* do *Bootstrap*) que contém uma tabela com todas as ações desse *workflow* (Figura 5.10). O funcionamento desta tabela é exatamente igual ao da tabela relativa aos *workflows*. Quando o administrador terminar a configuração da distribuição dos pontos, este poderá submeter as alterações através dos botões alterar. Os valores submetidos passam ainda por uma validação de forma a verificar que não são submetidos valores negativos. De seguida a base de dados é atualizada com a nova distribuição pontual.

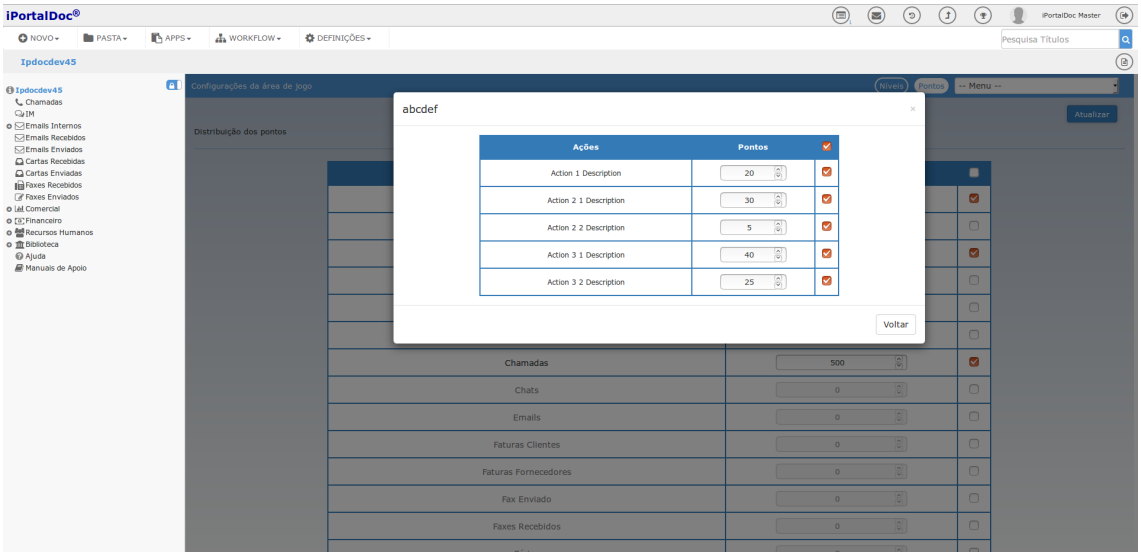


Figura 5.10: Interface de configuração de pontos da área de jogo relativos às ações

De forma a atribuir a pontuação consoante a distribuição de pontos, sempre que é realizada uma ação ou concluído um *workflow*, é necessário verificar se essa *ação* ou *workflow* têm uma pontuação associada. Nesta fase de implementação, essa verificação acontece ao nível de PHP. Caso esteja associada uma pontuação, esta é adicionada ao *User Achievement* da respetiva categoria de emblemas. O processo que se segue, de atualizar as conquistas, já foi exemplificado anteriormente.

A atualização de conquistas não acontece só quando são atribuídos pontos. Acontece também sempre que a configuração dos níveis é alterada, sendo dessa forma executada uma função que percorre todas das linhas da tabela referente à classe *User Achievement* e que atualiza os níveis conquistados com base nos novos parâmetros.

- **3ª fase de implementação**

  Esta última fase de implementação contemplou, maioritariamente, alterações ao nível de *back-end*, tendo sido apenas implementado o caso de uso de consulta de conquistas por parte do próprio utilizador. Até esta fase, todas as alterações que eram necessárias realizar na base de dados eram efetuadas em PHP:

  - Verificar se o utilizador era novo no sistema e, se fosse o caso, proceder à sua inserção na tabela de *Game User*;
  - Atualizar dados do *Game User*;
  - Criar ou alterar níveis de jogo por parte do administrador;
  - Atualizar conquistas, devido a alterações de níveis ou a aquisição de pontos;
  - Definir pontuação para *workflows* ou *ações*.

Contudo, esta abordagem obrigava a identificar no código do sistema *iPortalDoc*, o local onde, por exemplo, as ações eram definidas como concluídas. Por vezes, essas alterações ocorriam em diferentes situações, o que obrigava a introdução das funções PHP, que efetuavam as alterações na base de dados, em diversos locais, correndo também o risco de não englobar todas os casos. Desta forma, optou-se por realizar, tudo o que fosse possível, ao nível da base de dados. Isso foi implementado recorrendo à criação de *triggers* associados a tabelas, que executam determinadas funções desenvolvidas em linguagem PL/pgSQL. Os *triggers* implementados foram:

- ***insert_gameusers_trigger*:** Este *trigger* é desencadeado sempre que existe a introdução de uma linha na tabela *mailutilizador* (que contém todos os dados dos utilizadores do *iPortalDoc*). Executa a função *insert_gameusers_function()* que irá proceder à inserção na tabela *gameuser* dos dados relativos ao novo utilizador, com parâmetros *default* (por exemplo, para um novo utilizador o *nickname* será igual ao seu *login*). A função também irá também inicializar as conquistas associadas a esse novo utilizador, introduzindo nos *User Achievements* os emblemas de primeira ordem, de todo o conjunto de categorias de emblemas. Essas conquistas não foram ainda alcançadas, tendo sido inicializadas com a pontuação zero;
- ***update_levels_achievements_trigger*:** É responsável por atualizar os níveis de todos os utilizador de acordo com a configuração submetida. Este *trigger* é desencadeado sempre que exista uma inserção ou uma atualização de uma linha na tabela de emblemas, contudo, a função que é executada (*update_levels_achievements_function()*), termina logo no seu início, caso o emblema que foi introduzido ou alterado, não seja da categoria de níveis, ou se, mesmo com o *update*, a linha não tenha sofrido alterações. O procedimento consiste então em percorrer todos os utilizadores e, para cada um deles, existe dois casos possíveis, consoante o estado do nível introduzido/alterado. Se o nível em questão não estiver ativo, é preciso removê-lo das conquistas do utilizador. Caso esse *User Achievement* contenha a pontuação mais alta, é necessário guardar essa pontuação no nível ativo de ordem inferior, de forma a que a pontuação atual do utilizador não seja perdida. Se o nível introduzido/alterado estiver ativo, é então necessário verificar se o *User Achievement* do utilizador a ele associado, está conquistado ou não, com base nas novas configurações. Este procedimento poderá implicar novas conquistas de níveis por parte do utilizador. É necessário referir que esta função foi desenvolvida desta forma pelo facto de se saber que a inserção ou alteração de emblemas ocorre seguindo a respetiva ordem dentro da sua categoria.
- ***update_achievements _after_action_trigger*:** Este *trigger* ocorre sempre que existe um *update* de uma linha da tabela *accaorev*. Uma *accaorev* é uma ação de uma ocorrência de um determinado tipo de *workflow*. A função que é executada quando este *trigger* é desencadeado (*update_achievements_after_action_function()*), é responsável por atribuir eventuais pontuações ao utilizador que realiza a ação, e proceder à

atualização das suas conquistas. Esta função só continua o seu procedimento caso a ação alterada esteja agora finalizada e com uma data de realização válida. Serão então percorridas todas as distribuições pontuais que estejam definidas na tabela *pointsdistribution* para essa ação. Para cada uma dessas distribuições, os pontos serão então acumulados ao *User Achievement* do emblema com maior ordem, relativo à categoria identificada nessa distribuição pontual. Mais uma vez, caso a meta para esse emblema seja atingida, é criado um novo *User Achievement* para o emblema seguinte dessa categoria (se este existir e estiver ativo).

– ***update_achievements _after_workflow_trigger*:** Este *trigger* tem um procedimento semelhante ao anterior, mas neste caso é relativo à pontuação dos *workflows*. É desencadeado sempre que haja uma alteração da tabela *workflowrev* ou uma inserção (isto porque, em alguns casos, os *workflows* são diretamente inseridos como realizados). No início da execução da função correspondente a este *trigger*, é verificado se o *workflow* está terminado. Se este não estiver, ou se estiver, sendo que já estava finalizado aquando do *update*, a função termina. De seguida, para cada distribuição de pontos que esteja associada a este *workflow*, será atribuída a respetiva pontuação aos diversos utilizadores que nele participaram. Por cada um desses utilizadores, a atribuição de pontos e verificação de conquistas acontece da mesma maneira que foi apresentada para o caso das ações.

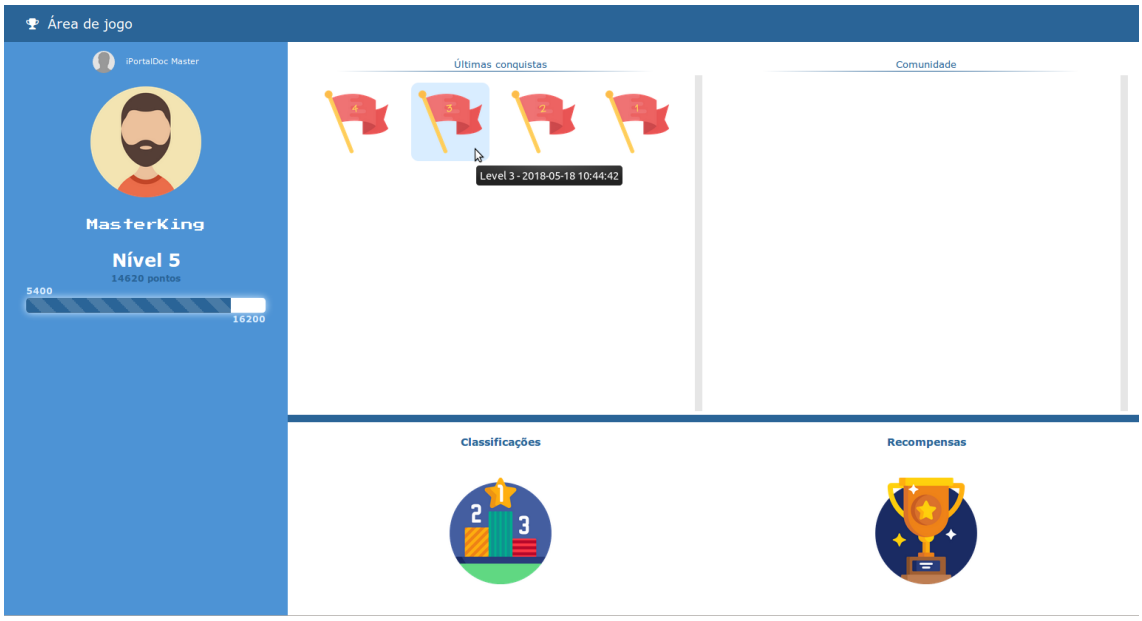


Figura 5.11: Interface da área de jogo com conquistas visíveis

Com a base de dados implementada a partir do diagrama de classes da segunda fase de implementação, e estando os *triggers* também em funcionamento, foi então possível apresentar as diversas conquistas do utilizador na sua área de jogo (Figura 5.11). A função que realiza

a *query* à base de dados é executada no carregamento da página. Como no sistema ainda só existem emblemas de níveis, só estes é que podem ser conquistados e apresentados na área de jogo. Contudo, caso o utilizador tivesse conquistado outro tipo de emblemas, estes seriam apresentados da mesma forma, com a respetiva imagem, nome, descrição e data de conquista. O nome e data de conquista pode ser visualizado ao passar o cursor por cima do emblema.

5.3 Página de estatísticas

Antes de se inicializar o desenvolvimento da página de estatísticas, foi necessário estudar qual a melhor solução para a implementação dos gráficos, de forma a satisfazer as especificações pedidas. Desta forma, a opção tomada recaiu na biblioteca de *JavaScript*, a *chart.js*. Isto porque é uma biblioteca *open source*, leve e que permite apresentar dados em oito tipos de gráficos, podendo estes ser configurados de várias formas. Os gráficos gerados por esta biblioteca são renderizados no elemento de HTML *Canvas*, sendo estes responsivos, quer no seu redimensionamento em função do tamanho da página, quer em interações que possam ocorrer com o próprio gráfico.

A interface de estatísticas foi mais uma vez desenvolvida utilizando a *framework Bootstrap*. Foi inicialmente estruturada toda a página, com as devidas secções e divisões, deixando o conteúdo de cada parâmetro estatístico para uma fase posterior. Na Figura 5.12 esta representada a interface da página de estatísticas apenas com a apresentação dos vários parâmetros que fazem parte do acordeão, mas sem que estas divisões contenham qualquer desenvolvimento. Na secção dos filtros é possível observar os campos destinados à data de início e data de fim. Ao *clicar* nesses campos, surge uma *popup* que permite a escolha da respetiva data. Isto é algo que já é usado no sistema *iPortalDoc*, e que está implementado com recurso ao *plugin Bootstrap DateTimePicker*. Desta forma, foi apenas necessário reutilizar a função que gera esse calendário, definindo, entre outros, o formato em que a data é apresentada, o valor inicial e o respetivo *range*. Tal como já foi referido, os valores iniciais para ambos os campos são definidos de forma a que a janela temporal seja relativa aos últimos trinta dias. Através de funções em *JavaScript/jQuery*, sempre que o valor de um dos campos é alterado, os limites de ambos os calendários são atualizados, não permitindo escolher uma data inicial mais recente que a data de fim, ou uma data final mais antiga que a data inicial.

O passo seguinte consistiu em implementar o "interior" de cada elemento do acordeão, ou seja, o gráfico e as respetivas configurações para cada parâmetro estatístico. Nesta dissertação foram apenas concluídas as estatísticas relativas ao número de ações realizadas e não realizadas, sendo que para os restantes parâmetros, o procedimento será praticamente o mesmo. O objetivo inicial deste passo foi criar um gráfico de barras que apresentasse um conjunto aleatório de dados. Sempre que se pretende gerar um gráfico, é necessário criar uma nova instância da classe *Chart*, cujo construtor recebe como parâmetros o elemento *Canvas* onde o gráfico irá ser renderizado, e um objeto que permite aplicar todo o tipo de configurações, que também inclui os dados que irão ser apresentados. Nesse objeto são normalmente incluídas as seguintes propriedades:

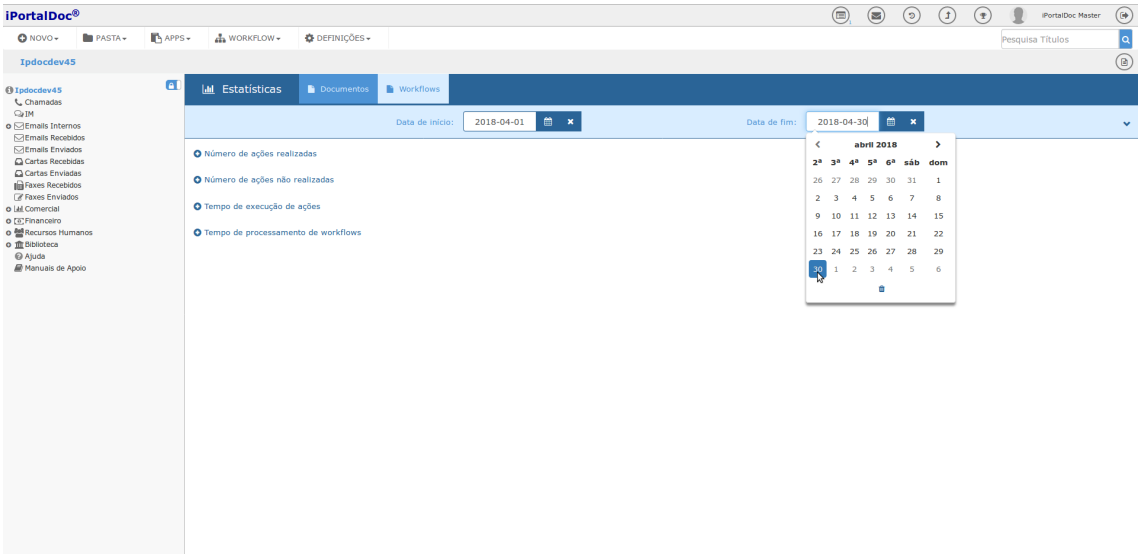


Figura 5.12: Fase inicial da interface da nova página de estatísticas do *iPortalDoc*

- ***type*** — Esta propriedade consiste numa *string* que indica o tipo de gráfico. A *string* poderá ser uma das seguintes: *"line"*, *"bar"*, *"radar"*, *"pie"*, *"polarArea"*, *"bubble"* ou *"scatter"*. Os gráficos de área não necessitam de indicar o tipo nesta propriedade;

- ***data*** — É um objeto que normalmente contém dois *arrays*: *labels* e *datasets*. Se tivermos em consideração um gráfico de barras, o *array labels* contém as descrições que serão apresentadas no eixo do x (por exemplo, Janeiro, Fevereiro, Março, etc.). O *array datasets* poderá conter vários conjuntos de dados. O número de conjunto de dados é equivalente ao número de barras que surgirão por cada mês, no exemplo que foi apresentado. Cada um desses conjunto de dados é um objeto que poderá conter um *array*, com os valores por cada *label* e ainda configurações como cor de fundo, cor da borda, espessura da borda, entre outros. No exemplo apresentado, essas configurações são relativas a cada uma das barras. Um conjunto de dados poderá ser apresentado com um tipo de gráfico diferente, permitindo desta forma obter gráficos mistos. Se voltarmos ao exemplo do gráfico de barras, se este envolver dois conjuntos de dados, um sem definir o tipo (assume que é de barras) e outro que é definido como *"line"*, o resultado será um gráfico de barras juntamente com um gráfico de linhas;

- ***options*** — Esta propriedade consiste num objeto que permite configurar as mais diversas opções. É possível apresentar a legenda do conjunto de dados, definir um título para o gráfico, alterar tipo de letra, definir cores e tamanhos dos eixos, etc. Basicamente, esta propriedade permite a personalização do gráfico que é apresentado.

Este objeto permite então configurar um gráfico em particular. Contudo, é também possível alterar as configurações gerais que são aplicadas a todos os tipos de gráficos, e que estão definidas

em *Chart.defaults.global*. Posto isto, e estando instanciado um objeto da classe *Chart*, foi possível observar o gráfico no respetivo *Canvas*.

Esta abordagem inicial apenas permitiu obter algum conhecimento relativamente ao funcionamento da biblioteca e de como eram gerados os gráficos. Mas, com as especificações que foram indicadas na proposta de solução, foi necessário uma maior complexidade em termos de procedimento. Pelo facto de a nova interface de estatísticas ter um comportamento de *single-page application*, todo o procedimento foi implementado em *JavaScript*. Uma *single-page application* consiste numa aplicação que é acedida através de um *browser* e que não necessita de *reload* durante a sua utilização. Todos os recursos (HTML, CSS e *scripts*) são carregados no início da aplicação. Através de AJAX é possível a troca de dados entre a aplicação e o servidor, e eventuais alterações à interface podem ocorrer sem a necessidade de um carregamento da página. A interface da área de jogo desenvolvida apresenta o mesmo comportamento.

O gráfico de cada parâmetro estatístico terá a si associado um *array* de objetos, em que cada objeto é relativo a um nível da hierarquia. Esses objetos foram criados para armazenar, de uma forma mais compacta, os dados e configurações que serão usadas em cada um dos níveis. Contemplam as seguintes propriedades:

- ***initialData*** — *Array* de objetos com todos os dados retornados da base de dados de acordo com os filtros aplicados, com o dado estatístico em questão e com o nível da hierarquia. Por exemplo, se o gráfico for relativo ao número de ações realizadas em função dos grupos, cada objeto do *array* irá conter o identificador, o nome e número de ações realizadas desse grupo;

- ***idDataSelected*** — Nas opções do gráfico é apresentada uma seleção com todos os elementos que estão presentes em *initialData*, em que o utilizador tem a possibilidade de escolher os que pretende visualizar. Esta propriedade consiste num objeto com todos os identificadores dos elementos, que têm uma *flag* associada, permitindo distinguir os que estão selecionados ou não;

- ***zeroValues*** — Booleano que indica se o gráfico irá apresentar elementos com valores nulos;

- ***dataForChart*** — *Array* semelhante ao *initialData*, mas apenas com os elementos que serão apresentados no gráfico. Existe uma função (que será descrita mais à frente) que, com base no objeto *idDataSelected* e na variável *zeroValues*, define os elementos deste *array*;

- ***colors*** — Objeto que contém o código RGB da cor de cada elemento do *initalData*. As cores são geradas aleatoriamente e são usadas para distinguir as diversas barras;

- ***type*** — Indica o tipo de gráfico para o nível em questão;

- ***dataId*** — O *array* de dados que é passado ao construtor da classe *Chart* apenas contém os valores de cada elemento. Como esses elementos poderão não estar na mesma posição que estão no *initialData*, este *array* permite guardar a identificação desses elementos na mesma posição;

- ***labels*** — *Array* com as *labels* que serão apresentadas no gráfico. Estão de acordo com os elementos da *dataForChart*;

- ***datasets*** — *Array* de objetos, em que cada objeto é um conjunto de dados. Os gráficos implementados na página de estatísticas apenas apresentam um conjunto de dados, que contêm quatro *arrays*: *data* — com os valores para cada barra; *backgroundColor* — com a cor de fundo de cada barra; *borderColor* — com a cor da borda de cada barra; *borderWidth* — com a espessura da borda de cada barra. Estes elementos são os que estão incluídos em *dataForChart*;

- ***title*** — *String* com o título que irá ser apresentado no gráfico.

Para além deste *array*, foi ainda necessário definir mais algumas variáveis de forma a guardar dados como: elemento HTML *Canvas*; data de início; data de fim, nível atual; níveis da hierarquia selecionados; sequência de elementos pela hierarquia (para saber que elementos foram selecionados pelo caminho até ao nível atual); e configurações para cada elemento de um determinado nível (ou seja, permite guardar as configurações por elemento, e não apenas por nível). Foram também desenvolvidas várias funções(estas funções são relativas ao gráfico do número de ações realizadas, mas que podem ser adaptadas para os restantes):

- ***initNumberActionsPerformedChart()*** — Função que permite criar o objeto a partir da classe *Chart*. Desta forma, inicia-se o gráfico do número de ações realizadas com base no nível atual e nas respetivas propriedades (nomeadamente: *type*, *labels*, *datasets* e *title*) definidas no *array* de objetos anteriormente apresentado. Na biblioteca *chart.js* existe um método que permite atualizar o gráfico que é apresentado. Contudo, através desse método, nem todas as opções eram atualizadas, como, por exemplo, o título do gráfico e as próprias *labels*. Por isso, sempre que se pretende atualizar o gráfico, este é eliminado através do método *destroy()* e depois é novamente executada esta função;

- ***randomRGB()*** — Função que retorna uma *string* com um código RGB aleatório;

- ***updateColorsNumberActionsPerformed()*** — Esta função verifica se, para cada elemento do *initialData* de um determinado nível, existe um código RGB correspondente em *colors*. caso não exista, é executada a função *randomRGB()* de forma a definir uma cor para esse elemento;

- ***updateAllIdDataSelectedNumberActionsPerformed()*** — Com base no *initialData*, esta função atualiza o objeto *idDataSelected*, associado a um determinado nível, sendo que, os elementos que já não fazem parte do *initialData* são removidos, e são adicionados os novos elementos com a *flag* ativa. Os restantes permanecem com as mesmas *flags*;

- ***updateSelectNumberActionsPerformed()*** — Esta função atualiza o cabeçalho das opções relativas à seleção de elementos, com base no nível da hierarquia atual (por exemplo, se o

nível passou de *Grupos* para *Atribuições*, o cabeçalho que será apresentado passa de *"Grupos selecionados"* para *"Atribuições selecionadas"*). Para além disso, atualiza os próprios elementos que são apresentados na respetiva lista com base nos elementos presentes em *idDataSelected*. Esta área de seleção foi implementada com recurso a um *plugin* baseado em *jQuery*, o *Bootstrap Multiselect*;

- ***updateZeroValuesNumberActionsPerformed()*** — Esta função apenas atualiza os *inputs* do tipo *radio*, associados aos valores nulos, consoante o estado do booleano *zeroValues* de um respetivo nível;

- ***updateDataForChartNumberActionsPerformed()*** — Esta é a função que atualiza os elementos do *array dataForChart* (que serão apresentados no gráfico), com base no *idDataSelected* e no *zeroValues*;

- ***updateDataChartNumberActionsPerformed()*** — Atualiza as propriedades *dataId*, *labels* e *datasets* a partir dos elementos presentes em *dataForChart*. Nos *datasets*, para além dos dados, são também definidos os *arrays* com as cores de fundo, cores das bordas e respetiva espessura;

- ***updateActionsNumberPerformedRequest()*** — Função que executa o pedido AJAX de forma a obter os dados estatísticos da base de dados. São passados parâmetros como localização e nome da função, que irá ser executada no lado do servidor, bem como as datas de início e fim, o nível atual e o *array elementIdSequence* com os elementos da sequência. A função no lado do servidor executa a *query* a base de dados tendo em conta estes parâmetros e retorna o resultado em JSON.

  Caso o retorno seja realizado com sucesso, esses dados são de seguida armazenados no *initialData*. Depois são executadas várias funções (cujo funcionamento foi descrito anteriormente) pela seguinte ordem:

  1. *updateColorsNumberActionsPerformed()*;
  2. *updateAllIdDataSelectedNumberActionsPerformed()*;
  3. *updateSelectNumberActionsPerformed()*;
  4. *updateZeroValuesNumberActionsPerformed()*;
  5. *updateDataForChartNumberActionsPerformed()*;
  6. *updateDataChartNumberActionsPerformed()*;
  7. *initNumberActionsPerformedChart()*.

  Durante a execução desta função, ou seja, durante a atualização do gráfico, surge um ícone de *reload* sobre o gráfico e todos os *inputs* que poderiam interferir com o processo de atualização são bloqueados (Figura 5.13).

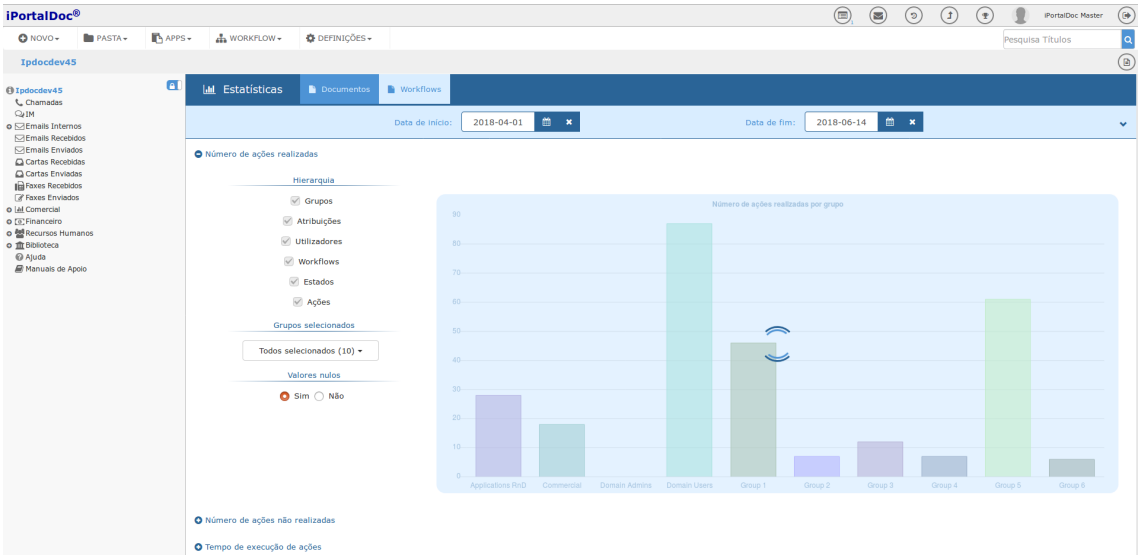


Figura 5.13: Interface da nova página de estatísticas durante a atualização de um gráfico

- ***getActionsNumberPerformedRequest()*** — Esta função é praticamente igual à anterior, só que é apenas executada quando a página de estatísticas é carregada. Houve a necessidade desta separação porque, sendo os primeiros dados a serem apresentados, não é preciso executar todas as funções que são utilizadas aquando da atualização do gráfico.

- ***updateBreadcrumbNumberActionsPerformed()*** — Em cima do gráfico é apresentado o caminho pelos diversos níveis da hierarquia até chegar ao nível atual. Para isso utilizou-se o componente *Breadcrumb* do *Bootstrap*. Esta função atualiza o *Breadcrumb* com base no nível atual, nos níveis da hierarquia escolhidos e pela sequência de elementos até ao nível em questão. Sempre que se *clique* nos elementos deste caminho (excluíndo o último elemento), é executada a função *changeNumberActionsPerformedLevel()* cujo argumento passado é o identificador do nível para o qual se pretende ir;

- ***changeNumberActionsPerformedLevel()*** — Esta função armazena as configurações relativas ao nível atual, alterando-o de seguida com base no valor passado como argumento. De forma a atualizar o *Breadcrumb* e o gráfico, são também executadas as funções *updateBreadcrumbNumberActionsPerformed()* e *updateActionsNumberPerformedRequest()*.

Na Figura 5.14 é possível observar o resultado final da nova interface de estatísticas. Como já foi referido, qualquer alteração que ocorra, quer nos filtros, quer nas opções para cada gráfico, implicará uma atualização daquilo que é visualizado. Nesse sentido, algumas das funções apresentadas estão associadas a eventos dos vários elementos da página.

Sempre que haja uma alteração nas datas de início e de fim, serão atualizados os gráficos, que no caso do número de ações realizadas, acontece através da execução da função *updateActionsNumberPerformedRequest()*). O mesmo se sucede caso seja *clicado* o botão de *refresh*, que se encontra no início do *Breadcrumb*. O *plugin Bootstrap Multiselect* permite várias configurações

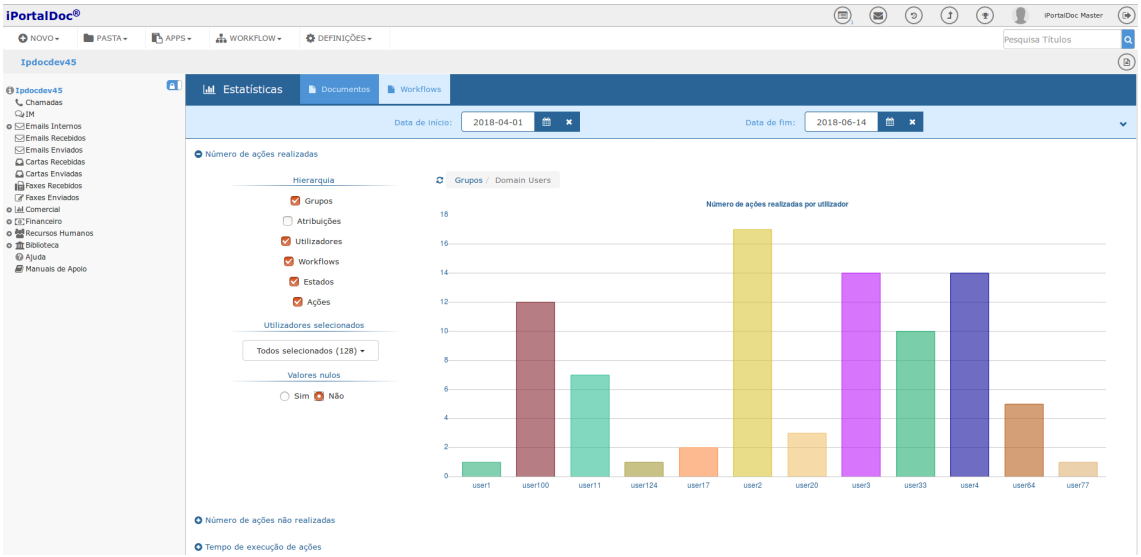


Figura 5.14: Interface final da nova página de estatísticas

como a existência de uma *checkbox* de selecionar todos e de um campo de pesquisa, limitação do número de elementos selecionados, entre outros. Também é possível definir funções associadas a eventos deste *plugin*, como é o caso dos eventos *onChange* e *onSelectAll*. Sempre que exista uma alteração dos elementos selecionados, é atualizado o objeto *idDataSelected* do nível em questão, sendo de seguida executadas as funções *updateDataForChartNumberActionsPerformed()*, *updateDataChartNumberActionsPerformed()* e *initNumberActionsPerformedChart()*, sucessivamente. Tal procedimento ocorre quando há uma alteração na opção de valores nulos, sendo que, neste caso, é atualizado o *zeroValues* e de seguida, são executadas as mesmas funções, pela mesma ordem.

Havendo uma interação com o gráfico, isto é, um *clique* numa das barras, é executado um determinado procedimento. Este consiste, numa primeira fase, em guardar as configurações relativas ao nível atual, sendo também armazenado no *elementIdSequence* (que guarda a sequência entre níveis), o identificador do elemento que foi *clicado*. Depois é atualizado o nível, sendo este, o nível seguinte com maior detalhe e que esteja ativo na respetiva hierarquia. Caso existam configurações deste nível associados ao elemento selecionado, estas serão carregadas. Por fim, são executadas as funções *updateBreadcrumbNumberActionsPerformed()* e *updateActionsNumberPerformedRequest()*, de forma a atualizar o *Breadcrumb* e o gráfico, respetivamente.

Por último, é também executada uma função sempre que uma das *checkboxes* da hierarquia seja alterada. O estado da *checkbox* é guardado no *hierarchySelected* (*array* que indica se um determinado nível está ativo ou não). Caso o utilizador desmarque todas os níveis, o primeiro nível é automaticamente selecionado de forma a que seja sempre apresentado o gráfico. Se o nível que sofreu a alteração seja maior (entenda-se por maior, um nível com mais detalhe) que o nível atual, o procedimento desta função termina logo após a atualização do *hierarchySelected*. Caso não seja, o nível seguinte passará a ser o nível hierarquicamente acima (menos detalhe) que esteja

ativo. Se porventura não haja nenhum nível ativo que seja menor que o nível atual, então o nível seguinte passa a ser o primeiro nível ativo maior. Após a definição do nível seguinte, do armazenamento das configurações do nível atual, e do carregamento das configurações do nível seguinte, são então novamente executadas as funções *updateBreadcrumbNumberActionsPerformed()* e *updateActionsNumberPerformedRequest()*.

5.4 Sumário

Neste capítulo foram descritas as implementações e respetivos resultados, para cada uma das áreas apresentadas na proposta de solução. Inicialmente foi implementado o conjunto de *feedbacks* com recurso ao *plugin bootstrap-tour*. Contudo, esta área não foi muito explorada nesta dissertação, tendo sido apenas implementado um *tour* para a interface principal. No entanto, o procedimento será o mesmo para a criação dos restantes *tours* que poderão vir a ser criados.

Relativamente à área de jogo, a implementação ocorreu em três fases. A primeira fase envolveu o desenvolvimento da interface da área de jogo, com as funcionalidades de alterar o *avatar* e o *nickname*, e ainda da interface de configuração dos níveis. Na segunda fase foi desenvolvida a interface de configuração de pontos, em que o administrador poderá definir quais os *workflows* e ações que irão contribuir pontualmente para o sistema de jogo, bem como a respetiva pontuação. Da primeira fase para a segunda houve uma alteração importante, do ponto de vista de implementação, passando o conceito de níveis a estar inserido nos emblemas. Isto é, existirá uma categoria de emblemas designada por *Níveis*, cujos emblemas dessa categoria representam os vários níveis do sistema de jogo. Após esta fase foi possível verificar que a realização de uma ação com uma determinada pontuação, implicava um aumento do número de pontos do utilizador que a realizava. Se os pontos atingissem a meta para um nível, este era conquistado. A última fase de implementação relativa à área de jogo, consistiu, maioritariamente, em alterações de *back-end*. Todos os procedimentos que eram executados ao nível do PHP e que podiam ser implementados ao nível da base de dados, foram alterados para tal, através de *triggers* e de funções em PL/pgSQL. Nesta fase também foi desenvolvida a secção da área de jogo onde são apresentados os emblemas conquistados pelo utilizador.

Por fim foi desenvolvida a nova página de estatísticas. Com recurso à biblioteca *chart.js* foi possível criar gráficos responsivos e que alteraram por completo o *design* e a *user experience* relativamente à antiga página de estatísticas. Foi também descrito todo o processo que permitiu apresentar dados estatísticos pelos diversos níveis hierárquicos, que é, talvez, o aspeto mais importante desta nova interface.

Capítulo 6

Validações

Todo o conjunto de implementações que foram realizadas nesta dissertação, tinham como objetivo corrigir os problemas identificados ao *iPortalDoc*, melhorando a experiência do utilizador com o sistema. Apesar das alterações propostas não terem sido totalmente implementadas, este capítulo irá descrever o processo e os métodos a usar de forma a validar, se de facto, a nova solução teve o impacto desejado. Numa fase inicial serão apresentados alguns métodos que poderão ser usados no âmbito da *user experience research*, sendo de seguida apresentadas as validações a serem realizadas após a implementação do sistema.

6.1 Métodos de *user experience research*

User research pode ser definido como o estudo sistemático dos objetivos, necessidades, e capacidades dos utilizadores, bem como, do design, construção, ou melhoramento de ferramentas que beneficia a forma como os utilizadores trabalham e vivem [18]. Isto foi o que foi realizado nesta dissertação, seguindo o processo de design centrado no jogador definido por Burke [3]. Este capítulo de validações consiste mesmo na especificação do que será realizado relativamente ao último passo desse processo.

Para que se possa avaliar a solução implementada, é necessário a recolha de dados. Esses dados podem ser quantitativos ou qualitativos. Os dados quantitativos permitem medir contagens específicas (Quanto? Quantas vezes? Com que frequência?), não havendo ambiguidade nos resultados obtidos. Contudo é necessário saber em que contexto se inserem, para serem bem interpretados. Número de produtos vendidos e número de reclamações são exemplos de dados quantitativos. Por outro lado, os dados qualitativos não podem ser quantificados. Estes não são precisos, sendo suscetíveis a várias análises. Normalmente são usados para descrever o porquê das coisas acontecerem [20].

Também existe a diferença de dados comportamentais e dados de atitude. Os dados comportamentais são obtidos através da observação daquilo que as pessoas fazem, enquanto que os dados de atitude são relativos à própria opinião das pessoas, ou seja, aquilo que elas pensam. De seguida serão apresentados os métodos mais comuns, e que são usados para recolher dados dos vários

tipos. Os métodos de *user experience research* podem ser usados em diversas fases do projeto: no início do desenvolvimento da solução para obter *feedback* dos *mockups* criados; nas fases finais do desenvolvimento, em que são disponibilizados protótipos aos utilizadores; e ainda após a versão final, de forma a comparar a solução com as suas versões antigas ou com soluções concorrentes.

- **Testes de usabilidade** — Os testes de usabilidade consistem em observar um conjunto de utilizadores reais, na realização de determinadas tarefas, com recurso ao sistema ou protótipo que se está a avaliar. Na generalidade, existem dois tipos de testes de usabilidade: formativos e sumativos. Os testes formativos estão relacionados com a procura e correção de problemas de usabilidade, enquanto que os testes sumativos permitem descrever a usabilidade da aplicação com o uso de métricas. Os testes formativos representam a maioria dos testes de usabilidade e, apesar de estarem mais relacionados com dados qualitativos, podem ser na mesma quantificados, por exemplo, em termos de frequência, ou em termos de tempo em que demoraram a realizar a tarefa. Os testes sumativos também podem ser divididos em dois tipos: *benchmarks* — permite descrever a aplicação relativamente a um conjunto de métricas padrão; comparativos — permite comparar resultados entre produtos distintos (versões diferentes ou mesmo produtos concorrentes). Nos testes comparativos, os mesmos utilizadores podem realizar tarefas nos diversos produtos (*within-subjects design*) ou diferentes conjuntos de utilizadores trabalharem em cada produto (between-subjects design)[17];

- **Testes A/B** — Nos testes A/B são apresentadas duas alternativas da solução aos utilizadores, e são medidos diversos parâmetros de forma a perceber qual das alternativas está mais de acordo com os objetivos propostos. Em muitos destes testes, as alternativas variam apenas em pequenos aspetos, como por exemplo a cor de um botão. Com a medição do número de vezes que aquele botão é *clicado* em ambos os casos, é possível escolher qual o melhor design a adotar;

- **Inquéritos** — Os inquéritos são a maneira mais simples que obter dados de atitude dos clientes. São normalmente constituídos por uma lista de questões, cujas respostas podem ser de sim ou não, de indicação de um determinado nível (por exemplo, indique de 1 a 5 o seu grau de satisfação) ou, eventualmente, de resposta aberta;

- **Entrevistas** — Conversas entre a pessoa responsável por projetar a solução e um participante, que pode ser um potencial utilizador final, uma parte interessada no negócio ou até mesmo um colega. São normalmente usadas para obter conhecimento sobre necessidades, desejos, motivações e frustrações das pessoas [20]. Nesta dissertação já foram apresentadas as principais ilações retiradas das entrevistas realizadas pela Ana Rita Mendes [11];

- **Grupos de foco** — Este método é praticamente igual ao das entrevistas só que em vez de ser apenas um participante, é um grupo de pessoas. Permite então ter uma maior perceção das ideias convergentes ou divergentes do grupo, em vez do ponto de vista individual;

- **Diário de estudo** — Neste método, é pedido aos utilizadores que registem todas as atividades, interações e atitudes que têm relacionadas com o sistema. Este registo detalha como as pessoas usam o sistema no contexto do dia a dia;
- ***Card Sorting*** — É uma forma que ajuda a perceber como é que as pessoas agrupam e categorizam a informação. Basicamente consiste em escrever palavras ou frases em cartões e pedir às pessoas que as agrupem por categorias. Este método é muito usado para ajudar a estruturar e organizar as várias navegações de um sistema;
- **Testes de árvore** — Este método é usado com o mesmo objetivo do *Card Sorting* e consiste no processo inverso. É apresentada às pessoas uma determinada estrutura e é pedido que estas encontrem um determinado elemento [20];
- **Acompanhamento do movimento dos olhos** — Este método consiste em utilizar um sistema que permite acompanhar o movimento dos olhos. Desta forma, é possível observar qual o impacto de uma determinada interface no utilizador, e saber o que o atrai mais;
- **Estudos no terreno** — Neste método, os potenciais utilizadores do sistema são observados no meio onde habitualmente se inserem, de forma a registarem os seus comportamentos, rotinas e dinâmicas. Este estudo, para além das observações, poderá também contemplar outros métodos, como por exemplo, entrevistas e inquéritos.

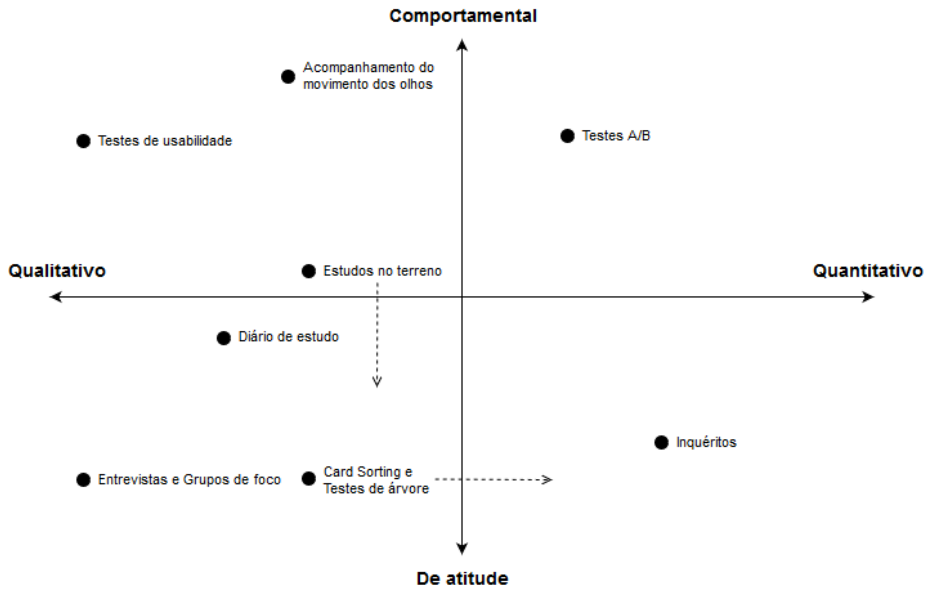


Figura 6.1: Enquadramento dos vários métodos de *user experience research* em função do tipo de dados (adaptada de [13])

Na Figura 6.1 é possível observar os métodos de *user experience research*, que foram apresentados, em função dos vários tipos de dados.

6.2 Validações a efetuar à solução implementada

Dado que as validações irão ocorrer numa fase em que o sistema já se irá encontrar finalizado, os testes de usabilidade e inquéritos serão os métodos que irão ser utilizados.

6.2.1 Testes de usabilidade

Relativamente aos testes de usabilidade, estes serão maioritariamente comparativos, de forma a comparar os resultados antes e após a solução implementada. Os dados terão de ser recolhidos, durante um período de tempo, em três alturas distintas: uma com o estado atual do sistema; outra logo após a implementação da solução; e por fim uma passados alguns meses depois da solução estar implementada. Desta forma, irá ser possível concluir se realmente a solução teve os efeitos desejados, e se os resultados obtidos numa fase inicial, não foram influenciados pelo facto da solução ser uma novidade para os utilizadores (*Novelty effect*). Os dados a recolher são:

- número de ações realizadas;
- número de ações não realizadas;
- tempo de execução das ações;
- tempo de conclusão dos *workflows*;
- número de acessos à página de estatísticas;
- número de relatórios estatísticos pedidos.

Como foi referido, um dos objetivos da solução era aumentar a produtividade dos seus utilizadores. Isso, em termos do sistema *iPortalDoc*, pode ser visto como o aumento do número de ações realizadas, a diminuição do número de ações não realizadas, a diminuição do tempo de execução das ações e a diminuição do tempo de conclusão dos *workflows*. Estes quatro parâmetros irão permitir avaliar se de facto, quer o conjunto de *feedbacks*, quer a área de jogo, contribuíram para esse aumento de produtividade. Por seu lado, o número de acessos à página de estatísticas e o número de relatórios estatísticos pedidos irão avaliar o impacto da nova página de estatísticas. Se após a implementação da solução, o número de acessos à página aumentar e o número de relatórios pedidos diminuir, quer dizer que o desenvolvimento da página teve o efeito desejado.

Para além destes dados recolhidos, é importante ter em conta todo o *feedback* que os utilizadores vão dando aquando da utilização do sistema. Apesar de em termos dos dados recolhidos, os resultados poderem ser satisfatórios, existe sempre algo a alterar, e que os utilizadores achem que pode melhorar.

6.2.2 Inquérito

Os testes de usabilidade são importantes no sentido de ser puder comparar dados entre o antes e o após solução, contudo, a realização de um inquérito irá permitir obter dados relativos à própria opinião dos utilizadores. Desta forma, nesta fase de validações, será usado um questionário de usabilidade *standard*, isto porque estes são objetivos, replicáveis e quantificáveis. Para além disso, os questionários *standard* são mais confiáveis que os questionários não *standard*.

O questionário usado será o *System Usability Scale* (SUS) desenvolvido por John Brooke [2]. Consiste numa escala simples de dez elementos, que oferece uma visão global das avaliações subjetivas da usabilidade. Cada questão tem cinco opções de resposta, desde *Discordo fortemente* até *Concordo fortemente*. As questões são as seguintes:

1. Eu gostava de usar este sistema com frequência?
2. O sistema é desnecessariamente complexo?
3. O sistema é fácil de usar?
4. Eu preciso de apoio técnico para poder usar este sistema?
5. As várias funções deste sistema foram bem integradas?
6. Existem muitas inconsistências no sistema?
7. A maioria das pessoas apreenderá a usar este sistema rapidamente?
8. O sistema é muito complicado de usar?
9. Sinto-me confiante a usar o sistema?
10. Eu preciso de apreender muitas coisas antes de usar o sistema?

O SUS é normalmente usado após o utilizador ter tido a oportunidade de usar o sistema que está a ser avaliado. A resposta a cada questão deve ser imediata e todas as questões devem ser respondidas. Caso não saiba responder a uma questão, a pessoa deve colocar, como resposta, o ponto central da escala (a escala de respostas é de 1 a 5).

Este inquérito produz, como resultado, um único número, que representa a usabilidade geral do sistema em estudo. Para calcular essa pontuação, primeiro é necessário somar as contribuições pontuais de cada questão. Para as questões com número ímpar, a contribuição pontual é o valor da escala menos 1, enquanto que para as questões com número par, a contribuição é de 5 menos o valor da escala. O resultado global do SUS é obtido multiplicando a soma das pontuações por 2,5. As pontuações do SUS variam entre 0 a 100. A média geral de resultados do SUS é de 68 com um desvio-padrão de 12,5 [18].

6.3 Sumário

Neste capítulo foram inicialmente apresentados os métodos de *user experience research*. Estes métodos consistem em formas de recolha de dados para avaliar um determinado sistema. Pelo facto da solução implementada ao longo desta dissertação não estar totalmente concluída, não foi possível validar com resultados concretos, a concretização dos objetivos propostos. Apesar disso, foram enumerados os diversos dados a recolher antes e após a implementação da solução, para se comparar os resultados de ambos os casos. Por fim, foi apresentado um inquérito a ser entregue aos utilizadores após a utilização do sistema com as novas funcionalidades, para se obter uma visão global das avaliações subjetivas da usabilidade.

Capítulo 7

Conclusões e Trabalho Futuro

Esta dissertação tinha como objetivo principal implementar, no sistema de gestão documental e de processos *iPortalDoc*, técnicas de *ludificação* que permitissem motivar e envolver os respetivos utilizadores. Contudo, pelo facto do *iPortalDoc* ser um sistema utilizado por diversas empresas, com objetivos distintos, a solução *ludificada* foi obrigada a ser o mais versátil e abrangente possível.

7.1 Satisfação dos Objetivos

O primeiro objetivo envolvia o estudo e análise da proposta de solução apresentada pela Ana Rita Mendes. Esse estudo permitiu retirar ideias para o que realmente foi implementado nesta dissertação, apesar de, como está justificado no Capítulo 4, ter-se optado por outra abordagem. Essa abordagem permitiu que o foco do utilizador se mantivesse no que realmente se pretende que ele faça. A proposta de solução envolvia desta forma um conjunto de *feedbacks*, uma área de jogo e uma nova página de estatísticas. Esta nova página de estatísticas foi incluída na proposta de solução após a análise efetuada ao sistema. Isto porque, concluiu-se que esta interface apresentava muitas lacunas e que já não estava adaptada ao sistema atual. O facto do módulo de estatísticas estar incluído no pacote base do *iPortalDoc*, aumentou ainda mais a necessidade de melhorias nesta interface.

Foram então desenvolvidas as respetivas interfaces e funcionalidades propostas. Contudo, por questões de complexidade e de limitação de tempo, não foi possível implementar tudo o que foi proposto. Por isso, procurou-se desenvolver uma base de forma a facilitar a implementação das restantes funcionalidades no sistema, num trabalho futuro.

O facto de o sistema *ludificado* não ter ficado totalmente concluído, impossibilitou a concretização do último objetivo, que passava por validar se, de facto, a solução proposta contribuía para a eficiência dos trabalhadores. Todas as áreas desenvolvidas foram apenas testadas numa máquina de testes, sendo necessária uma validação da solução num sistema de maiores dimensões e num ambiente mais real. Apesar disso, foram apresentados os métodos a utilizar numa futura validação do sistema.

7.2 Trabalho Futuro

Como já foi referido, não foi possível a implementação de tudo o que foi especificado, sendo necessária a continuidade do trabalho realizado nesta dissertação. Nenhuma das três áreas descritas na proposta de solução foram totalmente concluídas, contudo, como foi possível constatar, grande parte do trabalho está realizado, faltando apenas adaptar algumas das implementações já realizadas.

7.2.1 Conjunto de *feedbacks*

No que diz respeito ao conjunto de *feedbacks*, o trabalho restante passa por identificar quais as interfaces que suscitam mais dúvidas ao utilizador e implementar *tours* que permitem auxiliá-los nessas situações. A criação desses novos *tours* seguirá o mesmo processo que foi apresentado nesta dissertação. É também necessário haver um registo de quais utilizadores é que já visualizaram os *tours* de forma a que, numa próxima situação, estes só sejam apresentados caso o utilizador assim o pretenda.

Algo que não foi explorado nesta dissertação e que foi descrito na proposta de solução, é a apresentação de *feedbacks* em situações de erro, como por exemplo, 404 - página não encontrada. Para este tipo de *feedbacks* já não será necessária a implementação de *tours*. A solução poderá passar por ícones engraçados ou elementos lúdicos, como jogos simples, que até poderiam contribuir para a pontuação da área de jogo.

7.2.2 Área de jogo

A área de jogo é talvez a que necessite de mais trabalho num futuro. A interface da área de jogo está praticamente concluída, faltando apenas a apresentação das conquistas por parte dos restantes colegas. Contudo será necessário projetar e desenvolver as páginas relativas às classificações e recompensas, que não existem atualmente. Consequentemente será também necessária a criação de uma nova secção na interface de configuração de forma a que o utilizador possa definir e configurar as respetivas recompensas.

Tal como foi referido aquando da descrição da implementação, é necessária uma adaptação das interfaces de configuração de níveis e da distribuição de pontos. Isto porque os níveis representam apenas uma categoria de emblemas, sendo preciso uma área de configuração que permita, gerir as diversas categorias (permitindo a adição ou remoção), bem como configurar os diversos emblemas. Essa secção permitiria então adicionar e remover níveis a uma determinada categoria e alterar parâmetros como nome, descrição, pontos necessários para ser atingido, imagem do emblema, entre outros. A página de distribuição de pontos permite apenas definir a pontuação que um determinado *workflow* ou ação contribui para os emblemas de níveis, sendo necessária uma adaptação para que seja possível definir os pontos para as restantes categorias de emblemas.

Algo que não tem haver com aspetos de implementação mas que é relevante para esta área de jogo, é a própria definição dos níveis, dos emblemas e das recompensas. Isto é algo que será

configurável por parte dos administradores mas é necessário que estes tenham conhecimentos relativos ao conceito de *ludificação*. Desta forma será sempre necessário um suporte por parte da empresa IPBRICK aos clientes que adotem este módulo, pelo facto de o espaço de jogo e planeamento da jornada variar consoante os objetivos da empresa e dos respetivos colaboradores. Isto implica um estudo para cada caso de forma a que as configurações definidas sejam as mais indicadas para cada empresa.

7.2.3 Página de estatísticas

Relativamente à página de estatísticas, na secção dos dados estatísticos que dizem respeito aos *workflows*, falta apenas projetar os gráficos para os restantes parâmetros estatísticos, para além do número de ações realizadas e não realizadas. Após essa projeção (que envolve a definição da hierarquia e que tipo de gráfico será apresentado), a implementação seguirá os mesmos moldes que foram apresentados nesta dissertação. Se houver também uma adaptação das funções, que foram descritas no capítulo da implementação, de forma a que seja passado como argumento o respetivo gráfico, a implementação dos restantes gráficos será ainda mais rápida e de uma forma mais facilitada.

Como foi referido anteriormente, a secção destinada às estatísticas relativas aos documentos não foi sequer projetada. Contudo, essa secção será semelhante à dos *workflows*, cujo *layout* já está implementado e os procedimentos para a criação dos gráficos também. Por fim, seria importante que os dados e os gráficos apresentados pudessem ser exportados, para PDF ou ficheiros de folhas de cálculo.

Referências

[1] Richard Bartle. Hearts, clubs, diamonds, spades: Players who suit muds. *Journal of MUD research*, 1(1):19, 1996.

[2] John Brooke. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[3] B. Burke. *Gamify: How Gamification Motivates People to Do Extraordinary Things*. Bibliomotion, Incorporated, 2014.

[4] Yu–kai Chou. Octalysis – complete gamification framework, março 2015. Disponível em http://yukaichou.com/gamification-examples/octalysis-complete-gamification-framework [último acesso a 29-01-2018].

[5] D. DeHaven e S. Ferebee. Why gaming is working in higher ed, novembro 2012. Disponível em http://techonomy.com/2012/11/why-gaming-is-working-in-higher-ed/ [último acesso a 01-02-2018].

[6] S. Deterding, D. Dixon, R. Khaled, e L. Nacke. From game design elements to gamefulness: Defining "gamification". Em *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, MindTrek 2011*, páginas 9–15, 2011.

[7] P. Hilton. 10 workflow examples for signavio workflow accelerator, junho 2017. Disponível em https://www.signavio.com/post/10-workflow-examples/ [último acesso a 31-01-2018].

[8] M.M. Joy e M.J. Chiramel. *Fun is The Future: A Collection of Compelling Gamification Success Stories*. Educreation Publishing, 2017.

[9] S. Khan, U. Rani, B. V. N. Prasad, A. K. Srivastava, S. Selvi, e D. K. Gautam. Document management system: An explicit knowledge management system. Em *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, páginas 402–405, março 2015.

[10] J. Kumar, M. Herger, e R. Dam. Bartle's player types for gamification, agosto 2017. Disponível em https://www.interaction-design.org/literature/article/bartle-s-player-types-for-gamification [último acesso a 28-01-2018].

[11] Ana Rita Mendes. Gamification framework for a document management system. Tese de mestrado, Faculdade de Engenharia da Universidade do Porto, 2017.

[12] A. Mora, D. Riera, C. Gonzalez, e J. Arnedo-Moreno. A literature review of gamification design frameworks. Em *7th International Conference on Games and Virtual Worlds for Serious Applications (VS-Games)*, 2015.

[13] Christopher Murphy. A comprehensive guide to ux research, janeiro 2018. Disponível em https://www.smashingmagazine.com/2018/01/comprehensive-guide-ux-research/ [último acesso a 21-06-2018].

[14] Regina Nelson. Gamification mechanics, dynamics and components, abril 2015.

[15] R.M. Ryan e E.L. Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology*, 25(1):54–67, 2000.

[16] R.M. Ryan e E.L. Deci. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist*, 55(1):68–78, 2000.

[17] Jeff Sauro e James R. Lewis. *Quantifying the User Experience, Second Edition: Practical Statistics for User Research*. Morgan Kaufmann Publishers Inc., 2ª edição, 2016.

[18] Robert M. Schumacher. *Handbook of Global User Research*. Morgan Kaufmann, 2010.

[19] K. Seaborn e D.I. Fels. Gamification in theory and action: A survey. *International Journal of Human Computer Studies*, 74:14–31, 2015.

[20] Amanda Stockwell. Choosing the right ux research method, janeiro 2018. Disponível em https://uxmastery.com/choosing-right-ux-research-method/ [último acesso a 20-06-2018].

[21] S.P. Walz e S. Deterding. *The Gameful World: Approaches, Issues, Applications*. MIT Press, 2015.

[22] T. Wayman. *Document Management Overview: A guide to the benefits, technology and implementation essentials of digital document management solutions*. Laserfiche Institute, 5ª edição, setembro 2004.

[23] K. Werbach e D. Hunter. *For the Win: How Game Thinking Can Revolutionize Your Business*. Wharton Digital Press, 2012.