

Instituto Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Viseu



Instituto Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Viseu



RESUMO

Atualmente, muitas empresas de desenvolvimento de *software* ou de soluções tecnológicas são constituídas por equipas, em que parte delas integram colaboradores *outsourcing*, ou seja, são colaboradores que pertencem a outra empresa mas que diariamente realizam as suas funções nas instalações das empresas clientes. As empresas de consultoria fornecem programadores/consultores na área para trabalharem com os seus clientes, o que implica que diariamente, os consultores mudem de rotinas. Cada vez mais, existem empresas a dedicarem-se à prestação deste tipo de serviços. No entanto, tal como todas as outras empresas no mercado, precisam de crescimento.

Um grande problema que estas empresas de consultoria enfrentam é que grande parte dos seus colaboradores perdem o conhecimento/contacto com a empresa e com os restantes colegas. No entanto, uma empresa carece essencialmente do apoio dos seus colaboradores para crescer. O ideal seria os colaboradores sentirem-se motivados a apoiar esse crescimento, contudo tal desafio será abordado ao longo desta dissertação de forma que se idealize uma possível solução.

Ao longo da história, as pessoas foram cada vez mais educadas de modo a que a sua mentalidade possa ser facilmente influenciada por jogos, devido à competição e ao cumprimento de objetivos que motivam os utilizadores a jogar. *Gamification* é um conceito relativamente recente que aplica mecânicas de jogos em contextos não lúdicos, para motivar as pessoas a alcançar determinados objetivos. Quem não gosta de estar no topo de *rankings*? E de ter uma boa coleção de prémios derivados do seu esforço tornando uma boa imagem empresarial? São cada vez mais as empresas a aplicar esta metodologia para motivar os seus colaboradores a trabalharem em prol da empresa e com os colegas. Estima-se que em 2015, 70% das empresas mundiais terão uma plataforma interna de *gamification*, com vista a impulsionar a inovação, dar valor à informação “parada” e/ou gerar nova informação, mantendo os colaboradores ativos e motivados.

A XLM – *Innovation & Technology* é uma empresa com este carácter e paradigma. Atualmente, a empresa inclui uma gestão de IDI (Investigação, Desenvolvimento e Inovação), ou seja, possui ferramentas acessíveis aos colaboradores para que participem no crescimento da empresa colocando ideias. No entanto, a realidade é que a plataforma responsável pela bolsa de ideias é pouco movimentada. Além disso, existe pouca eficiência na partilha de informação entre os

colaboradores, entre outros fatores que reduzem a cooperação dos colaboradores no crescimento da empresa. A ideia para o presente projeto baseia-se no desenvolvimento de uma plataforma interna de *gamification*, integrando os sistemas já existentes, com a finalidade de motivar os colaboradores a participarem neste processo de crescimento e inovação, sem se tornar notoriamente intrusivo no dia-a-dia.

Um dos mecanismos usado num processo que emprega o conceito de *gamification* baseia-se na distinção dos colaboradores relativamente a conhecimentos adquiridos, boas práticas ou uma colaboração extraordinária para com a empresa. Estas distinções aumentam a “reputação” do colaborador perante os seus colegas e criam um sentido de reconhecimento do seu valor. Para a presente dissertação, pretende-se estudar e implementar os métodos de *gamification* mais adequados para a resolução dos problemas da empresa.

ABSTRACT

Nowadays, many software development or technological solution companies consist of multiple teams, and part of these teams integrate outsourcing collaborators, meaning these collaborators belong to another company but still do their daily job at the client companies. Consulting firms provide programmers/consultors in its respective area in order to work with client companies, which means that consultors often change routines daily. There is an increasingly amount of companies which provide this kind of service. Still, just like any other company available in the market, they need growth.

A big problem these consulting firms face is the fact that a large part of their collaborators lose contact or knowledge with the company and the rest of their colleagues. However, a company often needs the support of its collaborators to grow. The ideal would be for the collaborators to feel motivated to support this growth. This challenge will be addressed during this thesis in a way that reaches towards a possible solution.

In recent history, people have started to be trained in a way their mentality can be easily influenced by games, due to the competition and the accomplishment of objectives that motivate their users to keep playing. 'Gamification' is a relatively recent concept which applies game mechanics to non-recreational contexts, in order to motivate people to achieve certain objectives. I mean, who doesn't like to top the rankings? How about a nice collection of prizes obtained through effort while making a good business image? More and more companies are using this methodology in order to motivate its collaborators to work for the company and with their colleagues. It is estimated that in 2015, 70% of global companies will have an internal gamification platform, in order to propel innovation, evaluate information and/or generate new information, and keep its collaborators active and motivated.

XLM – Innovation & Technology is a company with this ethos and paradigm. Nowadays, the company includes an IDI (Investigation, Development and Innovation) management, which means it possesses accessible tools to its collaborators, so that they can participate in the company's growth by implanting ideas. However, the reality is that the platform responsible for the exchange of ideas is not often used. Also, there is little efficiency when sharing information

between collaborators, as well as other factors that reduce cooperation between collaborators for the company's growth. The idea for the current project is based on the development of an internal gamification platform, integrating the already existing system, with the goal of motivating collaborators to participate in this growth and innovation process, without becoming noticeable meddlesome in daily life.

One of the mechanisms used is a process which utilizes the concept of gamification is based on the relative distinction between collaborators to obtain knowledge, best practices or an extraordinary contribution to the company. These distinctions increase the "reputation" of the collaborator in the eyes of his or her colleagues and create a sense of recognition of his or her value. For this thesis, the objective is to study and implement the most appropriate methods of gamification in order to ameliorate the company's problems.

PALAVRAS-CHAVE

Gamification

Gestão de inovação

Integração de sistemas

Colaboradores

Motivação

KEYWORDS

Gamification

Innovation management

Systems integration

Cooperator

Motivation

AGRADECIMENTOS

Na conclusão deste projeto de mestrado vejo um reconhecimento importante academicamente, que resultou de um enorme esforço e dedicação ao longo deste últimos dois anos, período de frequência do Mestrado em Sistemas e Tecnologias de Informação para as Organizações, com acumulação de uma vida profissional.

Ao longo deste tempo superar os desafios só foi possível devido ao apoio de determinadas pessoas que direta ou indiretamente estiveram sempre presentes.

Quero começar por agradecer ao meu orientador, Engenheiro Francisco Ferreira Francisco, por toda a disponibilidade e profissionalismo no apoio, nas sugestões e nas orientações prestadas ao longo da realização de todo o projeto. Conseguiu, principalmente, simplificar e incentivar a realização de todas as tarefas, que inicialmente pareciam demasiado complicadas.

Quero agradecer à empresa para qual exerço a minha atividade profissional diariamente, XLM *Innovation & Technology*, pela proposta e possibilidade que ofereceu para trabalhar neste projeto. Agradeço, especialmente, ao Engenheiro André Ferro, por toda a disponibilidade e apoio que prestou. A sua ajuda foi fundamental para todas as fases do projeto, nomeadamente, no levantamento de requisitos, nos testes e na aceitação de todas as componentes desenvolvidas no projeto.

Agradeço também aos meus colegas de trabalho pela ajuda e sugestões que propuseram para o projeto.

Quero agradecer aos meus colegas do MSTIO por todo o apoio, ensinamentos e amizade ao longo destes últimos dois anos. Sem a amizade deles, todo este caminho teria sido muito mais complicado.

Por fim, agradeço à minha família por toda a compreensão e conselhos que me ofereceram durante estes últimos anos. Sem eles, este percurso académico não teria sido possível.

ÍNDICE

ÍNDICE DE FIGURAS	XV
ÍNDICE DE TABELAS	XIX
ABREVIATURAS E SIGLAS	XXI
1. Introdução.....	1
1.1. Contextualização	2
1.1.1. Atividade de <i>Outsourcing</i> e Problemas	3
1.1.2. A Empresa	4
1.2. Objetivos	5
1.3. Motivação	6
1.4. Metodologias e Planeamento de Trabalho.....	7
1.4.1. Calendarização.....	9
1.5. Estrutura da Dissertação.....	10
2. Estado da Arte.....	13
2.1. Motivação nos Indivíduos	14
2.2. <i>Gamification</i>	17
2.2.1. Jogos.....	18
2.2.2. Jogadores	19
2.2.3. Elementos e Mecanismos de Jogo.....	20
2.2.4. Áreas Aplicacionais	24
2.2.5. <i>Gamification</i> e as Organizações.....	25
2.3. Soluções Existentes	26
2.3.1. Sistemas vocacionados para clientes de uma organização.....	27
2.3.2. Sistemas vocacionados para colaboração de uma organização	29
2.3.3. Em Portugal.....	33
2.4. Enquadramento Legal	36
2.5. Conclusões.....	36
3. Análise de Requisitos.....	39
3.1. Enquadramento.....	40

3.2.	Sistemas Externos na Empresa	42
3.2.1.	LagoonIdeas	43
3.2.2.	Intranet	43
3.2.3.	LagoonSecurity	44
3.3.	Identificação dos Utilizadores	45
3.4.	Requisitos Funcionais.....	46
3.4.1.	API para Sistemas Externos.....	47
3.4.2.	<i>BackOffice</i>	48
3.4.3.	<i>FrontOffice</i>	59
3.4.4.	Comunicação com o LagoonSecurity	61
3.5.	Requisitos não Funcionais	63
3.5.1.	Desempenho.....	63
3.5.2.	Usabilidade	64
3.5.3.	Segurança.....	65
3.5.4.	Portabilidade.....	66
4.	Arquitetura e Conceção.....	67
4.1.	Arquitetura	68
4.1.1.	Arquitetura “ <i>Three Tier</i> ”	68
4.1.2.	Arquitetura MVC.....	69
4.1.3.	Arquitetura Alto-Nível	71
4.2.	Tecnologias Utilizadas.....	73
4.2.1.	PHP	74
4.2.1.1.	<i>CodeIgniter</i>	74
4.2.2.	HTML5 e CSS3	75
4.2.2.1.	<i>BootStrap</i>	76
4.2.3.	<i>JavaScript</i>	76
4.2.4.	AJAX.....	77
4.2.5.	MySQL	77
4.2.6.	Outros	77
4.3.	Casos de Uso	79
4.4.	Diagramas de Atividades	81

4.4.1.	Diagrama de Atividades da API	81
4.4.2.	Diagrama de Atividades para Gestão de Acessos e Permissões	83
4.4.3.	Diagrama de Atividades para a Gestão de Colaboradores	84
4.4.4.	Diagramas de Atividades para a Gestão de Eventos.....	86
4.4.4.1.	Gestão Eventos por Colaborador	86
4.4.4.2.	Adição de Colaborador em Evento	88
4.4.4.3.	Eliminação de Colaborador em Evento.....	89
4.5.	Modelo de Dados	90
4.5.1.	Modelo Entidade-Relação	90
4.5.2.	Entidades.....	91
5.	Desenvolvimento e Implementação.....	101
5.1.	O sistema	102
5.1.1.	API para Sistemas Externos	102
5.1.2.	<i>BackOffice</i>	105
5.1.2.1.	Estrutura e Funcionalidades	105
5.1.3.	<i>FrontOffice</i>	122
5.1.3.1.	Estrutura e Funcionalidades	123
5.1.4.	Comunicação com o LagoonSecurity.....	131
5.2.	Usabilidade e <i>Design</i> de Interação	135
5.3.	Segurança da Plataforma	143
5.4.	Testes Funcionais e de Usabilidade.....	145
6.	Conclusão.....	147
6.1.	Propostas Futuras.....	148
	REFERÊNCIAS.....	151
	Anexo A – Casos de Uso.....	157
	Anexo B – Modelo Físico de Dados	159

ÍNDICE DE FIGURAS

Fig. 1 - Fases da realização do projeto	7
Fig. 2 - Modelo de Kevin Werbach, retirado de (Alves, 2014)	20
Fig. 3 - Modelo de gamification retirado de (recyclebank.com, 2013).....	21
Fig. 4 - Apresentação LagoonSecurity	44
Fig. 5 - Exemplo conceitos LagoonSecurity	45
Fig. 6 - Diagrama das três camadas da arquitetura "Three Tier"	68
Fig. 7 - Diagrama arquitetura MVC no modelo "Three Tier"	70
Fig. 8 - Diagrama de alto nível da arquitetura proposta.....	71
Fig. 9 - Diagrama geral de módulos para os casos de uso.....	80
Fig. 10 - Diagrama de atividade da API.....	82
Fig. 11 - Diagrama de atividade de autenticação.....	83
Fig. 12 - Diagrama de atividades para gestão de colaboradores	85
Fig. 13 - Diagrama de atividade de gestão de eventos por colaborador.....	87
Fig. 14 - Diagrama de atividade de inserção de colaborador em evento.....	88
Fig. 15 - Diagrama de atividade de eliminação de colaborador em evento.....	89
Fig. 16 - Modelo Entidade-Relação.....	91
Fig. 17 - Exemplo de corpo de pedido json	103
Fig. 18 - Mensagem de resposta com erro quando identificação da aplicação não existe em sistema	104
Fig. 19 - Mensagem de resposta com erro quando atividade não existe em sistema ou não se relaciona com a aplicação.....	104
Fig. 20 - Mensagem de resposta com erro quando introduzido um colaborador que não existe em sistema ou não segue o formato identificado pela aplicação.....	104
Fig. 21 - Mensagem de resposta com sucesso.....	104
Fig. 22 - Estrutura base do backOffice.....	105
Fig. 23 - Formulário de acesso à plataforma.....	106
Fig. 24 - Página inicial do backOffice	107
Fig. 25 - Menu área "Colaboradores"	108
Fig. 26 - Listagem de colaboradores.....	108
Fig. 27 - Listagem de eventos por colaborador.....	109

Fig. 28 - Detalhes na gestão de um evento.....	109
Fig. 29 - Inserção de colaborador em evento	110
Fig. 30 - Criação de novo evento	111
Fig. 31 - Listagem de badges por colaborador.....	111
Fig. 32 - Atribuição manual de um badge a um colaborador.....	112
Fig. 33 - Listagem de badges para atribuição.....	113
Fig. 34 - Atribuição manual de um badge a vários colaboradores.....	113
Fig. 35 - Inserção de colaborador	115
Fig. 36 - Listagem das categorias	115
Fig. 37 - Listagem de atividades de uma categoria.....	116
Fig. 38 - Listagem de níveis de um jogo temporal aberto	117
Fig. 39 - Ferramenta OpenBadges - Desenho de badges	117
Fig. 40 - Gestão de métodos de atribuição automática de badges	119
Fig. 41 - Gestão de aplicação externa	120
Fig. 42 - Funcionalidade de estatísticas.....	122
Fig. 43 - Menu pessoal do backOffice	122
Fig. 44 - Estrutura base do frontOffice	123
Fig. 45 - Página inicial do frontOffice.....	124
Fig. 46 - Progressão nas diferentes categorias	125
Fig. 47 - Resumo de eventos realizados	126
Fig. 48 - Tabela de ranking	126
Fig. 49 - Resumo de colaboração.....	126
Fig. 50 - Resumo de badges	127
Fig. 51 - Feed com as últimas novidades.....	127
Fig. 52 - Informação de atividades existentes.....	128
Fig. 53 - Informação de badges	129
Fig. 54 - Informação de colaboração	129
Fig. 55 - Visualização de perfil de outro colaborador	130
Fig. 56 - Perfil de utilizador	131
Fig. 57 - Menu pessoal do frontOffice	131
Fig. 58 – E-mail de ativação de conta	133

Fig. 59 - Formulário de inserção de nova senha de acesso.....	133
Fig. 60 - E-mail de confirmação de ativação da conta.....	133
Fig. 61 - Formulário de solicitação de uma nova senha	134
Fig. 62 - E-mail de recuperação de senha	134
Fig. 63 - Menu principal do backOffice.....	137
Fig. 64 – Exemplo de formulário.....	140
Fig. 65 - Exemplo de interface de tabela.....	141
Fig. 66- Exemplo de mensagem de prevenção de erros	142
Fig. 67 - Mensagem de resposta de ação com sucesso.....	142
Fig. 68 - Mensagem de resposta de ação com insucesso.....	142
Fig. 69 - Diagrama caso de uso módulo "API"	3
Fig. 70 - Diagrama caso de uso módulo "Gestão de Acessos e Permissões"	4
Fig. 71 - Diagrama caso de uso módulo "Perfis de Colaboradores"	5
Fig. 72 - Diagrama caso de uso módulo "Gestão de Colaboradores"	6
Fig. 73 - Diagrama caso de uso módulo "Gestão Jogos Temporais"	7
Fig. 74 - Diagrama caso de uso módulo "Gestão Categorias"	8
Fig. 75 - Diagrama caso de uso módulo "Gestão Atividades"	9
Fig. 76 - Diagrama caso de uso módulo "Gestão Eventos"	10
Fig. 77 - Diagrama caso de uso módulo "Gestão Badges".....	11
Fig. 78 - Diagrama caso de uso módulo "Estatísticas"	12
Fig. 79 - Diagrama caso de uso módulo "Gestão Aplicações Externas".....	13

ÍNDICE DE TABELAS

Tabela 1 - Calendarização das fases	9
Tabela 2 - Resumo das teorias de conteúdo e seus pressupostos	15
Tabela 3 - Resumo das teorias de processos e seus pressupostos	16
Tabela 4 - Alguns mecanismos de gamification	22
Tabela 5 - Requisitos funcionais para a API.....	48
Tabela 6 - Requisitos para a gestão de jogos temporais	49
Tabela 7 - Requisitos funcionais para a gestão de categorias.....	50
Tabela 8 - Requisitos funcionais para a gestão de atividades	50
Tabela 9 - Requisitos funcionais para a gestão de colaboradores	52
Tabela 10 - Requisitos funcionais para a gestão de eventos.....	53
Tabela 11 - Requisitos funcionais para a gestão de badges	54
Tabela 12 - Requisitos funcionais para a gestão de aplicações externas	57
Tabela 13 - Outros requisitos funcionais para o backOffice	58
Tabela 14 - Requisitos funcionais para o frontOffice	59
Tabela 15 - Requisitos funcionais para a comunicação com o LagoonSecurity	62

ABREVIATURAS E SIGLAS

Ajax	<i>Asynchronous JavaScript and XML</i>
API	<i>Application Programming Interface</i>
CEO	<i>Chief Executive Officer</i>
CSS	<i>Cascading Style Sheets</i>
HTML	<i>HyperText Markup Language</i>
IDI	Investigação, Desenvolvimento e Inovação
JSON	<i>JavaScript Object Notation</i>
LS	<i>LagoonSecurity</i>
MDA	<i>Mechanics, Dynamics and Aesthetics</i>
MSTIO	Mestrado em Sistemas e Tecnologias de Informação para as Organizações
MVC	<i>Model-View-Controller</i>
PHP	<i>Hypertext Preprocessor</i>
REST	<i>Representational State Transfer</i>
SGBD	Sistema de Gestão de Base de Dados
SQL	<i>Structured Query Language</i>
TI	Tecnologia da Informação
URL	<i>Uniform Resource Location</i>
XML	<i>eXtensible Markup Language</i>
WWW	<i>World Wide Web</i>

1. Introdução

Todos os anos, o mundo assiste a novas evoluções tecnológicas. Pequenas e grandes equipas de desenvolvimento de *software* ou de outras soluções tecnológicas trabalham diariamente para sustentarem a manutenção e a inovação em ativo. Essa busca pela inovação tecnológica – TI – requer das organizações competências indisponíveis para a maioria delas. Portanto, muitas acabam por concluir que TI, ou parte de TI, não são de sua competência essencial e que não deve ser o foco de seus esforços na organização, levando-as à adoção de serviços de *outsourcing* de TI (Gay & Essinger, 2000), como uma forma de estratégia de gestão de conhecimento (Nicolau, 2002).

Uma empresa que dispõe serviços de *outsourcing* fornece consultores, no mínimo, com o conhecimento base adequado aos problemas que os clientes apresentam. Estas empresas têm a obrigação de manterem os seus colaboradores formados e preparados para trabalharem, diariamente ou não, com os diferentes clientes. Esta atividade traz diversas vantagens e desvantagens para a empresa prestadora. Nestas últimas, ressalta o “afastamento” dos colaboradores para as instalações dos clientes que origina uma “separação” entre os mesmos e a empresa. No entanto, tal como todas as outras empresas no mercado, a empresa prestadora carece de crescimento e inovação. O apoio dos colaboradores e a sua participação próxima e ativa são os “métodos” essenciais para permitir estes processos. Porém, torna-se inconveniente

se a empresa exigir mais dos trabalhadores do que as boas funções que têm de empenhar do lado dos clientes. A solução ideal seria os colaboradores sentirem-se motivados a colaborar em nestes processos de crescimento e inovação para com a sua empresa – *gamification*.

Para uma melhor compreensão sobre estes paradigmas, a primeira parte desta secção corresponde à contextualização da dissertação, com uma explicação mais detalhada sobre o que é a atividade de *outsourcing*, e os problemas que origina. Posteriormente, encontra-se uma breve descrição sobre a empresa interessada pela presente dissertação.

Seguidamente, os principais objetivos a realizar com esta dissertação são apresentados seguindo a motivação para a concretização dos mesmos, pois torna-se mais simples perceber-la, após a compreensão do contexto.

Posteriormente, são apresentadas as metodologias de investigação utilizadas nas diferentes fases desta dissertação. E, por fim, apresenta-se a estrutura dessas diversas fases – estrutura da dissertação.

1.1. Contextualização

A presente dissertação estuda e aborda os problemas que nascem numa organização que tenha no seu sistema atividades de *outsourcing*. A primeira parte deste subcapítulo tem a finalidade de apresentar isso mesmo. Apesar dos problemas que serão descritos se encontrarem na maioria das organizações com serviços de consultoria, a presente dissertação é baseada e desenvolvida focando-se numa empresa com esses paradigmas. A empresa em causa já inclui métodos na sua área de inovação destinada aos colaboradores para que participem no crescimento da empresa. Porém, não evidenciam o sucesso desejado. Existe um subcapítulo que refere um pouco a empresa em questão, bem como estes métodos já utilizados para a inovação e crescimento da empresa.

1.1.1. Atividade de *Outsourcing* e Problemas

A competitividade existente entre as empresas está cada vez mais viva nas atividades de serviços que interagem na economia. Antigamente, os recursos necessários para produção de riquezas eram apenas o capital, os recursos naturais e a mão-de-obra (Figueiredo, Bremer, & Maldonado, 2008). No entanto, nos dias de hoje, o conhecimento torna-se um dos recursos principais (Nicolau, 2002), relevando assim a importância dos trabalhadores do conhecimento (Drucker, 1999).

Nas décadas 50 e 60, a expressão “*make or buy*” era utilizada para definir o princípio que a melhor opção tomada pelas organizações na obtenção de um resultado, era através da produção interna. Atualmente, o princípio é o oposto (Walker & Weber, 1984). O trabalho de projetar, decidir, administrar e produzir determinados produtos consoante um período de tempo é dispendioso. Além disso, existe sempre a preocupação com a competitividade e com as competências essenciais e habilidades que determinam a produção interna desses mesmos produtos. As organizações começaram a abandonar uma gestão vertical e a adaptar uma horizontal (Klein, 2008), surgindo a decisão de comprar produtos ou serviços que auxiliem esse trabalho. O termo empregado para a compra de produtos ou serviços externos, designa-se *outsourcing*.

A atividade de *outsourcing* define-se como uma transferência de responsabilidades e conhecimentos, gerando um fornecimento de serviço contínuo para um terceiro, gerido por um determinado acordo dos níveis estabelecidos (Gay & Essinger, 2000). No *outsourcing* de TI, essa transferência menciona pessoas, processos ou aplicações. As pessoas, normalmente, sofrem transferência de instalações para as instalações do terceiro.

Grandes organizações começaram a adotar o *outsourcing* como base de seus esforços e para fortalecer as operações. Por outro lado, pequenas organizações adotaram o *outsourcing* como um modo de acompanhamento de uma grande organização, gerindo os custos e o tempo despendido na aquisição e na administração direta de cada colaborador (Adelakun, 2012). Hoje em dia, já existem grandes consultoras em que o *outsourcing* é o foco principal.

Contudo, a movimentação de pessoas que os serviços de *outsourcing* obriga, deu origem a diversos problemas para as empresas prestadoras.

Os colaboradores iniciam a sua atividade profissional na área de consultoria e rapidamente começam a realizar as suas funções nas instalações de uma organização terceira. Esta situação origina diversos paradigmas. O facto de os colaboradores não terem o contacto suficiente com a empresa prestadora, ocasiona que os mesmos acabam por não obterem conhecimento sobre a empresa e, nem a própria empresa consegue crescer ou inovar com esses mesmos colaboradores.

Os colaboradores estão espalhados pelas diversas instalações das empresas clientes. Esta colocação afeta o conhecimento e o relacionamento que os colaboradores desfrutam entre si, ou seja, os colaboradores conhecem outros colegas, mas apenas os que estão presentes na mesma empresa cliente. No entanto, os restantes colegas da empresa prestadora são pouco conhecidos ou mesmo desconhecidos.

Reunindo estas situações, o principal paradigma que as empresas prestadoras enfrentam consiste na dificuldade de crescimento com a falta de cooperação dos próprios colaboradores. As organizações com consultoria, nomeadamente, a empresa interessada por o presente projeto, para além das atividades de *outsourcing*, possuem outras áreas de trabalho distintas. Todas as áreas carecem da colaboração maioritária para incitar o crescimento e a inovação. No entanto, a empresa prestadora não pode forçar mais tarefas aos colaboradores de *outsourcing*, logo necessita do interesse e motivação dos mesmos no auxílio do crescimento e inovação organizacional, surgindo então o tema *gamification*.

1.1.2. A Empresa

A XLM - *Innovation & Technology* é uma empresa que apresenta soluções inovadoras no desenvolvimento de *software* e serviços de consultoria na área da indústria de TI. Um número muito significativo do total dos seus colaboradores está no ramo da consultoria, ou seja, diariamente, executam as suas funções nas diferentes empresas parceiras/clientes. Contudo, os

problemas descritos no subcapítulo 1.1.1 “Atividade de *Outsourcing* e Problemas” encontram-se na empresa.

Atualmente, a empresa inclui uma gestão de IDI (Investigação, Desenvolvimento e Inovação) (Morais, 2010) e com isso, procura métodos para resolver as situações. Existem ferramentas acessíveis aos colaboradores para participarem no crescimento da empresa colocando ideias. No entanto, a realidade é que a plataforma responsável pela bolsa de ideias é pouco movimentada.

A empresa organiza também atividades durante o ano, desde formações, concursos de desenvolvimento de artigos, entretenimento, entre outros. Estas atividades têm sempre a finalidade de obter cooperação entre os colaboradores e, principalmente, aumentar a inovação e crescimento da organização. Algumas destas atividades têm recompensas monetárias finais ou outro tipo de prémio. No entanto, não existe nenhum sistema em que guarde um histórico desse reconhecimento do colaborador. Isto poderá ser uma falta de motivação da aderência a atividades internas. Ao longo do tempo, estas atividades vão tendo alguma adesão mas não a suficientemente desejada, de modo a resolver os problemas anteriormente descritos.

O desenvolvimento deste projeto e dissertação foca-se na melhoria dos paradigmas detalhados aplicando o conceito de *gamification*.

1.2. Objetivos

Tendo em conta os problemas referidos anteriormente no subcapítulo “1.1. Contextualização”, o objetivo principal da realização deste projeto é motivar e conseguir uma atividade mais assídua por parte dos colaboradores no processo de crescimento e inovação da empresa, através do aumento da motivação pessoal dos mesmos e sem acarretar grandes custos. Além disso, também serão abordadas as resoluções de outros problemas, como o aumento do conhecimento dos colaboradores sobre a empresa e a colaboração com sentido de grupo.

Considerando o método decidido para a resolução dos problemas referidos, ou seja, a partir do estímulo da motivação dos colaboradores através do conceito de *gamification*, pretende-se o

desenvolvimento de uma versão inicial de uma plataforma com metodologias de *gamification* para os colaboradores e para a organização. Os clientes e as empresas clientes da organização não interferem com o sistema. Os únicos utilizadores são os colaboradores da empresa e os administradores responsáveis pela inovação na organização.

De referir que o objetivo principal não é tornar a plataforma de *gamification* desenvolvida muito movimentada, mas sim aumentar a motivação dos colaboradores para a colaboração no crescimento da empresa, a partir da gestão de IDI já empregada dentro da organização.

1.3. Motivação

Ao longo dos anos, o número de empresas prestadoras de serviços de consultoria tem vindo a aumentar, bem como o número de empresas com responsabilidades elevadas, principalmente na área de serviços. Como tal, estas têm vindo a apostar na compra de serviços de consultoria para as suas equipas de trabalho, de modo a diminuir os cargos internos. Este método de gestão de equipas torna-se bastante vantajoso em muitos níveis. No entanto, como já referido anteriormente na secção “1.1. Contextualização”, este tipo de atividade traz alguns paradigmas para as empresas de consultoria.

Como colaboradora de *outsourcing* nas tarefas diárias, foi questionado à XLM a existência de alguns projetos possíveis para estudo e desenvolvimento nesta dissertação. A empresa propôs vários temas e enunciou alguns problemas. No estudo para a resolução dos mesmos, o uso de *gamification* foi mencionado, acompanhado por algumas restrições: a nível monetário na compra de sistemas que já pudessem existir; e na possibilidade de interação e melhoramento no sistema de inovação já existente. Deu-se preferência a este projeto para a resolução destes problemas, pelo facto de existir uma grande noção dos mesmos no dia-a-dia.

Além disso, o termo *Gamification*, desde 2010, tem vindo a crescer a nível empresarial. Segundo o gráfico de ciclo de tecnologias emergentes da Gartner Research (Pasold, 2012), o *gamification* é um conceito que ainda está a chegar ao topo das expectativas. Foi também nomeado para a

palavra mais falada em 2014 (publico.pt, 2014), o que torna um tema atual e interessante para estudar e aplicar os seus métodos.

O sistema resultante da presente dissertação terá uma aplicabilidade para a empresa e para os colaboradores, o que torna um desafio mais motivador e mais relevante a nível pessoal.

1.4. Metodologias e Planeamento de Trabalho

Este subcapítulo inicia-se com a apresentação das diferentes fases do desenvolvimento desta dissertação. Cada fase expõe a metodologia de investigação utilizada assim como uma breve descrição do plano de trabalho realizado ao longo do tempo.

A estratégia realizada define-se como conceção e criação e as diferentes fases são: Investigação; Análise de Requisitos e Conceção; Desenvolvimento; Testes e Avaliação; Aceitação.

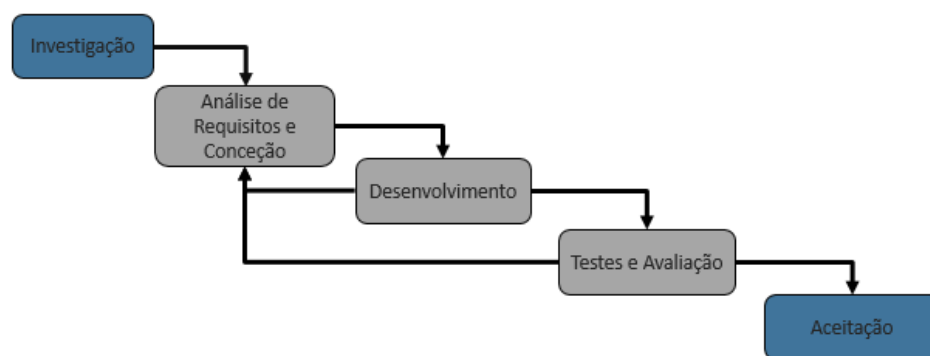


Fig. 1 - Fases da realização do projeto

Na fase inicial de investigação do projeto foi realizada uma análise de documentação e outras fontes de informação, como artigos, *sites* informativos de organizações que utilizam *gamification*. Igualmente, foram identificadas e estudadas diferentes plataformas existentes com o aplicabilidade deste conceito, de modo a verificar as melhores técnicas existentes que mais se

aproximam a dar soluções aos problemas em questão. Consoante essa análise foram argumentados, justificados e avaliados criticamente os diferentes resultados da investigação.

Os dados recolhidos com esta investigação foram consolidados, permitindo enumerar um conjunto de soluções eficazes para integrar na plataforma desenvolvida no projeto.

Em seguida, foram utilizadas novas técnicas de recolha de dados para a fase de análise de requisitos, nomeadamente, algumas entrevistas informais ao membro responsável pela inovação da XLM, que acompanhou todo o desenvolvimento do projeto. Esta fase, também definida por conceção do projeto, engloba a recolha e análise dos requisitos funcionais e não funcionais, a arquitetura e conceção do sistema e, por fim, a decisão das ferramentas e tecnologias utilizadas para o desenvolvimento do projeto.

Após uma análise de requisitos iniciais, as fases seguintes utilizaram uma metodologia ágil *scrum* (Vieira, 2014). Nesta metodologia, conceptualmente, existem ciclos temporais de desenvolvimento (*sprint*) para os diferentes requisitos detalhados na análise anterior. No entanto, esses mesmos requisitos podem sofrer alterações ao longo do desenvolvimento. Isso ocorreu, sendo necessário o plano de trabalhos passar por uma reanálise do(s) objetivo(s).

Voltando ao seguimento da análise de requisitos, posteriormente, todos os requisitos definidos foram colocados em diferentes *sprint's*, para a criação dos mesmos na fase de desenvolvimento.

Para cada requisito surgiu a fase de testes, sujeitos a avaliação, onde foram utilizadas novas técnicas de recolha e análise de dados. Foram realizados métodos de observação a alguns futuros utilizadores e pequenas entrevistas informais aos mesmos. Resultante disso, foram criadas conclusões com registos de ocorrências.

Depois da análise dos resultados dos testes, ou seja, depois dos requisitos serem submetidos a avaliação, os mesmos puderam ser aceites ou rejeitados. Da lista de requisitos associados ao *sprint*, os aceites foram integrados no sistema, ou seja, passaram à fase de aceitação. Os requisitos rejeitados regressaram à fase de análise de requisitos onde foram reavaliados e, posteriormente, abordados num outro *sprint*.

Esta metodologia ágil permitiu obter mais facilmente uma validação das opções tomadas ao longo do desenvolvimento do sistema. Além disso, ao longo do tempo, conseguiu-se obter uma noção se os resultados dos requisitos estavam de acordo com o que a empresa deseja. Este projeto é abrangente porque a sua aplicação, embora vocacionada para uma empresa em particular, pode facilmente ser reutilizado por outras empresas com problemas similares.

Resumidamente, o método teve por base uma estratégia de conceção e criação, com uso de uma triangulação de métodos para recolha de dados, e assentou numa abordagem qualitativa suportada por uma investigação no contexto de *gamification*.

1.4.1. Calendarização

De acordo com o referido nesta secção principal “1.4. Metodologias e Planeamento de Trabalho”, para as diferentes fases que estão associadas ao *scrum* não são calendarizadas de forma detalhada. Como se trata de um método ágil não se consegue obter definitivamente as datas finais. No entanto, a calendarização apresentada seguidamente foi definida de acordo com os objetivos principais a desenvolver neste projeto e dissertação. A tabela seguinte resume em alto nível a calendarização das diferentes fases.

Tabela 1 - Calendarização das fases

Fases		Data Provisória de Fim (Ano 2015)
Estudo e enquadramento (Investigação)		31 Janeiro
Análise de Requisitos e Conceção	Funcionais e não funcionais	14 Fevereiro
	Arquitetura	18 Fevereiro
	Ferramentas Utilizadas	21 Fevereiro
	Modelação/Conceção da Base de Dados	7 Março
Desenvolvimento/Testes	<i>BackOffice</i>	30 Abril
	Comunicação com o LagoonSecurity	15 Maio

	API - Integração com outros sistemas	15 Junho
	<i>FrontOffice</i>	30 Julho
Testes Finais		26 Setembro
Conclusão escrita da dissertação		16 Outubro

1.5. Estrutura da Dissertação

Após o presente capítulo – 1.Introdução -, a presente dissertação apresenta-se organizada separadamente por cinco capítulos.

No segundo capítulo é efetuado um estudo sobre a motivação dos indivíduos e como a mesma pode ser influenciada. Seguidamente, estuda-se o *gamification* e as suas características vocacionadas para proporcionar vantagens nas organizações. Posteriormente, são apresentados alguns sistemas existentes com implementações de *gamification*, bem como os resultados que proporcionaram. Por fim, apresenta-se um resumo das conclusões retiradas sobre o estudo efetuado comparando com os objetivos para o desenvolvimento do presente projeto.

No terceiro capítulo, apresenta-se um enquadramento semântico sobre os conceitos utilizados ao longo da dissertação. E, de seguida, são apresentados os sistemas externos existentes dentro da organização, que interagem com o sistema desenvolvido. Os requisitos funcionais e não funcionais adquiridos pelo estudo efetuado e pelo interesse da organização, também se encontram neste capítulo.

O quarto capítulo apresenta o planeamento e conceção para o desenvolvimento do sistema, nomeadamente, a (s) arquitetura (s) adotadas, as tecnologias utilizadas, os diagramas de casos de uso da plataforma bem como os diagramas de atividades mais relevantes no projeto, a conceção e a modelação da base de dados que dá suporte ao sistema também se encontra neste capítulo.

O quinto capítulo tem a finalidade de apresentar as principais funcionalidades implementadas no sistema, bem como algumas decisões refletidas sobre as mesmas. Ainda neste capítulo abordam-se as considerações de segurança tomadas e os testes realizados ao sistema.

Por fim, o sexto capítulo apresenta as conclusões retiradas após a realização do presente projeto.

2. Estado da Arte

Este capítulo apresenta alguns conceitos e fundamentos gerais relacionados com o *gamification* e qual a sua importância para o desenvolvimento do projeto. Além disso, foi realizado um pequeno estudo sobre a motivação estimulada nos indivíduos, que inicializa o presente capítulo.

Algumas soluções existentes no âmbito de *gamification* foram analisadas e abordadas neste capítulo. Estas soluções não se resumem apenas a plataformas semelhantes ao futuro sistema. Existem sistemas com mecanismos que podem ser utilizados no desenvolvimento do projeto e que não são, necessariamente, focados em produtividade nas organizações. Esta análise permitiu comparar o uso entre os melhores métodos nas ferramentas existentes e retirar determinadas conclusões.

Como o sistema proposto regista as atividades dos colaboradores, é realizada uma análise sobre o enquadramento legal do projeto, visando, essencialmente, os aspetos relacionados com os dados e informações pessoais.

2.1. Motivação nos Indivíduos

Desde há muitos anos que a motivação tem chamado a atenção de muitos pesquisadores (Vallerand, 2004) e, atualmente, já é considerada uma vantagem competitiva como uma estratégia em ambientes de trabalho (Tremblay, Blanchard, Taylor, Pelletier, & Villeneuve, 2009).

A palavra motivação surge de “*motivus*” em latim que indica uma causa que serve para movimentar, ou seja, estar motivado é ser movido a realizar uma ação (Werbach & Hunter, 2012).

O estado motivacional pode existir em três níveis de interação entre um indivíduo, uma tarefa e um ambiente (Vallerand, 2004):

- Situacional: refere-se à atividade num determinado momento, em que provoca expectativas de recompensas ou divertimento;
- Contextual: refere-se às atividades numa parte da vida do indivíduo. Geralmente promove atitudes de cooperação ou colaboração;
- Global: refere-se à personalidade de um indivíduo e não a nenhuma atividade em concreto. Segue orientação geral, porque o que importa é a forma de interagir num ambiente com a realização de uma atividade.

A motivação é um estado interno que impulsiona, dirige e mantém o comportamento (Lee, McInerney, Liem, & Ortiga, 2010). Se um indivíduo se sente desmotivado, não tem nenhum impulso ou inspiração para agir (Ryan & Deci, 2000).

Existem várias teorias herdadas sobre a motivação. Murray em 1967 classificou cinco teorias que originam motivação (Murray, 1967). No entanto, relativamente à motivação para o trabalho, algumas outras teorias importantes foram anunciadas, como “teorias de conteúdo” (*content theories*) e “teorias de processo” (*process theories*) conforme as tabelas 2 e 3. As teorias de conteúdo tratam de fatores que motivam os indivíduos enquanto as teorias de processo referem-se ao processo de motivação (Rocha, 2007).

Tabela 2 - Resumo das teorias de conteúdo e seus pressupostos

Autores	<i>Content Theories</i>
(Maslow, 1954)	Os indivíduos são motivados através de uma hierarquia de cinco tipos de necessidades sendo que as de nível superior só são satisfeitas quando as do nível imediatamente inferior se encontram satisfeitas.
(Herzberg, 1966)	Considera duas ordens de fatores associados à motivação, os fatores higiênicos e os fatores motivadores, sendo que os primeiros evitam a insatisfação enquanto os segundos conduzem à satisfação.
(Alderfer, 1972)	Defende três categorias de necessidades que conduzem à motivação: existência, relacionação e crescimento. Quando o indivíduo não consegue satisfazer um determinado tipo de necessidades tende a redobrar esforços no sentido de satisfazer a categoria de necessidades inferior, o que pode conduzir a frustração.
(McClelland, 1961)	Postula que a motivação ocorre pela satisfação de três tipos de necessidades: realização, afiliação e poder. Todos os indivíduos possuem esses três tipos de necessidades embora com predominância de uma em relação às restantes.
(Katz e Kahn, 1966)	Distinguem quatro tipos de fatores que influenciam o nível de motivação dos indivíduos: obediência à lei, satisfação instrumental, auto expansão e internacionalização, e valores, sendo que determinados incentivos conduzem a determinados tipos de comportamento.
(McGregor, 1960)	Apresenta duas abordagens antagônicas, as teorias x e y, acerca da administração: a teoria x considera que a natureza humana é

	indolente e não gosta de trabalhar; e a teoria y propõe que os seres humanos são bons e direcionados para o trabalho.
--	---

Fonte: (Galhanas, 2009)

Tabela 3 - Resumo das teorias de processos e seus pressupostos

Autores	Process Theories
(Vroom, 1964)	Sugere que a motivação depende da expectativa que o indivíduo tem em ser recompensado face ao trabalho desempenhado.
(Porter e Lawler, 1968)	Considera quatro variáveis que condicionam a motivação: o esforço, o desempenho, as recompensas e a satisfação.
(Adams, 1963)	Enfatiza a percepção pessoal do individuo sobre a razoabilidade ou justiça relativa na sua relação laboral com a organização.
(Locke, 1968)	Defende que a maior fonte de motivação reside na fixação de objetivos específicos e difíceis de atingir.

Fonte: (Galhanas, 2009)

Platão afirmou que existem motivos do corpo, como é o caso da fome, e os motivos da mente e alma, como a curiosidade e honra que motivam os seres (Reiss, 2012). Esta abordagem levou ao início da exploração da motivação intrínseca e extrínseca (Covington, 2000), visto que podem afetar o desempenho dos indivíduos.

Covington afirmou que a motivação intrínseca acontece quando um indivíduo realiza uma ação pela própria experiência, ou seja, as recompensas residem na satisfação em superar um desafio pessoal, ou na aprendizagem, ou no interesse. Contrariamente a isso, a motivação extrínseca ocorre quando um indivíduo se envolve numa atividade esperando outro tipo de recompensa, como financeira ou social (Covington, 2000). A motivação já não é diretamente ligada à ação, ou seja, realizar a atividade torna-se um meio para atingir um fim.

A motivação é um impulso que faz com que as pessoas ajam para atingir objetivos, tornando-a uma particularidade importante para a resolução dos problemas referidos nesta dissertação. A utilização de métodos de *gamification* tornou-se conveniente.

2.2. Gamification

Há muitos anos, desde que “os jogos” foram criados, no geral, começaram a preencher psicologicamente a necessidade de diversão aos jogadores. Por essa motivação que proporcionam aos indivíduos, atualmente os jogos são reconhecidos pelos métodos que oferecem a diferentes áreas de negócios como *marketing*, gestão de colaboradores, gestão com os clientes, inovação e sustentabilidade (Werbach & Hunter, 2012).

Existem milhões de pessoas em todo o mundo que passam milhões de horas a jogar. Os jogos começaram a ser desenvolvidos para diferentes tipos de dispositivos e formas (*online*, computador pessoal, consolas, dispositivos móveis, etc) e, para todas as idades e géneros (Werbach & Hunter, 2012). Os jogos representam uma parte significativa da vida contemporânea das pessoas (Foster, 2012).

Tudo isto implica que os indivíduos dediquem muito do seu tempo a jogar sem nenhum esforço e, apenas com a finalidade de obterem entretenimento (Nelson, 2012). Os jogos oferecem aos utilizadores um conjunto de experiências que ativam estímulos no cérebro semelhantes ao prazer (Werbach & Hunter, 2012). Por exemplo, quando um utilizador supera um desafio com sucesso, acontecem alterações neurológicas e fisiológicas nos jogadores que despertam uma sensação emocional positiva, podendo provocar efeitos na vida real (Foster, 2012).

Contrariamente a estes sentimentos, os jogos também podem provocar sensações negativas e de frustração. No entanto, é de referir que atualmente a maioria dos jogos criados impedem o utilizador de perder na totalidade, nem que ofereçam a possibilidade de repetir o objetivo ou uma outra forma de procurar soluções. Além do mais, existem jogos que se dedicam a oferecer sentimentos de relaxamento aos jogadores (Psychology & Professor, 2001). Os jogos que aplicam a teoria do *flow* (sentimentos de relaxamento em vez de frustração) (Mihaly, 2008) são exemplos desse tipo de jogos.

Tirando partido destes sentimentos, chegou-se à conclusão que a motivação humana pelos jogos, pode ser modelada para estimular as pessoas a fazerem certas atividades, tendo um princípio por de trás. Por exemplo, funcionários a aumentarem a produtividade, clientes a conhecerem

produtos, *marketing*, etc. A estes métodos de ir buscar as “coisas boas” dos jogos e aplicar na vida real dá-se o nome de *gamification*, que se tornou popular para realizar tarefas aparentemente indesejáveis mas importantes para as organizações (Passos, Medeiros, Neto, & Clua, 2011).

A maioria dos autores, define que *gamification* é um conceito de aplicações de mecânicas e de técnicas de *design* de jogos em contextos não-jogo (Deterding, 2011). No entanto, Werbach e Hunter afirmam que este conceito tem três dimensões: elementos de jogo (pequenos pedaços de um jogo); técnicas de *design* de jogos (combinações entre elementos); contextos de não-jogo que abrangem casos do mundo real (Werbach & Hunter, 2012). Esta última dimensão é a pretendida com o desenvolvimento da plataforma.

O conceito de *gamification* apenas começou a ser utilizado a partir de 2010 (Groh, 2012), embora já tenha sido abordado em 2002 por Nick Pelling (Marczewski, 2008).

O *gamification* não é usado apenas para o mundo das empresas (Groh, 2012). Muito pelo contrário, define-se como sendo a área que mais está em crescimento com a utilização dos métodos deste conceito. Estima-se que no final de 2015 cerca de 70% das empresas no mundo, terá uma plataforma interna com *gamification* (gartner.com, 2012) com a finalidade de resolução de determinados problemas, como publicidade, aumento de produtividade, entre outros.

2.2.1. Jogos

Segundo Deterding, a palavra “jogo” tem um significado caracterizado por determinadas regras, onde existe uma competição de indivíduos para alcançar determinados resultados específicos ou metas (Deterding, 2011).

Um jogo pode ser definido por um sistema com regras e estratégias, onde existem jogadores a participarem num ambiente artificial que alcançam um determinado resultado. Um jogo consegue criar efeitos positivos ou mudanças de comportamentos normais nos jogadores, e isso é um processo de *gamification* (Lucas, Gasparini, & Berkenbrock, 2013).

Os jogos podem ser aplicados nos mais diversos contextos, não tendo necessariamente o propósito de entretenimento. Inicialmente, os autores dos jogos “sérios” defendiam que o uso dos mesmos tinha a finalidade de aprendizagem (Deterding, 2011). Hoje em dia, os jogos são utilizados em diferentes contextos, tal como a aplicabilidade de *gamification*.

2.2.2. Jogadores

Todos os jogos necessitam de jogadores para poderem ser jogados. Estes indivíduos são uma parte importante a ter em atenção no desenvolvimento de um sistema com *gamification*, pois são eles os futuros utilizadores e torna-se conveniente ter um conhecimento generalizado sobre os seus comportamentos e os seus modos, visto que têm características e motivações diferentes.

Existem quatro tipos básicos de jogadores (Zichermann & Cunningham, 2011):

- *Achievers* (Conquistadores)
Jogador que procura realizar todos os desafios, cumprir 100% todas as missões e completar todos os prémios possíveis.
- *Explorers* (Exploradores)
Jogador que tem curiosidade em descobrir todo o jogo, explorando todos os recursos, áreas e segredos.
- *Socializers* (Sociais)
Jogador que gosta de jogar pela possibilidade de interagir com outros jogadores.
- *Killers* (Competitivo)
Jogador com perfil mais competitivo. Desejo de mostrar poder ou habilidade perante outros jogadores.

No desenvolvimento de um sistema é importante entender que tipo de jogadores serão os utilizadores, pois facilita a decisão de elementos a utilizar.

Os problemas referidos anteriormente, que a presente dissertação procura diminuir, abrangem estes quatro tipos de jogadores. Portanto, convém perceber bem quais os mecanismos e

elementos mais vantajosos para o sistema, tendo em conta as restrições mencionadas pela empresa.

2.2.3. Elementos e Mecanismos de Jogo

Os elementos baseiam-se em padrões regulares que se podem retirar dos jogos (Werbach, 2014). Pode-se comparar estes elementos como uma caixa de ferramentas que se consegue utilizar no desenvolvimento de uma solução com *gamification* (Alves, 2014). Estes elementos podem ser conjugados de diferentes maneiras adequando-se às soluções que se pretende obter. O que faz o sucesso dos elementos é como são usados de modo a provocar a experiência ao jogador.

Kevin Werbach realizou um modelo, em forma de pirâmide, para definir os elementos mais comuns retirados dos jogos. No entanto, ele mesmo refere que não existe nenhum elemento que seja mais eficaz do que outro, mas sim a finalidade para que o elemento é aplicado (Alves, 2014).

A pirâmide é constituída por 3 partes: na parte superior a dinâmica, no meio a mecânica e na parte inferior os componentes. O entorno da pirâmide é composto pela experiência resultante da relação de todos os elementos existentes no sistema de *gamification*. Grande parte desta experiência resulta da estética do sistema, pois sem esse componente o jogador não consegue jogar (Alves, 2014).

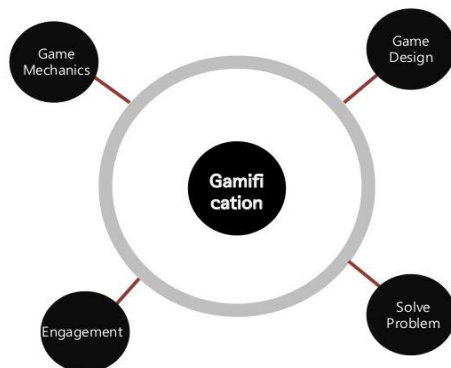


Fig. 2 - Modelo de Kevin Werbach, retirado de (Alves, 2014)

A dinâmica é constituída por elementos responsáveis de atribuir coerência e padrões regulares à experiência (Werbach, 2014). Estes elementos podem-se confundir com as regras, porém as regras são apenas definidas para cada jogo. A dinâmica tem o conjunto de elementos mais conceptuais, por exemplo, as emoções que provoca, os progressos visíveis no jogo, os relacionamentos entre os jogadores, a narrativa da história do jogo, entre outros.

A mecânica é constituída por os elementos que promovem a ação dos jogadores (Alves, 2014). Os *quests* (desafios) existentes fazem parte desta camada. Igualmente, os termos cooperação e a competição que promovem aos jogadores o desejo de jogar com outras pessoas na mesma atividade. O *feedback* no jogo é fundamental pois permite ao jogador compreender os objetivos supostos. Os recursos, as recompensas e as transações que permitem aos jogadores comprarem, ou venderem ou trocarem recursos com a finalidade de progredir no jogo. Todos estes elementos podem fazer parte da mecânica do jogo.

Num sistema com *gamification* é (são) a (s) mecânica (s) adotada(s) que podem tornar o jogo interessante, divertido e viciante para os utilizadores (Schonfeld, 2010). As mecânicas utilizadas refletem-se nas dinâmicas, ou seja, aquilo que mais interage com o sentimento e perceção do utilizador e, é isso que pode tornar o jogo ou o sistema motivante. Antes de se escolherem as mecânicas a utilizar deve-se ter em vista quais as dinâmicas que se pretendem obter.



Voltando à pirâmide de Kevin Werbach, a camada de componentes corresponde aos elementos que fazem mesmo funcionar e que circulam no jogo, como por exemplo, as pontuações, as conquistas, brindes, medalhas e coleções, avatares, *badges*, poderes, habilidades, líderes, equipas, entre outros.

Marc LeBlanc também acrescentou oito tipos de diversão retirados dos jogos muito relevantes. Criou uma tabela similar à pirâmide de Kevin Werbach, designando-a como MDA (mecânica, dinâmica e estética) (Hunicke, LeBlanc, & Zubek, 2004).

Concluindo, a união de todos os elementos permite oferecer uma experiência ao jogador, com o objetivo de resolver um problema ou uma atividade com motivação. O ideal é determinar os elementos mais adequados aos problemas.

Fig. 3 - Modelo de gamification retirado de (recyclebank.com, 2013)

Segundo Schonfeld existem mecanismos que podem levar à adoção da dinâmica desejada (Schonfeld, 2010). A tabela seguinte indica alguns desses mecanismos que são relevantes para a realização do sistema.

Tabela 4 - Alguns mecanismos de gamification

Mecanismos	Definição
Realização	Representação virtual ou física de o jogador ter realizado algo. Normalmente, são vistos como recompensas (pontos, subida de nível, medalha, etc.).
Dinâmica de compromisso	Determinação de um tempo pré-definido para o jogador realizar certas funções diárias ou não.
<i>Momentum Behavioral</i>	A tendência que os jogadores aparentam para continuarem a realizar as mesmas atividades ao longo do tempo.
<i>Blissful Productivity</i>	A ideia de que jogar um jogo faz mais feliz um indivíduo do que trabalhar. Aplicação ao trabalho de uma gratificação e um significado.
<i>Communal Discovery</i>	A dinâmica do jogo, em que toda a comunidade ou uma equipa se reúne para resolver um enigma, um problema ou um desafio.
Contingência	Determinação de um problema que o jogador deve superar para obter uma recompensa ou uma passagem de nível.
Contagem regressiva	Limitação dos jogadores com uma certa quantidade de tempo para executar alguma atividade.
<i>Cross Situational Leader-boards</i>	Aplicação de mecanismos de classificação (<i>rankings</i>).
Jogos indeterminados	Não aplicar um fim explícito a um jogo. Não colocar a existência de limitação de recompensas.
Inveja	O desejo do jogador possuir também o que os outros têm.
Significado épico	Fazer acreditar que existe uma recompensa muito grande para obter uma motivação elevada aos jogadores.
<i>Fixed Ratio</i>	Esquema de recompensas após uma sequência de atividades.

<i>Reward Schedule</i>	
<i>Free Lunch</i>	A dinâmica em que o jogador recebe recompensas grátis por alguém ter realizado uma ação. Por exemplo, inserir um comentário numa ideia.
<i>Fun Once, Fun Always</i>	O conceito de que uma ação é agradável para repetir o tempo todo. Geralmente isso relaciona-se com ações simples.
Lotaria	A dinâmica do jogo em que o vencedor é determinado unicamente por acaso. Pode surgir o sentimento de sorte para um jogador, mas de revolta para os restantes.
<i>Modifiers</i> (Modificadores)	Criação de itens que, quando usados afetam outras ações. Geralmente modificadores são ganhos depois de ter completado uma série de desafios ou funções essenciais.
Orgulho	Sentimento de orgulho por o jogador ser recompensado.
Privacidade	O conceito de que determinada informação é privada, não para distribuição pública. Isso pode ser um desmotivador ou motivador se as atividades forem públicas.
Progressão dinâmica	Dinâmica em que o sucesso é exibido e medido através do processo de conclusão das tarefas detalhadas.
Resposta	Ações que são esperadas dos jogadores sem ser necessário existir informação para realizar essas ações.
Horários de Recompensa	Existência de calendário para entrega de recompensas consoante atividades que os jogadores tenham feito até ao momento.
<i>Physical Goods</i>	Entrega de prémios a qualquer jogador que apresente bons resultados numa determinada característica.
<i>Shell Game</i>	Iludir o jogador com a possibilidade de escolha com uma situação que na verdade já é desejado pela organização.

Fábrica social	A ideia de que os jogadores aumentam a confiança entre eles, após terem realizado uma determinada atividade juntos.
<i>Status</i>	Determinar um nível a cada jogador para existir motivação de progressão.
Mecânicas Virais de Jogos	Contruir parte de jogos que requerem vários jogadores.

2.2.4. Áreas Aplicacionais

Como já referido, só a partir de 2010 é que começou a existir uma forte atração por este tema nas mais diversas áreas. No entanto, conceitos de *gamification* já se utilizavam no dia-a-dia pela população. Por exemplo, desde há muitos anos atrás, a forma de educar as crianças passava pelo método de “manipular” com desejos das crianças em troca de bons comportamentos e/ou ações. Outro caso que exemplifica são os programas de pontos em cartões de crédito e pontos em cartão para troca de equipamentos, também já comuns há alguns anos.

A aplicação de elementos de jogos em diversos contextos visando um objetivo de negócio através da diversão tem aumentado nas diferentes áreas (learningtech.com, 2013). Entre as quais, alguns exemplos:

- Integração entre pessoas e grupos de interesse comum, nomeadamente, através de redes sociais;
- Educação, alunos se tornam jogadores e interagem desenvolvendo e aprendendo conteúdo;
- *Design*, elementos de jogos vão alterando a forma de *design* e estética, trazendo um ambiente de jogo para a vida real;
- Cidadania, jogos são criados para incentivar os valores cívicos e sociais;
- Desporto, plataformas com *rankings* pessoais para estimular a prática de desporto;
- *Marketing*, jogos são criados para publicitar novos produtos;

- Empresas.

No subcapítulo “2.3. Soluções existentes” encontra-se alguns casos de sucesso inspiradores.

2.2.5. *Gamification* e as Organizações

Nos finais de 2010, existiam já algumas soluções com aplicações de mecânicas de jogos e com características típicas de aplicações sociais em ambientes empresariais, principalmente em páginas *Web* de empresas.

Outra área de aplicação que se tornou importante para as organizações, relaciona-se com a motivação dos funcionários. Neste contexto, a utilização de mecânicas de jogos para o engajamento à colaboração, trouxe um impacto ainda significativo para as organizações cujos colaboradores trabalham distantes, nomeadamente, colaboradores de *outsourcing*.

O *Gamification* tem-se tornado uma parte essencial de qualquer estratégia de negócio como forma de motivar as pessoas digitalmente e ultrapassar as barreiras de escala, tempo, distância, conexão e custo (Jonathan, 2014). Uma pesquisa de Aberdeen Group revela que organizações que implementaram conceitos de *gamification* melhoraram 48% no envolvimento dos seus colaboradores e aumentaram 36% os seus negócios (Paharia, 2015).

Uma pesquisa realizada pela *NICE Systems* mostrou que as técnicas de *gamification* podem reduzir o tempo de realização de uma tarefa por um funcionário até 90%. Foram entrevistados diferentes colaboradores de empresas que adotaram práticas de *gamification*. Quase 80% disseram que se sentiam mais motivados para trabalhar e, 88% afirmaram que entendem que a atuação deles para com a empresa tem impacto direto (Bazolli, 2013).

Um estudo publicado na *Harvard Business Review* constata também que colaboradores motivados são 31% mais produtivos, geram 37% mais vendas e são 300% mais criativos do que colaboradores desmotivados. Além disso, ainda é revelado que 70% dos líderes de empresas ou de equipas de trabalho têm noção dos benefícios existentes pelo desempenho individual de cada colaborador e pela motivação que o proporciona (Araújo, 2014).

2.3. Soluções Existentes

A aplicação de métodos de *gamification* na resolução de problemas tem proporcionado resultados positivos nas mais diversas áreas, como já referido. Existem aplicações que foram desenvolvidas com determinados objetivos que tiveram grande sucesso. Algumas das quais se nomeiam como:

- O **Foldit** foi em 2010 um jogo *online* de quebra-cabeças sobre o dobramento de proteínas que permitia aos indivíduos descobrirem várias estruturas de proteínas dentro de determinados critérios. O número de jogadores chegou rapidamente aos 240 mil e, foi encontrada uma solução para a estrutura da M-PMV em 10 dias, criando um grande avanço no campo de pesquisa AIDS (zmescience.com, sem data).
- O **NikePlus** monitoriza as corridas dos utilizadores e propõe novas conquistas e metas para novos *records*. Oferece aos utilizadores estatísticas de todo o histórico aumentando a motivação que os utilizadores precisam para irem mais longe como nunca (emmawynne09, 2013).
- O **Pain Squad** foi um jogo desenvolvido em 2013 para ajudar as crianças com cancro e os médicos que as acompanhavam. Este jogo permitia às crianças realizarem várias missões policiais expondo as dores e os sentimentos que tinham durante o dia. Teve imenso sucesso nos hospitais de crianças com cancro pois proporcionava divertimento às crianças e ao mesmo tempo facilitava o trabalho dos médicos (Cundari, 2013).
- O **RecycleBank** foi um mecanismo de *gamification* abordado em 2013 para incentivar os indivíduos a reciclar mais de modo a reduzir o aterro de lixo nas cidades. Eram lançados questionários de sustentabilidade e promessas, cujas respostas eram recompensadas com pontos que poderiam ser trocados por bens reais (recyclebank.com, 2013).
- O **Voki** foi desenvolvido para permitir aos seus utilizadores expressarem-se através de avatares. Através desta ideia, foi desenvolvido um sistema de gestão escolar onde o docente podia inserir as turmas e lançar desafios. Cada aluno tinha a possibilidade de usar um avatar para responder às atividades propostas (Simões, 2012).

Existem muitas outras aplicações com uso de métodos de *gamification* nas diferentes áreas com sucesso.

A nível empresarial o uso de *gamification* também tem apresentado resultados muito positivos, desde plataformas e serviços destinados aos clientes das empresas como plataformas para o aumento da produtividade e aprendizagem dos colaboradores.

O sistema da presente dissertação enquadra-se mais nas plataformas destinadas aos colaboradores. Os casos de estudo explicados neste estado da arte apresentam uma descrição mais focada a este tipo de plataformas. No entanto, existem elementos importantes e interessantes encontrados em sistemas também destinados a clientes, cujos mais relevantes são abordados seguidamente.

2.3.1. Sistemas vocacionados para clientes de uma organização

No geral, os sistemas com *gamification* utilizados para “atração” de clientes são integrações em páginas *Web* ou em aplicações móveis de organizações (Schonfeld, 2010). No entanto, apresentam características e métodos pertinentes para o sistema do presente projeto.

Os exemplos a seguir são sistemas/serviços desenvolvidos que podem ser utilizados por outras organizações.

- **Gigya**

O Gigya oferece serviços a organizações que pretendem integrar um sistema de *gamification* dentro da sua página *Web* ou em aplicações móveis nativas. O principal objetivo foca-se no aumento de clientes das organizações que usam este sistema/serviço. Oferece uma forte escalabilidade para suportar um grande volume de operações em simultâneo, lidar com milhões de perfis de clientes e autenticar milhares de clientes por segundo (gigya.com, 2014).

O serviço é construído em cima de uma API REST, fornecendo acesso programático a funcionalidades da plataforma.

Os elementos e métodos que este sistema/serviço usa no âmbito de *gamification* focam-se:

- Na existência de um perfil para cada cliente/utilizador com um último histórico de progressão no jogo, a pontuação e as recompensas que ganhou;
- As atividades resumem-se em colocar comentários e “likes” na página *Web* ou na aplicação móvel;
- As recompensas pelas atividades são pontos e pequenos *badges*.

▪ **BigDoor**

Igualmente ao caso anterior, o BigDoor oferece serviços de *gamification* a organizações para integrar em sistemas da empresa, nomeadamente, na *Web*. Os mecanismos utilizados são semelhantes aos do Gigya, no entanto oferece um serviço interessante à organização para acompanhar o sucesso do programa, através de gráficos (bigdoor.com, 2014).

▪ **LevelUp**

O LevelUp oferece serviços a organizações vendedoras ou de restauração, para publicitarem as suas campanhas em aplicações móveis integrando com as redes sociais (thelevelup.com, 2014).

O LevelUp foca-se nos pagamentos das campanhas, portanto quantas mais visitas, partilhas ou compras os clientes fizerem, mais recompensas desbloqueiam. Os clientes das empresas têm uma pequena janela com as recompensas desbloqueadas. No aniversário os clientes são presenteados e, quando estão muito tempo sem visitar as campanhas é enviado ao cliente um lembrete.

Para os clientes organizacionais, o LevelUp oferece um serviço de acompanhamento das campanhas.

Perante os problemas a resolver com a presente dissertação, este tipo de sistemas é incompleto. Várias falhas são identificadas neste tipo de sistema, às quais se destacam:

- Estes sistemas focam-se apenas numa única aplicação, onde o *gamification* é aplicado para aumentar o uso das funcionalidades dessa aplicação;
- Outra falha nestes sistemas é a limitação na gestão de atividades realizadas pelos colaboradores. Os sistemas apenas permitem gerir as atividades que se encontram incorporadas no sistema com *gamification*. Esta situação não é o desejado para o presente projeto;
- Outra falha nestes sistemas encontra-se no objetivo principal dos sistemas se focarem nas ações que os jogadores/utilizadores realizam numa única plataforma *online*. Contudo, como referido no subcapítulo 1.2 “Objetivos”, as ações principais pretendidas para a presente dissertação apresentam-se como o oposto.

2.3.2. Sistemas vocacionados para colaboração de uma organização

No geral, os sistemas com implementação de *gamification* utilizados com destino a colaboradores têm sempre a intenção de a empresa aumentar a produtividade dos mesmos ou a aprendizagem interna. Os dois primeiros exemplos a seguir apresentam-se como sistemas/serviços que podem ser disponibilizados para outras organizações, direcionados para os seus colaboradores. No entanto, as mesmas organizações também podem aplicar esses serviços em sistemas vocacionados para os seus clientes.

- **Bunchball**

O Bunchball Nitro oferece serviços a organizações que desejam integrar um sistema destinado aos seus clientes ou apenas destinado aos seus colaboradores (bunchball.com, 2014).

A nível de aplicações corporativas, este serviço oferece bons métodos de equipa para organizações de vendas.

Das metodologias implementadas de *gamification* que apresenta, destacam-se as seguintes:

- Oferece aos colaboradores uma experiência baseada em recompensas;

- Existem muitos desafios constantemente lançados pela empresa;
- Existe uma avaliação das resoluções de desafios;
- Existe uma página com um *feed* notícias das atividades dos colaboradores;
- Monitorização com redes sociais. Esta característica é interessante pois consegue verificar determinadas ações dos colaboradores em redes sociais e colocar ou agradecer na plataforma de *gamification*.

Este sistema além de apresentar boas metodologias e ter um grande número de clientes, não é aceitável para todos os problemas que se desejam resolver com o presente projeto. Por exemplo, é importante o sistema permitir a integração de pontuações com outras atividades externas da plataforma de *gamification*, nomeadamente, ações realizada noutras plataformas internas ou atividades extra para a organização.

- **Badgeville:**

O Badgeville, igualmente ao Bunchball Nitro, oferece serviços a organizações que desejam aumentar a fidelidade dos seus clientes ou melhorar o desempenho dos seus funcionários, com aplicações corporativas, redes sociais e em comunidades (badgeville.com, 2013).

As soluções que proporciona têm como finalidade obter mais visitas de utilizadores, mais documentos criados na plataforma e mais *downloads*.

Dentro das características encontradas pelas diferentes soluções com *gamification* que já ofereceu, destacam-se:

- A motivação que entusiasma os utilizadores foca-se na reputação e no reconhecimento individual;
- Os perfis dos utilizadores contêm os dados pessoais, a pontuação geral, os *badges* recebidos, uma pequena lista de médias de atividades que o utilizador realizou e, por fim, um gráfico com percentagens consoante as respostas a questões, comentários e votos que o utilizador efetuou;

- As recompensas podem ser personalizadas de acordo com os utilizadores, podendo os mesmos obterem recompensas únicas;
- A utilização da metodologia de missões para o progresso no jogo através de níveis;
- Como o Bunchball Nitro, existe uma monitorização com as redes sociais.
- A implementação de múltiplos sistemas de pontos. O total de pontuações que os utilizadores vão angariando podem ser trocados por recompensas tangíveis;
- A utilização de tabela de líderes contextuais. Os casos implementados com esta mecânica destacam-se mais para a área dos clientes. Por exemplo, a criação de uma tabela de líderes com pontos ou recompensas ganhas em torno de um determinado produto, assunto ou conteúdo;
- Notificações positivas em tempo real quando os utilizadores são recompensados.

Em 2013, Kendall-Jackson, propriedade Californiana produtora de vinhos, começou a usar uma aplicação para facilitar a escolha de vinhos para os seus clientes. Mais tarde, Kendall-Jackson acrescentou à aplicação uma solução de Badgeville. Após esta integração, os resultados das mudanças foram extremamente positivos. Em um ano, houve um aumento de 65% da utilização da aplicação e um aumento de 38% de *downloads*: 8200 novos utilizadores; mais de 27000 ações de utilizadores registados; mais de 1500 ações de partilha nas redes sociais (Hanifin, 2014).

Claramente, esta plataforma apresenta características interessantes. No entanto, os resultados finais são individualistas. Não existe grande motivação para trabalhos em equipa.

▪ SuMo

A Glenroy é uma empresa que fabrica embalagens e películas aderentes. Para o aumento do número de vendas realizadas pelos seus colaboradores, a empresa apostou numa aplicação inovadora - *SuMo* de *CloudApps* (empresa que criou a aplicação) -, com reforços positivos de encorajamento e mecânicas de *gamification* para motivar e ensinar melhores comportamentos a ter durante as vendas dos seus produtos (yareah.com, 2015).

A aplicação proporciona aos utilizadores agirem como treinadores de vendas constantemente, incentivando as ações corretas a ter nos momentos certos. A pontuação da sua experiência é incrementado durante o jogo, onde existe um *ranking* dos colaboradores.

Desde que o SuMo foi aplicado na organização, a Glenroy obteve um aumento de 295% na atividade das vendas; 275% na qualidade das qualificações das vendas; e 200% na qualidade das propostas (yareah.com, 2015).

▪ DirecTV

A DirecTV é uma empresa líder em entretenimento de televisão nos Estados Unidos e na América Latina. Esta organização teve o objetivo de aumentar o conhecimento interno e a experiência. (Greengard, 2014).

A partir de 2012, experimentaram a partilha de vídeos através de um canal interno, com a finalidade de promover a aprendizagem e a inovação. Posteriormente, foram desenvolvidos vários *quizzes* para os colaboradores interagirem tendo pontos e emblemas de recompensa. Com o número de adesão ao jogo, em 2013, a DirecTV criou uma plataforma de *gamification* para facilitar a aprendizagem, a transferência de conhecimentos e o desenvolvimento de negócios. Os colaboradores dão ideias, comentam, partilham informações, oferecem o seu ponto de vista e, em troca, recebem pontos e recompensas (Greengard, 2014).

Depois que o sistema de *gamification* foi implementado, o volume de ideias cresceu exponencialmente. Cerca de 97% dos colaboradores participam no processo de aprendizagem e inovação, proporcionando um crescimento de aproximadamente 300% comparando com a participação que tinham anteriormente (Greengard, 2014).

- **PeerSpace**

O PeerSpace é um ambiente de aprendizagem *online*, que oferece uma rede de apoio com o objetivo de encorajar os estudantes de ciência de computação numa fase inicial. Este sistema não está desenhado para colaboradores, no entanto abrange uma área de alunos onde se pode considerar que existem objetivos são semelhantes: aprendizagem.

Esta rede lança várias atividades, como: revisões de código-fonte de outros alunos; exercícios de grupos; *quizzes* individuais e fóruns de discussão de ideias. Todas as atividades têm o objetivo de incentivar a colaboração entre os alunos.

No entanto, inicialmente, o uso de PeerSpace não era tão utilizado como tinha sido programado. Surgiu, então, a ideia de aplicar técnicas de *gamification* na plataforma, tais como: a implementação de pequenos jogos para serem jogados entre os alunos; a atribuição de pontos nas participações; a inserção de níveis e um *ranking* (Li, Dong, Untch, & Chasteen, 2013).

Na avaliação desta proposta de *gamification*, foram realizados testes com uma equipa sem as técnicas descritas e outra equipa com o uso das técnicas. Os resultados foram triplicados: o ambiente normal obteve um total de 91 inserções no sistema enquanto o ambiente com *gamification* obteve um total de 265 inserções (Li, Dong, Untch, & Chasteen, 2013).

2.3.3. Em Portugal

Em Portugal, o termo *gamification* ou, em português, gamificação, ainda está em crescimento. No entanto, já se vêem algumas empresas a apostarem neste ramo, como é o caso do projeto apresentado na presente dissertação.

- **skilleo**

O skilleo é uma *startup* portuguesa que construiu uma plataforma muito interessante. Atualmente, o skilleo oferece serviços para solucionar problemas nas organizações a nível de

aprendizagem interno. Porém, não é a característica mais pertinente. O skilleo é também uma plataforma que apresenta objetivos não destinados a nenhum tipo de organização, mas ao interesse dos indivíduos que gostam de programar.

Esta plataforma apresenta metodologias e características relevantes que chamam a atenção para o desenvolvimento deste projeto, apesar de não se relacionar com os objetivos finais. Das características que contém focam-se as seguintes:

- Apresenta desafios de codificação de alto nível aos indivíduos registados;
- O utilizador pode colecionar *badges* de acordo com os desafios que alcança. Contém um perfil com semelhanças aos perfis da rede social LinkedIn, relativamente ao ramo da profissão e ao *ranking* das tecnologias que tem conhecimento;
- Permite o registo de empresas na plataforma e podem gerar desafios e anúncios de ofertas de trabalho;
- Existem cursos disponíveis para os utilizadores.

O Skilleo apresenta-se como uma plataforma que qualquer indivíduo pode usar, de modo a apresentar os seus conhecimentos a nível profissional. Estes conhecimentos tornam-se uma mais-valia na ajuda ao recrutamento para as empresas (skilleo.me,2014).

▪ Mercado de Ideias

A Portugal Telecom é uma organização que tem uma história de sucesso com a prática de métodos de *gamification* e inovação. Esta organização gere três tipos de inovação: inovação exploratória (projetos estruturais, de longo prazo e alto risco); inovação planeada (desenvolvimento de negócios a médio prazo) e inovação incremental (de baixo risco e de curto prazo). A Portugal Telecom reconhece o potencial de cada funcionário como sendo uma parte da sua equipa de inovação, especialmente na inovação incremental.

O mercado de ideias é uma plataforma *online* que a organização incrementou, onde se aplicam métodos de *gamification*. Nesta plataforma, cada funcionário tem uma bolsa virtual

de ideias e entre 2 a 3 meses são lançados desafios pelo CEO. Cada utilizador pode apresentar as suas ideias novas ou ideias para a resolução dos desafios e pode realizar comentários sobre as ideias existentes no mercado. Com estas atividades são lhes oferecidas “moedas” virtuais e, posteriormente, esses utilizadores podem trocar essas moedas por prémios.

Desde que esta plataforma foi lançada em 2009, já foram inseridas mais de 7000 ideias, cujas 300 foram implementadas em diversas áreas, incluindo melhorias na experiência ao cliente, produtos e serviços, cultura organizacional e redução de custos.

A Portugal Telecom tem lucrado com alguns resultados destas ideias, por exemplo: cerca de 25.000.000€ de lucro anualmente a partir de renovação e reutilização de telefones e *Set Top Boxes*; mais de 20% de redução no uso de papel; 125.000€ em iniciativas de iluminação e de uso de computador; entre outros (Kaufman, 2013).

Antes do mercado de ideias, apenas 58% dos funcionários afirmaram que se sentiam alinhados com a estratégia da empresa. Após 4 anos do uso desta plataforma, a percentagem subiu para 84% (Kaufman, 2013).

Como referido anteriormente, a empresa XLM, já tem incorporado na sua gestão um sistema de ideias – o LagoonIdeas - com os objetivos semelhantes ao Mercado de Ideias referido. Esta ferramenta apresenta-se mais à frente na presente dissertação.

Em Portugal, ao longo do tempo, têm existido alguns cursos/formações para aprender a utilizar estratégias de *gamification* no dia-a-dia, principalmente na área de educação e turismo (gamificationportugal.com, 2015).

Acredita-se que já existem, em Portugal, alguns sistemas com métodos de *gamification* em diversas áreas, porém não são conhecidos. No entanto, a aplicabilidade deste termo nas empresas tem como finalidade o sucesso empresarial interno, tornando o uso desta estratégia vantajoso se não for público.

2.4. Enquadramento Legal

O sistema desenvolvido tem uma aplicabilidade interna. Este facto torna o sistema mais “livre” no que se refere às leis atuais, nomeadamente, à lei n.º 67/98 de proteção de dados pessoais (cnpd.pt, 1998).

Todas as informações que circulam no sistema são apenas informações básicas e internas que podem ser de conhecimento de todos os colaboradores, nomeadamente, o nome, uma fotografia, *e-mail* interno, morada e contacto telefónico. No entanto, todos os colaboradores que não desejarem que alguma desta informação, embora disponível a qualquer colaborador interno, não seja pública internamente, podem comunicar com a administração ou responsável pelo sistema e estes dados não serão visíveis no sistema para os restantes colaboradores.

Por outro lado, como refere o próximo capítulo “3. Análise de Requisitos”, a plataforma disponibiliza uma funcionalidade que permite aos utilizadores partilharem informação de conquistas de *badges* numa rede social. Como estas informações possíveis de partilha correspondem apenas ao colaborador que as partilhará, se o assim desejar, implica que essa responsabilidade parta desse mesmo colaborador. Essas informações disponíveis para partilha externa serão dados credíveis inseridos pela administração ou responsável pela plataforma.

2.5. Conclusões

Pela investigação realizada, verifica-se que não é apenas na área empresarial que existem grandes projetos inovadores, mas sim nas mais diversas áreas, em que o resultado das junções de certas características podem surgir grandes soluções de *gamification*.

A descrição e análise realizada sobre sistemas existentes com soluções que mais se aproximam da resolução dos problemas descritos para o desenvolvimento deste projeto, mostra que nenhum é totalmente completo para a resolução dos três principais problemas (apoio ao crescimento da empresa; colaboradores não conhecem a empresa; colaboradores não se conhecem entre si).

As soluções existentes possuem sempre o objetivo de motivarem os utilizadores a gastarem muito tempo com a plataforma. O que se espera do sistema desenvolvido com o uso de *gamification* não é nem de perto esse o objetivo. Espera-se um sistema:

- Que motive os colaboradores a utilizarem com mais frequência as ferramentas internas já existentes na empresa;
- Que motive a colaboração com a empresa em atividades e contribuições extra para o seu crescimento nas diferentes áreas;
- Que permita aumentar o conhecimento sobre a empresa perante os colaboradores e aumentar o envolvimento entre os mesmos.

Perante as análises retiradas conclui-se que é possível inovar em determinadas características para o sistema do presente projeto. Os serviços oferecidos para plataformas de *gamification* apresentam algumas limitações, nomeadamente:

- A possibilidade de integrar as atividade de vários sistemas existentes numa organização;
- Resumem-se a um sistema de ideias ou apenas em lançamento de desafios para os colaboradores;
- Existe pouca motivação para a colaboração em equipa, ou seja, os colaboradores não interagem muito entre si. Relativamente a sistemas vocacionados para colaboradores, não se encontra nenhum método aplicado com incentivo à realização de atividades em conjunto;
- A maioria dos sistemas são limitados às atividades do sistema, ou seja, não permitem inserção de novos reconhecimentos para atividades que os colaboradores executem fora da empresa, mas que são ações de valor. Por exemplo, um colaborador ser formador esporadicamente;
- Não se encontram plataformas que apostem para além de uma “pontuação” total, como uma pontuação por categorias, de modo a motivar os utilizadores a participarem nas diferentes áreas de interesse da empresa;
- A ideia de “currículo interno na empresa” não está muito abordada;

- O objetivo do termo “jogo” também não se encontra bem definido a nível temporal. Os sistemas estudados dão a ideia que o jogo em si é “infinito”, diminuindo o interesse para novos utilizadores.

Para o desenvolvimento do presente projeto, pretende-se implementar algumas características implementadas já noutros sistemas. No entanto, além disso, procura-se também desenvolver soluções que incluem estas particularidades identificadas como limitadas, tornando o sistema inovador e com aplicabilidade.

3. Análise de Requisitos

De acordo com as investigações efetuadas e com os requisitos solicitados pela organização XLM, verificou-se a existência de determinadas necessidades e objetivos que, dificilmente, se conseguem alcançar com apenas a utilização de uma aplicação existente no mercado. O desenvolvimento de uma plataforma interna capaz de responder a todas as necessidades, com aplicabilidade de mecânicas e conceitos de *gamification*, tornou-se essencial para o presente projeto.

A partir das mecânicas estudadas e da restante abordagem de *gamification*, para a realização deste projeto são utilizados diversos conceitos próprios. O primeiro subcapítulo do presente capítulo refere e define esses conceitos, enquadrando-os com o projeto.

Seguidamente, o segundo subcapítulo identifica os sistemas externos existentes na empresa, cuja interação entre os mesmos e a plataforma desenvolvida é um requisito.

Por fim, o terceiro e o quarto subcapítulos apresentam os requisitos funcionais e não funcionais da plataforma, implementados nas diferentes fases do *scrum*.

3.1. Enquadramento

Antes de se prosseguir para uma identificação detalhada dos requisitos do projeto, é importante compreender todas as necessidades gerais requeridas pela empresa. No início do projeto, foram solicitados diversos interesses funcionais por parte da empresa. Porém, ao longo do tempo, novos requisitos foram surgindo nas diferentes fases do *scrum*.

O presente projeto resume-se no desenvolvimento de um sistema informático capaz de responder a uma série de necessidades que a empresa requer com aplicabilidade de *gamification*. Atendendo a essas necessidades gerais, o sistema deve ser composto por quatro componentes principais, nomeadamente:

- A criação de uma API que comunique com as ferramentas já existentes na empresa;
- O desenvolvimento de um *backOffice* que permita efetuar os processos de gestão internos relacionados com o *gamification* na empresa;
- O desenvolvimento de um *frontOffice* onde os colaboradores podem consultar o seu estado atual de *gamification* dentro da empresa;
- A comunicação e a utilização de diversas funcionalidades do LagoonSecurity (ferramenta interna da empresa com capacidades de gestão de acessos e permissões de utilizadores).

Abrangendo estes quatro componentes, existem requisitos com conceitos aplicados que, embora possam ter uma definição diferente, são utilizados para representarem determinadas características. De seguida, apresenta-se uma lista desses conceitos, facilitando a compreensão do âmbito do projeto.

- **Atividade**

Como já referido na introdução, no subcapítulo “1.1.2. A Empresa”, a XLM organiza diversas atividades destinadas aos seus colaboradores, para incentivar a colaboração e a inovação dentro da organização. A competição na realização de artigos, formações, criações de ideias, participação num desafio, entre outros, são exemplos de atividades.

Resumindo, uma atividade é uma tarefa realizada por um colaborador.

▪ Categoria

A partir das atividades existentes, torna-se pertinente agrupá-las. A possibilidade de as atividades apresentarem-se em grupos no sistema, permite o aumento do incentivo na realização de diferentes tipos atividades. Estes agrupamentos definem-se como categorias na presente dissertação. Uma categoria pode ser: inovação, entretenimento, educação, etc.

▪ Evento

Um evento acontece quando um ou vários colaboradores realizam uma atividade, numa determinada data.

▪ Pontuação

Por cada evento realizado, é atribuída uma pontuação administrativa para cada colaborador que participou nesse evento.

▪ Jogo Temporal

Na vida real nenhum jogo é infinito. Um jogo é o limite temporal definido para os colaboradores realizarem eventos e receberem as devidas recompensas. Apesar de um mecanismo de *gamification* se apresentar como a ilimitação de jogos, foi decidido contrariar este mecanismo. O objetivo principal de existir um início e um fim temporal baseia-se no aumento do incentivo aos novos colaboradores que vão entrando na empresa concretizarem eventos, dando mais hipóteses em concorrer com os antigos colaboradores. Se não existissem vários jogos, seria mais complicado os novos indivíduos na organização terem uma forte reputação, e conseqüentemente, seria mais desmotivante para a realização de novos eventos. Cada jogo é definido com um nome, acompanhado com uma descrição, para contextualizar os jogadores (colaboradores) do âmbito do jogo.

- **Níveis**

Os níveis servem para gerir as pontuações obtidas por cada colaborador. Existem diferentes níveis que correspondem à pontuação geral em cada jogo. Igualmente, existem diversos níveis para cada categoria de cada jogo. Cada colaborador encontra-se num determinado nível geral e num outro nível em cada categoria do corrente jogo.

- **Badge**

Crachá que, normalmente, será criado pela organização. Um *badge* é atribuído a um colaborador consoante diversos requisitos, mas sempre relacionado com a realização de eventos.

- **Sistema Externo**

Um sistema externo, tal como o nome indica, é um sistema/aplicação independente da plataforma desenvolvida no presente projeto. Estes sistemas podem pertencer à organização ou não. Este termo é utilizado para referir todos os sistemas que contêm integrado atividades que os colaboradores podem realizar, tornando necessário o registo desses eventos na plataforma de *gamification*.

3.2. Sistemas Externos na Empresa

Um grande requisito para o projeto é o desenvolvimento de uma potencialidade que permita outros sistemas externos comunicarem com a plataforma. Atualmente, no sistema organizacional da empresa existem diferentes plataformas onde os colaboradores podem e devem fazer diferentes atividades. No entanto, como já referido, estas plataformas não têm o uso desejado. O LagoonIdeas e o Intranet são casos de plataformas disponíveis aos colaboradores mas com pouca movimentação.

3.2.1. LagoonIdeas

O LagoonIdeas é uma plataforma desenvolvida pela XLM que permite a gestão do processo de ideias - gestão de IDI (Morais, 2010) de uma forma simples e intuitiva.

O LagoonIdeas é uma plataforma que visa a procura de novas ideias ou soluções para determinados problemas, dentro e fora da organização, de modo a maximizar a resolução de problemas e a criação de novas soluções para a organização.

Esta plataforma oferece um *interface* que permite a cooperação entre os colaboradores que registam as ideias e a organização que gere todo o processo. A plataforma gere desde a receção de ideias, à sua análise e avaliação, até à entrega para implementação.

Um dos objetivos para o projeto de *gamification* é permitir a comunicação dos eventos que os colaboradores realizam no *LagoonIdeas*. Esta plataforma é desenvolvida em *ruby on rails* (Kehoe, 2013) e ainda não contém uma API própria. No entanto, como requisito, a API incorporada no projeto de *gamification*, deve estar preparada para receber os pedidos via REST (Fielding, 2000) que serão enviados no futuro por esta plataforma.

3.2.2. Intranet

O Intranet é uma pequena plataforma desenvolvida pela XLM que permite a inserção/gestão das picagens diárias pelos utilizadores. Diariamente, os colaboradores devem aceder a esta plataforma quando entram e saem do estabelecimento de trabalho. Esta plataforma ainda é recente dentro da organização e o movimento nela ainda não é o ideal. Pretende-se que todos os colaboradores sejam assíduos nesta tarefa. Portanto, é um requisito esta plataforma também comunicar com a plataforma de *gamification*.

O Intranet é desenvolvido em *drupal* (Schaffer, 2013) e ainda não contém uma API incorporada. Porém, tal como no LagoonIdeas, a API incorporada no projeto de *gamification*, deve estar preparada para receber os pedidos via REST que no futuro serão enviados pelo Intranet.

A organização estuda a possibilidade do fim da utilização de *drupal* dentro da organização. Portanto, existem ainda futuras decisões a tomar pela empresa sobre a possibilidade de integrar o *frontOffice* desenvolvido no projeto de *gamification* na página principal do Intranet ou vice-versa.

3.2.3. LagoonSecurity

O LagoonSecurity ou LS, referido deste modo daqui em diante, é uma plataforma também desenvolvida pela XLM que permite gestão de perfis de utilizadores e permissões nos mais diversos sistemas, de uma forma simples e intuitiva (lagoon.pt, 2013). Este sistema integra-se facilmente em qualquer aplicação capaz de efetuar pedidos REST. A grande vantagem do uso deste sistema é a redução do tempo de desenvolvimento de outras aplicações, relativamente ao processo de autenticação e permissões dos diversos utilizadores.

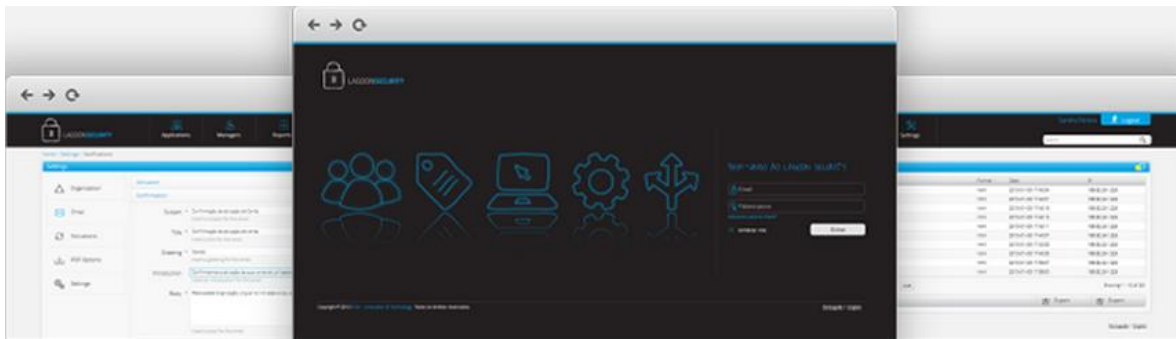


Fig. 4 - Apresentação LagoonSecurity

A nível interno, o LS funciona tendo em conta aplicações, ambientes, utilizadores, módulos, ações, pontos de ação, utilizadores e perfis.

Um utilizador registado pode registar e gerir várias aplicações. Cada aplicação registada tem de ter no mínimo um ambiente associado. O ambiente configurado permite aos pedidos externos recebidos, serem autenticados no LS. A nível de segurança, este sistema exige autenticação de todos os pedidos efetuados. Ainda no âmbito do ambiente, o LS permite escolher e gerir os *e-mails* que são enviados aos diversos utilizadores, consoante os pedidos REST que são solicitados ao sistema.

Um módulo é constituído por várias ações e cada ação pode ter diversos pontos de ação. Estes conceitos são essenciais para identificar os perfis de utilizadores existentes na aplicação.

No âmbito do projeto de *gamification*, um exemplo da aplicação destes conceitos, pode ser: o acesso ao menu de configurações existente no *backOffice* é um módulo. Gerir os colaboradores que se encontra dentro desse módulo é uma ação. A opção adicionar novo colaborador é um ponto de ação.

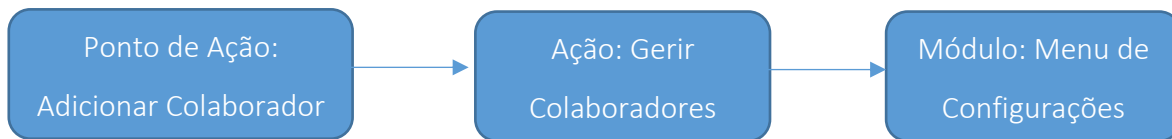


Fig. 5 - Exemplo conceitos LagoonSecurity

Um perfil de utilizador é composto por diversos pontos de ação geridos pelo (s) administrador (es) do sistema. Através destes perfis, o LagoonSecurity gere as permissões de cada utilizador.

O LagoonSecurity permite que cada ambiente tenha diversos utilizadores, cujos podem ter um ou vários perfis de utilizadores. No entanto, como requisito, a aplicação de *gamification* limita um tipo de perfil associado, a cada colaborador registado.

A utilização desta ferramenta para o projeto de *gamification* é um requisito essencial. Para o futuro, a empresa planeia que todos os sistemas internos estejam incorporados no LagoonSecurity, tornando os utilizadores que estão registados serem os mesmos para todas as aplicações - colaboradores.

3.3. Identificação dos Utilizadores

Como referido, a plataforma é composta por quatro componentes principais: API, *backOffice*, *frontOffice* e comunicação LS. Excetuando este último, para todos os restantes componentes existe um tipo de utilizador associado. Contrariamente, a comunicação com o LS não contempla nenhum utilizador em concreto, pois o sistema de *gamification* do presente projeto é que se apresenta como um utilizador do LS.

- Utilizador Externo

Este tipo de utilizadores são as aplicações externas que enviam eventos à plataforma através da API. Estes utilizadores são caracterizados como utilizadores indiretos.

- Administrador

Pode existir um administrador ou vários no sistema. Um administrador é um colaborador com funções de gerir todo o sistema relacionado com o *gamification* dentro da empresa. Estes utilizadores são caracterizados como utilizadores diretos com acesso ao *backOffice*.

- Colaborador Normal

Um utilizador considerado colaborador normal pode visualizar toda a informação que lhe é associada no presente jogo temporal corrente. Além disso, pode realizar outras funcionalidades que se encontram mais à frente identificadas. Estes utilizadores são caracterizados como utilizadores diretos com acesso ao *frontOffice*.

3.4. Requisitos Funcionais

Como referido anteriormente, pretende-se obter uma plataforma com recursos ao *gamification*, capaz de responder às necessidades da empresa, permitindo a gestão de todo o processo do sistema e, com a finalidade de incentivar a colaboração dentro da organização.

Após a recolha de informação obtida através da pesquisa realizada e das informações transmitidas pela empresa, às quais definem-se como os principais requisitos para este projeto, verifica-se que são necessários construir as quatro grandes componentes referidas (API, *backOffice*, *frontOffice* e a comunicação com as funcionalidades do LS). Os requisitos funcionais encontram-se explicados separadamente para estes quatro componentes no presente subcapítulo.

Os requisitos solicitados alargam-se em todos os conceitos referidos. Contudo, como em tudo, existe sempre prioridades de desenvolvimento, portanto todos os requisitos referentes estão classificados como essenciais, importantes ou desejáveis:

- Requisito essencial tem de estar mesmo aplicado no sistema.
- Requisito importante deve estar aplicado no sistema.
- Requisito desejável não tem de estar aplicado ao sistema, mas seria vantajoso.

3.4.1. API para Sistemas Externos

Como referido no subcapítulo “3.2. Sistemas Externos na Empresa” existem ferramentas dentro da organização onde os colaboradores já realizam determinadas atividades diariamente. Um dos objetivos principais, ao recurso de *gamification*, é o aumento do incentivo para a utilização destas ferramentas que atualmente apresentam pouca adesão, como referido no subcapítulo “1.1.2 A Empresa”.

Reparando nesta necessidade, torna-se essencial que a plataforma desenvolvida no presente projeto monitorize as atividades realizadas nas outras ferramentas. Portanto, um requisito principal para o presente projeto é permitir outros sistemas externos comunicarem com a plataforma de *gamification*, os eventos realizados. Apesar de o subcapítulo “3.2. Sistemas Externos na Empresa” referir apenas os dois sistemas atuais na organização que necessitam de comunicação de eventos com o projeto de *gamification*, o desenvolvimento desta API permite que qualquer sistema desejável consiga comunicar também com a plataforma.

Nem todos os sistemas externos podem comunicar com a plataforma sem existir uma devida autorização. A plataforma necessita de conter incorporado uma gestão dos sistemas que podem comunicar, bem como as atividades que os mesmos podem enviar. Essa gestão está integrada na componente do “3.4.2. *BackOffice*”, referida seguidamente.

A tabela seguinte apresenta os requisitos necessários no desenvolvimento da API.

Tabela 5 - Requisitos funcionais para a API

Requisito	Descrição	Prioridade
Inserção de eventos realizados noutros sistemas	Quando um colaborador realiza um evento noutra ferramenta externa, permitir a adição desse evento na plataforma de <i>gamification</i> .	Essencial
Informar as aplicações externas do resultado da comunicação.	Cada vez que uma aplicação externa solicita um novo evento à plataforma, a mesma tem de dar uma resposta com o resultado do pedido (se obteve sucesso ou insucesso e qual).	Essencial
Validar as aplicações externas que comunicam os eventos	Evitar que qualquer aplicação possa somente enviar pedidos de inserção de novos eventos sem estar identificada.	Importante

3.4.2. *BackOffice*

O *backOffice* da plataforma destina-se apenas aos colaboradores ou às pessoas responsáveis por gerir todo o sistema relacionado com o *gamification* na empresa, ou seja, utilizadores do tipo administradores.

Este módulo define-se como sendo o requisito principal do projeto, pois um dos grandes objetivos é permitir uma gestão de todas as atividades que os colaboradores realizam para a empresa, tanto a partir das ferramentas existentes (por exemplo, a inserção de uma ideia no LagoonIdeas) como a realização de atividades espontâneas (por exemplo, a escrita de um artigo). Os requisitos do *backOffice* agrupam-se em diferentes áreas. As próximas tabelas apresentam todos os requisitos necessários no desenvolvimento do *backOffice*, separados por essas diversas áreas.

Gestão de jogos temporais:

Tabela 6 - Requisitos para a gestão de jogos temporais

Requisito	Descrição	Prioridade
Adicionar jogo	Adicionar um nome, uma descrição – história -, a data de início, a data de fim do jogo e o estado do jogo.	Essencial
Remover jogo	Permitir a remoção do jogo sempre que não existem eventos associados.	Essencial
Editar jogo	Editar todos os dados de cada jogo temporal.	Essencial
Só permitir um jogo em aberto	Os jogos temporais têm um estado de “aberto” ou “fechado”. Apenas só pode estar um jogo em aberto no sistema.	Essencial
Adicionar nível ao jogo	Adicionar nome e descrição de cada nível.	Essencial
Atribuir pontuação mínima a cada nível	Permitir que cada nível tenha uma pontuação mínima para ser atingido.	Essencial
Adicionar imagens a cada nível do jogo	É pertinente se cada nível tiver uma imagem associada. Deste modo, os utilizadores em vez de visualizarem apenas uma pontuação total no jogo conseguem ser, por exemplo, uma personagem naquele jogo.	Importante
Remover nível do jogo	Permitir a qualquer altura remover um nível do jogo, sem interferir nos restantes níveis.	Essencial
Editar nível do jogo	Editar todas as informações de cada nível.	Essencial
Listar todos os jogos	Listar as informações de todos os jogos temporais em sistema.	Importante

Gestão de categorias:*Tabela 7 - Requisitos funcionais para a gestão de categorias*

Requisito	Descrição	Prioridade
Adicionar categoria	Adicionar um nome de uma nova categoria.	Essencial
Remover categoria	Permitir a remoção de categorias sempre que não existem atividades associadas.	Essencial
Editar categoria	Editar o nome das categorias.	Essencial
Adicionar nível à categoria	Adicionar nome e uma pontuação para a inserção de níveis numa determinada categoria.	Essencial
Atribuir pontuação mínima a cada nível	Permitir que cada nível tenha uma pontuação mínima para ser atingido.	Essencial
Remover nível da categoria	Permitir a qualquer altura remover um nível da categoria, sem interferir nos restantes níveis e caso nenhum utilizador esteja presente nesse nível.	Essencial
Editar nível da categoria	Editar todas as informações de cada categoria.	Essencial
Listar todas as categorias	Listar as informações de todas as categorias em sistema.	Importante

Gestão de atividades:*Tabela 8 - Requisitos funcionais para a gestão de atividades*

Requisito	Descrição	Prioridade
Adicionar atividade numa categoria	Adicionar uma descrição e pontuação de uma nova atividade. Permitir a adição de uma atividade sempre dentro de uma categoria existente no sistema.	Essencial
Remover atividade	Permitir a remoção de atividades sempre que não	Essencial

	existem eventos associados a essa atividade.	
Editar atividade	Editar todos os dados das atividades.	Essencial
Atribuir pontuação fixa à atividade	Permitir a adição de uma pontuação a cada atividade. Desta forma, cada vez que é realizado um evento é automaticamente atribuída essa pontuação ao (s) colaborador (es) que realizaram-na.	Essencial
Atribuir peso e pontuação ponderada	Permitir um peso quantitativo às atividades assim se o desejarem. Algumas atividades podem ter uma pontuação fixa, enquanto outras podem ter um peso associado onde a pontuação pode variar. Por exemplo, um colaborador que seja orador numa formação, o peso é o número de horas. Para estas situações é necessário gerir qual esse peso bem como o mínimo e o máximo do mesmo.	Essencial
Listar todas as atividades por categoria	Listar as informações de todas as atividades em sistema das diferentes categorias.	Importante

Gestão de colaboradores:

Os requisitos que se seguem relacionam-se com os colaboradores, ou seja, com os utilizadores da plataforma. No entanto, apenas referem-se a nível de configurações individuais. Existem mais requisitos que fazem parte da gestão dos colaboradores, todavia relacionam-se mais com a gestão dos eventos e dos *badges*.

Tabela 9 - Requisitos funcionais para a gestão de colaboradores

Requisito	Descrição	Prioridade
Adicionar colaborador	Permitir a adição manual de novos colaboradores. Quando se adiciona um novo utilizador permitir escolha do tipo de utilizador (administrador ou colaborador normal).	Essencial
Adicionar colaborador ao LS	Quando se adiciona um novo colaborador é necessário comunicar e adicioná-lo ao LS para, posteriormente, esse colaborador conseguir autenticar-se na plataforma.	Essencial
Enviar <i>e-mail</i> ao colaborador com <i>link</i> para atribuição de uma senha de acesso	Quando se cria um novo colaborador é necessário o envio de <i>e-mail</i> com um <i>link</i> da plataforma, onde o novo colaborador adiciona a nova senha que permite a ativação da conta de acesso à plataforma.	Essencial
Remover colaborador	Permitir a eliminação de um colaborador do sistema, desde que não tenha eventos nem <i>badges</i> associados.	Essencial
Remover colaborador do LS	Permitir a eliminação de um colaborador do LS, de modo a desativar o colaborador impossibilitando o mesmo de se autenticar no sistema.	Essencial
Editar colaborador	Permitir editar todos os dados de cada colaborador.	Essencial
Listar todos os colaboradores	Listar as informações de todos os colaboradores em sistema.	Essencial

Gestão de eventos:*Tabela 10 - Requisitos funcionais para a gestão de eventos*

Requisito	Descrição	Prioridade
Adicionar evento manualmente	Permitir a inserção manual de um evento realizado por um ou vários colaboradores.	Essencial
Adicionar eventos individuais ou em grupo	Aquando da inserção de um novo evento, permitir ao utilizador escolher a tipologia de evento: individual ou em grupo. Caso se deseje inserir um evento em grupo, permitir a escolha dos restantes colaboradores, possibilitando sempre a remoção dos mesmos.	Essencial
Remover evento	Permitir a eliminação de um evento. Quando esta ação é executada ter em conta a diminuição da pontuação nos diferentes níveis do (s) colaborador (es) que realizaram esse evento.	Essencial
Editar evento	Permitir a edição das informações básicas de um evento, nomeadamente, a data e os colaboradores que realizaram esse evento.	Essencial
Listar informação de cada evento	Permitir visualizar toda a informação detalhada de cada evento.	Essencial
Adicionar colaborador a evento	Permitir a adição posterior de outros colaboradores num determinado evento.	Essencial
Remover colaborador de evento	Permitir a remoção de um ou vários colaboradores de um determinado evento. Apenas permitir a remoção de outros colaboradores associados ao evento, ou seja, impossibilitar a eliminação do colaborador principal associado ao evento. Desta	Essencial

	forma, impossibilita a existência em sistema de um evento sem colaboradores.	
Listar todos os eventos por colaborador	Permitir a listagem de todos os eventos de cada colaborador.	Essencial
Pesquisar eventos	Permitir a pesquisa de eventos por colaborador, por jogo temporal e por categoria.	Importante

Gestão de *badges*:

A gestão de *badges* divide-se em duas vertentes: a parte de configurações e da criação dos *badges* e a parte de atribuição dos mesmos. A tabela seguinte apresenta os requisitos direcionados para essas duas vertentes.

Tabela 11 - Requisitos funcionais para a gestão de *badges*

Requisito	Descrição	Prioridade
Adicionar <i>badge</i>	Permitir a inserção de um novo <i>badge</i> no sistema, sendo necessário guardar o nome, a descrição e a imagem relativamente ao mesmo.	Essencial
Relacionar cada <i>badge</i> com o jogo temporal corrente	Cada vez que um <i>badge</i> é adicionado ao sistema, automaticamente deve ser associado ao jogo corrente. Desta forma, possibilita aos colaboradores ganharem o mesmo <i>badge</i> mas em jogos diferentes.	Essencial
Remover <i>badge</i>	Permitir a eliminação de um <i>badge</i> do sistema, desde que não exista colaboradores com esse <i>badge</i> já conquistado.	Essencial
Editar <i>badge</i>	Permitir a edição os dados de cada <i>badge</i> , inclusive a imagem.	Essencial

Desenhar <i>badge</i>	Apesar dos <i>badges</i> que estarão no sistema serem desenhados internamente pela empresa é interessante a possibilidade de desenhar alguns modelos através da aplicação.	Desejável
Listar todos os <i>badges</i>	Permitir a listagem de todos os <i>badges</i> existentes em sistema, com a possibilidade de filtrar os <i>badges</i> existentes em cada jogo temporal.	Essencial
Atribuição manual de <i>badge</i>	Cada vez que um <i>badge</i> é criado, permitir que automaticamente seja possível ao utilizador fazer atribuições desse <i>badge</i> , aos colaboradores, de forma manual.	Essencial
Criar vários métodos de atribuição de cada <i>badge</i> e fazer a gestão desses métodos	Além da atribuição manual de <i>badges</i> , permitir que cada <i>badge</i> possa ser atribuído em outros diferentes métodos. Por exemplo, um colaborador que realize 100 pontos numa categoria pode conquistar automaticamente um determinado <i>badge</i> . O sistema deve permitir ao utilizador a possibilidade de gerir os vários métodos de atribuição de cada <i>badge</i> . Os métodos de atribuição automáticos devem-se relacionar com as pontuações obtidas ou o número máximo de eventos numa determinada atividade ou categoria.	Essencial
Adicionar método de atribuição	Permitir ao utilizador adicionar, sempre que desejar, um novo método de atribuição de um determinado <i>badge</i> .	Essencial
Remover método de atribuição	Permitir ao utilizador remover, sempre que desejar, um método de atribuição de um determinado <i>badge</i> .	Essencial

Listar todos os métodos de atribuição de cada <i>badge</i>	Permitir a listagem de todos os métodos de atribuição associados a cada <i>badge</i> .	Essencial
Atribuir um <i>badge</i> a um colaborador manualmente	Permitir a atribuição manual de um <i>badge</i> a um determinado colaborador. O foco deve ser o colaborador, ou seja, primeiro o utilizador escolhe o colaborador e só depois o <i>badge</i> que irá atribuir.	Essencial
Listar todos os <i>badges</i> de cada colaborador	Permitir a listagem de todos os <i>badges</i> de cada colaborador, com a possibilidade de filtrar os <i>badges</i> conquistados em cada jogo temporal.	Essencial
Atribuir um <i>badge</i> a vários colaboradores manualmente	Permitir a atribuição manual de um determinado <i>badge</i> a um ou a vários colaboradores simultaneamente. O foco nesta situação deverá ser o <i>badge</i> , ou seja, primeiro o utilizador deve escolher o <i>badge</i> e só depois os colaboradores.	Importante
Indicar o total de atribuições de cada <i>badge</i>	Permitir a visualização do total de conquistas de cada <i>badge</i> .	Desejável

Gestão de aplicações externas:

Como indicado, anteriormente, nos requisitos “3.4.1. API para Sistemas Externos”, existe a necessidade de identificar as aplicações que efetuam um pedido à API. Contudo, torna-se essencial existir uma gestão das aplicações externas com permissões de comunicação com a API. Essa gestão tem de ser gerida pelo (s) administrador (es). A tabela seguinte apresenta os requisitos para responder a todas as necessidades dessa gestão.

Tabela 12 - Requisitos funcionais para a gestão de aplicações externas

Requisito	Descrição	Prioridade
Adicionar aplicação	Permitir a inserção de novas aplicações externas no sistema. Uma aplicação não consegue comunicar com a plataforma de <i>gamification</i> , nem nenhum pedido que seja solicitado é processado, se a aplicação não estiver identificada no sistema. Portanto, é necessário atribuir à aplicação um identificador único externo.	Essencial
Remover aplicação	Permitir a eliminação de uma determinada aplicação externa.	Essencial
Editar aplicação	Permitir a edição dos dados referentes a cada aplicação externa.	Essencial
Listar todas as aplicações	Permitir a listagem de todas as aplicações externas autorizadas a comunicar com o sistema.	Essencial
Adicionar ações às aplicações	Em cada sistema externo são realizadas apenas determinadas atividades. A possibilidade de gerir essas atividades torna-se relevante, limitando as atividades que podem ser realizadas em cada sistema externo. Portanto, o sistema deve permitir a adição e identificação de determinadas ações, que as aplicações podem solicitar à API. Se a ação solicitada não existir no sistema, ou se existir mas não se relacionar com a aplicação, nenhum pedido solicitado é processado.	Essencial
Remover ações das aplicações	Permitir a remoção, sempre que desejável, de ações associadas a uma determinada aplicação externa.	Essencial

Listar todas as ações de cada aplicação externa	Permitir a listagem de todas as ações associadas às diversas aplicações externas.	Essencial
---	---	-----------

Outros Requisitos:

Para além dos requisitos anteriormente referidos, que se enquadram dentro de gestões conceituais, existem também outros requisitos importantes solicitados pela empresa.

Tabela 13 - Outros requisitos funcionais para o *backOffice*

Requisito	Descrição	Prioridade
Ter em conta sempre o jogo corrente	Muitos dos requisitos pedem listagens de informações. Essas listagens devem sempre corresponder às informações relacionadas com o jogo corrente e não com os jogos temporais anteriores. No entanto, o sistema deve permitir ao utilizador pesquisar informações relativas a jogos que estejam já num estado fechado.	Essencial
Visualizar tabela de pontuações gerais	Permitir a visualização, na página inicial do <i>backOffice</i> , de uma tabela com as pontuações gerais ordenadas ascendentemente, dos diferentes colaboradores no jogo temporal corrente.	Importante
Visualizar <i>feed</i> de últimas atividades	Permitir a visualização, na página inicial do <i>backOffice</i> , de uma lista com as últimas atividades realizadas pelos colaboradores bem como as atribuições de <i>badges</i> conquistadas pelos mesmos.	Importante
Visualizar estatísticas	Permitir a visualização de estatísticas e resultados obtidos através de vários filtros de pesquisa, nomeadamente por jogo, datas, categoria,	Importante

	atividade e por colaborador.	
--	------------------------------	--

3.4.3. *FrontOffice*

O *frontOffice* da plataforma destina-se aos colaboradores normais, ou seja, aos colaboradores que realizam os eventos no sistema.

Nesta componente, os colaboradores conseguem, essencialmente, visualizar a sua progressão no jogo corrente. A próxima tabela apresenta os diversos requisitos que abrangem esta progressão.

Como já referido no subcapítulo “3.2.2. Intranet” existe uma plataforma onde os colaboradores interagem diariamente. A empresa está a estudar e a avaliar as possibilidades de colocar os detalhes existentes no *frontOffice* do presente projeto, nessa plataforma diária com alguns outros componentes ou vice-versa. Dadas as circunstâncias, este requisito não foi considerado um objetivo principal, mas essencial pois toda esta componente põe em prática os métodos de *gamification* aplicados no restante sistema.

Tabela 14 - Requisitos funcionais para o *frontOffice*

Requisito	Descrição	Prioridade
Visualizar progressão atual no jogo corrente	Permitir ao colaborador a visualização da sua progressão geral no jogo corrente. Se os níveis do jogo corrente possuem imagens, o sistema deve mostrar o nível em que o colaborador se encontra, bem como a imagem do nível e a pontuação total atual.	Essencial
Visualizar pontuações e posições nos níveis das diferentes categorias	Permitir ao colaborador a visualização do seu estado atual em cada categoria do sistema, ou seja, o sistema deve indicar a pontuação e a posição do colaborador para cada categoria, no jogo temporal corrente.	Essencial

Visualizar os eventos realizados	Permitir ao colaborador a visualização dos eventos que realizou no presente jogo.	Essencial
Visualizar informações de eventos nas diferentes categorias	Permitir ao colaborador verificar informações dos diversos eventos, nomeadamente, a descrição e a pontuação que ganham se realizarem eventos, nas diferentes categorias.	Importante
Visualizar os <i>badges</i> ganhos	Permitir ao colaborador a visualização dos diferentes <i>badges</i> que conquistou ao longo do presente jogo.	Essencial
Visualizar todos os <i>badges</i> conquistáveis	Permitir ao colaborador visualizar todos os <i>badges</i> que podem ser conquistados, bem como as informações de como o podem conseguir.	Importante
Visualizar a colaboração realizada com outros colaboradores	Permitir ao colaborador verificar quantos eventos em equipa já realizou e quais e com quem realizou os eventos.	Essencial
Visualizar o perfil dos colaboradores que já realizaram eventos juntos	Permitir ao colaborador visualizar o perfil de outros colaboradores, ou seja, os seus estados atuais no jogo corrente. O sistema apenas pode permitir a visualização de perfis dos colaboradores que já realizaram eventos juntos.	Essencial
Visualizar um <i>feed</i> com as últimas atividades realizadas por todos os colaboradores	Permitir ao colaborador verificar os últimos eventos realizados bem como os últimos <i>badges</i> conquistados, por todos os colaboradores no sistema.	Desejável
Visualizar a posição atual no jogo	Permitir ao colaborador verificar a sua posição atual num <i>ranking</i> , através de uma tabela de pontuações.	Essencial

corrente	Nesta tabela, o sistema apenas deve permitir ao colaborador verificar a sua posição e a sua pontuação geral atual, bem como as pontuações relativas ao colaborador que se posiciona anteriormente e ao colaborador que se posiciona seguidamente no <i>ranking</i> .	
Gerir dados pessoais	Permitir ao colaborador alterar os seus dados pessoais assim que o desejar.	Essencial
Publicar <i>badges</i> ganhos	Permitir ao colaborador partilhar distinções obtidas na plataforma e integrar esta informação em rede (s) social (is) adequadas, tornando-se uma forma de enriquecer o seu currículo com elementos credíveis das suas capacidades.	Desejável
Visualizar na página inicial os últimos dois eventos, os últimos dois <i>badges</i> ganhos e o colaborador com quem realizou mais eventos juntos	Permitir ao colaborador visualizar na página inicial, ou página de perfil, os últimos dois eventos que realizou, os últimos dois <i>badges</i> que conquistou e o colaborador com quem realizou mais eventos juntos.	Importante

3.4.4. Comunicação com o LagoonSecurity

Como já referido, o LS apresenta diversas funcionalidades vantajosas no desenvolvimento de um sistema. A utilização desta ferramenta para a gestão de acessos e permissões na plataforma é um requisito essencial para o presente projeto.

A tabela que se segue apresenta a lista com os requisitos necessários na comunicação com o LS.

Tabela 15 - Requisitos funcionais para a comunicação com o LagoonSecurity

Requisito	Descrição	Prioridade
Restringir uma autenticação por colaborador	Nenhum colaborador pode ser vários utilizadores ao mesmo tempo. Se um administrador apresentasse como um colaborador normal, então este terá de conter duas entradas de sessão.	Essencial
Restringir o registo de utilizadores	Nenhum colaborador pode-se registar diretamente na plataforma. Se o colaborador ainda não existir no LS, então só um administrador pode registar um colaborador no sistema.	Essencial
Autenticação via LS	Quando um utilizador se autentica para entrar na plataforma, o sistema deve pedir essa autenticação ao LS, que se responsabiliza por validar cada utilizador e por fazer a gestão de permissões de cada um. Além disso, logicamente, o sistema antes de pedir autenticação externa, deve garantir que o utilizador seja válido na base de dados que suporta o sistema.	Essencial
Permissões de acesso a funcionalidades geridas a partir do LS	As permissões para as diversas funcionalidades, impostas pelos requisitos descritos, podem variar consoante o tipo de utilizador. Todas as permissões dos utilizadores, dentro do <i>backOffice</i> e do <i>frontOffice</i> , devem ser geridas na plataforma do LS por um administrador ou outra pessoa responsável pela gestão. Não é da responsabilidade da plataforma fazer essa gestão. O sistema apenas deve estar preparado para permitir ou impedir as diversas funcionalidades, independentemente das	Essencial

	permissões ou das alterações futuras das mesmas.	
--	--	--

3.5. Requisitos não Funcionais

Para a concretização do projeto, além dos requisitos funcionais, é também necessário considerar alguns requisitos não funcionais, de forma a garantir a qualidade do sistema. O presente subcapítulo aborda esses requisitos que são necessários cumprir.

3.5.1. Desempenho

Qualquer sistema, por mais funcionalidades interessantes que tenha implementado, precisa sempre de ser eficiente relativamente ao desempenho. A resposta das funcionalidades implementadas não devem ser demoradas para os utilizadores e, para isso, é necessário minimizar a quantidade de recursos que cada ação efetuada pelo utilizador consome.

A lógica principal de todas as funcionalidades do sistema deve estar do lado do servidor e não do cliente, desta forma, a responsabilidade de existir um processamento rápido fica maioritariamente do servidor e não da máquina do cliente. Para o desenvolvimento do projeto é utilizada a *framework* MVC – *CodeIgniter*. Esta *framework* tem uma arquitetura server-side, ou seja, todo o processamento é executado do lado do servidor e termina a gerar o HTML adequado e envia-o para o *browser* dos utilizadores. Além desta vantagem, esta *framework* disponibiliza diversas funções otimizadas já implementadas e preparadas para oferecer respostas rápidas aos utilizadores.

Além do uso da *framework* referida, também é utilizada a tecnologia AJAX para algumas funcionalidades do projeto, permitindo diminuir carregamentos sucessivos das “vistas” visíveis aos utilizadores, evitando assim, novos pedidos ao servidor.

As explicações destas tecnologias encontram-se no subcapítulo “4.2. Tecnologias Implementadas”.

No funcionamento deste projeto, apenas se justifica um pior desempenho na comunicação com o LS, pois a responsabilidade não depende só do sistema de *gamification* mas também deste sistema. Além disso, é necessário considerar sempre o estado da rede nos momentos da comunicação.

A estrutura da base de dados deve ser desenvolvida de modo a conseguir responder às necessidades dos utilizadores de uma forma rápida e eficaz, não pondo em causa a integridade em que os dados são apresentados.

3.5.2. Usabilidade

Uma vez que o projeto tem como objetivo o desenvolvimento de uma aplicação *online* torna-se bastante importante focar alguma atenção no *design* do *interface* desta. As funcionalidades da aplicação devem-se apresentar de uma forma simples e intuitiva. Espera-se que os utilizadores consigam efetuar as ações pretendidas sem perderem muito tempo na procura das funcionalidades que pretendem. Este define-se como o requisito principal, a nível de usabilidade, requerido pela empresa.

Outro requisito a nível de usabilidade, solicitado pela XLM, baseia-se na necessidade de focar sempre o colaborador, no âmbito do *backOffice*. Isto é, pretende-se que o sistema foque sempre primeiro o colaborador e só depois os eventos e os *badges*. Por exemplo, na inserção de um novo evento, não se pretende que o utilizador escolha uma opção de “Eventos” e, posteriormente, escolha o (s) colaborador (es) para associar. Pretende-se sim que primeiro seja escolhido o colaborador e só depois os restantes detalhes necessários para a inserção de um evento no sistema.

Além dos requisitos de usabilidade pedidos pela empresa, a informação e os menus devem ser agrupados e apresentados por relação de proximidade ou hierarquia. Os menus devem estar sempre visíveis, de forma o utilizador conseguir de modo fácil e rápido aceder a qualquer outra informação.

Sendo uma plataforma que será utilizada por diversos tipos de pessoas, o sistema deve apresentar uma simplicidade estética, ou seja, um *interface* com menus simples com vista a agradar a todos os tipos de utilizadores que irão utilizar a plataforma.

Relativamente aos erros que, eventualmente, possam ocorrer durante a execução das diversas funcionalidades, como por exemplo no preenchimento de formulários, o sistema deve apresentar mensagens de erro ao utilizador. O mesmo deve acontecer quando o utilizador termina a tarefa com sucesso. Os formulários da plataforma devem conter um “*” para todos os campos de preenchimento obrigatório, aumentando assim a atenção do utilizador. A simplicidade linguística também deve ser uma característica no sistema, ou seja, a utilização de textos e expressões compatíveis com os vários tipos de utilizadores.

3.5.3. Segurança

A segurança em qualquer sistema é uma componente bastante importante a considerar, principalmente quando se coloca em questão vários tipos de utilizadores, cujos dados pessoais são administrados, como é o caso do presente projeto.

Sendo uma plataforma que suporta vários tipos de utilizadores com vários tipos de acessos e permissões, é necessário efetuar um controlo desses privilégios de cada utilizador através do registo e da autenticação na plataforma. Como já referido no subcapítulo “3.4.4. Comunicação com o LagoonSecurity”, o LS tem a maior responsabilidade de gerir os diversos tipos utilizadores bem como as permissões dos mesmos. Existe no LS uma área com uma lista dos tipos de utilizadores que existem e uma área com todas as funcionalidades da aplicação onde é feita a gestão dos privilégios de cada utilizador. Um administrador ou um indivíduo responsável pela plataforma de *gamification* pode entrar na plataforma do LS e adicionar ou retirar as permissões que cada tipo de utilizador possui. Com isto, a responsabilidade de segurança, para estas situações, encontra-se maioritariamente do lado do LS.

No acesso à plataforma – processo de autenticação – também é solicitado uma validação ao LS. Todos os utilizadores que são inseridos no sistema têm de se encontrar também no sistema do

LS, com determinados dados pessoais, inclusivo o *username* e a senha de acesso. Além disto, existe um estado de “ativação” de cada utilizador, ou seja, o utilizador pode estar ativo ou inativo. Caso o utilizador esteja inativo nunca conseguirá autenticar-se na plataforma de *gamification*. O sistema, logicamente, antes de pedir autenticação externa ao LS, deve garantir que o utilizador é válido na base de dados que suporta o sistema, pois todos os utilizadores para se autenticarem também têm de estar registados na base de dados.

Além da importância da segurança no processo de autenticação e na gestão de permissões, existem outras formas de aumentar a segurança no sistema a partir do desenvolvimento do mesmo. Estas características de implementação encontram-se mais à frente no subcapítulo de segurança no capítulo “5. Desenvolvimento e Implementação” do sistema.

3.5.4. Portabilidade

A nível de portabilidade, a XLM solicitou que a plataforma deve ser acedida num *Desktop*, com acesso à Internet através de um *browser*, principalmente nos mais comuns (*Google Chrome*, *Firefox* e *Internet Explorer*). No entanto, com a utilização de HTML5 e CSS3 no desenvolvimento do projeto, referidos no capítulo “4. Arquitetura e Conceção”, a plataforma permite ser acedível em qualquer dispositivo com acesso à Internet via *browser*, facilitando a interação dos utilizadores.

4. Arquitetura e Conceção

Após um estudo ao tema *gamification* e ao levantamento de requisitos para o presente projeto, estão reunidas todas as condições para planear o desenvolvimento da plataforma. Este planeamento, ao longo do tempo, sofreu bastantes alterações devido à metodologia *scrum*, pois os requisitos foram sofrendo alterações e novas necessidades também foram nascendo. O presente capítulo aborda a união das diversas fases do *scrum*.

O início do presente capítulo apresenta e explica a arquitetura adotada para o desenvolvimento do projeto. De seguida, refere as tecnologias utilizadas para o desenvolvimento das diferentes partes da arquitetura. Posteriormente, são apresentados os diagramas e cenários mais importantes para a compreensão do funcionamento lógico da plataforma. Por fim, este capítulo apresenta a explicação do modelo-de-dados criado, nomeadamente, o modelo Entidade-Relação, que dá suporte a todo o funcionamento da plataforma. Desta forma, consegue-se obter, com mais facilidade, uma perspetiva da estruturação e armazenamento de dados no sistema de *gamification*.

4.1. Arquitetura

A arquitetura do sistema resulta da análise de todos os requisitos referidos no capítulo anterior “3. Análise de Requisitos”, incluindo o legado existente dentro do sistema interno da XLM.

4.1.1. Arquitetura “Three Tier”

A plataforma, como já referido, é uma aplicação *Web*. Como tal, a arquitetura baseia-se numa arquitetura cliente-servidor (Rouse, 2008) onde se aplica, a nível global, uma arquitetura “*Three Tier*”, que já é aplicada nos sistemas atuais existentes na empresa.

O modelo “*Three Tier*” (Ramirez, 2000), como ilustra a figura seguinte, é constituído por três camadas onde se encontram separados os diversos componentes da aplicação.

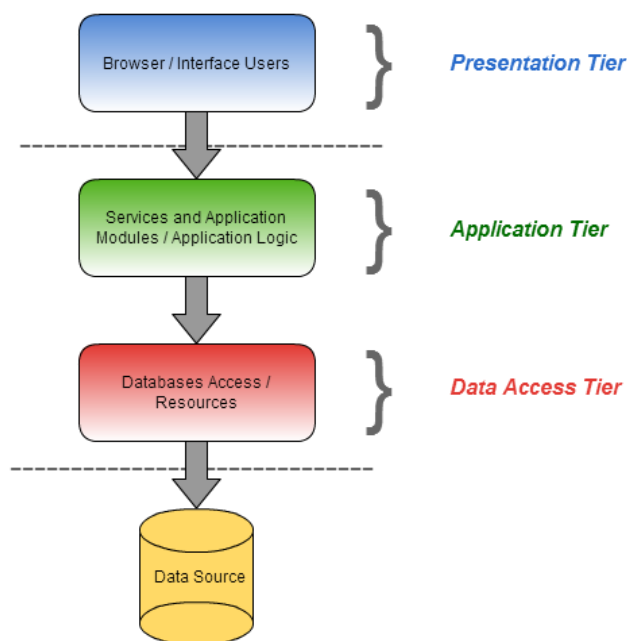


Fig. 6 - Diagrama das três camadas da arquitetura "Three Tier"

A camada de apresentação – *Presentation Tier* – é responsável por disponibilizar todos os dados das camadas inferiores aos utilizadores. Esta camada proporciona diversos controlos simples, direcionados à interação dos utilizadores com a camada da aplicação.

A camada de aplicação – *Application Tier* – é a camada intermédia e caracteriza-se como sendo a camada mais importante da plataforma. Conhecida como o “servidor” da aplicação, esta camada é a responsável por gerir e processar todos os pedidos recebidos da camada de apresentação. É nesta camada que é realizado o processamento de maior parte da lógica da aplicação. Nesta camada também é permitido o acesso aos serviços da camada de dados.

A camada de acesso aos dados – *Data Access Tier* – é responsável por aceder e interagir com os dados armazenados na base de dados. Esta é a camada de acesso ao SGBD (Furtado, 2013). Os serviços desta camada estão protegidos pelo acesso da primeira camada, ou seja, só a partir da camada de aplicação consegue-se aceder a esta camada de dados.

4.1.2. Arquitetura MVC

Para a realização do projeto foi utilizada uma arquitetura MVC, através da *framework CodeIgniter*. Esta *framework* encontra-se referida com mais detalhe no subcapítulo “4.2. Tecnologias Utilizadas”. No entanto, torna-se pertinente abordar o MVC relacionando com a arquitetura “*Three Tier*”.

MVC – *Model, View and Controller* - é um padrão de arquitetura de *software* para o desenvolvimento de aplicações *Web*. MVC divide toda a lógica da aplicação e representação de informação. Este padrão é composto por três partes: Modelos, Vistas e Controladores.

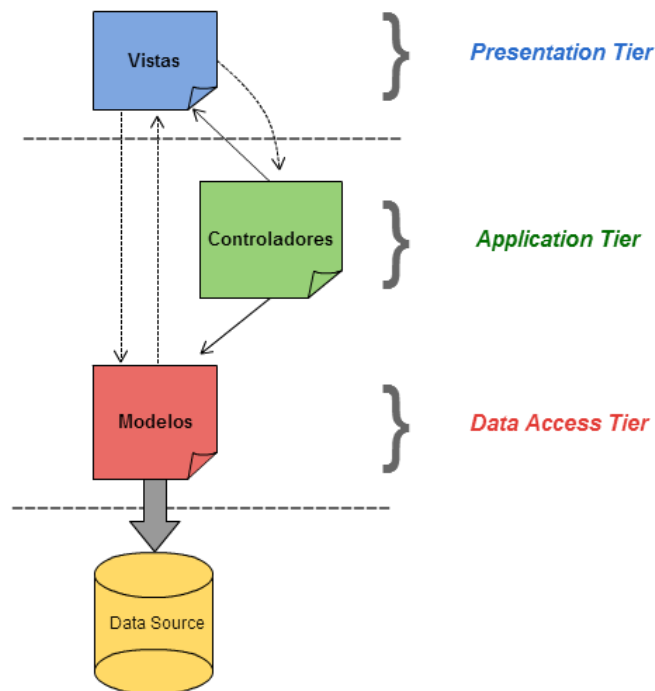


Fig. 7 - Diagrama arquitetura MVC no modelo "Three Tier"

As vistas têm a função de representar os dados do modelo para os utilizadores. Na plataforma, estes componentes comunicam com a camada de apresentação.

Os modelos definem-se como o nível mais baixo do padrão MVC. Um modelo é responsável por manter os dados da base de dados e manipulá-los assim que indicado, nomeadamente, fazer adições, eliminações, edições e consultas de dados. Estes componentes comunicam com a camada de acesso aos dados.

Os controladores são os intermediários entre os modelos e as vistas, convertendo e processando tudo o que é necessário na comunicação entres ambos. Um controlador é responsável por responder aos pedidos dos utilizadores, valida-os e executa as operações requisitadas e, por fim, comunica ao modelo para manipular os dados recebidos. Os diferentes controladores encontram-se na camada da aplicação.

Resumindo, um utilizador quando executa uma determinada ação, faz sempre um pedido a um controlador, que por sua vez trabalha com os modelos e envia os respetivos dados de resposta às vistas.

4.1.3. Arquitetura Alto-Nível

Após a compreensão das três camadas da arquitetura e o que elas englobam, torna-se mais simples entender o que acontece do lado do servidor tendo em conta os requisitos. A imagem seguinte representa o diagrama de alto-nível do sistema desenvolvido.

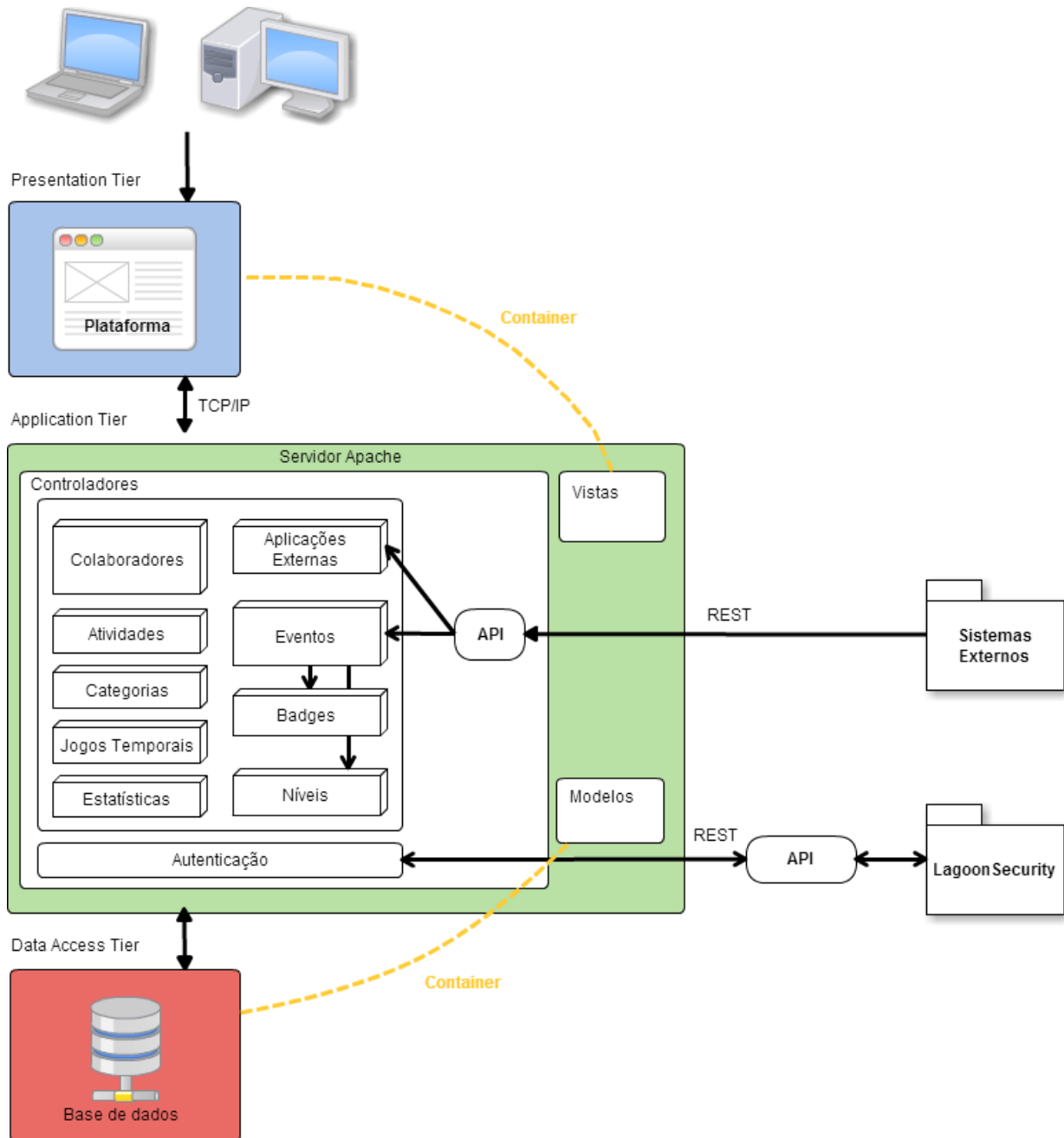


Fig. 8 - Diagrama de alto nível da arquitetura proposta

A partir do diagrama de alto-nível, encontram-se as diferentes camadas do sistema anteriormente referidas. De uma forma resumida, verifica-se:

- A camada de apresentação, onde os utilizadores interagem com a plataforma;
- A camada de aplicação, onde se encontram todos os processos lógicos do sistema;
- A camada de acesso aos dados, responsável pela comunicação com a base de dados.

Analisando o diagrama da arquitetura, verifica-se que, dentro da camada de aplicação, existe o servidor apache responsável por manter todos os componentes ativos. Ainda dentro desta camada, existem os controladores que contêm os módulos principais, enquanto as vistas e os modelos das outras camadas encontram-se vazios. Ambos se apresentam desta forma, porque no desenvolvimento do projeto foi criado um modelo para cada módulo dos controladores, não fazendo sentido colocar no diagrama módulos repetidos. As vistas encontram-se nesta camada de aplicação por, tecnicamente, estarem dentro do servidor apache, porém são uma parte da camada de *software* que indiretamente fazem parte da camada superior (camada de apresentação). O mesmo acontece para os modelos que pertencem indiretamente na camada inferior (camada de acesso aos dados). Concluindo, todos os componentes MVC encontram-se dentro do servidor apache.

Ainda no diagrama da arquitetura, pode-se verificar a existência de uma base de dados central que fornece suporte constante a todos os serviços. A base de dados encontra-se detalhada no subcapítulo “4.5. Modelo de Dados”.

API para Sistemas Externos

Tendo em conta o requisito que implica a criação de uma API, para os sistemas externos conseguirem comunicar os eventos realizados pelos colaboradores, existe dentro da camada de aplicação um controlador responsável por receber e processar os pedidos externos REST. Para o processamento dos pedidos, existe sempre comunicação dessa API com as aplicações externas para validação das permissões. Existe também comunicação com o módulo dos eventos, para

proceder-se à inserção do evento recebido no pedido externo. Consequentemente, os módulos dos *badges* e dos níveis no jogo são afetados com essa inserção.

BackOffice

Para o *backOffice*, existe uma série de controladores responsáveis pela gestão de todos os requisitos referidos no subcapítulo “3.4.2. *BackOffice*”. Como se verifica no diagrama, os diferentes módulos que se encontram dentro da caixa “controladores” têm essa função.

FrontOffice

Como se verifica no subcapítulo “3.4.3. *FrontOffice*” não existe nenhuma gestão em concreto como requisito, para além da gestão dos dados pessoais. Portanto, os controladores apenas têm a função de mostrar dados consoante os colaboradores. No entanto, todos eles relacionam-se com os módulos referidos no diagrama da arquitetura.

Autenticação

Como já existe uma aplicação legada na empresa responsável por fazer a autenticação dos seus colaboradores, no atual projeto para efetuar a entrada dos colaboradores no sistema, são efetuados diversos pedidos REST ao LS, pedindo autorização de acesso e permissões. Logicamente, antes de se proceder a este processo, existem também determinadas validações na base de dados do sistema. A partir destas permissões são criadas as sessões que mantêm o utilizador autorizado ao longo da navegação na plataforma.

4.2. Tecnologias Utilizadas

No desenvolvimento do sistema foram utilizadas diversas tecnologias, para além da utilização da ferramenta LS, indicada no subcapítulo “3.2.3. LagoonSecurity”.

4.2.1. PHP

O PHP (Praciano, 2014) é uma linguagem utilizada na criação e desenvolvimento de páginas *Web*. A utilização desta linguagem apresenta bastantes vantagens para os programadores.

O PHP é uma linguagem gratuita que permite ao programador não acarretar muitos gastos com o desenvolvimento de um novo sistema. O PHP foi projetado para ser processado em cima de Linux e Apache, sendo ambos de código aberto e livre. Apresenta grande margem de compatibilidade em diferentes sistemas e em diversas versões de *browsers*, facilitando o desenvolvimento. O PHP caracteriza-se como sendo uma linguagem robusta. Permite também desenvolver funcionalidades complexas de uma forma rápida e eficaz devido à existência de diversas classes e funções próprias.

Outra grande vantagem do uso desta linguagem é a facilidade de aprendizagem. Existe uma grande comunidade à volta do PHP, facilitando a resolução dos problemas que se vão encontrando ao longo do desenvolvimento. Além disso, existem diversas *frameworks* em PHP disponíveis, que apresentam integradas determinadas funções comuns e de fácil uso. Fornecem também suporte e configurações já organizadas, entre outras vantagens que auxiliam bastante o desenvolvimento de um novo sistema, como é o caso, como já referido, da *framework* utilizada para o presente projeto – *CodeIgniter*.

4.2.1.1. CodeIgniter

O *CodeIgniter* (ellislab.com, 2015) é uma *framework* PHP com recursos simples e completos para criar aplicações *Web*. Oferece um conjunto de bibliotecas já construídas para o desenvolvimento das tarefas mais comuns nas aplicações, evitando que seja necessário desenvolver no projeto algumas funcionalidades, bastando assim chamar as devidas bibliotecas do *codeigniter*. Além desta grande vantagem, existe uma larga comunidade à volta desta *framework*, tornando o seu uso ainda mais apropriado e fazendo com que seja ainda mais simples desenvolver com *CodeIgniter*.

Esta *framework* foi desenvolvida olhando para a programação orientada a objetos com uso à arquitetura MVC.

4.2.2. HTML5 e CSS3

O HTML5 (*Hypertext Markup Language*, versão 5) é uma linguagem de marcação utilizada para estruturar e apresentar conteúdos para a WWW. É a quinta versão da linguagem HTML. O desenvolvimento dos documentos/páginas possuem etiquetas (*tags*). Estas *tags*, definem-se como comandos de formatação da linguagem onde se pode adicionar outras propriedades que normalmente estão estruturadas em forma de árvore. O HTML5, comparando com as versões anteriores, facilita com mais eficácia a manipulação dos elementos. Algumas *tags* foram alteradas, outras criadas e outras descontinuadas. As versões anteriores do HTML não eram padronizadas para criação de seções comuns e específicas, como rodapé, cabeçalho, menus, etc. A forma como se desenvolve o código-fonte e se organizam as páginas também foi modificada. As páginas passaram a ser mais semânticas e com menos código-fonte associado, melhorando assim a performance das aplicações, entre outras vantagens.

O CSS (*Cascading Style Sheets*) é uma linguagem de estilo utilizada para organizar e definir a apresentação das páginas e/ou documentos escritos numa linguagem de marcação, como o HTML. A maior vantagem estabelece-se na separação entre a formatação e os conteúdos dos documentos, pois em vez de estar tudo num só documento, cria-se uma ligação para outra página que contém todos os outros estilos, e faz-se o mesmo processo para as restantes páginas. Deste modo, quando se deseja alterar alguma formatação é só necessário alterar um único documento.

O CSS3 é a última versão do CSS. O CSS3 trouxe novos estilos para as páginas da *Web 2.0* (Bookwalter, 2011) em todos os aspetos de *design* do *layout*, nomeadamente, na abolição de imagens de fundo, as transições e efeitos de criação de animações, as bordas arredondadas, entre outros.

Uma grande vantagem da utilização do HTML5 e do CSS3 é o facto de tornarem as páginas desenvolvidas preparadas para diferentes tipos de dispositivos.

4.2.2.1. *BootStrap*

O BootStrap (getbootstrap.com, 2015) é uma *framework* desenvolvida já com uso de HTML5 e CSS3. No desenvolvimento de uma aplicação *Web*, a utilização desta *framework* é bastante vantajosa pois permite a um programador reduzir tempo relativamente à programação de *frontend*, permitindo desta forma dar mais importância ao desenvolvimento de funcionalidades internas da aplicação.

Quando se desenvolve uma aplicação, a parte de *interface* e usabilidade são sempre dois aspetos relevantes. O BootStrap ajuda bastante os programadores nesta vertente, pois tem incorporado uma série de formatações já constituídas, como por exemplo, botões, tipos de letras, cores, etc. É muito simples aplicar as formatações disponíveis nos elementos de um projeto e, a qualquer altura o programador pode alterar essas formatações.

Resumindo, a utilização desta *framework* no desenvolvimento de uma aplicação é interessante, simples e bastante vantajosa.

4.2.3. *JavaScript*

O *JavaScript* (javascript.com, 2015) é uma linguagem de *script* e é atualmente a principal linguagem de programação cliente-servidor de páginas *Web*. Foi desenvolvida para ser uma linguagem *script* orientada a objetos baseada em protótipos e funções de primeira classe. Assim sendo, faz suporte a programações funcionais e apresenta recursos em linguagens mais populares como o Java e o C++.

Inicialmente, foi implementado com o objetivo dos *scripts* puderem ser executados do lado do cliente e interajam com o utilizador sem ser necessário o *script* ir ao servidor, controlando apenas o *browser* e atualizando todas as alterações no documento desejado.

4.2.4. AJAX

O Ajax (*Asynchronous JavaScript e XML*) (jquery.com, 2015), embora na teoria não seja uma tecnologia, é um termo que abrange o uso de diversas tecnologias em simultâneo, nomeadamente, HTML, CSS, *JavaScript*, XML, XMLHttpRequest, entre outros. O Ajax faz chamadas ao servidor de modo a obter informações rapidamente que permite fazer determinadas atualizações sem que as páginas *Web* sejam recarregadas do lado dos utilizadores.

4.2.5. MySQL

A base de dados do presente projeto foi criada com MySQL. MySQL é um sistema de gestão de base de dados (SGBD) (mysql.com, 2015) que utiliza a linguagem SQL (*Structured Query Language*) como *interface*. Este SGBD continua a ser um dos mais utilizados na criação e administração de bases de dados para aplicações *Web*. O seu uso apresenta diversas vantagens, nomeadamente, o facto de ser gratuito; compatível por muitas linguagens de programação; portabilidade; rápido e confiável.

4.2.6. Outros

Além das tecnologias referidas, no desenvolvimento do projeto foram utilizadas algumas aplicações livres:

- XAMPP v3.2.1

É utilizado para simular o servidor da plataforma. XAMPP (apachefriends.org, sem data) é um servidor *Web* livre, composto por base de dados MySQL, servidor *Web* Apache e interpretadores para linguagens PHP e Perl. Com estes componentes surge então o nome: X - por funcionar em qualquer sistema operativo; A –Apache; M – MySQL, P – PHP, P – Perl. XAMPP é utilizado para servir *sites Web online*, como o caso da plataforma.

- **REST Console e Postman – REST Client**

Aplicações disponíveis *online* que permitem efetuar pedidos REST (getpostman.com, sem data) (restconsole.com, sem data). Ambas as aplicações apresentam um *interface* simples e de fácil utilização. Oferecem funcionalidades como a inserção do URL da API destino; a escolha do tipo de pedido (GET, POST, PUT, DELETE); a autenticação para aceder ao destino se for necessário; a escolha do tipo de corpo do pedido (JSON, XML, etc.) bem como a inserção dos dados no corpo do mesmo; entre muitas outras funcionalidades relacionadas a pedidos REST.

- **Mozilla OpenBadges**

Software livre que proporciona, a qualquer organização, técnicas de criar, emitir e verificar *badges online*. Mozilla OpenBadges (openbadges.org, sem data) disponibiliza uma API externa que permite comunicar uma série de atividades. Uma funcionalidade dessa API, nomeadamente, uma ferramenta para desenhar *badges* foi utilizada no desenvolvimento do presente projeto. Esta API apresenta muitas funcionalidades pertinentes, no entanto as suas implementações não se tornaram prioritárias pois seria necessário que todos os *badges* no sistema fossem criados dentro deste *software*. Contudo, como indicado nos requisitos funcionais, isso não é um objetivo.

- **LinkedIn's Javascript SDK**

API que permite integrar na rede social do LinkedIn acessos e partilhas de informações a partir de outras aplicações externas. Esta API foi utilizada para permitir a partilha de informação quando um utilizador ganha um determinado badge (developer.linkedin.com, 2015).

4.3. Casos de Uso

Casos de uso são “narrativas” que descrevem uma determinada unidade funcional de um sistema. Um diagrama de casos de uso serve para representar esses casos de uso, os atores (possíveis utilizadores) e a relação existente entre ambos. No fundo, um diagrama deste tipo tem a finalidade de representar os requisitos funcionais, aos olhos dos utilizadores. O diagrama indica para que serve o sistema sem referir os métodos ou funcionalidades internas.

Como referido no subcapítulo “3.3. Identificação dos Utilizadores” existem três tipos de utilizadores no presente sistema: Colaboradores do tipo administradores; Colaboradores normais e utilizadores externos, que não sendo diretamente uma pessoa, interagem com o sistema via API. Para cada tipo de utilizador representa-se um ator de caso de uso. Além destes três atores, existe o LS que embora não se apresente como um utilizador do sistema, é responsável pela parte da gestão de acessos e permissões, citando-se também como um ator para os casos de uso.

Subentendendo dos capítulos anteriores, o presente projeto divide-se em quatro componentes: API para sistemas externos; *backOffice*; *frontOffice* e comunicação com o LS. A estrutura do sistema, como se verifica no subcapítulo anterior “4.1.3 Arquitetura Alto-Nível”, encontra-se dividida por diversos módulos, de modo a facilitar a compreensão do sistema e a tornar a implementação do mesmo mais simples. Todos os módulos indicados encontram-se dentro dos quatro componentes referidos. A construção de diagramas de casos de uso tornou-se importante no processo de planeamento para simplificar o entendimento dos diversos módulos e a relação com cada ator. A figura seguinte apresenta o diagrama de casos de uso geral do sistema dividido por diferentes módulos, ligados com os respetivos atores. Cada um dos módulos representa uma parte ou uma gestão importante dos quatro componentes do projeto.

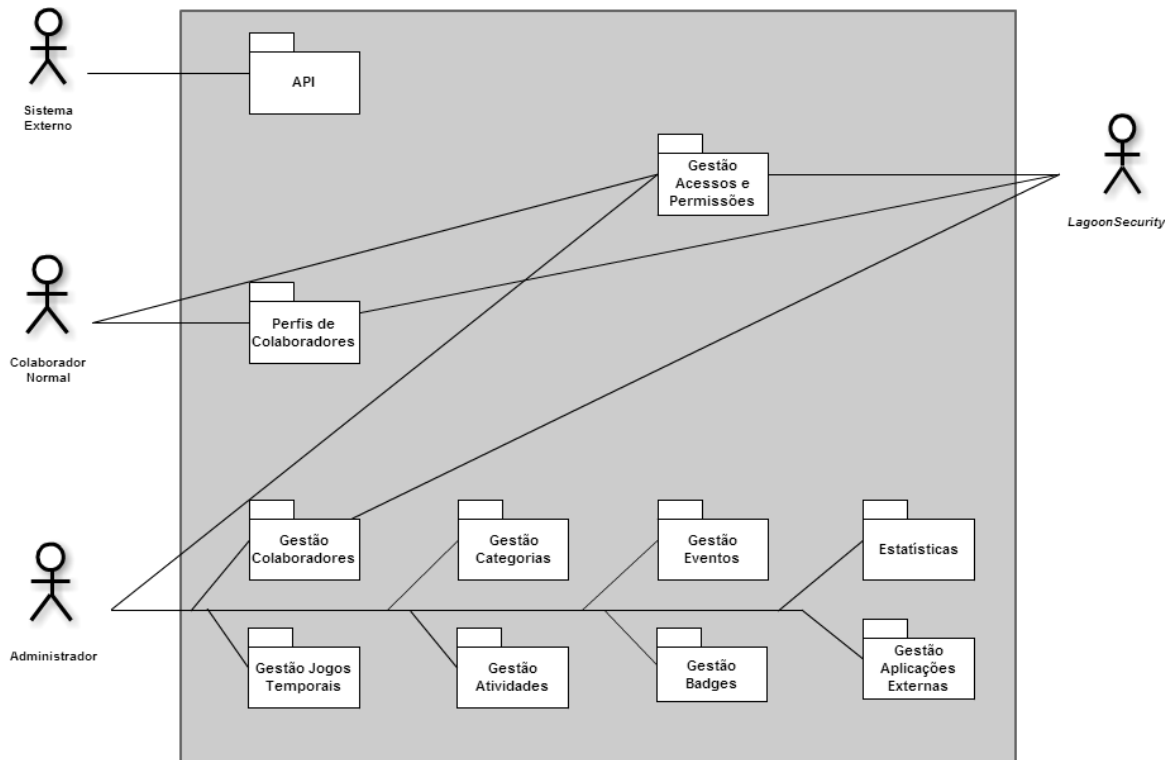


Fig. 9 - Diagrama geral de módulos para os casos de uso

A partir do diagrama, pode-se verificar que:

- O ator “Sistema Externo” apenas se liga com o módulo da API.
- O ator “LagoonSecurity” é responsável pelo módulo de gestão de acessos e permissões. Além disso, também se interliga com o módulo de perfis de colaboradores, pois nenhum ator “Colaborador Normal” tem acesso a este módulo se o ator “LagoonSecurity” não autorizar e, igualmente acontece para as restantes gestões disponíveis nos restantes módulos para o ator “Colaborador Administrador”. No entanto, no digrama deu-se enfase ao relacionamento do “LagoonSecurity” com o módulo de gestão de colaboradores, pois quando um administrador adiciona um novo colaborador, edita dados ou elimina o próprio colaborador, estas ações interagem diretamente com o LagoonSecurity.
- O ator “Colaborador Normal” tem ação com o módulo de perfis de colaboradores, responsável pela parte do *frontOffice*.

- Por fim, o ator “Colaborador Administrador” tem ação com os restantes módulos de gestão do sistema.

Os digramas de casos de uso que explicam os diferentes módulos referidos encontram-se no anexo A - Casos de Uso.

4.4. Diagramas de Atividades

Um diagrama de atividades, tal como o nome indica, tem a função de representar todo um fluxo que ocorre no sistema, com a realização de uma determinada atividade. Estes diagramas representam, detalhadamente, os fluxos de trabalho existentes no funcionamento do sistema. Enquanto os diagramas de casos de uso, ou outros diagramas, dão ênfase ao fluxo do controlo de um objeto para outro, os diagramas de atividades preocupam-se em realçar o fluxo de controlo de uma atividade para a outra.

Não querendo inferiorizar a importância de outros diagramas de modelação UML (Janssen, 2010), para a presente dissertação deu-se relevância aos diagramas de atividades, pois além de permitirem detalhar as operações que um ator pode fazer, detalham também todo o processamento paralelo realizado pelo servidor. O presente subcapítulo indica os diagramas de atividades com mais importância no sistema.

4.4.1. Diagrama de Atividades da API

Para o módulo da API, existe apenas uma grande vertente que é a inserção de eventos existentes em outros sistemas externos à plataforma. O diagrama seguinte demonstra detalhadamente o que ocorre entre uma aplicação externa e o sistema de *gamification*.

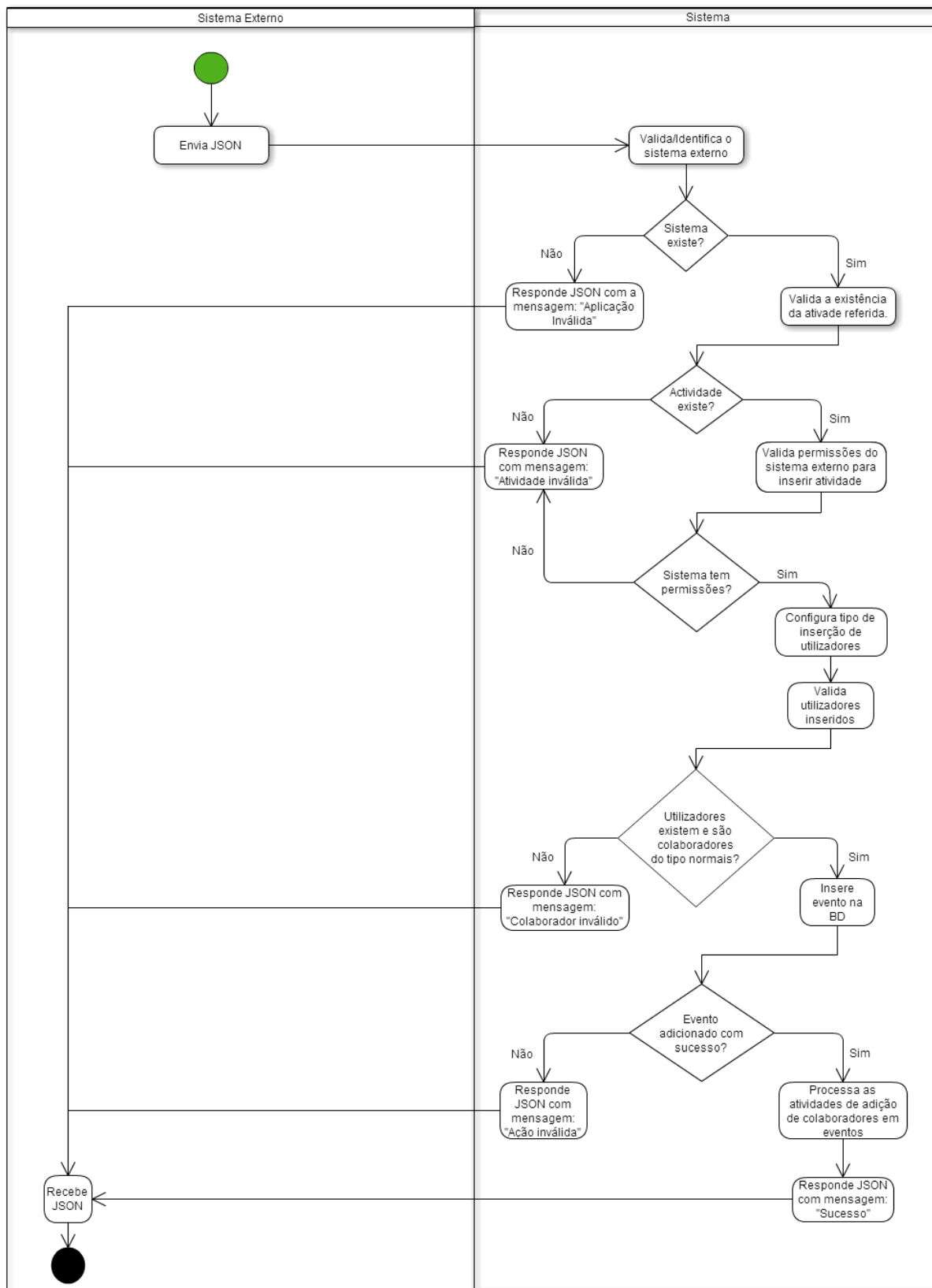


Fig. 10 - Diagrama de atividade da API

4.4.2. Diagrama de Atividades para Gestão de Acessos e Permissões

Para o módulo da gestão de acessos e permissões, o principal interveniente neste âmbito é o LS, no entanto o sistema também executa determinadas validações e funções internas necessárias neste processo. O diagrama seguinte corresponde à atividade de autenticação, quando um determinado utilizador se autentica na plataforma.

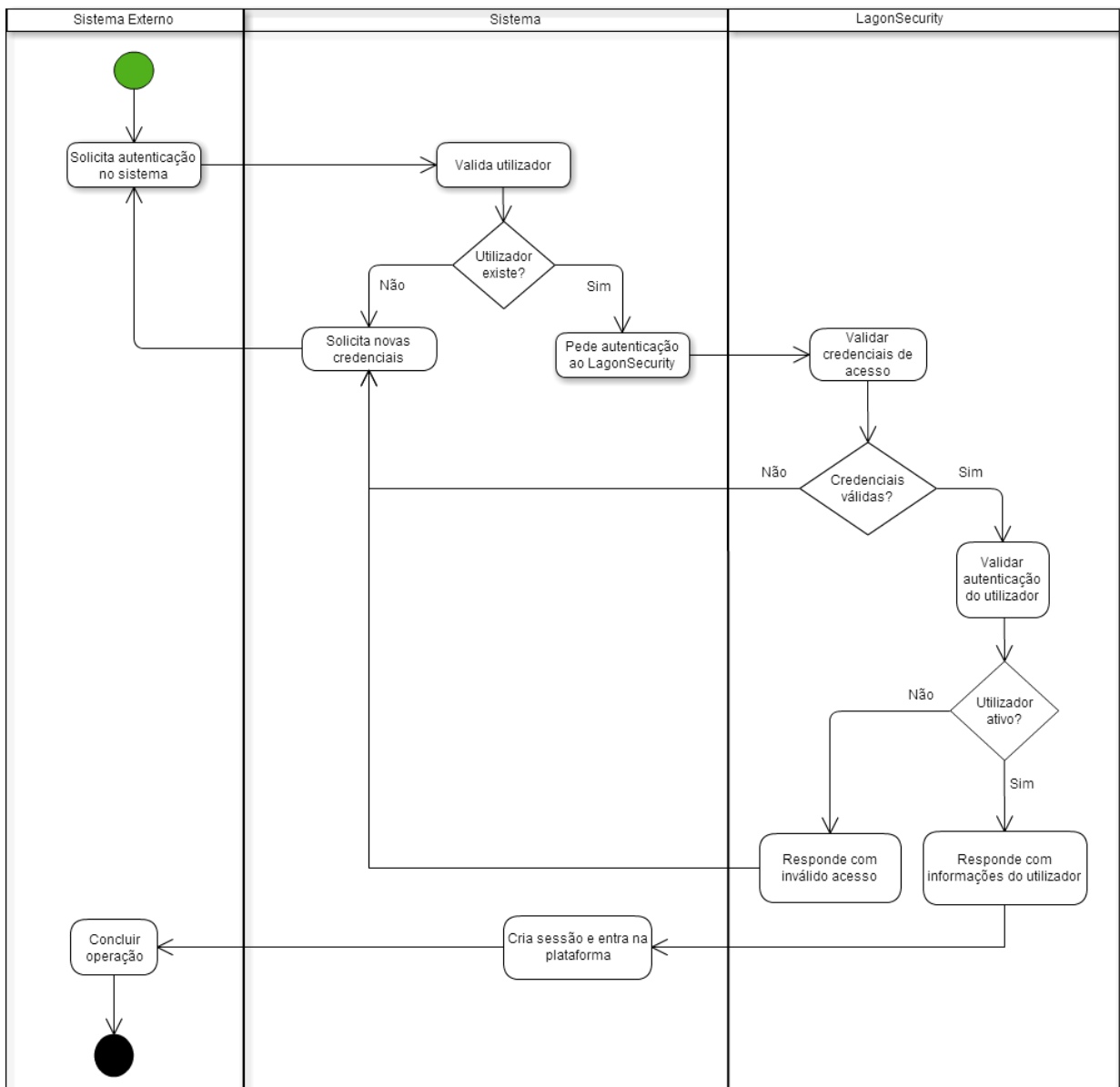


Fig. 11 - Diagrama de atividade de autenticação

4.4.3. Diagrama de Atividades para a Gestão de Colaboradores

Para o módulo da gestão de colaboradores, existem vários intervenientes. O (s) administrador (es) da plataforma de *gamification* solicita (m) os diversos pedidos. O sistema tem a função de processar todos os pedidos recebidos do utilizador e comunicar com o LS. Este sistema tem uma vertente importante neste módulo, pois o sistema não consegue adicionar novos colaboradores nem eliminá-los sem comunicar e receber validações do LS.

A adição de um novo colaborador requer comunicação com o LS. Todos os colaboradores, registados no sistema de *gamification*, têm obrigatoriamente de estar também inseridos no LS, juntamente com determinados dados, como por exemplo, o nome de utilizador (*username*), a senha e o *e-mail*, podendo assim o LS proceder ao envio de *e-mails* para os novos utilizadores, permitindo aos mesmos autenticarem-se na plataforma e terem permissões de acesso e navegação nas diferentes funcionalidades.

Quando são solicitadas alterações de dados é necessário validar se esses dados também existem no LS e, logicamente, se existem devem igualmente ser atualizados.

Quando um colaborador é eliminado do sistema de *gamification*, torna-se desnecessário o LS manter os dados desse colaborador, no ambiente. Portanto, a comunicação entre ambos continua a ser obrigatória no processo de eliminação de colaboradores no sistema.

O diagrama seguinte demonstra, detalhadamente, o que ocorre entre o utilizador, o sistema e o LS na gestão dos colaboradores, nomeadamente, nas configurações dos mesmos, ou seja, no processo de adição de um novo colaborador, edição de dados e eliminação dos mesmos.

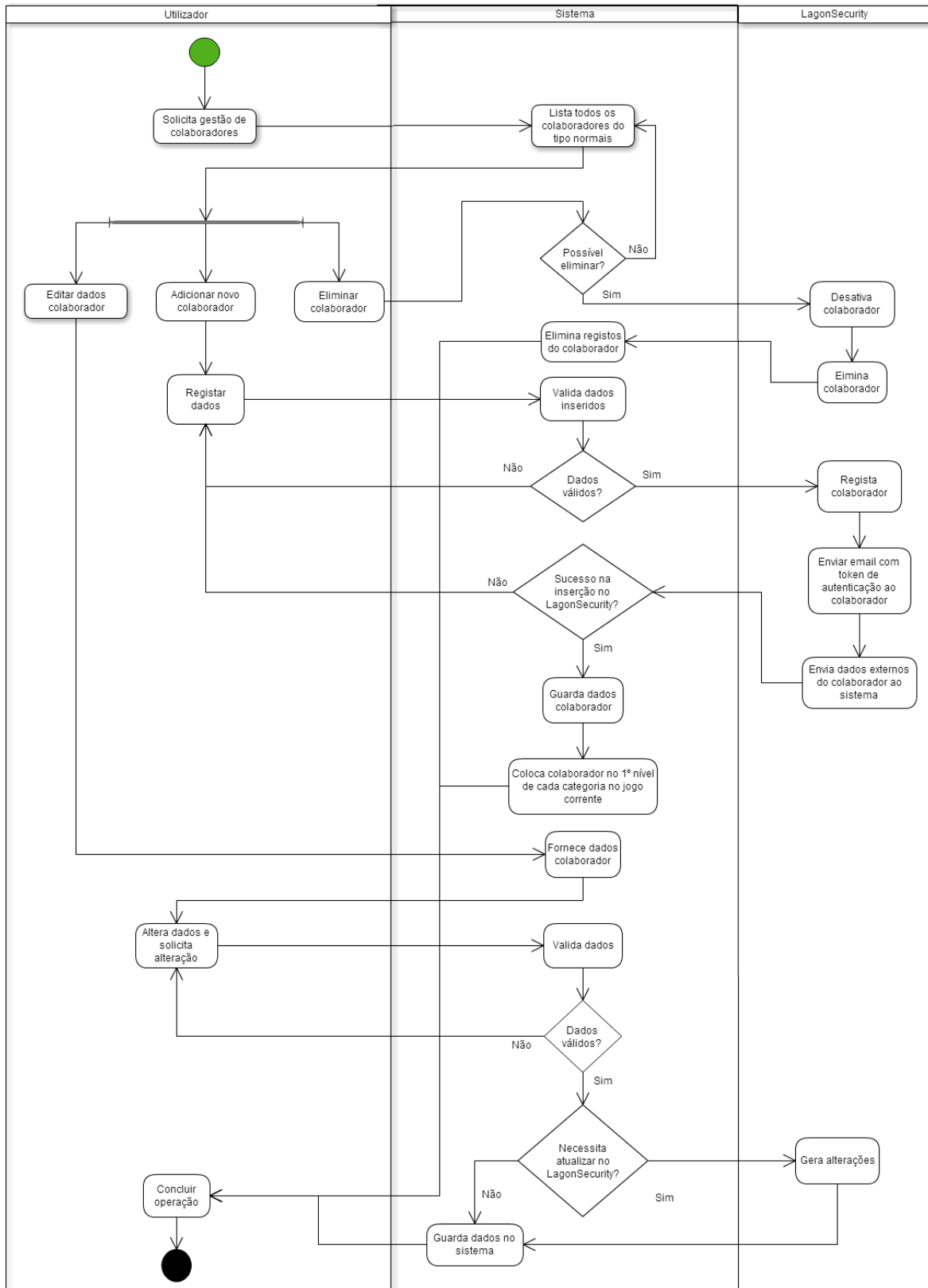


Fig. 12 - Diagrama de atividades para gestão de colaboradores

4.4.4. Diagramas de Atividades para a Gestão de Eventos

Como se pode verificar ao longo da presente dissertação, os eventos são a componente mais importante neste projeto de *gamification*, pois é através da realização dos mesmos, que os colaboradores conseguem alcançar objetivos ao longo dos jogos temporais.

Os eventos apresentam-se como uma área que quando existem ações, muitas outras áreas do sistema são envolvidas. Quando ocorre uma inserção ou remoção de um evento, existe interferência com as pontuações de cada colaborador que realizou o evento e, conseqüentemente, com a atribuição ou remoção de *badges* automáticos.

O sistema executa uma série de operações na gestão de eventos. Os diagramas seguintes demonstram, detalhadamente, o que ocorre entre o utilizador e o sistema. Embora existam mais detalhes e funcionalidades na plataforma sobre a gestão de eventos, deu-se ênfase aos pormenores mais relevantes no âmbito de aplicabilidade de *gamification* no projeto.

4.4.4.1. Gestão Eventos por Colaborador

A plataforma permite ao (s) administrador (es) gerir (em) os eventos de duas formas distintas:

- Focando-se nos eventos, ou seja, foca-se no evento e só depois gere os colaboradores que o realizam;
- Focando-se nos colaboradores, isto é, primeiro foca-se no colaborador principal e só depois gere os eventos desse colaborador.

O método que sucedeu mais interesse por parte da empresa foi o segundo. Portanto, o diagrama seguinte detalha os diferentes processos na gestão de eventos através do segundo método.

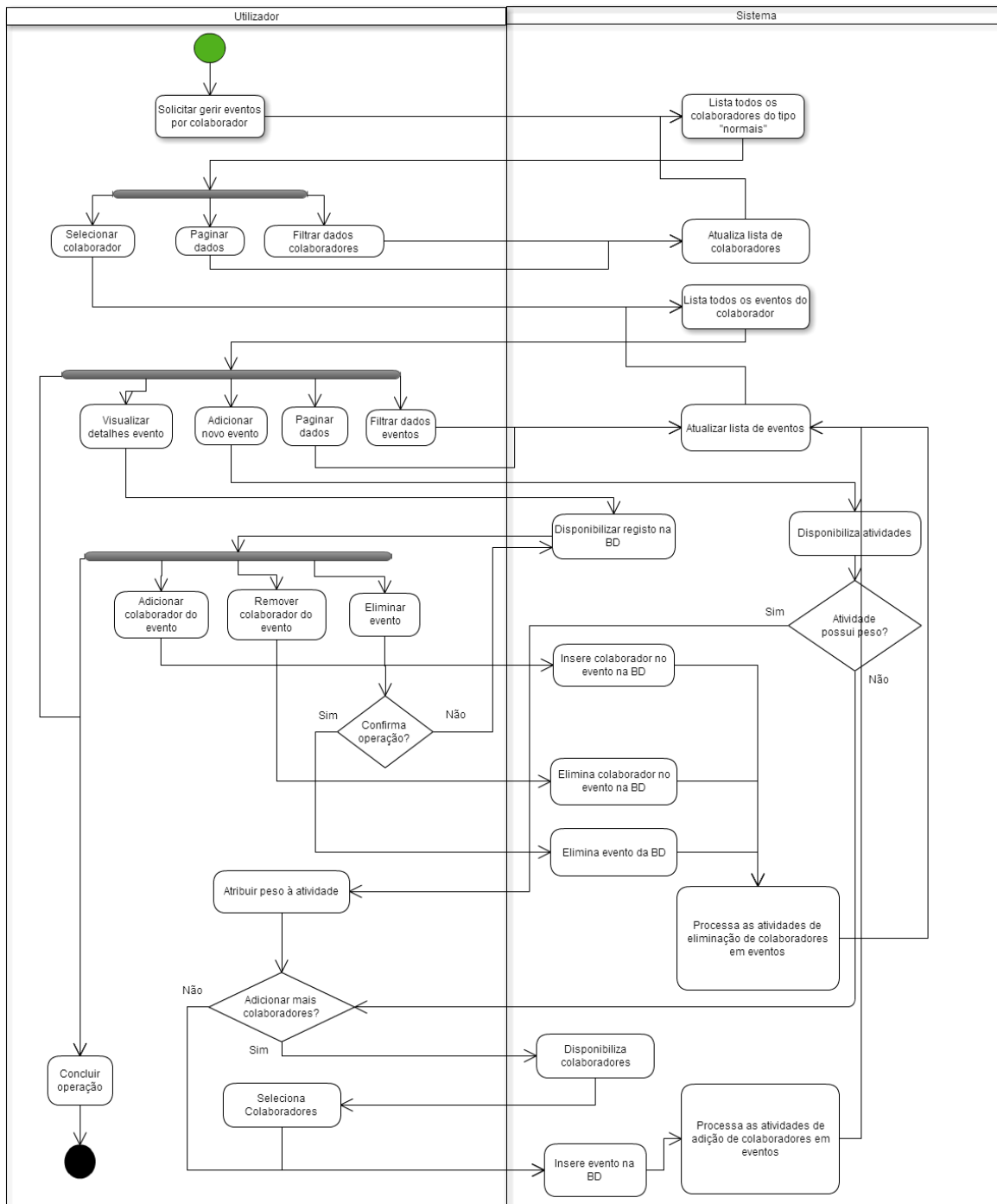


Fig. 13 - Diagrama de atividade de gestão de eventos por colaborador

A partir do diagrama, pode-se verificar duas componentes gerais: “Processa as atividades de eliminação de colaborador em evento” e “Processa as atividades de adição de colaborador em evento”, que se apresentam seguidamente.

4.4.4.2. Adição de Colaborador em Evento

O diagrama seguinte descreve as variantes que acontecem quando um determinado colaborador realiza um evento. As pontuações, os níveis e os *badges* são os elementos que podem variar com esta ação.

O processo de atribuição automática de *badges* também se verifica no seguinte diagrama.

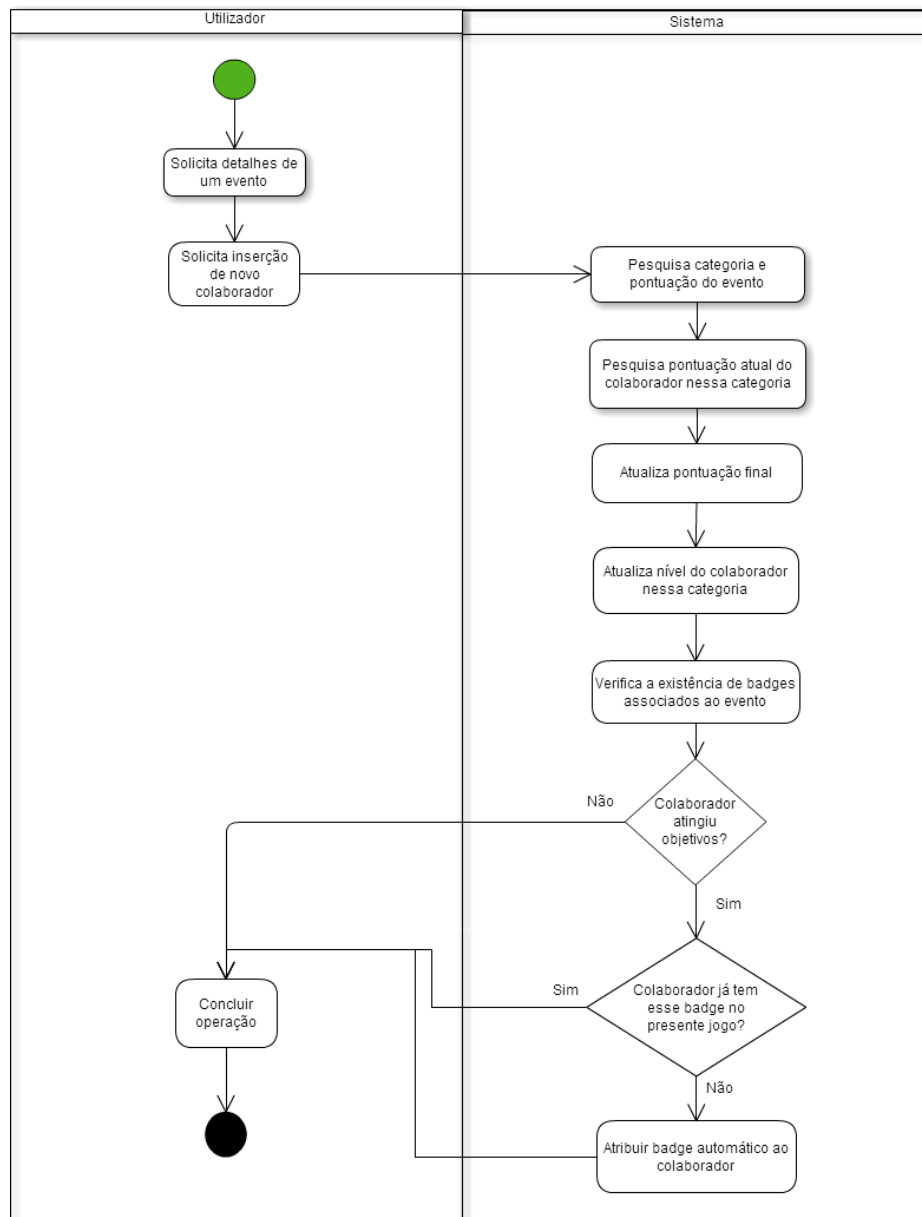


Fig. 14 - Diagrama de atividade de inserção de colaborador em evento

4.4.4.3. Eliminação de Colaborador em Evento

O próximo diagrama descreve as variantes que acontecem quando um determinado colaborador é eliminado de um evento. Igualmente ao diagrama anterior da adição de um colaborador num evento, as pontuações, os níveis e os *badges* são os elementos que podem variar com esta ação.

O processo de eliminação automática de *badges* também se verifica no seguinte diagrama.

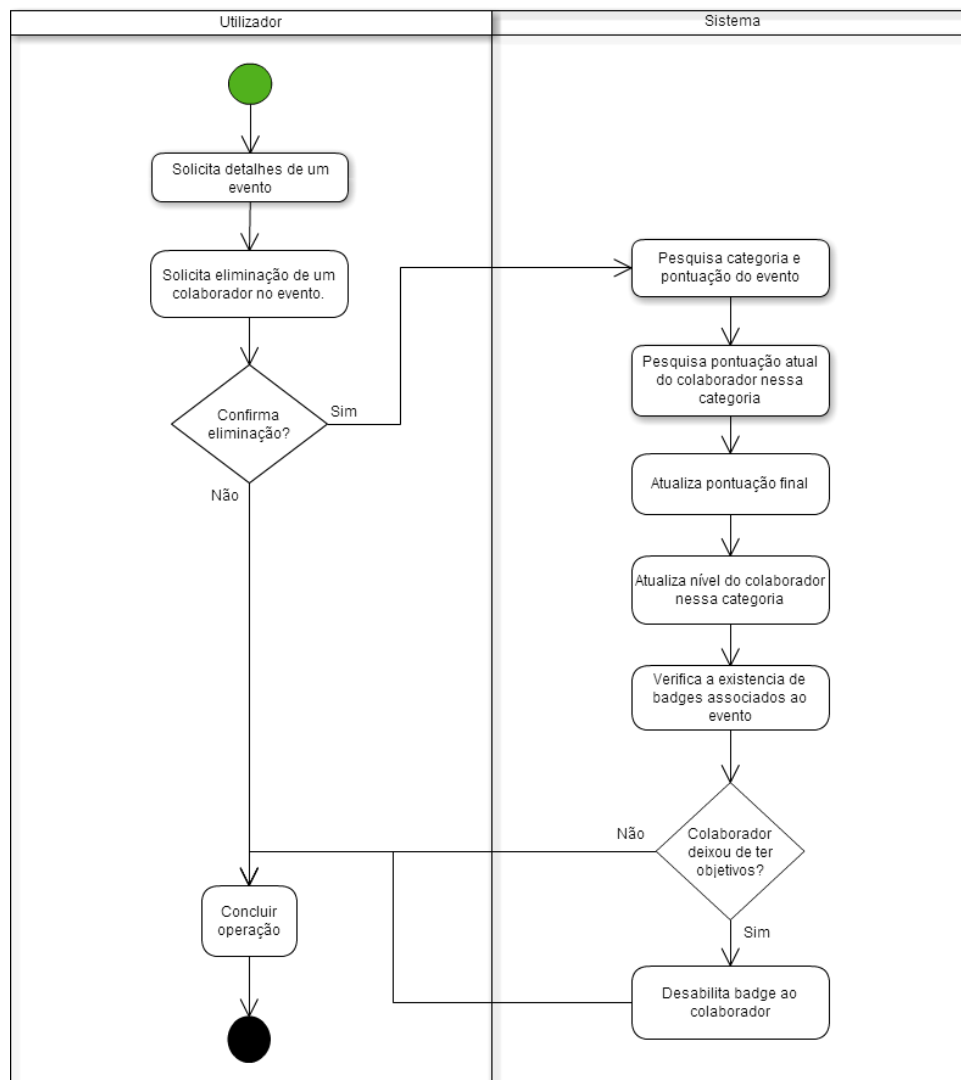


Fig. 15 - Diagrama de atividade de eliminação de colaborador em evento

4.5. Modelo de Dados




Como já referido, existe uma base de dados que dá suporte constante ao sistema. A modelação dos dados é uma componente muito importante para o funcionamento do sistema, desde ao nível de lógica à performance do mesmo.

No presente subcapítulo pretende-se transmitir a base de conhecimento das informações presentes no sistema, de modo a possibilitar uma melhor compreensão do funcionamento interno do mesmo. Além disso, pretende-se com este subcapítulo facilitar eventuais manutenções ou futuras atualizações do projeto.

4.5.1. Modelo Entidade-Relação

O modelo entidade-relação, tal como o nome indica, identifica todas as entidades existentes no sistema e expõe as relações existentes entre elas. Esta relação entre as diversas entidades podem ter diversas tipologias, a nível de obrigatoriedade e uso de dados.

A figura seguinte representa o modelo entidade-relação utilizado no sistema e, como se verifica, as relações entre as diversas entidades possuem diferentes tipologias. De seguida, apresentam-se as diferentes tipologias utilizadas no diagrama do projeto bem como a explicação das mesmas:

- Nenhum ou muitos registos 
- Obrigatoriamente 1 e 1 registo 
- Nenhum ou 1 registo 

Aproveitando ainda o modelo, por exemplo, a relação existente entre a entidade “Categorias” e a entidade “Atividades”, lê-se:

- Uma categoria pode ter zero ou muitas atividades;
- Uma atividade corresponde obrigatoriamente a uma e uma só categoria.

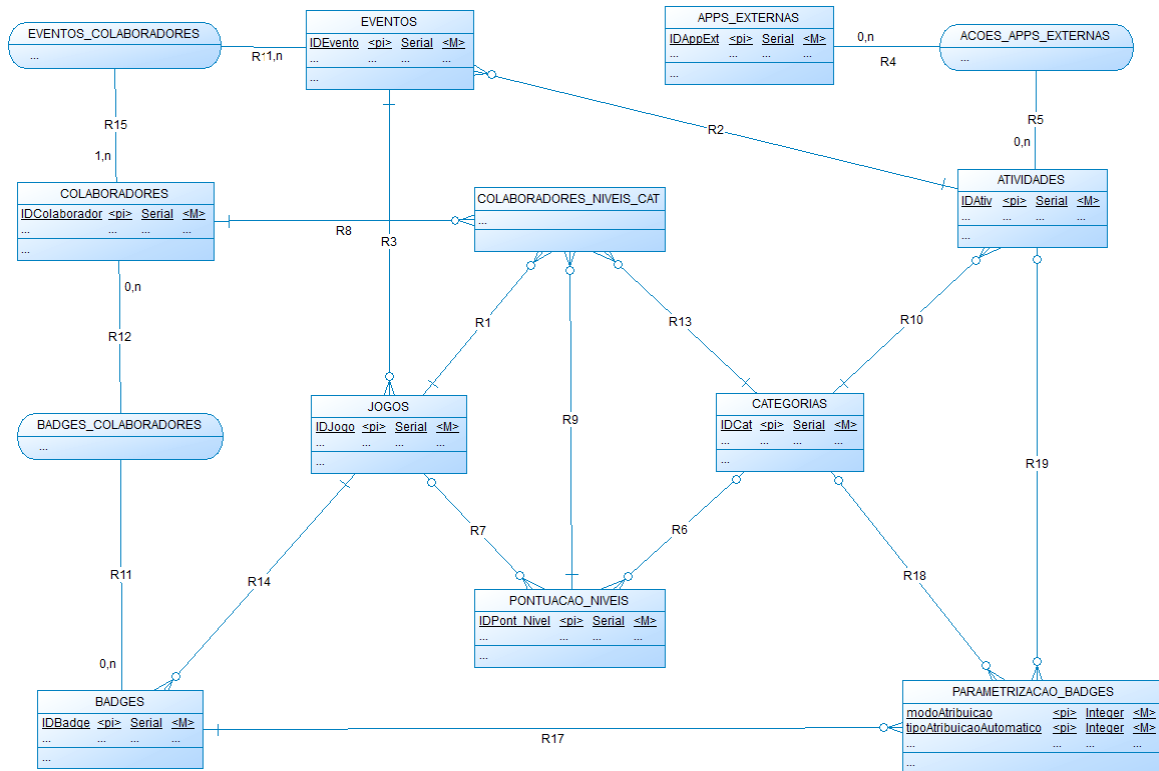


Fig. 16 - Modelo Entidade-Relação

No anexo B, encontra-se o modelo físico de dados com todas as entidades, compostas pelos diversos atributos e chaves que as integram.

4.5.2. Entidades

Para uma melhor compreensão do modelo de dados, anteriormente referido, torna-se importante perceber a importância principal de cada entidade. O presente subcapítulo apresenta uma descrição de cada tabela existente no diagrama. Para mais detalhe de cada tabela, pode-se verificar no anexo B – *Modelo físico de dados* - todas as tabelas com referência a cada campo que as constituem, bem como os tipos de dados e a obrigatoriedade de dados.

▪ Colaboradores

A tabela “colaboradores”, associada ao módulo de gestão de colaboradores/utilizadores, é responsável por guardar todos os dados referentes a cada colaborador/utilizador do sistema. Com base nos dados guardados nesta tabela, os utilizadores podem entrar nas diferentes partes do sistema. Além disso, é através destes registos que se torna possível adicionar eventos e atribuir *badges*, ou seja, a aplicação de *gamification* destina-se aos colaboradores registados nesta tabela.

O campo “IDColaborador” é o identificador único de cada colaborador, ou seja, representa-se como a chave primária da tabela. Existe um campo “IDExterno” responsável por interligar o colaborador ao registo do mesmo no LS. Outro campo essencial para a entrada em sistema e que se relaciona com esta plataforma é o campo “flagActivate”. Este campo, tal como o nome indica, guarda informação se o utilizador está ativo ou inativo. Caso o campo se apresente como verdadeiro então o utilizador encontra-se ativo no LS, tendo permissões de acesso no sistema, caso contrário, o utilizador não consegue autenticar-se na plataforma. Quando um colaborador é adicionado ao sistema, esta *flag* só muda para verdadeiro após o mesmo colaborador se autenticar pela primeira vez através de um *link* de acesso que é redirecionado automaticamente para o *e-mail* do mesmo.

Esta tabela guarda o *username* do utilizador, para validar a existência do mesmo antes de se conectar com o LS e para permitir a recuperação de senhas. Este campo, também serve para identificar utilizadores que são recebidos pela API, se a aplicação externa estiver configurada para tal. Igualmente, o *e-mail* apresenta a mesma finalidade, e além disso, também é essencial para o envio de *e-mails* quando necessário perante a conexão com o LS. O campo “senha” serve para o sistema conseguir comunicar com o LS, quando o utilizador deseja alterar a sua senha de acesso.

O campo “tipoUser” surgiu na necessidade de distinguir internamente os utilizadores do tipo normais dos administradores, para além das permissões geridas através do LS.

O campo “numColaborador” é um número interno que o colaborador representa na empresa. Esta informação tem de ser inserida manualmente pelo administrador e não podem existir números iguais no sistema.

Exceto o “nome” e o “clienteAtual” que são essenciais para facilitar a distinção entre os colaboradores, os restantes campos não são de preenchimento obrigatório. No entanto, não evidencia que sejam irrelevantes, pois todos os dados guardados são importantes para o sistema.

▪ Jogos

A tabela “jogos” associa-se a todo o âmbito da plataforma e ao módulo de gestão de jogos temporais. Como já referido anteriormente, todo o contexto da plataforma, tanto o *interface* do *frontOffice* como do *backOffice*, refere-se apenas a um jogo temporal corrente. Esta tabela guarda todas as informações dos diferentes jogos temporais ao longo do tempo.

O campo “dataInicioJogo” é de preenchimento obrigatório quando inserido no sistema e mantido como estado “em aberto”. Quando um jogo é inserido como “fechado” ou quando se procede ao encerramento de um jogo então o campo “dataFimJogo” também se torna obrigatório. O campo “estadoJogo” é o responsável por guardar em que estado se encontra ao jogo temporal. De modo a limitar o âmbito do sistema, apenas se pode encontrar um jogo temporal em estado “aberto”.

O campo “IDJogo” é a chave primária da tabela “jogos”.

▪ Categorias

A tabela “categorias”, associada ao módulo de gestão de categorias, guarda apenas a informação de todas as categorias existentes atualmente no sistema. Esta tabela serve para agrupar os diferentes eventos que os colaboradores vão realizando ao longo do tempo.

O campo “IDCat” é a chave primária da tabela “categorias”.

Uma nota importante foi a decisão/requisito de as categorias não necessitarem de se relacionar com os jogos, de modo a evitar uma nova gestão de categorias, atividades e níveis por cada vez que se inicia um novo jogo. Além disso, apenas se pretende que os colaboradores consigam visualizar o seu histórico no jogo corrente, não sendo necessário este relacionamento.

▪ **Atividades**

A tabela “atividades”, associada ao módulo de gestão de atividades, guarda as informações de todas as atividades existentes atualmente no sistema. Cada atividade tem de estar relacionada com uma categoria, fazendo com que haja obrigatoriamente uma chave estrangeira da tabela “categorias”.

Cada atividade tem uma pontuação associada, portanto o campo “pontuacao” é de preenchimento obrigatório. Além disso, pode ter também um peso associado ou não. Por exemplo, uma atividade que seja a inserção de uma ideia na plataforma de ideias, não fará sentido a atividade ter um peso associado. Contrariamente, se a atividade é uma formação, o número de horas pode variar e então fará sentido ter um peso associado. Desta forma o campo “peso” apenas guarda uma variável para verificar a existência de peso, o campo “descricaoPeso” serve para identificar qual o peso, os campos “valorMin” e “valorMax” guardam o valor mínimo e máximo do peso que a atividade pode ter, por exemplo, uma e cinco horas.

O campo “IDAtiv” é a chave primária da tabela “atividades”.

▪ **Eventos**

A tabela “eventos”, associada ao módulo de gestão de eventos, tem a responsabilidade de guardar as informações quando uma atividade é realizada num determinado jogo temporal. Contudo, existem duas chaves estrangeiras para as tabelas “atividades” e “jogos”. Esta tabela

guarda, obrigatoriamente, a data da realização da atividade, o valor do peso se assim o tiver, bem como a pontuação final obtida.

O campo “IDEvento” guarda o identificador de cada evento, ou seja, é a chave primária da tabela “eventos”.

▪ Eventos_Colaboradores

A tabela “eventos_colaboradores” está também associada ao módulo de eventos e surge na necessidade de identificar qual ou quais os colaboradores que realizaram um determinado evento. Normalmente, a pontuação atribuída a cada colaborador é igual, no entanto pode surgir a necessidade de atribuir pontuações diferentes. Para isso, o campo “pontuacaoObtida” guarda qual a pontuação adquirida por cada colaborador que realizou um determinado evento.

Os campos “IDColaborador” e “IDEvento” são chaves estrangeiras das tabelas “colaboradores” e “eventos” e, simultaneamente, caracterizam-se como sendo a chave primária composta da tabela “eventos_colaboradores”.

▪ Pontuacao_Niveis

A tabela “pontuacao_niveis”, associada ao módulo de gestão de categorias e de jogos temporais, surge na necessidade de atribuir níveis às diferentes categorias e aos diferentes jogos temporais. O facto de estes componentes conterem diferentes níveis associados, segundo as características de *gamification*, a motivação dos colaboradores é estimulada para a realização de novos eventos, de modo a alcançarem novos níveis. Com isto, para cada categoria pode existir um número indeterminado de níveis, consoante o desejado pelos administradores da plataforma. O mesmo ocorre para cada jogo temporal. O campo “nivel” guarda o nome de cada nível. O campo “pontuacaoMin” indica qual a pontuação mínima que um utilizador tem de alcançar obrigatoriamente para atingir o nível em causa. O campo “imagem” só se associa aos jogos temporais, isto é, apenas os níveis de cada jogo podem ter

imagens associadas se assim o desejarem. Desta forma, cada colaborador poderá ser uma “personagem” ao longo do jogo.

Os campos “IDCat” e “IDJogo” são chaves estrangeiras das tabelas “categorias” e “jogos” e na teoria não se caracterizam como sendo de preenchimento obrigatório, pois cada registo associa-se apenas a um tipo individualmente. No entanto, quando um administrador faz a gestão dos níveis, o sistema associa sempre a uma categoria ou a um jogo, tornando estes campos obrigatórios na prática.

O campo “IDPont_Nivel” é o identificador de cada nível, sendo a chave primária da tabela “pontuacao_niveis”.

▪ Colaboradores_Niveis_Cat

A tabela “colaboradores_niveis_cat”, associada ao módulo de gestão de categorias, surge com a finalidade do sistema não sobrecarregar cada vez que tem de indicar a pontuação atual e os diferentes níveis em que se encontra cada colaborador. Esta tabela tem a função de indicar os diferentes níveis de categorias onde se encontra cada colaborador, no presente jogo temporal, guardando igualmente em histórico para os restantes jogos.

Os campos “IDColaborador”, “IDPont_Nivel”, “IDJogo” e “IDCat” são chaves estrangeiras das tabelas “colaboradores”, “pontuacao_niveis”, “jogos” e “categorias”. Cada colaborador apenas pode presenciar-se num nível por categoria num jogo, tornando o “IDColaborador”, “IDJogo” e “IDCat” a chave primária composta da tabela “colaboradores_niveis_cat”.

Embora através do “IDPont_Nivel” seja possível identificar o “IDCat”, é necessário este campo existir nesta tabela, pois vários ID’s únicos da tabela “Pontuacao_Niveis” podem corresponder à mesma categoria.

▪ Badges

A tabela “*badges*”, associada ao módulo de gestão de *badges*, guarda a informação de cada *badge* existente em sistema, inclusivo a imagem que apresenta. Cada colaborador só pode ganhar cada *badge* uma vez por jogo. Portanto, foi decidido que cada *badge* inserido no sistema deve estar associado a um jogo temporal, obrigando a existência de uma chave estrangeira “IDJogo” da tabela “jogos”. Esta ligação permite ainda que fique registado como histórico todos os *badges* existentes nos diferentes jogos temporais. Deste modo, a gestão dos *badges* torna-se mais simples, quando se pretende eliminar algum *badge* do sistema ou editar informações dos mesmos.

O campo “IDBadge” é o identificador de cada *badge*, sendo então a chave primária da tabela “*badges*”.

▪ Parametrizacao_Badges

A tabela “*parametrizacao_badges*” está também associada ao módulo de gestão de *badges* e surge pela necessidade de o sistema permitir a atribuição de *badges* automáticos. O sistema poderia ter internamente definido os objetivos que permitiriam as atribuições automáticas de *badges*. No entanto, decidiu-se tornar essas decisões dinâmicas conforme os desejos dos administradores do sistema, tornando assim a tabela “*parametrizacao_badges*” essencial.

Esta tabela guarda todos os modos de atribuição de cada *badge*, obrigando assim a existência de uma relação com a tabela “*badges*”. Quando um novo *badge* é inserido no sistema, automaticamente é inserido o modo de atribuição manual, ou seja, um administrador pode atribuir um *badge* manualmente a um colaborador. Contudo, definiu-se que o ideal seria existir uma parametrização que permitisse a atribuição automática consoante o número de eventos realizados e/ou a pontuação obtida em categorias e em eventos, ou seja, quatro métodos para a atribuição automática de *badges*.

O campo “modoAtribuicao”, como o nome indica, guarda apenas o modo de atribuição do *badge*, modo manual ou automático. O campo “tipoAtribuicaoAutomatico” indica qual o tipo

de atribuição (nº de eventos numa categoria; nº de eventos numa atividade; pontuação obtida numa categoria; pontuação obtida numa atividade).

Os campos “IDCat” e “IDAtiv” são chaves estrangeiras das tabelas “categorias” e “atividades” e na teoria não se caracterizam como sendo de preenchimento obrigatório, pois cada registo associa-se apenas a uma categoria ou a uma atividade individualmente. No entanto, quando um administrador faz a gestão da parametrização de cada *badge*, quando o utilizador escolhe modo automático então o sistema obriga sempre a escolha da categoria ou de uma atividade, tornando estes campos obrigatórios na prática.

Um *badge* não pode conter vários tipos iguais de inserção em simultâneo, tornando os campos “IDBadge”, “modoAtribuicao” e “tipoAtribuicaoAutomatico” a chave primária composta da tabela “parametrizacao_badges”.

▪ Badges_Colaboradores

A tabela “*badges_colaboradores*”, associada também ao módulo de gestão de *badges*, surge na necessidade de atribuir um *badge* a um determinado colaborador. Contudo, esta tabela guarda as informações necessárias quando isso ocorre, nomeadamente, a data de atribuição e uma descrição sobre a atribuição.

Tendo em conta o requisito essencial de cada colaborador só poder ganhar cada *badge* uma vez por jogo, poder-se-ia relacionar o identificador do jogo na presente tabela. No entanto, o facto da ligação com o jogo existir na tabela “*badges*”, permite uma melhor gestão das configurações de cada *badge*, principalmente, na parte das parametrizações para as atribuições automáticas e na eliminação dos *badges* quando se pretende que determinados *badges* deixem de existir no jogo temporal corrente. Além disso, a ligação com a tabela “*badges*” permite ainda a vantagem de guardar um histórico de todas as configurações dos *badges* nos diferentes jogos temporais.

Os campos “IDBadge” e “IDColaborador” são chaves estrangeiras das tabelas “*badges*” e “*colaboradores*” e, simultaneamente, caracterizam-se como sendo a chave primária composta da tabela “*badges_colaboradores*”.

▪ **Apps_Externas**

A tabela “*apps_externas*”, associada ao módulo de gestão de aplicações externas, surge na necessidade de identificar os pedidos recebidos através da API construída. De modo a evitar que qualquer aplicação seja enviada, existe o campo “*identificador*” que reconhece a aplicação externa. Os pedidos enviados através da API indicam a realização de um evento por um ou vários colaboradores, deste modo o campo “*tipoInsercaoUsers*” guarda de que tipo serão identificados os utilizadores, ou pelo *username* ou pelo *e-mail*, guardados na tabela “*colaboradores*”. Os restantes campos apresentam-se como não sendo de preenchimento obrigatório no entanto são importantes para citar a aplicação externa.

O campo “*IDAppExt*” é o identificador de cada aplicação/sistema externo, sendo a chave primária da tabela “*apps_externas*”.

▪ **Acoes_Apps_Externas**

Após a identificação das aplicações externas que enviam para o sistema a realização de eventos pela API, torna-se necessário identificar quais as atividades também recebidas. A tabela “*acoes_apps_externas*”, associada ao módulo de gestão de aplicações externas, surge para identificar e limitar quais as atividades que cada aplicação externa pode enviar. O campo “*identificador*” serve para reconhecer a atividade recebida e identificar com uma atividade registada no sistema. Todas estas ações são geridas pelos administradores do sistema, através do *backOffice*. De referir que, a API só permite a inserção de eventos de atividades que não contém peso. Deste modo, não é necessário registar nem enviar no pedido essa informação.

Os campos “IDAppExt” e “IDAtiv” são chaves estrangeiras das tabelas “apps_externas” e “atividades” e, simultaneamente, caracterizam-se como sendo a chave primária composta da tabela “acoes_apps_externas”.

5. Desenvolvimento e Implementação

O presente capítulo apresenta as funcionalidades principais desenvolvidas, que resultaram da implementação de todos os requisitos identificados e preconizados ao longo dos capítulos anteriores. As diferentes funcionalidades implementadas são apresentadas neste capítulo, bem como as decisões tomadas, tendo em conta, principalmente, no aumento de motivação para a realização de eventos com uso de *gamification*.

Esta fase de desenvolvimento inclui:

- A criação de todas as funções e procedimentos essenciais na resposta às necessidades dos quatro grandes componentes do sistema (a API para aplicações externas, o *backOffice*, o *frontOffice* e a comunicação com o LagoonSecurity);
- A criação dos objetos de *design* e *layouts* da plataforma, suportados também por essas funções;
- A criação das funções de acesso e manipulação de informação da base de dados de suporte, apresentada detalhadamente no subcapítulo “4.5 Modelo de Dados”;
- A implementação de métodos de segurança;
- A realização de testes funcionais e de usabilidade.

5.1. O sistema

Como já referido anteriormente, o sistema desenvolvido no presente projeto, integra quatro componentes principais: a criação de uma API para sistemas externas; a criação de um *backOffice*; a criação de um *frontOffice*; a comunicação com o LagoonSecurity para a gestão de autenticações e permissões dos utilizadores. De seguida, encontram-se os principais desenvolvimentos realizados para estas quatro componentes, bem como, as eventuais decisões tomadas na realização dos mesmos.

De referir que todos os dados apresentados são fictícios.

5.1.1. API para Sistemas Externos

A API foi desenvolvida para permitir às aplicações externas comunicarem quais os eventos realizados pelos colaboradores. Assim sendo, quando um determinado colaborador realiza uma atividade numa aplicação já existente, evita-se que exista sempre a necessidade de inserir manualmente um novo evento no sistema de *gamification*.

A API desenvolvida está preparada para receber pedidos POST via REST. Os pedidos são efetuados para a URL “<servidor>/Gamification/API/insertEvent”. Cada pedido enviado à API, tem de estar apresentada em formato JSON. O corpo do pedido deve ser composto pelos seguintes componentes:

- “application”

Identifica a aplicação externa que está a efetuar o pedido. Este identificador tem de estar registado no sistema, caso contrário o pedido não tem sucesso.

- “action”

Identifica a atividade do evento que está a ser inserido. Este identificador tem de estar registado no sistema e associado à aplicação que está a efetuar o pedido, caso contrário o pedido não tem sucesso.

- “date”

Identifica a data do evento.

- “employees”

Identifica todos os colaboradores que realizaram a atividade. Os colaboradores podem estar apresentados no pedido, através do *username* ou do *e-mail*. No entanto, cada aplicação externa registada no sistema, só pode ter associado um tipo de inserção de colaboradores. Por exemplo, uma aplicação que tenha registado como método de envio de colaboradores pelo *username*, não pode identifica-los através de *e-mail*, pois o pedido não terá sucesso. Igualmente, o mesmo ocorre quando os colaboradores identificados não existem em sistema.

O corpo de cada pedido efetuado tem de estar apresentado semelhantemente à figura seguinte.

```
{
  "application": "lagoonIdeas",
  "action": "insercaoIdeia",
  "date": "2015-07-29",
  "employees": ["maria", "andre"]
}
```

Fig. 17 - Exemplo de corpo de pedido json

Como se pode verificar na figura anterior, não existe um campo que identifique o peso da atividade. Foi decidido que as atividades recebidas pelas aplicações externas não têm peso associado. O peso reflete-se em atividades que podem ter características dinâmicas, por exemplo, o tempo de duração. As atividades que os colaboradores realizam nas aplicações externas caracterizam-se como sendo diretas, não sendo necessário terem um peso definido.

Quando um sistema efetua um pedido à API, é lhe sempre obtido uma resposta com identificação se o pedido foi realizado com sucesso ou insucesso. Essa mensagem de resposta apresenta-se sempre em formato *json*. Caso o pedido obtenha sucesso, o corpo da mensagem

de resposta apenas é constituída por "Success". Caso contrário, a mesma é composta por a mesma identificação de "Success" e "Message" que informa o motivo da falha do pedido.

As possíveis falhas que um pedido pode obter são referidas no diagrama de atividades apresentado no subcapítulo "4.4.1. Diagrama de Atividades da API". Contudo, as figuras seguintes exibem as diferentes mensagens de resposta que cada pedido à API pode obter.

```
{  
  "Success": "FALSE",  
  "Message": "Aplicacao invalida."  
}
```

Fig. 18 - Mensagem de resposta com erro quando identificação da aplicação não existe em sistema

```
{  
  "Success": "FALSE",  
  "Message": "Atividade invalida."  
}
```

Fig. 19 - Mensagem de resposta com erro quando atividade não existe em sistema ou não se relaciona com a aplicação

```
{  
  "Success": "FALSE",  
  "Message": "Colaborador invalido."  
}
```

Fig. 20 - Mensagem de resposta com erro quando introduzido um colaborador que não existe em sistema ou não segue o formato identificado pela aplicação

```
{  
  "Success": "TRUE"  
}
```

Fig. 21 - Mensagem de resposta com sucesso

5.1.2. BackOffice

O *backOffice* é um componente do sistema que foi desenvolvido para permitir, essencialmente, a existência de uma gestão interna das atividades que os colaboradores realizam ao longo do tempo, dentro da organização XLM, com recurso ao *gamification*.

5.1.2.1. Estrutura e Funcionalidades

A estrutura do *backOffice* foi desenvolvida tendo por base os requisitos anteriormente identificados. Para a estruturação desta componente, teve-se em consideração algumas práticas de *design* (Molin & Abreu, 2011) e a organização de conteúdos por grupos lógicos. No entanto, deu-se bastante relevância às preferências requeridas pela organização, nomeadamente, à ordem de apresentação dos conteúdos e às informações que necessitam de ter visibilidade.

O esquema seguinte apresenta a estrutura base criada para o *backOffice*, seguindo da explicação detalhada de cada constituinte identificado.



Fig. 22 - Estrutura base do *backOffice*

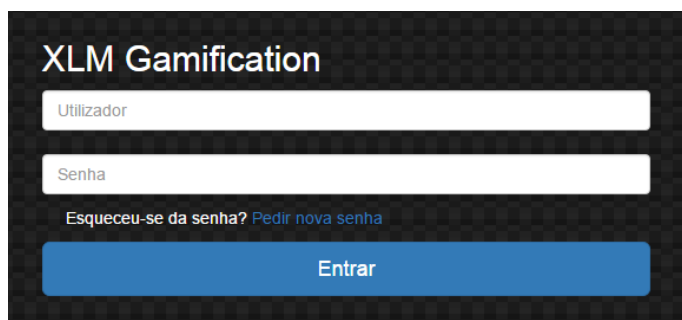
De relembrar que as permissões de todas as características e funcionalidades apresentadas neste subcapítulo, podem variar pela gestão no LS.

Autenticação

O processo de autenticação – *login* - é uma funcionalidade essencial na plataforma, constituída por um formulário de validação de utilizadores para acesso à aplicação. Através da introdução de um nome de utilizador e uma senha, o sistema, com o auxílio do LS, verifica se o utilizador é válido. Caso seja, consoante o tipo de utilizador que esteja a aceder, o sistema permite a entrada na plataforma e às funcionalidades que o utilizador tem permissão de acesso. Caso contrário, se as credenciais inseridas não estão previamente registadas no sistema, então o sistema não permite o acesso.

Neste processo de autenticação, em termos técnicos, é necessário que o nome de utilizador esteja corretamente registado na base de dados de suporte. No entanto, embora na mesma base de dados se encontre a senha de acesso encriptada, a responsabilidade de validar a senha encontra-se do lado do LS e não do sistema. Esta informação da senha, como já referido, é necessária existir na base de dados para o processo de recuperação de senhas, que se encontra explicado no subcapítulo “5.1.4. Comunicação com o LagoonSecurity”.

A figura seguinte apresenta o formulário de acesso à plataforma, comum para todos os tipos de utilizadores.



The image shows a login form for 'XLM Gamification'. It features a dark background with white text and input fields. At the top, the title 'XLM Gamification' is displayed in white. Below the title, there are two white input fields: the first is labeled 'Utilizador' and the second is labeled 'Senha'. Underneath the 'Senha' field, there is a link that reads 'Esqueceu-se da senha? Pedir nova senha' in a light blue color. At the bottom of the form, there is a prominent blue button with the white text 'Entrar'.

Fig. 23 - Formulário de acesso à plataforma

Página Inicial

A página inicial do *backOffice* é composta, essencialmente, por três zonas principais. No lado esquerdo é disponibilizado o menu de navegação com acesso a todas as funcionalidades principais do *backOffice*. Este menu encontra-se sempre disponível ao utilizador.

No centro da página encontra-se do lado direito um *feed* com as últimas vinte e cinco novidades que ocorreram no sistema. Os últimos eventos realizados e as conquistas dos *badges* fazem parte deste *feed*. Do lado direito encontra-se uma tabela com o *ranking* dos colaboradores no jogo temporal corrente. A página inicial preocupa-se apenas de oferecer dados informativos aos utilizadores. Esta decisão foi preferenciada pela organização.

Por fim, ainda na parte superior nomeia-se a plataforma – *XLM Plataforma de Gamification* – com ligação a esta primeira página do *backOffice*. Também se encontra referido o nome do utilizador logado, onde daí surge um novo pequeno menu que permite ao utilizador alterar dados pessoais ou sair da plataforma.

Na figura seguinte pode-se observar a página inicial do *backOffice*.

Nome	Atividade	Data
André Santos	Conquistou o Badge: Este Badge é um badge de Teste... mas ganha-se se se se fizer y	2015-08-13
Maria Martins	Realizou a atividade: Formação Pontuação: 40	2015-08-11
Maria Martins	Conquistou o Badge: SSSS	2015-08-11
Ana Oliveira	Realizou a atividade: Aprovação de ideias Pontuação: 5	2015-07-31
André Santos		2015-07-31

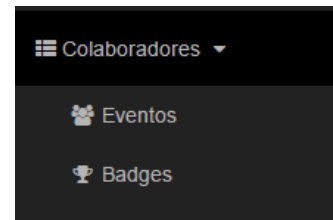
Nome	Pontuação
Maria Martins	1162
André Santos	67
Ana Oliveira	17
Luis Martins	5
João Oliveira	0

Fig. 24 - Página inicial do *backOffice*

Colaboradores

A área “Colaboradores”, encontrada no menu principal referido, abrange duas grandes constituintes do projeto: gestão de eventos e gestão de atribuição de *badges*. Cada constituinte pode ser acedido através do menu.

Esta área apresenta-se desta forma, de modo a obedecer ao requisito essencial de usabilidade, que pede ao sistema o foco inicial dos colaboradores e, só posteriormente, oferecer as



devidas funcionalidades dentro do âmbito de um determinado colaborador. Deste modo, o sistema permite aos utilizadores fazerem a gestão de eventos e/ou *badges*, pensando apenas num colaborador em concreto.

Fig. 25 - Menu área "Colaboradores"

Ambas as páginas principais, de gestão de eventos ou gestão de atribuição de *badges*, de um determinado colaborador são semelhantes. O utilizador pode visualizar todos os colaboradores registados no sistema e seleccionar o que deseja gerir, tendo a opção de filtrar pelo nome de utilizador. A figura seguinte apresenta o *interface* dessa página.

Colaboradores Histórico de Atividades

Introduzir Nome		Q
Número Colaborador	Nome	
2	João Oliveira	+
15	André Santos	+
32	Luis Martins	+
80	Maria Martins	+
102	Ana Oliveira	+
Número Colaborador	Nome	

Fig. 26 - Listagem de colaboradores

Na área de gestão de eventos por colaborador, após a seleção de um determinado colaborador, é apresentado ao utilizador os eventos que esse colaborador realizou no jogo corrente. O utilizador pode visualizar o histórico dos eventos do colaborador, filtrando para outros jogos temporais. Existe também um filtro de pesquisa de eventos por categoria.

Histórico Maria Martins

Jogo: Categoria:

Data Inicio	Atividade	Categoria	Jogo	
2015-08-11	Formação	Educação	Nome JogoAtual	<input type="button" value="+"/>
2015-07-31	Aprovação de ideias	Inovação	Nome JogoAtual	<input type="button" value="+"/>
2015-07-29	Criação de ideias	Inovação	Nome JogoAtual	<input type="button" value="+"/>
2015-07-29	Criação de ideias	Inovação	Nome JogoAtual	<input type="button" value="+"/>
2015-07-29	Criação de ideias	Inovação	Nome JogoAtual	<input type="button" value="+"/>
2015-07-29	Criação de ideias	Inovação	Nome JogoAtual	<input type="button" value="+"/>
2015-07-29	Criação de ideias	Inovação	Nome JogoAtual	<input type="button" value="+"/>
2015-07-29	Criação de ideias	Inovação	Nome JogoAtual	<input type="button" value="+"/>
2015-07-29	Criação de ideias	Inovação	Nome JogoAtual	<input type="button" value="+"/>

Fig. 27 - Listagem de eventos por colaborador

A cada evento visível é possível visualizar as informações associadas com mais detalhe, alterar determinados dados ou mesmo eliminar esse evento. A figura seguinte apresenta o *interface* com as informações de um evento, selecionado anteriormente.

Detalhes Evento

Jogo: Data:

Categoria: Atividade:

Peso: Quantas horas totais? Pontuação da Atividade: Pontuação Ponderada:

Colaboradores * Pontuação Obtida *

Número	Colaborador	Pontuação Obtida
4	Maria Martins	40
Número	Colaborador	Pontuação Obtida

Fig. 28 - Detalhes na gestão de um evento

Como referido nos requisitos funcionais no subcapítulo “3.4.2. *BackOffice*”, o sistema permite ao utilizador eliminar um evento ou fazer alterações de dados, nomeadamente, na data, no peso do evento e nos utilizadores que realizaram o evento, como se pode verificar na figura anterior. Ainda no mesmo *interface*, o sistema permite ao utilizador adicionar ou remover outros colaboradores na realização de um determinado evento.

A figura seguinte representa o resultado da inserção de um colaborador num evento.

Detalhes Evento Editar

Guardar Eliminar Evento Cancelar

Jogo: Nome JogoAtual

Data: 2015-08-11

Categoria: Educação

Atividade: Formação

Peso: Quantas horas totais? 2

Pontuação da Atividade: 20

Pontuação Ponderada: 40

Colaboradores * Escolher Colaborador

Pontuação Obtida *

Adicionar Colaborador

Número	Colaborador	Pontuação Obtida
4	Maria Martins	40
41	João Oliveira	20

Fig. 29 - Inserção de colaborador em evento

Regressando ao *interface* com o histórico de eventos de um determinado colaborador, verifica-se a existência de uma opção que permite a criação de um novo evento, cujo (s) realizador (es) inclui sempre o colaborador do âmbito atual. O *interface* de inserção de novos eventos é semelhante ao *interface* que representa os detalhes de um evento. Caso o utilizador seleccione a opção indicativa que a realização do evento é em grupo, o sistema apresenta ao utilizador uma lista com os restantes colaboradores.

A nível lógico do sistema, para a gestão de eventos foram implementadas todas as características detalhadas encontradas nos diagramas de atividades existentes no subcapítulo “4.4.4. Diagramas de Atividades para a Gestão de Eventos”.

Eventos Adicionar Novo Evento

[Guardar](#) [Cancelar](#)

Colaborador
Maria Martins

Jogo *
Nome JogoAtual

Data *
[Calendar Icon]

Categoria *
Escolher Categoria

Atividade *
Escolher Atividade

Peso
[Dropdown]

Pontuação da Atividade
[Input]

Pontuação Ponderada
[Input]

Tipo de Evento
Individual

Fig. 30 - Criação de novo evento

Regressando ao menu na área de “Colaboradores”, encontra-se a subárea de gestão de atribuição de *badges* por colaboradores. Como referido, esta gestão inicializa com um *interface* semelhante ao da figura 25. Após a seleção de um determinado colaborador, é apresentado ao utilizador a lista de todos os *badges* que esse colaborador conquistou no jogo temporal corrente. O utilizador pode visualizar o histórico de *badges* ganhos pelo colaborador, filtrando para outros jogos temporais.

Como se verifica na figura seguinte, no mesmo *interface* que apresenta os *badges*, também é permitido a remoção da atribuição dos mesmos.

Histórico de Badges Maria Martins

Jogo
Escolher Jogo

[Pesquisar](#) [Atribuir Badge](#) [Cancelar](#)









Badge	Nome	Descrição	Data	
	AAAAA	SSSS	2015-08-11	
	Badge3	Badge3	2015-07-01	
	Badge2	Badge2	2015-06-30	
	Badge1	Este Badge é um badge de Teste... mas ganha-se se se fizer y	2015-05-12	

Fig. 31 - Listagem de badges por colaborador

Ainda no mesmo *interface*, o utilizador pode prosseguir para uma atribuição manual de um determinado *badge* ao colaborador em âmbito.

The screenshot shows a web interface titled "Badges Atribuir Badge". At the top right, there are two buttons: "Guardar" (blue) and "Cancelar" (grey). Below the title, there are two input fields: "Badges *" with a dropdown menu showing "Escolher Badge" and "Data *" with a date input field and a calendar icon. Below these are two text areas: "Descrição" and "Descrição Adicional".

Fig. 32 - Atribuição manual de um *badge* a um colaborador

Badges

A área “*Badges*”, encontrada no menu principal anteriormente referido, permite a atribuição de *badges* manualmente. A diferença desta funcionalidade com a anterior, referida na área “Colaboradores”, resume-se no âmbito. A área anterior focava um colaborador, permitindo a atribuição manual de um *badge* a esse colaborador. Enquanto, esta área “*Badges*”, caracteriza-se como sendo uma funcionalidade que permite a atribuição manual de um *badge* para vários colaboradores em simultâneo.

A decisão para o desenvolvimento desta funcionalidade verificou-se na possibilidade de se criarem atividades eventuais, em que a organização decida criar um *badge* especial para a realização dessas atividades. Torna-se mais prático e vantajoso para o utilizador, a existência desta funcionalidade para estas situações. Deste modo, o sistema permite também aos utilizadores fazerem a atribuição de *badges*, pensando apenas num determinado *badge* e não num colaborador em concreto, como na funcionalidade anterior.

A figura seguinte apresenta o *interface* com a lista de *badges* possíveis de atribuir no jogo temporal corrente, que é mostrada ao utilizador.

Badges Atribuições Gerais

Introduzir Nome Badge






Imagem	Nome	Total	
	Badge1	2	<input type="button" value="Atribuições"/>
	Badge2	3	<input type="button" value="Atribuições"/>
	Badge3	3	<input type="button" value="Atribuições"/>
	AAAAA	2	<input type="button" value="Atribuições"/>
	SDS	0	<input type="button" value="Atribuições"/>

Fig. 33 - Listagem de badges para atribuição

A partir da figura anterior, pode-se verificar que o utilizador pode filtrar os *badges* pelo nome dos mesmos. O sistema ainda indica o número total de vezes que cada *badge* foi atribuído no corrente jogo.

O utilizador, após a seleção do *badge* a atribuir, pode prosseguir para a escolha dos colaboradores que conquistaram o *badge* bem como a inserção dos dados necessários no processo de atribuição de *badges*.

A figura seguinte apresenta o *interface* responsável por permitir o registo de uma atribuição de um determinado *badge* a vários colaboradores.

Atribuir Badge Badge1

Data *

Descrição Adicional

Colaboradores *

- Ana Oliveira
- André Santos
- João Oliveira
- Luis Martins
- Maria Martins

Fig. 34 - Atribuição manual de um badge a vários colaboradores

Eventos

A área “Eventos”, encontrada no menu principal anteriormente referido, é constituída por todas as funcionalidades integradas nos “Eventos” na área dos “Colaboradores”. A principal diferença entre estas funcionalidades é o âmbito. Enquanto a área dos “Colaboradores” foca, inicialmente, a gestão por um colaborador, esta área de “Eventos” permite ao utilizador gerir cada evento sem estar diretamente no âmbito de um determinado colaborador. As funcionalidades de pesquisa de informações, inserção de um novo evento, remoção e edição de dados funcionam de uma forma idêntica às funcionalidades da área anterior.

Configurações

A área “Configurações”, encontrada no menu principal anteriormente referido, é constituída por todas as funcionalidades que permitem a gestão de todas as configurações básicas necessárias para o devido funcionamento do sistema. As configurações desta área abrangem os dados dos colaboradores (colaboradores do tipo normais e administradores), as categorias bem como os níveis e as atividades que nelas incluem, os *badges*, os jogos temporais e os seus níveis. Para o efeito, foram desenvolvidos formulários necessários para a adição de novos registos, eliminação e alteração de dados correspondentes.

Do conjunto de configurações, como se pode verificar na figura 21 da estrutura base do *backOffice*, existem quatro opções, cujas gestões se distinguem ligeiramente das restantes: A área dos colaboradores; das categorias; dos *badges* e a área da gestão dos jogos temporais.

Relativamente à área dos colaboradores, distingue-se pelo facto de existirem interferências e ações do LS. Quando um utilizador adiciona, edita ou remove um utilizador, existe sempre comunicação com o LS. A nível lógico do sistema foram implementadas todas as características detalhadas no diagrama de atividades “4.4.3. Diagramas de Atividade para a Gestão de Colaboradores”.

Como referido, existem dois tipos de utilizadores diretos com autenticação na plataforma: o administrador e o colaborador normal. Como ambos os utilizadores são colaboradores, um administrador pode gerir os dois tipos de utilizadores nesta área. A imagem seguinte apresenta o *interface* para inserção de um novo colaborador.

Colaboradores Adicionar Novo Colaborador

Fig. 35 - Inserção de colaborador

A área das categorias, como se pode verificar na figura seguinte, permite a gestão de cada categoria incluindo a gestão dos diversos níveis da mesma e das atividades que nela preenchem.

Categorias Gestão

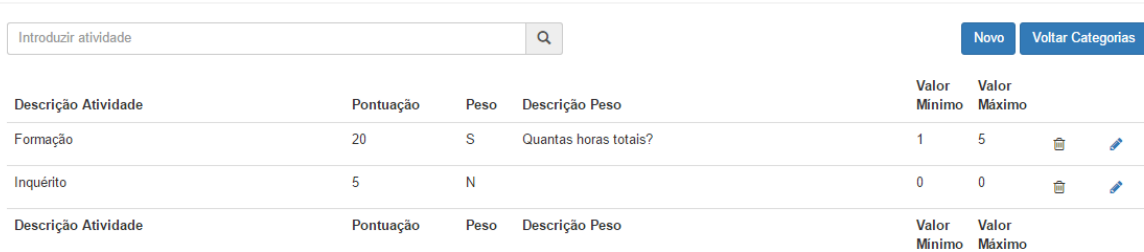
Introduzir categoria	Q	Novo
Categoria		
Educação		Níveis Atividades
Entretenimento		Níveis Atividades
Inovação		Níveis Atividades
Categoria		

Fig. 36 - Listagem das categorias

Como referido anteriormente, cada atividade contém obrigatoriamente uma pontuação que pode ser ganha por cada colaborador quando realiza essa atividade. O peso que cada

atividade pode conter ou não também pode ser gerido pelo utilizador nesta fase. A figura seguinte apresenta as atividades incluídas numa categoria. Este *interface* também pode ser acedido a partir da opção “atividades” na área “Configurações”, tendo apenas o utilizador de selecionar primeiro a categoria.

Atividades Educação



Descrição Atividade	Pontuação	Peso	Descrição Peso	Valor Mínimo	Valor Máximo		
Formação	20	S	Quantas horas totais?	1	5		
Inquérito	5	N		0	0		
Descrição Atividade	Pontuação	Peso	Descrição Peso	Valor Mínimo	Valor Máximo		

Fig. 37 - Listagem de atividades de uma categoria

A gestão dos níveis de cada categoria é semelhante à gestão dos níveis de cada jogo temporal. Os níveis podem ser representados por números (nível 1, nível 2,...) ou texto (por exemplo, iniciante, intermédio, experiente,...). A nível técnico o que difere de ambas as gestões são as imagens. Os níveis dos jogos podem ter imagem associadas, enquanto aos níveis de cada categoria não é permitido. Um dos objetivos das categorias é somar e permitir a visualização dos pontos obtidos por cada colaborador, portanto a imagem não é necessária.

A figura seguinte apresenta o *interface* para a gestão dos níveis do jogo temporal corrente. Como a nível geral a plataforma apenas funciona no âmbito do jogo corrente, para os jogos temporais que se encontrem num estado fechado o utilizador apenas tem acesso à visualização dos níveis.

Os níveis caracterizam-se como um elemento de *gamification* na camada de componentes no modelo de Kevin Werbach, referido no subcapítulo “2.2.3. Elementos e Mecanismos de Jogo”. Foi decidido o sistema integrar este componente nas categorias e nos jogos temporais, para incentivar o aumento na realização de eventos, permitindo aos colaboradores alcançarem limites (níveis) e não obterem apenas uma determinada pontuação.

Gestão Níveis Nome JogoAtual






Imagem	Nível	Pontuação Mínima	Descrição	
A	Criança	0	Neste nível o colaborador é uma criança	 
B	Adolescente	100	Após alguns eventos o colaborador é adolescente	 
C	Adulto	200	O colaborador chegou à fase adulta	 
Imagem	Nível	Pontuação Mínima	Descrição	

Fig. 38 - Listagem de níveis de um jogo temporal aberto

Na área das “configurações”, a opção “Badges” apresenta formulários e listagens semelhantes aos anteriores. No entanto, existem dois aspectos relevantes nesta área: as imagens dos *badges* no sistema e os métodos de atribuição dos mesmos.

Os *badges* que serão inseridos no sistema prevê-se que sejam construídos e desenhados pela organização. Porém, foi integrado no sistema o acesso a uma pequena ferramenta que permite o utilizador desenhar *badges* e fazer o respetivo *download* da imagem. Posteriormente, o utilizador pode inserir essa imagem no sistema apresentado como sendo um *badge*. Esta ferramenta é disponibilizada pelo serviço *OpenBadges*, referido no subcapítulo “4.2. Tecnologias Utilizadas”, mais precisamente, no subcapítulo “4.2.6. Outros”. A imagem seguinte apresenta um exemplo do uso dessa ferramenta.

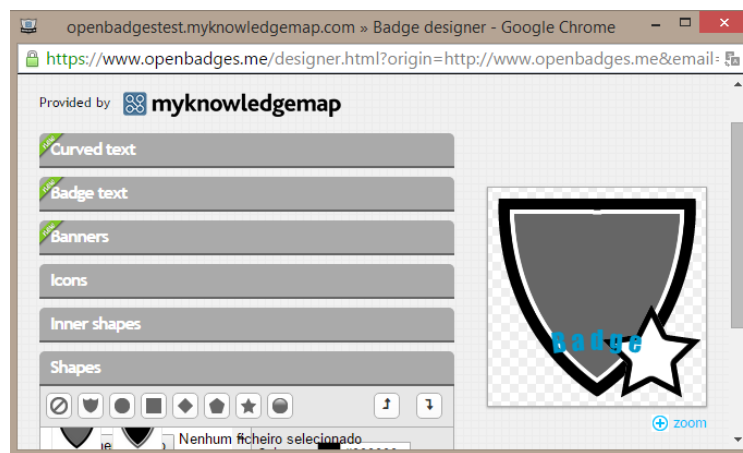


Fig. 39 - Ferramenta OpenBadges - Desenho de badges

O processo de atribuição de *badges* pode ser realizado pelo método manual ou atribuído automaticamente pelo sistema. O método manual foi referido anteriormente no presente subcapítulo. De lembrar, quando um novo *badge* é inserido, o sistema automaticamente possibilita a atribuição desse mesmo *badge* pelo método manual. Contrariamente, os métodos de atribuição automática pelo sistema são dinâmicos mas têm de ser geridos pelo (s) utilizador (es). Conforme os requisitos, o sistema permite ao utilizador gerir quatro tipos de métodos de atribuição automática:

- Quando um colaborador atingiu o total de pontos a realizar uma determinada atividade;
- Quando um colaborador atingiu o total de pontos a realizar uma determinada categoria;
- Quando um colaborador realizou um número máximo de eventos numa determinada atividade;
- Quando um colaborador realizou um número máximo de eventos numa determinada categoria.

Foram decididos estes quatro métodos de modo a permitir uma junção lógica entre os componentes categorias e atividades. Desta forma, a organização pode “jogar” com a motivação dos utilizadores pelos *badges*, quando preferência que sejam realizados com mais frequência certas atividades. As preferências solicitadas pela organização também influenciaram a decisão destes quatro métodos de atribuição automática. Não existe nenhum processo de atribuição automática relacionado com níveis. No entanto, tendo em conta que os níveis das categorias são classificados com pontuação, é possível adicionar no sistema *badges* de níveis, colocando a pontuação máxima igual à registada pelos níveis das categorias. A qualquer momento, tanto as pontuações máximas de níveis como as pontuações parametrizadas nos *badges* podem ser alteradas.

Esta funcionalidade é uma grande vantagem na gestão de *badges*. O utilizador não necessita de fazer verificações dos eventos que os colaboradores vão realizando ao longo do tempo.

Com esta funcionalidade, o utilizador tem o poder de decidir em que circunstâncias os *badges* do sistema são atribuídos de forma automática. O sistema é inteligente e adapta-se às decisões tomadas pelos utilizadores, tornando o processo de atribuição automática de *badges* dinâmico.

A próxima figura apresenta um exemplo do *interface* na inserção de um novo método de atribuição automática para um determinado *badge*. De referir, que o sistema limita os métodos de atribuições, por exemplo, não faria sentido o mesmo *badge* ser atribuído quando um colaborador realiza 10 eventos de inserção de ideias e 5 eventos de formações.

Gestão de Métodos de Atribuições Badge3

Adicionar
Voltar

Modo Atribuição

Automático ▼

Categoria

Entretenimento ▼

Método Atribuição Automático

Pontuação Obtida numa Categoria ▼

Pontuação Máxima Obtida

50

Modo	Método	Categoria	Atividade	NºEventos	Pontuação Obtida
Manual					
Automático	Nº Máximo de Eventos numa Categoria	Educação		20	
Modo	Método	Categoria	Atividade	NºEventos	Pontuação Obtida

Fig. 40 - Gestão de métodos de atribuição automática de *badges*

Aplicações Externas

A área “Aplicações Externas”, encontrada no menu principal anteriormente referido, tem bastante importância para o funcionamento da API desenvolvida. Como referido, os sistemas externos conseguem comunicar com a API, apenas se estiverem devidamente registados no sistema e com as devidas permissões de acesso.

A figura seguinte apresenta o *interface* com a funcionalidade que permite ao (s) administrador (es) gerir (em) as aplicações externas, bem como as ações que podem realizar via API.

Detalhes Aplicação Lagoon Ideias

Guardar Eliminar Aplicação Cancelar

Nome <input type="text" value="Lagoon Ideias"/>	Identificador Externo <input type="text" value="lagoonIdeas"/>
Descrição <input type="text" value="Portal de Ideias"/>	Tipo Recepção/Inserção Utilizadores <input type="text" value="Username"/> ▼

Categoria <input type="text" value="Escolher Categoria"/> ▼	Atividade <input type="text" value="Escolher Atividade"/> ▼
Identificador Externo Ação <input type="text"/>	Descrição Ação <input type="text"/>

Adicionar Ação

Atividade	Identificador Externo	Descrição
Criação de ideias	insercaoIdeia	🗑️
Aprovação de ideias	aprovacaoIdeia	🗑️

Atividade	Identificador Externo	Descrição
------------------	------------------------------	------------------

Fig. 41 - Gestão de aplicação externa

Estatísticas

A área “Estatísticas”, encontrada no menu principal anteriormente referido, é uma funcionalidade desenvolvida com a intuição de fornecer informações que podem ser relevantes para a organização, nomeadamente, na avaliação dos resultados dos jogos temporais.

Esta funcionalidade permite ao (s) administrador (es) visualizarem resultados por números de eventos realizados ao longo do tempo. Para a organização o interesse subsiste no número de eventos realizados e quem os realiza, pois um grande objetivo do sistema é aumentar esse número ao longo do tempo. Os elementos existentes na plataforma, ou seja, os *badges*, as pontuações e os níveis, destinam-se para o interesse dos colaboradores que realizam os eventos.

Como se pode verificar na figura seguinte, existem vários filtros que os utilizadores podem reunir para obterem resultados. Qualquer pesquisa oferece sempre como resposta um gráfico

de barras e uma tabela com os resultados finais, de modo a tornar a leitura dos dados mais compreensíveis.

A existência de todos os filtros de pesquisa apresentados teve um propósito, nomeadamente:

- Jogos Temporais: Possibilita a consulta do histórico de outros jogos, permitindo comparar as diferenças existentes entre os jogos;
- Categorias: Permite a consulta de quais as categorias e as atividades nelas incluídas com mais e menos eventos realizados, possibilitando à organização decidir possíveis alterações nestas áreas;
- Atividades: Possibilita a consulta de quais os períodos que se realizam mais e menos eventos, permitindo também à organização decidir possíveis alterações nestas áreas;
- Data de início e data de fim: Todos os jogos temporais contêm uma data de início e uma data de fim. No entanto, o utilizador pode necessitar de pesquisar resultados num período em concreto. Por exemplo, se a organização lança um desafio de escrita de artigos com um determinado prazo, torna-se pertinente a existência destes dois filtros, para mais tarde os utilizadores poderem consultar nesse período e comparar resultados;
- Colaboradores: É um filtro essencial para a organização ter conhecimento da atividade e da evolução de cada colaborador.

A figura seguinte apresenta esta funcionalidade com um exemplo de pesquisa acompanhado com os resultados que devolve, designadamente, a procura no jogo corrente dos eventos que um determinado colaborador realizou nas diferentes atividades de uma dada categoria.

Estatísticas Resultados por Eventos

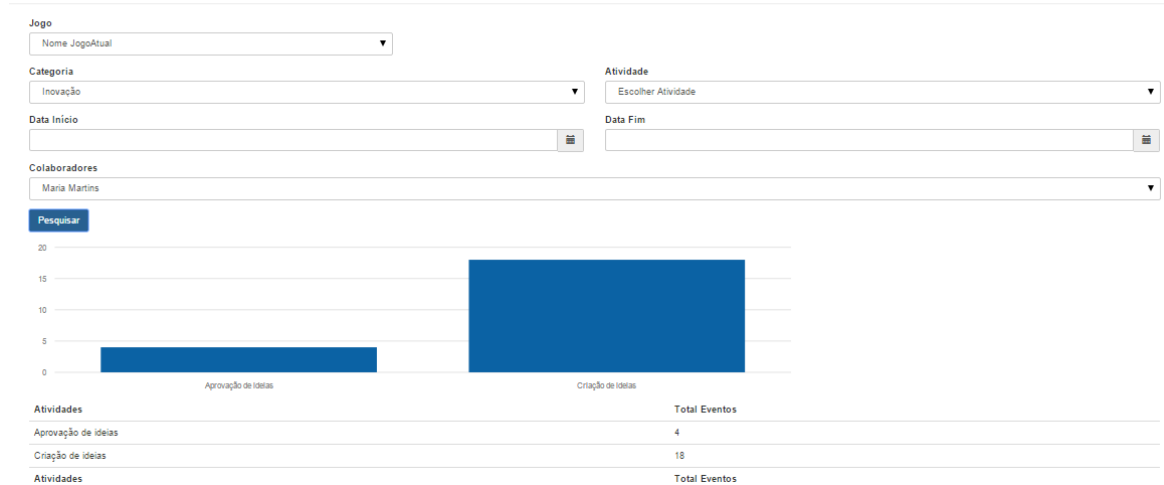


Fig. 42 - Funcionalidade de estatísticas

Outros

Além das funcionalidades referidas, existe um pequeno menu a partir do nome do utilizador logado, que permite ao mesmo alterar os seus dados pessoais, a sua senha de acesso ou sair da plataforma. A figura ao lado apresenta esse menu.

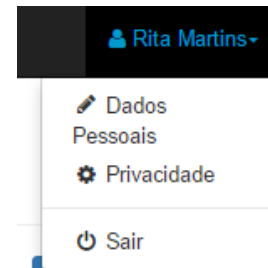


Fig. 43 - Menu pessoal do backOffice

5.1.3. FrontOffice

O *frontOffice* é um componente do sistema desenvolvido para permitir que os colaboradores da organização acompanhem as suas progressões e conquistas, ao longo dos jogos temporais ativos. Se o *backOffice* desenvolvido serve para gerir o *gamification* dentro da empresa, o *frontOffice* expõe os resultados dessa gestão, com a finalidade de os colaboradores sentirem-se mais motivados a cooperarem com a organização.

Relembrando o referido no subcapítulo “3.4.3. *frontOffice*” existe uma plataforma interna que os colaboradores interagem diariamente e, a organização estuda e avalia as hipóteses de entregar as funcionalidades desta componente nessa plataforma ou vice-versa.

5.1.3.1. Estrutura e Funcionalidades

A estrutura do *frontOffice* foi desenvolvida tendo por base os requisitos anteriormente identificados no subcapítulo “3.4.3. *frontOffice*”. Esta componente tem como finalidade mostrar todas as informações relativamente aos progressos de cada colaborador no jogo temporal corrente. Portanto, este componente agrega menos conteúdos que o componente do *backOffice*, tornando a estrutura base mais pequena.

O esquema seguinte apresenta a estrutura base criada para o *frontOffice*, seguindo da explicação detalhada de cada constituinte identificado.

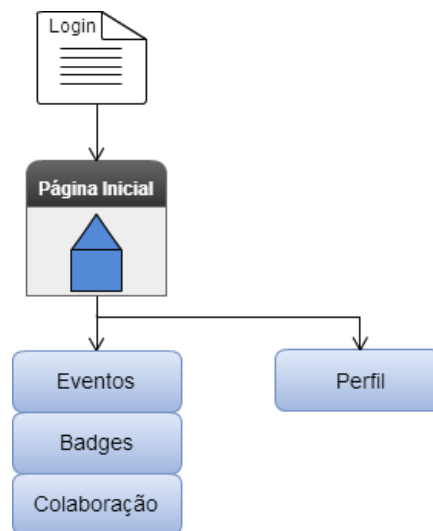


Fig. 44 - Estrutura base do *frontOffice*

De relembrar que as permissões de todas as características e funcionalidades apresentadas neste subcapítulo, podem variar pela gestão no LS.

Autenticação

O processo de autenticação – *login* - é uma funcionalidade essencial para aceder a este componente do projeto. Este processo apresenta-se bastante semelhante ao processo de autenticação do *backOffice*, referido no subcapítulo anterior. O *interface*, as funções e os métodos invocados são exatamente iguais. A única característica que os distingue é somente o tipo de utilizador que diferencia os destinos finais após autenticarem-se.

Página Inicial

A página inicial do *frontOffice* é composta por sete zonas principais. Embora, esta quantidade de zonas possa parecer demasiado em apenas um *interface*, excetuando uma dessas zonas, as restantes são, simplesmente, progressões e resumos do jogo temporal. Os resumos, aqui expostos, apresentam as informações mais relevantes para os colaboradores e, por isso, foi decidido como requisito que a página inicial do *frontOffice* deveria apresentar todos esses pequenos detalhes.

A figura seguinte representa a página inicial do *frontOffice*, cujas informações apresentadas variam consoante o colaborador logado.

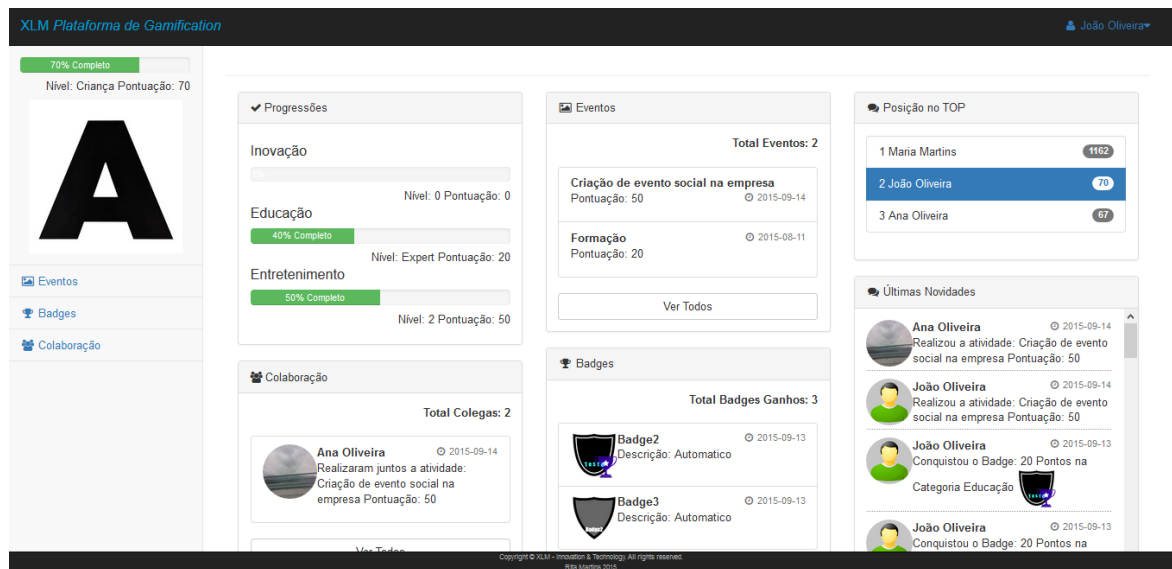


Fig. 45 - Página inicial do *frontOffice*

Como se pode verificar na imagem anterior, no lado esquerdo da página inicial encontra-se uma zona com indicação do nível do jogo temporal que o colaborador se encontra, juntamente com a pontuação que já ganhou até ao momento. A imagem exposta, na situação da figura, representa a imagem do nível do jogo que o colaborador se encontra. Por exemplo, se a organização implementar um jogo com diversas personagens nos diferentes níveis, é pertinente que essa imagem seja exposta. No entanto, se o nível do jogo não tiver imagem, será apresentada a imagem de perfil do utilizador.

Ainda na mesma zona é disponibilizado um menu de navegação com acesso a outras funcionalidades e a informações com mais detalhe, nomeadamente, às informações dos eventos, dos *badges* e dos eventos realizados em grupo (Colaboração).

Esta zona encontra-se sempre disponível ao utilizador. E, por esse motivo foi decidido que a informação da pontuação geral do colaborador estivesse colocada nesta zona, permitindo ao utilizador visualizar facilmente a sua progressão geral no jogo.

No centro da página inicial encontra-se as restantes zonas. A primeira zona corresponde às informações das progressões do colaborador em cada categoria. Esta zona tem bastante importância porque um grande objetivo na construção do presente projeto, para além de ser o aumento da motivação dos colaboradores para realizarem eventos, é simultaneamente, motivá-los para realizarem vários tipos de atividades e não serem somente muito participativos em apenas um tipo de atividade.



Fig. 46 - Progressão nas diferentes categorias

As barras de progressão apresentadas são calculadas tendo em conta as pontuações ganhas anteriormente. Quando um colaborador atinge o limite de pontos que permite alcançar o próximo nível, a barra de progressão não se apresenta nos 0% desse nível. Deste modo, os utilizadores não se sentem desmotivados ou frustrados por se encontrarem numa posição a

zero. Com isto, o sistema dá sempre a ideia que “falta pouco” para o colaborador atingir o próximo objetivo. A barra de progressão do jogo temporal também apresenta esta característica.

A zona dos Eventos indica o total de eventos que o colaborador realizou e, apresenta resumidamente os últimos dois.

Existe um botão de atalho com acesso ao *interface* responsável pelos eventos, referido mais à frente.



Fig. 47 - Resumo de eventos realizados

A página inicial também oferece ao utilizador uma tabela de *ranking*, relativamente às posições totais no jogo temporal corrente. Um *ranking* é um elemento relevante de *gamification*, pois facilmente cria motivações aos indivíduos para

fazerem certas ações. No entanto, se um colaborador se encontra em posições muito baixas no *ranking*, este elemento em vez de motivar, irá desmotivar. Desta forma, foi decidido que a tabela de *ranking* apresentado aos colaboradores deve indicar apenas três posições: a posição do colaborador, a anterior e a seguinte. O utilizador tendo esta limitação, dificilmente se sentirá

frustrado e, contrariamente, poderá sentir-se com mais motivação para ultrapassar no *ranking* o colega que está à sua frente. E, para isso, irá realizar mais eventos no sistema.

A zona de Colaboração indica o total de eventos em grupo que o colaborador realizou. Além disso, também apresenta o respetivo colaborador com quem mais



Fig. 48 - Tabela de ranking

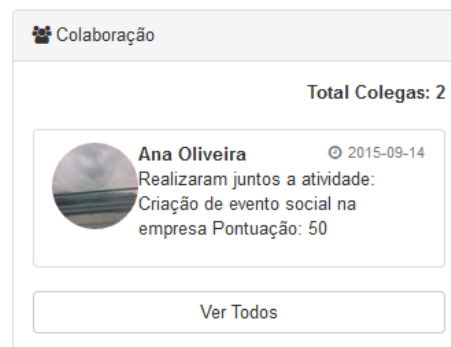


Fig. 49 - Resumo de colaboração

eventos realizou, juntamente com a informação do último evento.

Existe um botão de atalho com acesso ao *interface* responsável pela colaboração, referido mais à frente.

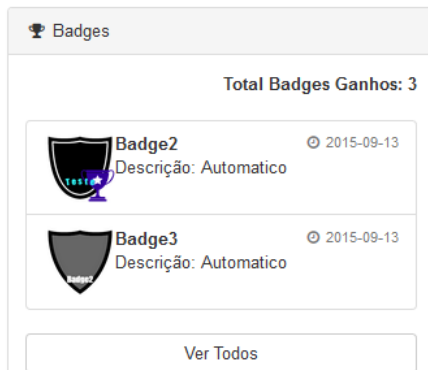


Fig. 50 - Resumo de badges

A zona de *Badges* indica o total de *badges* que o colaborador conquistou e, apresenta resumidamente os últimos dois.

Existe um botão de atalho com acesso ao *interface* responsável pelos *badges*, referido mais à frente.

A página inicial oferece a possibilidade do utilizador visualizar um *feed* com as últimas vinte e cinco novidades que ocorreram no sistema. Os últimos eventos realizados e as conquistas dos *badges* fazem parte deste *feed*. Esta lista é semelhante ao *feed* encontrado na página inicial do *backOffice*, mas em tamanho mais reduzido.



Fig. 51 - Feed com as últimas novidades

Por fim, na parte superior nomeia-se a plataforma – *XLM Plataforma de Gamification* – com ligação a esta primeira página do *frontOffice*. Também se encontra referido o nome do utilizador logado, onde daí surge um novo pequeno menu que permite ao utilizador alterar os seus dados pessoais, a sua senha de acesso ou sair da plataforma.

A organização destas zonas na página inicial foi decidida tendo em conta a relevância da informação. Existem zonas com informação mais pertinente para os colaboradores visualizarem, tendo em conta o interesse da empresa. Por exemplo, as progressões nas diferentes categorias é informação relevante porque interage diretamente com a motivação dos colaboradores. Enquanto a zona do *feed* com as últimas novidades não tem esse impacto. Considerando que a leitura das páginas é lida da esquerda para a direita, a organização das zonas foi contruída pelo grau de interesse da informação para a empresa.

Eventos



Fig. 52 - Informação de atividades existentes

A área “Eventos”, encontrada no menu principal anteriormente referido, é constituída por uma lista de todos os eventos que o utilizador logado realizou. Além disso, como a página inicial faz referência a pontuações, esta área de “Eventos” contém informações relativamente às atividades existentes nas diversas categorias, bem como as pontuações que oferecem pelas suas realizações. Deste modo, o utilizador tem acesso a todas as informações que necessita para realizar eventos

e conseguir progredir no jogo.

Badges

A área “Badges”, encontrada no menu principal anteriormente referido, tem a função de apresentar todos os *badges* que o colaborador logado conquistou no jogo temporal corrente. Como se pode verificar na próxima figura, além de indicar os *badges* conquistados, também mostra os restantes *badges* sombreados existentes no sistema possíveis de o colaborador conquistar.

O utilizador pode clicar em cada *badge* para visualizar as informações de como cada badge pode ser conquistado. Juntamente a essa informação de cada *badge* ganho, o sistema

disponibiliza um botão que permite ao utilizador publicar a sua conquista na rede social “Linkedin”. O utilizador pode-se logar nesta rede social através do botão da plataforma. Foi decidido a partilha de informação nesta rede, pois atualmente considera-se a maior rede profissional do mundo, tornando relevante e motivador para os colaboradores que realizam os eventos para conquistarem *badges*. Além disso, o facto destas distinções partilhadas serem atribuídas pela empresa onde o colaborador exerce as suas funções e não por um grupo de “amigos”, torna esta funcionalidade mais interessante para o colaborador, pois permite publicar informações credíveis sobre a sua atividade profissional.

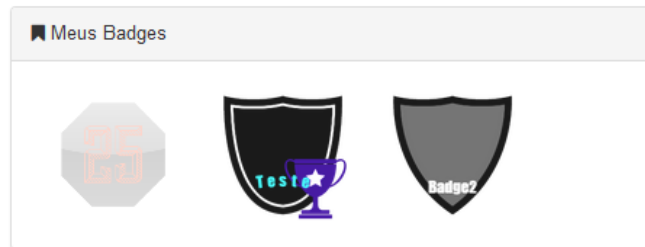


Fig. 53 - Informação de badges

Colaboração

A área “Colaboração”, encontrada no menu principal anteriormente referido, tem a função de apresentar todos os outros colaboradores que realizaram atividades juntos com o utilizador logado. Pode-se chamar a esta área a “zona de amigos”.



Fig. 54 - Informação de colaboração

Como se pode verificar na figura anterior, juntamente a cada colaborador apresenta-se o total de eventos que realizaram juntos. O utilizador pode verificar a informação de todos esses eventos, clicando em cada colaborador apresentado. Juntamente a essa informação, o sistema disponibiliza um botão que permite ao utilizador visualizar o perfil dos outros colaboradores. A figura seguinte é um exemplo das informações que o utilizador pode visualizar dos outros colaboradores.

The screenshot displays a user profile for Ana Oliveira. On the left, there is a progress indicator showing '67% Completo' and 'Nível: Criança Pontuação: 67'. Below this is a large black letter 'A' representing the profile picture. The main profile section includes a small landscape image, the number of collaborators (102), the name (Ana Oliveira), and the current location (XLM). To the right, personal details are listed: Date of Entry (2010-10-15), Date of Birth (1992-06-25), Address (Morada Aveiro), and Contact Information (Email: ritamar_3@hotmail.com, Phone: 965818681).

The profile is divided into several sections:

- Progressões:** A list of progress bars for 'Inovação' (45% complete, Level 1, 12 points), 'Educação' (10% complete, Level: Iniciante, 5 points), and 'Entretenimento' (50% complete, Level: 2, 50 points).
- Eventos:** A list of events with 'Total Eventos: 9'. Two events are visible: 'Criação de evento social na empresa' (50 points, 2015-09-14) and 'Aprovação de ideias' (5 points, 2015-07-31).
- Posição no TOP:** A ranking of top collaborators: 2 João Oliveira (70), 3 Ana Oliveira (67), and 4 André Santos (67).
- Últimas Novidades:** A recent activity feed showing 'Ana Oliveira' (2015-09-14) for 'Realizou a atividade: Criação de evento social na empresa Pontuação: 50'.
- Badges:** A section for earned badges, currently empty.

Fig. 55 - Visualização de perfil de outro colaborador

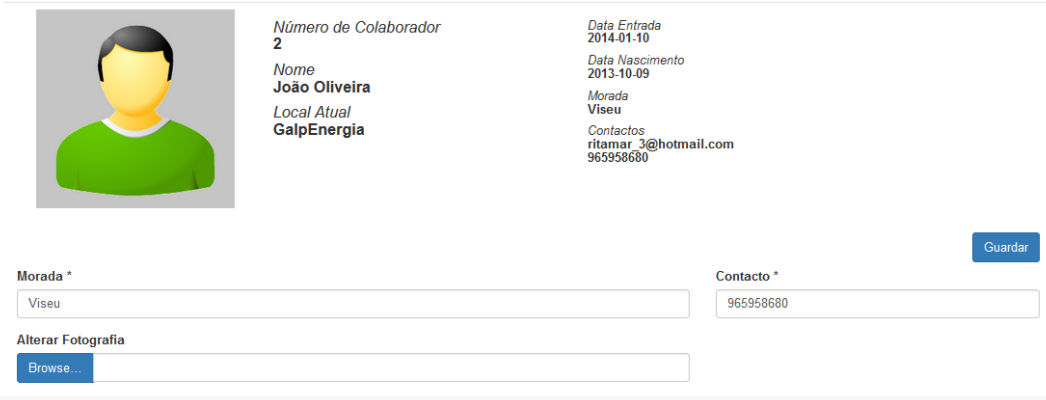
O sistema permite ao utilizador logado visualizar apenas os restantes colaboradores que realizaram eventos com ele, com a finalidade de limitar as informações que o utilizador pode visualizar sobre os seus colegas. Um dos objetivos do desenvolvimento do projeto é fazer com que os colaboradores conheçam-se um pouco melhores, interagindo entre si. O facto de o sistema fazer esta limitação, implica que os colaboradores tenham de realizar eventos com outras pessoas para poderem ver as informações delas. Desta forma, o sistema motiva os colaboradores na realização de eventos em equipa.

A organização pode reunir esta funcionalidade com a dos *badges*, despertando ainda mais a motivação para a realização de atividades em equipa. Por exemplo, criar e atribuir *badges* de equipa a colaboradores que tenham uma “zona de amigos” bem composta.

Perfil

A área de perfil, encontrada num pequeno menu que nasce a partir do nome do utilizador logado, serve para apresentar ao utilizador como os restantes colaboradores visualizam o seu perfil. Além disso, também permite a alteração de determinados dados, nomeadamente, a morada, o contacto e a fotografia.

Meus Dados de Perfil



Número de Colaborador 2	Data Entrada 2014-01-10
Nome João Oliveira	Data Nascimento 2013-10-09
Local Atual GalpEnergia	Morada Viseu
	Contactos ritamar_3@hotmail.com 965958680

Morada * Contacto *

Alterar Fotografia

Fig. 56 - Perfil de utilizador

Outros

O pequeno menu da figura ao lado nasce a partir do nome do utilizador logado e permite ao mesmo alterar os seus dados pessoais visíveis no seu perfil, a sua senha de acesso ou sair da plataforma.

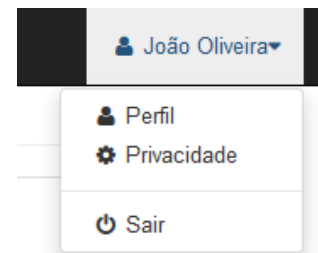


Fig. 57 - Menu pessoal do frontOffice

5.1.4. Comunicação com o LagoonSecurity

Como referido anteriormente, o LS é um sistema externo à plataforma mas com determinadas responsabilidades no processo de autenticação e nas permissões dos utilizadores para acederem às funcionalidades.

O LS guarda algumas informações de cada colaborador registado no sistema. Cada um desses colaboradores tem no LS um estado (ativo ou inativo). Este estado influencia a autenticação no sistema, pois os utilizadores só conseguem ter acesso à plataforma se estiverem num estado ativo.

No sistema desenvolvido, existem, principalmente, duas situações em que o processo das funcionalidades depende também do LS. A inserção de um novo utilizador e a alteração ou recuperação da senha de acesso são funcionalidades, em que o LS também atua com algumas das suas funcionalidades internas, como por exemplo o envio de *e-mails* aos utilizadores.

O LS é um sistema de segurança. Cada vez que o sistema comunica com o LS é necessário o mesmo autenticar-se. Foi criado no projeto um ficheiro de configurações que permite a qualquer altura um administrador do projeto alterar os dados. Neste ficheiro existe as informações necessárias que possibilitam a autenticação do sistema no LS. Este ficheiro de configurações guarda todos os dados técnicos que são necessários ao longo do projeto. Assim, cada vez que existe alterações técnicas a fazer apenas é necessário alterar os dados neste ficheiro sem modificar código-fonte do sistema.

Inserção de Novo Utilizador

Quando um administrador insere no sistema um novo utilizador, o sistema tem a função de guardar no sistema e comunicar ao LS a criação de um novo utilizador bem como de que tipo se tipifica. O LS recebe os dados do novo utilizador, valida-os, atribui-lhe um identificador externo, guarda-o no seu sistema e coloca-lhe num estado inativo. O LS responde ao sistema as informações externas, nomeadamente, o identificado externo atribuído ao novo utilizador e as permissões de acesso de acordo a sua tipologia.

Para o utilizador ter acesso no sistema, precisa de se encontrar num estado ativo. O LS, simultaneamente ao processo anterior, envia um *e-mail* ao novo utilizador com um URL que direciona para uma página do sistema, onde o utilizador pode introduzir uma nova senha de acesso permitindo a ativação da sua conta. A figura seguinte apresenta um exemplo do *e-mail*

que é enviado ao utilizador nesta fase. E, seguidamente, verifica-se uma nova figura que representa o formulário da página de ativação da conta.

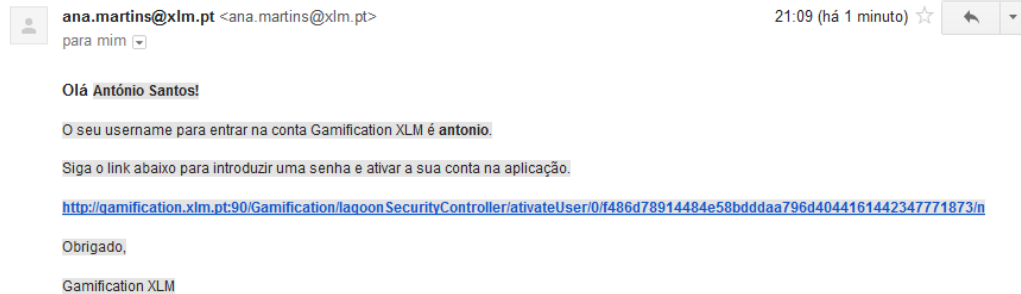


Fig. 58 – E-mail de ativação de conta

The screenshot shows a form titled 'XLM Gamification'. It has two input fields: 'Introduzir nova senha de acesso' and 'Confirmar senha de acesso'. Below the fields is a blue button labeled 'Guardar Senha'.

Fig. 59 - Formulário de inserção de nova senha de acesso

Após a inserção da nova senha, o utilizador além de receber uma notificação no sistema a indicar se a senha foi registada com sucesso ou não, o LS também envia ao utilizador um novo e-mail de confirmação da ativação da conta, como se pode visualizar na figura seguinte.

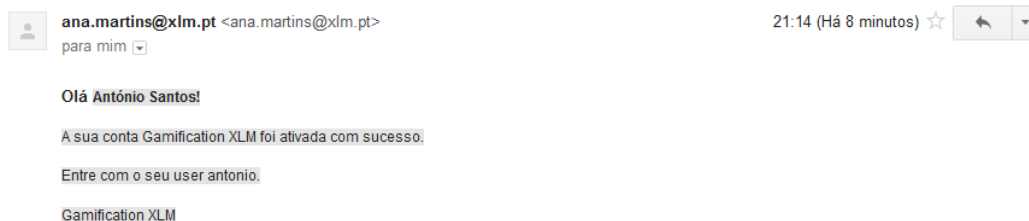


Fig. 60 - E-mail de confirmação de ativação da conta

Recuperação/Alteração da Senha

Quando um utilizador deseja recuperar a sua senha de acesso ou simplesmente alterá-la, o sistema tem obrigatoriamente de comunicar com o LS. No processo de recuperação da senha, o utilizador deve solicitar ao sistema com a introdução do seu nome de utilizador, numa página que pode ser acedida a partir da página de autenticação. A figura seguinte apresenta o pequeno formulário que permite a solicitação de uma nova senha.

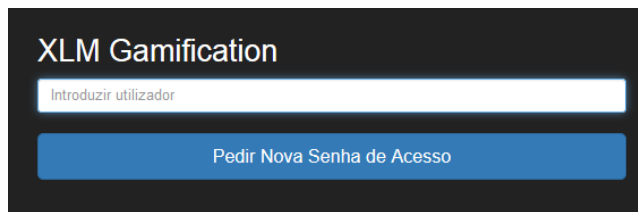
The image shows a dark-themed web form for XLM Gamification. At the top, the text 'XLM Gamification' is displayed in white. Below it is a white input field with the placeholder text 'Introduzir utilizador'. Underneath the input field is a blue button with the white text 'Pedir Nova Senha de Acesso'.

Fig. 61 - Formulário de solicitação de uma nova senha

Após a solicitação efetuada pelo utilizador, o sistema de *gamification*, através do *e-mail* desse utilizador, informa o LS que existe um utilizador que deseja fazer uma recuperação da senha. O LS processa o pedido e envia um *e-mail* para esse utilizador com um URL que o direciona para uma página, semelhante ao *interface* da figura 55, onde o utilizador pode colocar uma nova senha de acesso. A figura seguinte representa um exemplo desse *e-mail*.

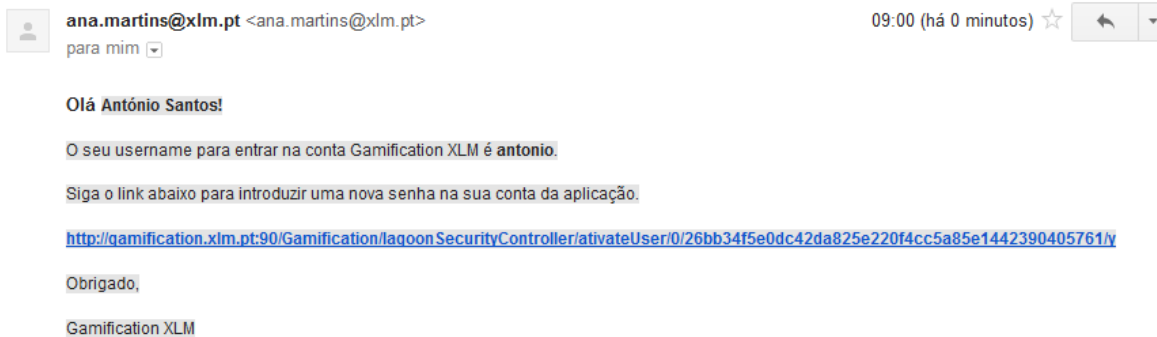


Fig. 62 - E-mail de recuperação de senha

Quando o utilizador introduz uma nova senha, recebe uma notificação de confirmação e um *e-mail* do LS também com a confirmação de sucesso. Posteriormente a isso, o utilizador volta a

conseguir autenticar-se na plataforma com a nova senha introduzida da alteração efetuada com sucesso.

No processo de apenas alteração da senha, sem incluir uma recuperação, o sistema apenas comunica ao LS a antiga e a nova senha inserida, de modo o LS registar essa alteração. Depois disso, igualmente ao processo anterior, o utilizador recebe uma notificação e um *e-mail* do LS com a confirmação de sucesso.

Permissões

Todas as funcionalidades da plataforma têm uma validação se o utilizador logado tem permissão para acedê-las. Caso contrário, essas funcionalidades não são visíveis ao utilizador.

Quando o utilizador se autentica na plataforma, como referido anteriormente, o sistema comunica ao LS esse pedido de autenticação e, é este sistema que valida se o utilizador pode ou não aceder à plataforma. Quando o LS responde ao sistema que o utilizador é válido, juntamente envia uma lista de todas as ações que o mesmo pode realizar na plataforma. Essa lista são as permissões de acesso às funcionalidades na plataforma.

5.2. Usabilidade e *Design* de Interação

Usabilidade é um termo usado para definir a facilidade de utilização de um determinado produto. Usabilidade pode ser encontrado em qualquer produto de utilização. A nível informático, se uma aplicação é simples de utilizar, o utilizador facilmente realiza as tarefas que deseja, cometendo poucos erros e memorizando as ações que vai fazendo.

Design de interação é um termo que exprime a ideia do *design* de qualquer produto, que seja interativo e dê suporte às atividades das pessoas. Winograd, em 1997, descreveu que *design* de interação é como “o projeto de espaços para comunicação e interação humana” (Resende, 2008). Resumidamente, pode-se dizer que este termo, numa aplicação, baseia-se no *interface*

desenvolvido que permite aos utilizadores interagirem com o sistema, criando experiências que resultam da interação com um produto.

A relação entre estes dois termos, encontra-se no facto de numa aplicação, a criação do *design* de interação necessitar de aplicar métodos de usabilidade, para que todas as funcionalidades da aplicação sejam utilizadas como o desejável.

Uma aplicação que crie boas experiências é um princípio para os utilizadores voltarem a usá-la. Além do presente projeto se tratar de uma plataforma interna, estes termos também são relevantes para a experiência que é oferecida aos diferentes tipos de utilizadores. Do lado do *backOffice*, para os administradores fazerem a gestão de uma forma rápida, simples e agradável. Do lado do *frontOffice*, para os colaboradores do tipo normais desejarem aceder constantemente à plataforma, para seguirem os seus percursos nos diversos jogos temporais.

A plataforma foi desenvolvida tendo em conta alguns princípios de *design* de interação e usabilidade (Molin & Abreu, 2011), nomeadamente, no que toca à eficácia, eficiência, segurança, utilidade, aprendizagem e, por fim, a usabilidade. A eficácia refere-se ao quanto o sistema é bom em fazer o que se espera dele. A eficiência refere-se à forma como o sistema facilita o utilizador na realização das suas tarefas. A segurança refere-se aos métodos de prevenção de situações indesejáveis em determinadas ações que o utilizador pode realizar. A utilidade indica se as funcionalidades do sistema são adequadas para os utilizadores. A aprendizagem refere-se à simplicidade para os utilizadores conhecerem as funcionalidades básicas, de modo a facilmente conseguirem aceder às funcionalidades mais complexas.

De seguida apresentam-se alguns exemplos e métodos integrados no desenvolvimento da plataforma que respondem aos princípios de *design* de interação e usabilidade.

Menus

Os menus da plataforma foram construídos tendo em conta as estruturas base decididas de cada componente, nomeadamente, o menu do *backOffice* e *frontOffice*. No entanto, no desenvolvimento dos menus considerou-se o facto de as estruturas serem amplas e com

pouca profundidade de níveis hierárquicos, para evitar que os utilizadores se percam na procura de funcionalidades. Quando o utilizador navega para um nível mais baixo da hierarquia, o sistema realça a opção inicial selecionada, com o objetivo de evitar que o utilizador perca o contexto.

Outra característica relevante no menu do *backOffice* verifica-se nas opções se apresentarem ordenadas por ordem decrescente do uso de funcionalidades. Por exemplo, prevê-se que as funcionalidades mais utilizadas serão a gestão de eventos e de *badges*. Portanto, faz sentido ambas as funcionalidades se encontrarem no topo do menu. Ainda neste menu, verifica-se que todas as configurações básicas necessárias que permitem que o sistema funcione normalmente, encontram-se agrupadas na opção de “Configurações”.

Todos os conteúdos dos menus da plataforma estão acompanhados por diferentes ícones representativos que permitem interação e que possuem uma ação agregada. Os ícones escolhidos tentam aproximar-se o mais real do significado de cada ação a que estão associados, tornando as opções apresentadas mais interativas e perceptíveis aos olhos dos utilizadores.

Os menus principais da plataforma encontram-se sempre visíveis, permitindo que de uma forma fácil e rápida o utilizador consiga mudar de funcionalidade.

Formulários

Os formulários caracterizam-se como sendo o meio principal de manipulação de dados entre os utilizadores e o sistema. Os formulários criados contêm diferentes tipos de entrada de dados, tais como texto, imagens e opções de seleção a partir de uma lista de dados. Com isto,

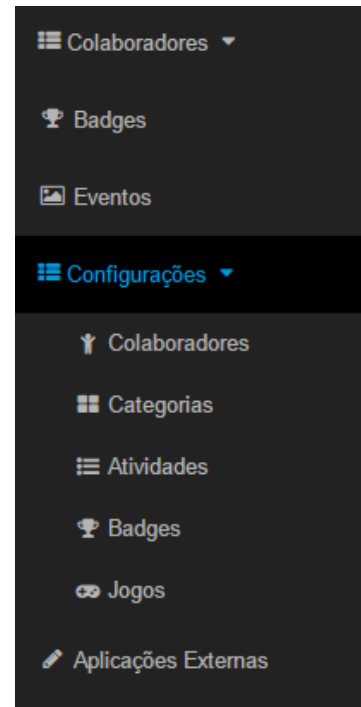


Fig. 63 - Menu principal do *backOffice*

teve-se em consideração diferentes métodos que permitissem diminuir a probabilidade dos utilizadores cometerem erros no preenchimento dos diferentes formulários, nomeadamente:

- Todas as entradas de dados que o sistema consiga obter a informação antes do utilizador preencher o formulário, apenas são apresentados esses dados de modo o utilizador certificar que são válidos. Por exemplo, na inserção de um novo evento manual pela área dos “Colaboradores”, primeiro o utilizador seleciona qual o colaborador para realizar a inserção de eventos e, posteriormente, procede para o preenchimento dos dados devidos. O sistema apresenta no formulário o nome do utilizador que realiza o evento mas não permite alteração dessa informação;
- Todos os dados de preenchimento obrigatório encontram-se acompanhados com “*”. Deste modo, diminui-se os casos do utilizador tentar guardar o processo no sistema sem as informações estarem devidamente todas preenchidas e, conseqüentemente, levando o utilizador a erro obtendo uma mensagem de erro. Além disto, nos dados de entrada de dados previstos, o sistema automaticamente já preenche os formulários com essa informação. Por exemplo, a inserção de um evento para o jogo temporal corrente;
- Todos os campos possuem um título curto, indicando quais as informações que o utilizador deve inserir;
- Teve-se o cuidado de avaliar todas as entradas de informação e, sempre que possível desenhou-se caixas de listagem em vez de caixas de texto. As caixas de listagem evitam os utilizadores de cometerem erros de escrita e não obrigam aos mesmos a memorizarem informação;
- Nos formulários com uso de datas é utilizado um calendário em que o utilizador consegue navegar nele e escolher uma data fixa. Deste modo, evita-se erros de introdução de datas ou em formatos inadequados;
- A partir das caixas de texto dos formulários, na introdução de dados, que representam formatos ou são de tipos comuns, são feitas determinadas validações. Por exemplo, a

introdução de *e-mail* obedece determinadas regras universais. O sistema faz validação destas situações gerais;

- Quando o utilizador comete um erro no preenchimento do formulário, o sistema indica onde se encontra o erro com uma linguagem simples e direta. Os restantes dados que estavam corretamente inseridos permanecem visíveis no formulário;
- Relativamente à organização dos objetos e das informações presentes no *interface*, todos os formulários são coerentes. Os botões principais de finalizar a ação, por exemplo, de guardar ou cancelar, encontram-se no canto superior direito. Teve-se o cuidado de colocar, praticamente, todos os objetos do *interface* justificados. Além disso, existem determinados formulários dinâmicos, ou seja, certos objetos de entrada de dados só são visíveis ao utilizador se forem mesmo necessários. Por exemplo, na inserção manual de um evento individual, o utilizador não necessita de inserir outros colaboradores, tornando esse objeto dinâmico consoante o tipo de evento;
- O botão de retorno, ou seja, o botão com acesso à saída do formulário voltando para o *interface* anterior, encontra-se em todas as páginas no canto superior direito com uma cor mais neutra do que os restantes botões de conclusão da ação;
- Quando um utilizador termina o preenchimento de um determinado formulário e guarda as informações, o sistema volta diretamente à página anterior que abriu o formulário e, mostra uma mensagem informativa ao utilizador com a indicação do resultado da ação, ou seja, se obteve-se sucesso ou insucesso.

A figura seguinte apresenta um formulário com algumas das características referidas.

Eventos Adicionar Novo Evento

Colaborador: Maria Martins

Jogo *: Nome JogoAtual

Data *: 2015-09-14

Categoria *: Escolher Categoria

Atividade *: Escolher Atividade

Peso: [dropdown]

Pontuação da Atividade: [input]

Pontuação Ponderada: [input]

Tipo de Evento: Individual

Su	Mo	Tu	We	Th	Fr	Sa
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3

Fig. 64 – Exemplo de formulário

Tabelas

As tabelas permitem reunir e apresentar aos utilizadores grandes quantidades de informação de uma forma concisa e organizada. Maioritariamente das tabelas desenvolvidas, que alegam fortes probabilidades de apresentar grandes quantidades de informação, estão acompanhadas com filtros de pesquisa. Um exemplo são as tabelas que expõem a informação de eventos. A funcionalidade de filtrar dados tornando as tabelas e as listagens que apresentam dinâmicas, aumenta a interação do utilizador, permitindo que de um modo simples e rápido encontre a informação que pretende. Os filtros de pesquisa existentes nas diversas tabelas de listagens variam consoante a relevância das informações.

As tabelas desenvolvidas encontram-se com os conteúdos organizados de uma forma distinta. O tipo de conteúdo a apresentar é relevante para a organização dos dados. Por exemplo, a apresentação de eventos numa tabela é mais interessante se o sistema mostrar, inicialmente, quais os últimos eventos realizados. Deste modo, a tabela encontra-se organizada descendentemente por data. Enquanto, na apresentação da lista de colaboradores, provavelmente, é mais interessante se os dados se encontrarem organizados ascendentemente pelo nome.

A paginação nas tabelas é outra funcionalidade implementada que permite aos utilizadores navegarem entre os dados apresentados, ajudando-os a encontrar a informação pretendida. Nem todas as tabelas possuem paginação, pois a quantidade de informação que apresentam não se justifica. Esta funcionalidade aparece visível aos utilizadores quando as tabelas apresentam mais do que 10 registos, ou seja, linhas de informação na tabela.

A imagem seguinte apresenta um exemplo de uma tabela com estas características.

Histórico Maria Martins

Jogo
Escolher Jogo

Categoria
Escolher Categoria

Pesquisar

Novo Evento

Cancelar

Data	Atividade	Categoria	Pontuação Obtida	
2015-06-29	Formação	Educação	40	+
2015-06-29	Formação	Educação	20	+
2015-06-29	Formação	Educação	20	+
2015-06-29	Formação	Educação	20	+
2015-06-29	Formação	Educação	20	+
2015-06-29	Formação	Educação	20	+
2015-06-29	Formação	Educação	20	+
2015-06-29	Formação	Educação	20	+
2015-06-29	Formação	Educação	20	+
2015-06-29	Formação	Educação	20	+
2015-06-29	Formação	Educação	20	+

<
2
3
4
>

Fig. 65 - Exemplo de interface de tabela

Outros

Além das características referidas, existem outros pormenores implementados na plataforma que obedecem aos princípios de *design* de interação e usabilidade, nomeadamente:

- O acesso às funcionalidades mais complexas, incluindo os atalhos que existem ao longo da plataforma, aparecem sobressaídos com uma cor mais forte. Com isto, os utilizadores conseguem visualizar mais rapidamente o acesso às diversas funcionalidades.
- O botão de acesso a funcionalidades que implicam um novo registo é coerente em todos os *interfaces*, pois encontra-se sempre no canto superior direito;

- Todos os utilizadores podem realizar ações que na verdade não eram o que pretendiam. Muitas dessas ações podem levar a situações que dificilmente o utilizador consegue recuperar. O sistema tenta prevenir estes erros que podem levar a consequências, mostrando ao utilizador mensagens de aviso ou de confirmação. Por exemplo, quando o utilizador seleciona opção de eliminação de dados, o sistema mostra ao utilizador uma pequena janela sobressaída na *interface*, com uma mensagem para o utilizador confirmar a ação;

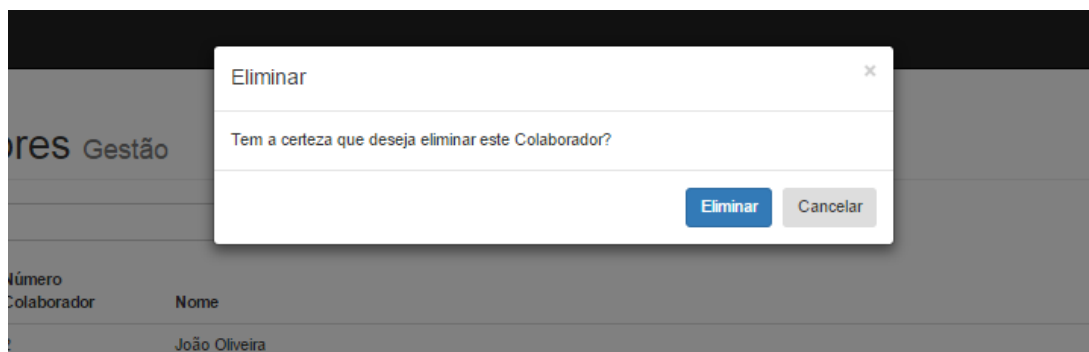


Fig. 66- Exemplo de mensagem de prevenção de erros

- As ações que cada utilizador realiza, no fim, têm sempre pelo menos uma mensagem de resposta visíveis, caso o resultado da ação tenha sido realizado com sucesso ou insucesso. Estas mensagens são fundamentais para o utilizador se manter informado dos resultados obtidos pelas ações que realiza bem como sobre o estado do sistema. As imagens seguintes representam exemplos dessas mensagens de resposta, tanto de sucesso como insucesso.

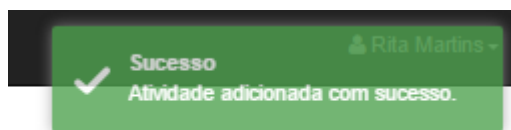


Fig. 68 - Mensagem de resposta de ação com insucesso

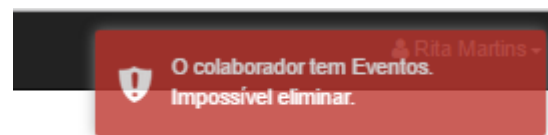


Fig. 67 - Mensagem de resposta de ação com sucesso

5.3. Segurança da Plataforma

No desenvolvimento de uma aplicação é necessário, paralelamente, ter sempre em consideração a importância da segurança. Hoje em dia, as informações *online*, inclusive dados pessoais, encontram-se expostas a qualquer tipo de intruso que ataque com sucesso os sistemas. Atualmente, existem diversos tipos de ataques (Campos, Neto, & Canto, sem data) que afetam a segurança das páginas Web, onde facilmente a informação pode ser exibida.

A segurança numa plataforma pode apresentar várias vertentes. A importância da segurança de um sistema pode encontrar-se desde o ambiente da própria plataforma, à credibilidade, à suscetibilidade que provoca aos utilizadores e, principalmente, à segurança da informação que contempla a plataforma. Um sistema que esteja protegido com os mais diversos métodos de segurança, nunca se caracterizará como um sistema totalmente seguro.

No desenvolvimento do presente projeto teve-se, principalmente, em consideração a importância de manter as informações que existem no sistema protegidas. Para tal, aspetos como a autenticação, a confidencialidade, a integridade e a verificabilidade foram refletidos e colocados em prática durante o desenvolvimento do sistema.

O primeiro mecanismo de segurança implementado no desenvolvimento do sistema foi a utilização de uma *framework*, nomeadamente, o CodeIgniter como já referido. A utilização de uma *framework* traz muitas vantagens, inclusive na área de segurança. O CodeIgniter, por padrão, já oferece uma série de opções e configurações que ajudam a proteger a nossa aplicação em desenvolvimento, às quais se destacam:

- Limita os componentes de cada URL, diminuindo a possibilidade de diminuir ataques que possam surgir através de URL, por exemplo, executar scripts no URL – ataque Document Object Model (DOM) Based XSS (acunetix.com, 2013);
- Disponibiliza uma classe que permite filtrar os dados recebidos por POST, cookies, URL's, etc, de modo validar se os dados recebidos apresentam-se como duvidosos;
- Disponibiliza uma classe de validação de campos, por exemplo, preenchimento de formulários pelo utilizador;

- Disponibiliza uma função capaz de determinar o tipo de dados que estão a ser manipulados na base de dados de suporte, e caso não apresentem plicas nas situações de inserção de informações a mesma função acrescenta-as. Este mecanismo protege o sistema de ataques, especialmente, de *SQL Injection* (php.net, sem data);
- Disponibiliza uma configuração que previne o sistema de ataques *CSRF (Cross Site Request Forgery)* (DuPaul, sem data), prevenindo situações em que terceiros mal-intencionados conseguem aceder a métodos de inserção de dados (formulários, URL's, ...) no sistema e façam alterações aos dados, em momentos que estão a ser preenchidos pelos utilizadores reais;
- Disponibiliza mecanismos de encriptação de dados, de forma simples e rápida. Estes mecanismos de criptografia são muito úteis quando se deseja guardar informações confidenciais, por exemplo, senhas de acesso. No desenvolvimento do projeto, como referido anteriormente, o processo de autenticação no sistema interage com o LS. No entanto, a senha de acesso dos utilizadores também é guardada na base de dados de suporte encriptada pelo algoritmo SHA-1 (movable-type.co.uk, sem data);
- Disponibiliza configurações de variáveis de sessões que permite uma gestão das sessões criadas no sistema de uma forma simples. Quando um utilizador acede à plataforma é criada uma sessão com um tempo de vida de duas horas. Após esse tempo, o sistema automaticamente mata a sessão, simulando a saída automática do utilizador da plataforma.

Além destes mecanismos aplicados e desenvolvidos no sistema, existem também outros componentes, abordados anteriormente, que têm a função de manter o sistema em segurança.

Relativamente à API desenvolvida, que permite a comunicação de outros sistemas externos, contém um mecanismo de validações sucessivas de cada informação que compõe o corpo dos pedidos solicitados à API, bem como a gestão de permissões de cada aplicação externa. Deste modo, aumenta a segurança da comunicação que pode chegar via API.

Relativamente ao LS, que tem grande responsabilidade no processo de autenticação na plataforma e nas permissões dos utilizadores, também é um componente muito importante que se relaciona com o âmbito de segurança no sistema. É o LS que tem a maioria da responsabilidade de limitar ou permitir o acesso de utilizadores à plataforma. Se o sistema LS não apresentasse boa segurança, seria perigoso para o sistema do projeto.

Concluindo, os mecanismos de segurança são características que devem ser sempre refletidas e aplicadas, de modo a permitir que o sistema no fundo seja confiável para os utilizadores e para a organização.

5.4. Testes Funcionais e de Usabilidade

A prática de testes num desenvolvimento de qualquer tipo de produto é sempre um procedimento importante e essencial. A realização de testes permite identificar problemas nas funcionalidades desenvolvidas ou possibilita o reconhecimento de falta de outras funcionalidades. A usabilidade é uma característica muito importante que também deve ser testada com testes.

O contacto de um produto com os futuros utilizadores é o princípio para o mesmo ser bem utilizado e ter sucesso no futuro. Se houver a possibilidade de serem efetuados testes durante o desenvolvimento de um produto melhor, pois consegue-se obter resultados reais, identificar problemas e fazer as devidas alterações antes de terminar a aplicação.

Como já referido anteriormente no subcapítulo “1.4. Metodologias e Planeamento de Trabalho”, após a principal recolha de requisitos, no desenvolvimento do presente projeto foi utilizado a metodologia ágil *scrum*. O uso desta metodologia permitiu que ao longo do desenvolvimento fossem efetuados testes com futuros utilizadores às diferentes funcionalidades implementadas. Este procedimento trouxe grandes vantagens. Em cada *sprint* adaptado permitiu a realização de testes, onde foram identificados problemas, possibilitando fazer as devidas alterações ainda no mesmo *sprint*. Deste modo, evitou-se que só no fim do

desenvolvimento do projeto fossem realizadas as devidas modificações, que por vezes obrigam a alteração de outras funcionalidades mexendo com grande parte do projeto.

Os testes efetuados pelos utilizadores foram realizados com a presença da autora deste projeto que permitiu observar e identificar falhas de usabilidade. Foram pedidos aos utilizadores que realizassem determinadas funcionalidades, de modo a analisar se os utilizadores eram capazes de efetuar o pedido, bem como o tempo de demora e os erros que cometiam até alcançarem o objetivo.

A partir dos testes realizados, tanto a nível de funcionalidades como de usabilidade, houve alguns casos que implicaram novas reavaliações dos requisitos que tinham sido anteriormente definidos e, conseqüentemente, novos desenvolvimentos e novos testes. Um exemplo que chumbou nos testes realizados foi a inserção de novos eventos no sistema, que caracteriza-se como sendo uma função principal no projeto. O utilizador preocupou-se em encontrar um sítio para selecionar o colaborador desejado e só depois inserir o evento no sistema. O primeiro desenvolvimento da gestão de eventos permitia a inserção das informações do evento e, posteriormente, a seleção dos colaboradores que realizavam-no. Foi a partir dos resultados dos testes efetuados a este *sprint*, que surgiu o requisito de adaptar a plataforma de maneira a permitir aos utilizadores fazerem a gestão no âmbito de um colaborador, ou seja, primeiro escolhem um colaborador e depois geram os eventos ou os *badges* desse colaborador.

Resumindo, além dos testes realizados pelo autor do projeto foi uma grande vantagem ter a possibilidade de testar com futuros utilizadores do sistema.

6. Conclusão

A realização deste projeto permitiu adquirir conhecimentos sobre o tema *gamification* e as relevâncias e vantagens que oferece quando implementado num sistema. Na introdução da presente dissertação revela-se que o projeto surgiu de uma lista de propostas apresentadas pela XLM à autora. Perante os projetos e os problemas que a organização anunciou, deu-se preferência ao projeto desenvolvido por existir grande noção desses problemas por parte da autora. Além disso, o facto do conceito *gamification* estar em evolução ocasionou mais interesse e motivação em estudar este âmbito. No fim desta dissertação, consegue-se concluir que, de facto, a aplicabilidade deste conceito é sem dúvida interessante para os utilizadores que torna os sistemas mais cativantes e vantajoso para as organizações.

A realização do presente projeto permitiu ainda aprofundar conhecimentos técnicos a níveis de desenvolvimento em arquitetura MVC, da *framework* CodeIgniter e, sobretudo, de desenvolvimento para a *Web*, que não é de perto o que a autora realiza diariamente.

A plataforma desenvolvida é composta por os requisitos base que permitem iniciar um processo de *gamification* dentro da organização. Verifica-se que todos os requisitos inicialmente previstos foram concluídos com sucesso. No entanto, existem muitas outras ideias de funcionalidades que serão uma mais-valia para o sistema. Porém, devido ao tempo limitado essas funcionalidades

ainda não foram implementadas. Contudo, um bom sistema não tem fim, pois novas necessidades e novas ideias aparecem diariamente. O desenvolvimento deste projeto não termina nesta fase. Prevê-se que ao longo do tempo, o sistema continuará a ser explorado e trabalhado de modo a ficar cada vez mais completo e adaptado aos utilizadores e às necessidades da organização.

A presente dissertação não apresenta resultados de testes concretos, pois implicaria que a plataforma entrasse em sistema ainda dentro do tempo limite e fossem realizados eventos suficientes que permitissem analisar o impacto da plataforma. O ideal seria a plataforma estar em ativo e disponível aos utilizador cerca de um ano, para conseguir-se comparar resultados e a evolução dos objetivos de uma forma fidedigna. Porém, o tempo escassa e não possibilita este acontecimento. No entanto, entre o período de entrega e a defesa do projeto, se for possível a obtenção de resultados, os mesmos serão apresentados.

Ao fim desta dissertação, como se pôde verificar, concluiu-se que o *gamification* consegue oferecer resultados muito positivos em muitos problemas que as organizações nos dias de hoje apresentam. O sistema desenvolvido, decerto, proporcionará bons resultados à organização. De referir que a plataforma está preparada para ser utilizada na XLM, contudo de uma forma rápida e simples também se adequa a outras organizações.

6.1. Propostas Futuras

Como referido, o presente projeto caracteriza-se por conter as funcionalidades base, de modo a permitir o sistema iniciar-se dentro da organização. No entanto, existem muitas outras ideias de novas funcionalidades que devido à escassez de tempo ficaram pendentes. Pretende-se que ao longo do tempo algumas delas sejam implementadas dependendo, logicamente, das preferências da organização.

A nível dos jogos temporais, para o presente projeto, este requisito não foi dado como prioritário, pois o que se pretende apenas é limitar a gestão de eventos. No entanto, para um melhoramento de currículo de cada colaborador e como preferência pessoal, sugere-se a criação

de um método capaz de transferir automaticamente todas as atividades bem como todas as categorias para os novos jogos. Deste modo, a visualização de histórico do perfil de cada colaborador poderá enriquecer, permitindo visualizar as informações de jogos temporais fechados.

Ainda dentro do componente *frontOffice*, pretende-se implementar mais funcionalidades que permitem a interação entre os utilizadores, tornando a plataforma mais motivadora, embora o objetivo do sistema não é que os colaboradores movimentem constantemente este sistema, mas sim os restantes. No entanto, a organização poderá ganhar se ao longo do tempo esse objetivo for alterado. Por exemplo, na partilha de informações internas, na publicidade de eventos e desafios, entre outros.

A área de estatísticas também pode conter novas funcionalidades que possibilitam melhores análises de informação. Por exemplo, a implementação de filtros dinâmicos que permitem a escolha de resultados pelos utilizadores, ou seja, os utilizadores criam os seus filtros de pesquisa em vez de aparecer visivelmente todos os filtros possíveis.

Na componente da API sugere-se a criação de novas funções que permitem uma melhor interação entre sistemas. Por exemplo, a eliminação e alteração de informação de eventos, a atribuição de *badges*, permitindo aos utilizadores ganharem *badges* fora do sistema mas visíveis dentro dele.

De uma forma resumida, ao longo do tempo pretende-se estudar e implementar novas funcionalidades e/ou aperfeiçoar as já existentes de modo a adequarem-se às necessidades da organização e aos utilizadores, sempre com a finalidade de alcançar cada vez mais impactos e resultados positivos com o sistema desenvolvido.

REFERÊNCIAS

- acunetix.com. (2013). DOM-based Cross-Site Scripting (XSS) Explained. Obtido de <http://www.acunetix.com/blog/articles/dom-xss-explained/>
- Adelakun, O. (2012). *Outsourcing Best Practices: Sprint Process Improvement Initiative*. Obtido de http://r.search.yahoo.com/_ylt=A7x9UnLO6RpWNSwAeBhHjAx.;_ylu=X3oDMTBydWpobjZlBHNIYwNzcgRwb3MMDMQRjb2xvA2lyMgR2dGikAw--/RV=2/RE=1444633166/RO=10/RU=http%3a%2f%2fwww.iaop.org%2fDownload%2fDownload.aspx%3fid%3d2408/RK=0/RS=73G_t1Qsc2k23ITkwp7kE8Uqv94-
- Alves, F. (2014). *Gamification: Como criar experiências de aprendizagem engajadoras. Um guia completo: do conceito à prática*. DVS Editora.
- apachefriends.org. (sem data). XAMPP Installers and Downloads for Apache Friends. Obtido 20 de Março de 2015, de <https://www.apachefriends.org/index.html>
- Arata, S. (2014). Mihaly Csikszentmihalyi: Estado de Flow (Fluxo) como elemento de realização e alta performance. Obtido 10 de Janeiro de 2014, de <http://www.arataacademy.com/port/mihaly-csikszentmihalyi-estado-de-flow-fluxo-como-elemento-de-realizacao-e-alta-performance/>
- Araújo, D. (2014). Tecnologia para motivar equipas.. em Portugal! *novaweb*. Obtido de <http://novaweb.pt/blog/tecnologia-para-motivar-equipas-em-portugal/>
- badgeville.com. (2013). In-App Gamification [Text]. Obtido 5 de Janeiro de 2015, de <https://badgeville.com/products/in-app-gamification/>
- Bazolli, A. (2013). Gamification motiva e amplia a produtividade em contact centers. Obtido de <http://www.televendasecobranca.com.br/estrategia-e-modelagem/gamification-motiva-e-amplia-a-produtividade-em-contact-centers-27577/>
- bigdoor.com. (2014). Our Products. Obtido 5 de Janeiro de 2015, de <http://bigdoor.com/products/>
- Bookwalter, J. R. (2011, Abril). What is CSS3? Obtido 24 de Março de 2015, de http://www.maclife.com/article/features/what_css3
- bunchball.com. (2015). Nitro. Obtido 10 de Dezembro de 2014, de <http://www.bunchball.com/products/nitro>
- Campos, F., Neto, N. S., & Canto, W. F. (sem data). Web 2.0 Sob a Perspectiva da Segurança. IPT - Instituto de Pesquisas Tecnológicas do Estado de São Paulo. Obtido de <http://dainf.ct.utfpr.edu.br/~maziero/lib/exe/fetch.php/ceseg:2009-sbseg-mc6.pdf>
- cnpd.pt. (1998, Setembro 24). Lei n.º 67/98 de 26 de Outubro. Obtido 13 de Setembro de 2015, de http://www.cnpd.pt/bin/legis/nacional/lei_6798.htm
- Covington, M. V. (2000). Intrinsic Versus Extrinsic Motivation in Schools A Reconciliation. *Current Directions in Psychological Science*, 9(1), 22–25. <http://doi.org/10.1111/1467-8721.00052>
- cundari.com. (2014). Cundari. Obtido 16 de Dezembro de 2014, de <http://www.cundari.com/cases/pain-squad/>
- Deterding, S. (2011). Gamification: Using Game Design Elements in Non-Gaming Contexts. Apresentado na Conference on Human Factors in Computing Systems.
- developer.linkedin.com. (2015). Getting Started: Javascript SDK | LinkedIn Developer Network. Obtido 15 de Agosto de 2015, de <https://developer.linkedin.com/docs/js-sdk>
- Drucker, P. F. (1999). *Sociedade Pós-Capitalista*. Pioneira.
- DuPaul, nEIL. (sem data). Cross-Site Request Forgery Guide: Learn All About CSRF Attacks and CSRF Protection. Obtido 21 de Agosto de 2015, de <http://www.veracode.com/security/csrf>

- ellislab.com. (2015). CodeIgniter / EllisLab. Obtido 13 de Março de 2015, de <https://ellislab.com/codeigniter>
- emmawynne09.wordpress.com. (2013). Nike Plus Running, Gamification in Practice. Obtido de <https://emmawynne09.wordpress.com/2013/03/13/nike-plus-running-gamification-in-practice/>
- Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. UNIVERSITY OF CALIFORNIA, IRVINE. Obtido de <http://jpkc.fudan.edu.cn/picture/article/216/35/4b/22598d594e3d93239700ce79bce1/7ed3ec2a-03c2-49cb-8bf8-5a90ea42f523.pdf>
- Figueiredo, R. M. da C., Bremer, C. F., & Maldonado, J. C. (2008). Evolução dos modelos de outsourcing: o estado da arte da literatura dos novos provedores de serviços de aplicativos. *Perspectivas Em Ciência Da Informação*, 8(1). Obtido de <http://portaldeperiodicos.eci.ufmg.br/index.php/pci/article/view/373>
- Foster, J. A. (2012). Gamification as a Strategy for Promoting Deeper Investigation in a Reverse Engineering Activity. Obtido de http://r.search.yahoo.com/_ylt=A7x9Un8zwxpWkS8AWjBHjAx;_ylu=X3oDMTBydWpobjZlBHNIYwNzcgRwb3MMDMQRjb2xvA2lyMgR2dGlkAw--/RV=2/RE=1444623283/RO=10/RU=http%3a%2f%2fwww.asee.org%2fpublic%2fconferences%2f8%2fpapers%2f5456%2fdownload/RK=0/RS=6h1SCO2jfvmk114IBH97RnKMWzk-
- Furtado, G. (2013, Abril). O que é um SGBD? *Dicas de Programação*. Obtido de <http://www.dicasdeprogramacao.com.br/o-que-e-um-sgbd/>
- Galhanas, C. R. G. (2009). A motivação dos recursos humanos nos novos modelos de gestão da administração pública. Obtido de <https://www.repository.utl.pt/bitstream/10400.5/1757/1/TESE%20DE%20MESTRADO%20-%20CARLA%20GALHANAS%20-%20VERS%C3%83O%20FINAL.pdf>
- gamificationportugal.com. (2014). Gamification | Gamificação Portugal. Obtido de <http://gamificationportugal.com/>
- gartner.com. (2012). Gartner's 2012 Hype Cycle for Emerging Technologies Identifies. Obtido 2 de Janeiro de 2015, de <http://www.gartner.com/newsroom/id/2124315>
- Gay, C. L., & Essinger, J. (2000). *Inside Outsourcing: The Insider's Guide to Managing Strategic Sourcing*. Nicholas Brealey Pub.
- getbootstrap.com. (2015). Bootstrap. Obtido 2 de Março de 2015, de <http://getbootstrap.com/2.3.2/index.html>
- getpostman.com. (sem data). Postman. Obtido 4 de Abril de 2015, de <https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcdcbncdddomop>
- gigya.com. (2014). Gamification API. Obtido 19 de Dezembro de 2014, de <http://www.gigya.com/gamification/gamification-api/>
- Greengard, S. (2014). DIRECTV Channels Gamification and Crowdsourcing. Obtido de <http://www.baselinemag.com/innovation/directtv-channels-gamification-and-crowdsourcing.html>
- Groh, F. (2012). *Gamification: State of the Art Definition and Utilization* (Artigo). Institute of Media Informatics. Obtido de http://www.click4it.org/images/7/7c/Gamification_Sate_of_the_art_deifinition_and_utilization_Groh.pdf#page=39
- Hanifin, B. (2014). Badgeville & Kendall-Jackson Increase Customer Engagement by 65% - See more at: [http://loyalty360.org/loyalty-today/article/badgeville-kendall-jackson-increase-customer-engagement-by-65](http://loyalty360.org/loyalty-today/article/badgeville-kendall-jackson-increase-customer-engagement-by-65#sthash.uEarS4fQ.dpuf)
- Hunicke, R., LeBlanc, M., & Zubek, R. (2004). MDA: A Formal Approach to Game Design and Game Research. Apresentado na Proceedings of the AAAI Workshop on Challenges in Game AI (pp. 04–04). Obtido de <http://www.aaai.org/Papers/Workshops/2004/WS-04-04/WS04-04-001.pdf>

- Janssen, C. (2010). What is Unified Modeling Language (UML)? - Definition from Techopedia. Obtido 18 de Junho de 2015, de <http://www.techopedia.com/definition/3243/unified-modeling-language-uml>
- javascript.com. (2015). JavaScript.com. Obtido 20 de Março de 2015, de <https://www.javascript.com>
- Jonathan, B. (2014). *Gamification for Human Factors Integration: Social, Education, and Psychological Issues: Social, Education, and Psychological Issues*. IGI Global.
- jquery.com. (2015). jQuery.ajax() | jQuery API Documentation. Obtido de <http://api.jquery.com/jquery.ajax/>
- Kaufman, A. (2013). A gamification and Innovation Success Story: Portugal Telecom. Obtido de <http://ecoopportunity.net/2013/06/a-gamification-and-innovation-success-story-portugal-telecom/>
- Kehoe, D. (2013). What is Ruby on Rails? · RailsApps. Obtido de <http://railsapps.github.io/what-is-ruby-rails.html>
- Klein, P. G. (2008). The Make-or-Buy Decisions: Lessons from Empirical Studies. Em C. Ménard & M. M. Shirley (Eds.), *Handbook of New Institutional Economics* (pp. 435–464). Springer Berlin Heidelberg. Obtido de http://link.springer.com/chapter/10.1007/978-3-540-69305-5_18
- lagoon.pt. (2013). Lagoon® - Transforme o seu negócio com o nosso software! Obtido 17 de Agosto de 2015, de <http://www.lagoon.pt/lagoonsecurity/>
- learningtech.com. (2013). Learning Teach gamify your business. Obtido 13 de Dezembro de 2014, de <http://www.learningtech.com.br/gamification.html>
- Lee, J. Q., McInerney, D. M., Liem, G. A. D., & Ortiga, Y. P. (2010). The relationship between future goals and achievement goal orientations: An intrinsic–extrinsic motivation perspective. *Contemporary Educational Psychology*, 35(4), 264–279. <http://doi.org/10.1016/j.cedpsych.2010.04.004>
- Li, C., Dong, Z., Untch, R. H., & Chasteen, M. (2013). Engaging Computer Science Students through Gamification in an Online Social Network Based Collaborative Learning Environment. Obtido de <http://www.ijiet.org/papers/237-T125.pdf>
- Lucas, C., Gasparini, I., & Berkenbrock, C. (2013). Investigando o uso de gamificação para aumentar o egajamento em sistemas colaborativos. Obtido de <http://ceur-ws.org/Vol-1051/paper4.pdf>
- Marczewski, A. (2008). *Gamification: A Simple Introduction*. Andrzej Marczewski.
- Molin, F., & Abreu, M. (2011). *Princípios de Design de Interação*. Obtido de <http://pt.slideshare.net/FelipeMolin/principios-de-design-de-interao-9599420>
- Morais, R. (2010). A importância da gestão de IDI na gestão estratégica. Obtido de <http://repositorio.ucp.pt/handle/10400.14/3680>
- movable-type.co.uk. (sem data). SHA-1 Cryptographic Hash Algorithm | Movable Type Scripts. Obtido 28 de Setembro de 2015, de <http://www.movable-type.co.uk/scripts/sha1.html>
- Murray, E. J. (1967). *Motivação e Emoção*. Zahar.
- mysql.com. (2015). MySQL :: The world's most popular open source database. Obtido 20 de Março de 2015, de <https://www.mysql.com/>
- Nelson, M. J. (2012). Soviet and American Precursors to the Gamification of Work. Em *Proceeding of the 16th International Academic MindTrek Conference* (pp. 23–26). New York, NY, USA: ACM. <http://doi.org/10.1145/2393132.2393138>
- Nicolau, I. (2002). *Gestão do Conhecimento nas Organizações e Mercado de Serviços*. Lisboa: INDEG/ISCTE. Obtido de <http://cedo.ina.pt/docbweb/MULTIMEDIA/ASSOCIA/INTERNO/ELECTRON/E162.PDF>
- openbadges.org. (sem data). Mozilla Open Badges. Obtido 30 de Agosto de 2015, de <http://openbadges.org/>

- Paharia, R. (2015). Bunchball: Gamification Blog: Gamification Trends to Watch in 2015. Obtido de <http://www.bunchball.com/blog/post/1616/5-gamification-trends-watch-2015>
- Pasold, A. (2012). Acompanhando as tecnologias e tendências emergentes «. Obtido de <http://www.blogit.com.br/blog/2012/08/acompanhando-as-tecnologias-e-tendencias-emergentes/>
- Passos, E., Medeiros, D., Neto, P., & Clua, E. (2011). Turning Real-World Software Development into a Game. Apresentado na SBC - Proceedings of SBGames. Obtido de http://www.sbgames.org/sbgames2011/proceedings/sbgames/papers/comp/full/30-91552_2.pdf
- php.net. (sem data). PHP: SQL Injection - Manual. Obtido 22 de Agosto de 2015, de <http://php.net/manual/en/security.database.sql-injection.php>
- Praciano, E. (2014). Os benefícios e as vantagens do PHP. Obtido de <http://elias.praciano.com/2014/02/15-beneficios-e-vantagens-do-php/>
- Psychology, C. R. S. E. W. D. P. of C., & Professor, D. of P. both at U. of K.-L. S. J. L. A. (2001). *Handbook of Positive Psychology*. Oxford University Press.
- publico.pt. (2014). «Corrupção», «banco», «legionela», «selfie» e «xurdir» entre as candidatas a Palavra do Ano 2014. Obtido 2 de Janeiro de 2015, de <http://www.publico.pt/n1678052>
- Ramirez, A. O. (2000, Julho). Three-Tier Architecture. *Linux Journal*. Obtido de <http://www.linuxjournal.com/article/3508>
- recyclebank.com. (2013). Recyclebank. Obtido 3 de Janeiro de 2015, de <https://www.recyclebank.com/>
- Reiss, S. (2012). Intrinsic and Extrinsic Motivation. *Teaching of Psychology*, 39(2), 152–156. <http://doi.org/10.1177/0098628312437704>
- Resende, T. (2008). Design de interação: O que é? Obtido de <https://tanila.wordpress.com/2008/09/03/design-de-interacao-o-que-e/>
- restconsole.com. (sem data). REST Console. Obtido 5 de Abril de 2015, de <https://chrome.google.com/webstore/detail/rest-console/cokgbflfommojglbmbpenpphpikmonn>
- Rocha, J. A. de O. (2007). *Gestão de Recursos Humanos na Administração Pública*. Escolar Editora. Obtido de <https://books.google.es/books?hl=pt-PT&lr=&id=fCIPIR0rI7IC&oi=fnd&pg=PA13&dq=Gest%C3%A3o+de+Recursos+Humanos+na+Administra%C3%A7%C3%A3o+P%C3%BAblica.+Oliveira+Rocha&ots=APyNLSv8-6&sig=pCYfeiVlhExXVyMH4lLUgf1RLQ#v=onepage&q=Gest%C3%A3o%20de%20Recursos%20Humanos%20na%20Administra%C3%A7%C3%A3o%20P%C3%BAblica.%20Oliveira%20Rocha&f=false>
- Rouse, M. (2008). client/server (client/server model, client/server architecture). Obtido de <http://searchnetworking.techtarget.com/definition/client-server>
- Ryan, R. M., & Deci, E. L. (2000). Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions. *Contemporary Educational Psychology*, 25(1), 54–67. <http://doi.org/10.1006/ceps.1999.1020>
- Schaffer, J. (2013). What is Drupal? | Blue Coda. Obtido de <http://www.bluecoda.com/blog/what-is-drupal>
- Schonfeld, E. (2010). SCVNGR's Secret Game Mechanics Playdeck. Obtido de <http://techcrunch.com/2010/08/25/scvngr-game-mechanics/>
- Simões, J. (2012). Education & eLearning 2.0: Criar Avatares com Voki. Obtido de <http://edulearning2.blogspot.pt/2012/08/criar-avatares-com-voki.html>
- thelevelup.com. (2014). LevelUp. Obtido 23 de Dezembro de 2014, de <https://www.thelevelup.com/campaigns>

- Tremblay, M. A., Blanchard, C. M., Taylor, S., Pelletier, L. G., & Villeneuve, M. (2009). Work Extrinsic and Intrinsic Motivation Scale: Its value for organizational psychology research. *Canadian Journal of Behavioural Science/Revue canadienne des sciences du comportement*, 41(4), 213–226. <http://doi.org/10.1037/a0015167>
- Vallerand, R. J. (2014). Intrinsic and Extrinsic Motivation in Sport. Em *Encyclopedia of Applied Psychology* (Vol. 2, pp. 427–435). Université du Québec à Montréal, Montréal, Canada. Obtido de <http://www.er.uqam.ca/nobel/r26710/LRCS/papers/128.pdf>
- Vieira, D. (2014). Scrum: A Metodologia Ágil Explicada de Forma Definitiva. Obtido 18 de Junho de 2015, de <http://www.mindmaster.com.br/scrum/>
- Walker, G., & Weber, D. (1984). A Transaction Cost Approach to Make-or-Buy Decisions. *Administrative Science Quarterly*, 29(3), 373–391. <http://doi.org/10.2307/2393030>
- Werbach, K. (2014). (Re) Defining Gamification. *Springer Lecture Notes in Computer Science*, 8462. Obtido de http://works.bepress.com/kevin_werbach/3/
- Werbach, K., & Hunter, D. (2012). *For the Win: How Game Thinking Can Revolutionize Your Business*. Wharton Digital Press.
- yareah.com. (2015). Tech. Glenroy unwraps gamification solution to motivate sales performance. Obtido 6 de Julho de 2015, de <http://www.yareah.com/2015/05/05/tech-glenroy-unwraps-gamification-solution-to-motivate-sales-performance/>
- Zichermann, G., & Cunningham, C. (2011). *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps*. O'Reilly Media, Inc.
- zmescience.com. (sem data). Gamers solve decade old HIV puzzle in ten days. Obtido de <http://www.zmescience.com/research/studies/gamers-solve-decade-old-hiv-puzzle-in-ten-days/>

Anexo A – Casos de Uso

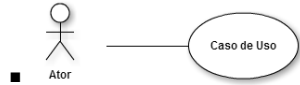
ÍNDICE

1. Introdução	2
2. Diagrama de casos de uso do módulo “API”	3
3. Diagrama de casos de uso do módulo “Gestão de Acessos e Permissões”	4
4. Diagrama de casos de uso do módulo "Perfis de Colaboradores"	5
5. Diagrama de casos de uso de módulo "Gestão de Colaboradores"	6
6. Diagrama de casos de uso do módulo "Gestão Jogos Temporais"	7
7. Diagrama de casos de uso do módulo "Gestão Categorias"	8
8. Diagrama de casos de uso do módulo "Gestão Atividades"	9
9. Diagrama de casos de uso do módulo “Gestão Eventos”	10
10. Diagrama de casos de uso do módulo “Gestão Badges”	11
11. Diagrama de casos de uso do módulo "Estatísticas"	12
12. Diagrama de casos de uso do módulo "Gestão Aplicações Externas"	13

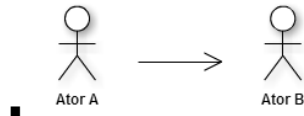
1. Introdução

O presente documento apresenta os módulos referidos na dissertação no subcapítulo “4.3. Casos de Uso”, através dos respetivos diagramas de casos de uso com mais detalhe.

Os diagramas representados no presente anexo contém três tipos de relacionamentos:



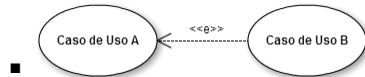
Define uma funcionalidade direta do sistema do ponto de vista do utilizador.



Significa que os casos de uso disponíveis ao ator B, também são disponíveis ao ator A.



Significa que o caso de uso B é essencial para o caso de uso A.



Significa que o caso de uso B não é essencial para o caso de uso A. No entanto, a partir do caso de uso A pode surgir o caso de uso B.

Alguns diagramas apresentam ainda “*Extension points*”, normalmente são referidos em listagens que fazem parte de uma outra gestão, ou seja, de um outro módulo.

2. Diagrama de casos de uso do módulo “API”

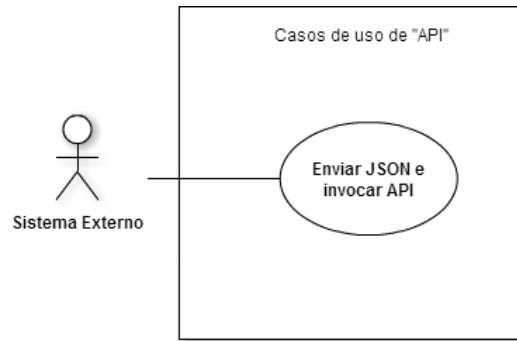


Fig. 69 - Diagrama caso de uso módulo "API"

3. Diagrama de casos de uso do módulo “Gestão de Acessos e Permissões”

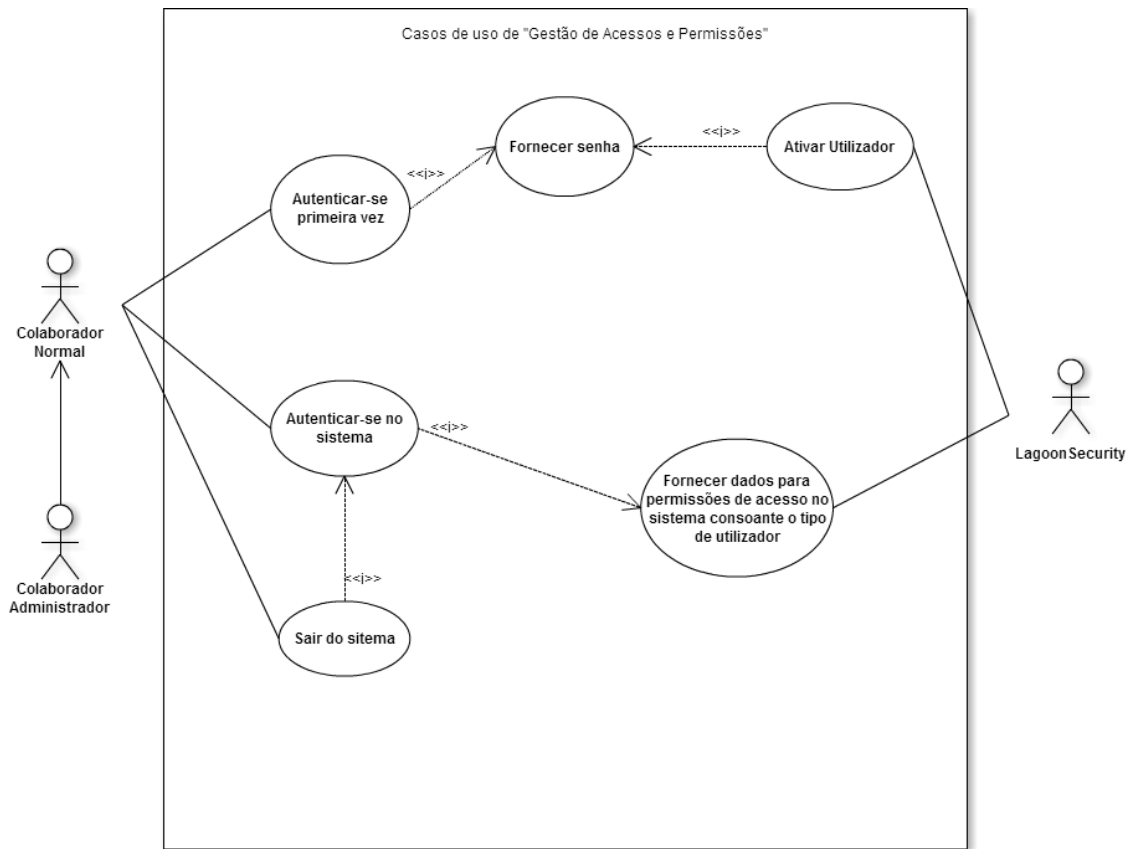


Fig. 70 - Diagrama caso de uso módulo "Gestão de Acessos e Permissões"

4. Diagrama de casos de uso do módulo "Perfis de Colaboradores"

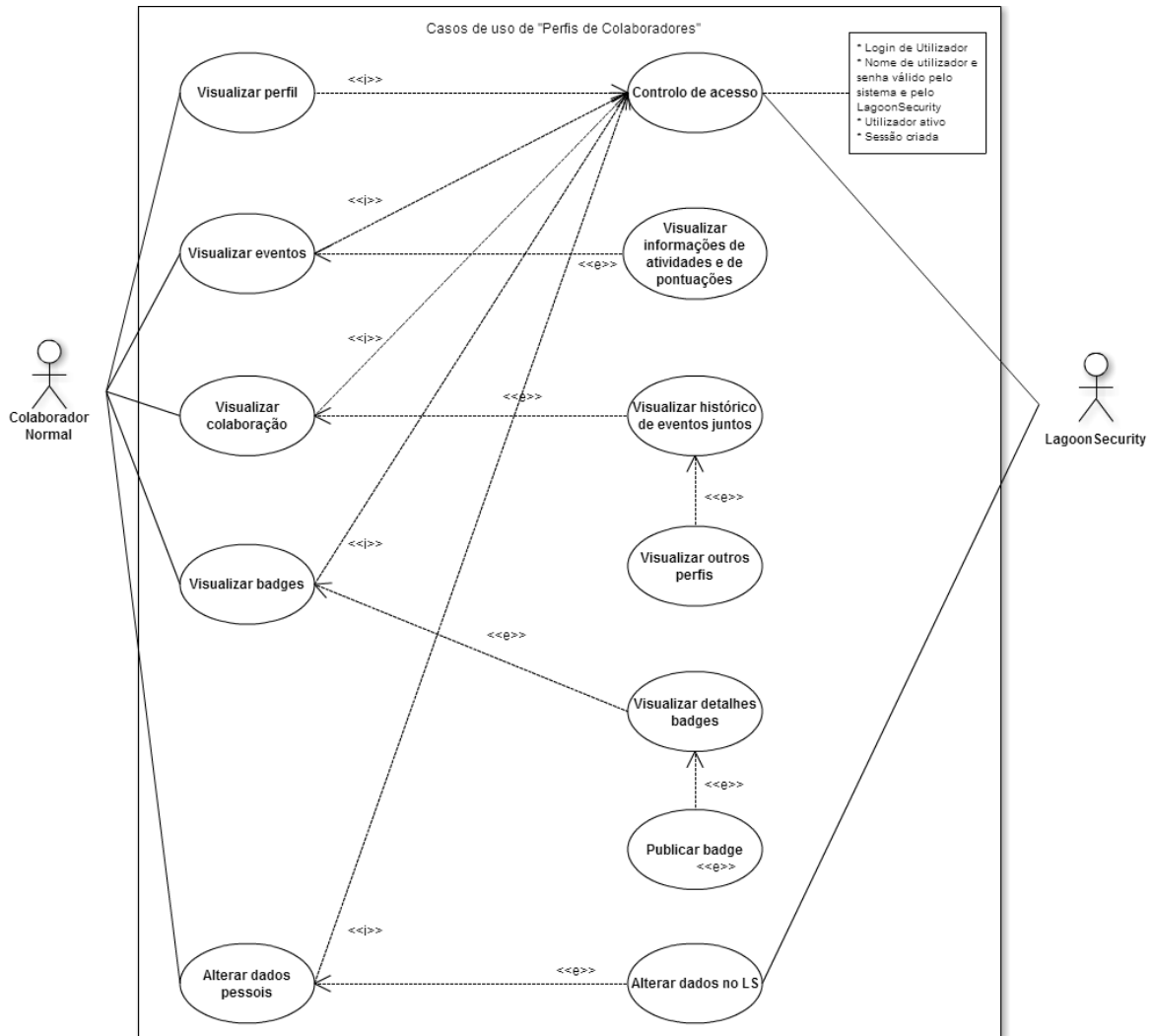


Fig. 71 - Diagrama caso de uso módulo "Perfis de Colaboradores"

Notas:

O caso de uso “Controlo de acesso” está presente para todas as funcionalidades e tem de abranger todos os requisitos referidos na janela que se relaciona.

O caso de uso “Visualizar colaboração” permite ao utilizador ver os outros colaboradores, com quem já realizou eventos juntos.

5. Diagrama de casos de uso de módulo "Gestão de Colaboradores"

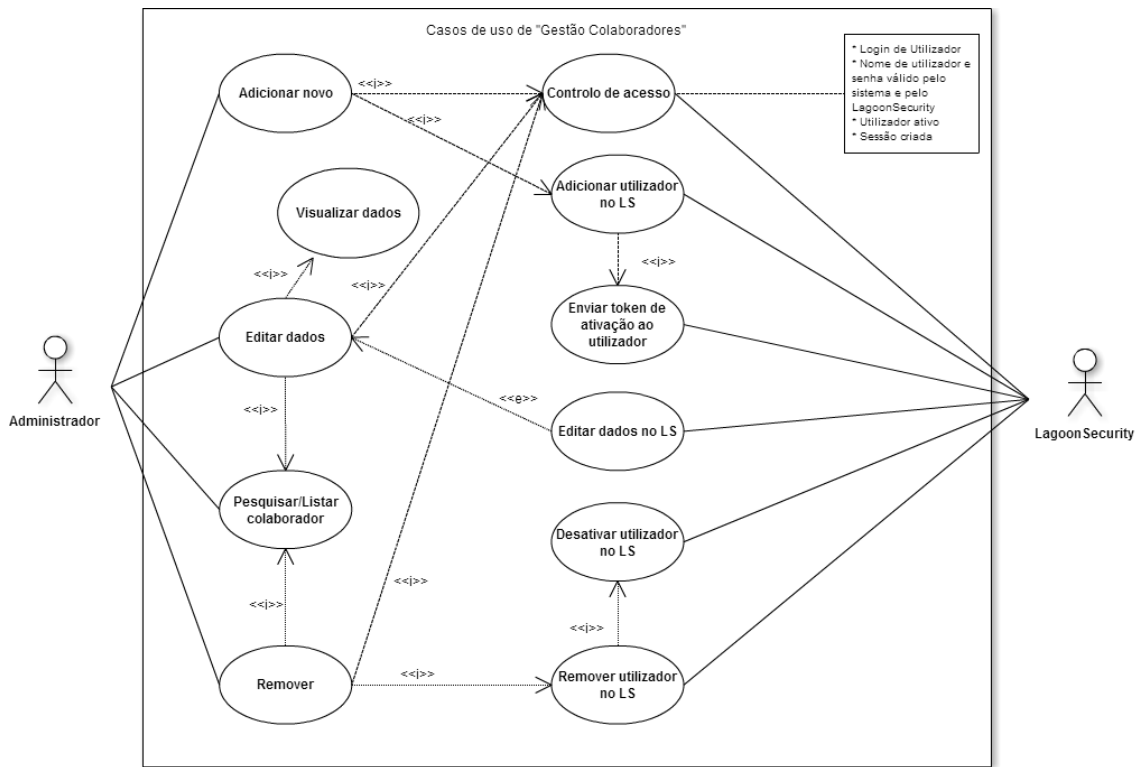


Fig. 72 - Diagrama caso de uso módulo "Gestão de Colaboradores"

6. Diagrama de casos de uso do módulo "Gestão Jogos Temporais"

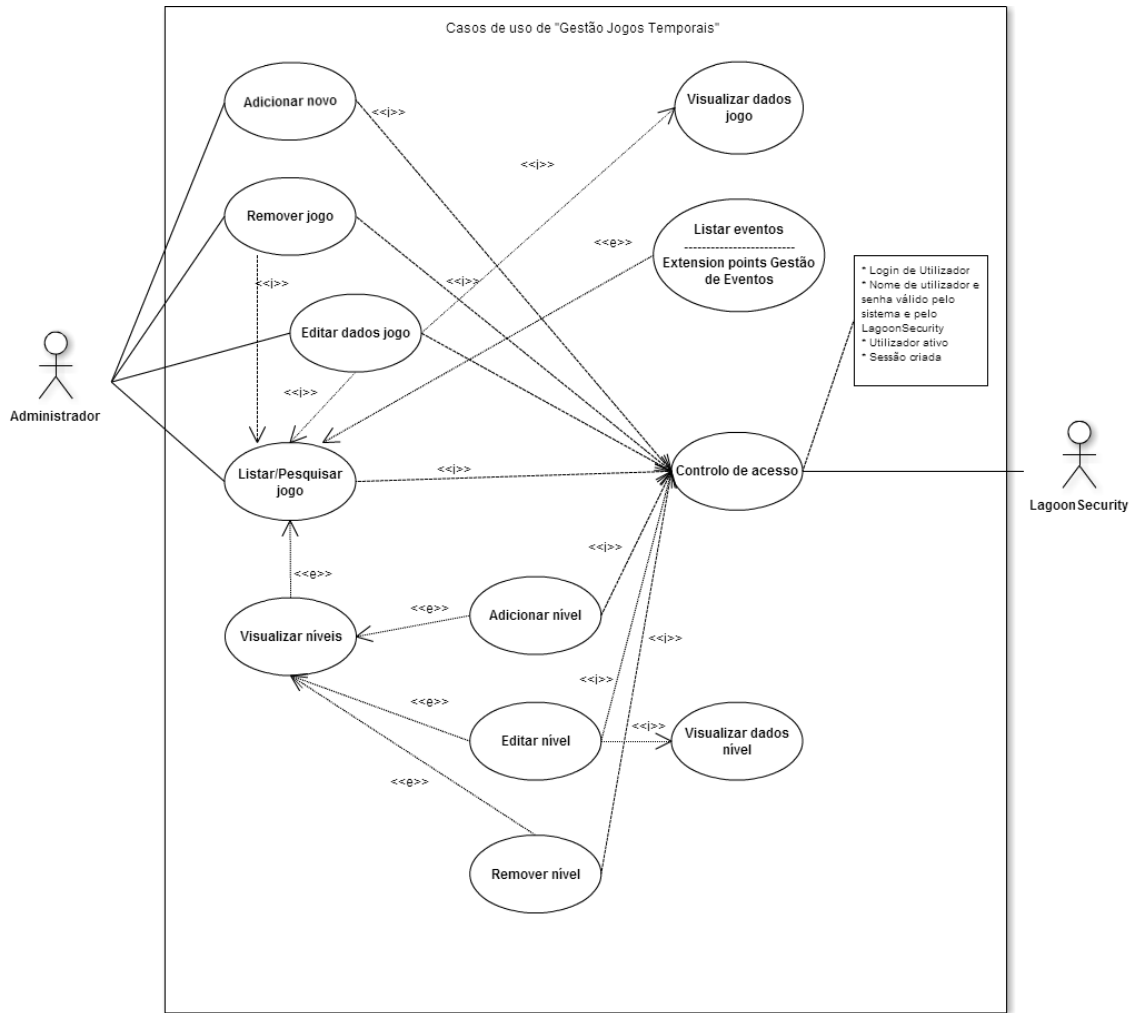


Fig. 73 - Diagrama caso de uso módulo "Gestão Jogos Temporais"

7. Diagrama de casos de uso do módulo "Gestão Categorias"

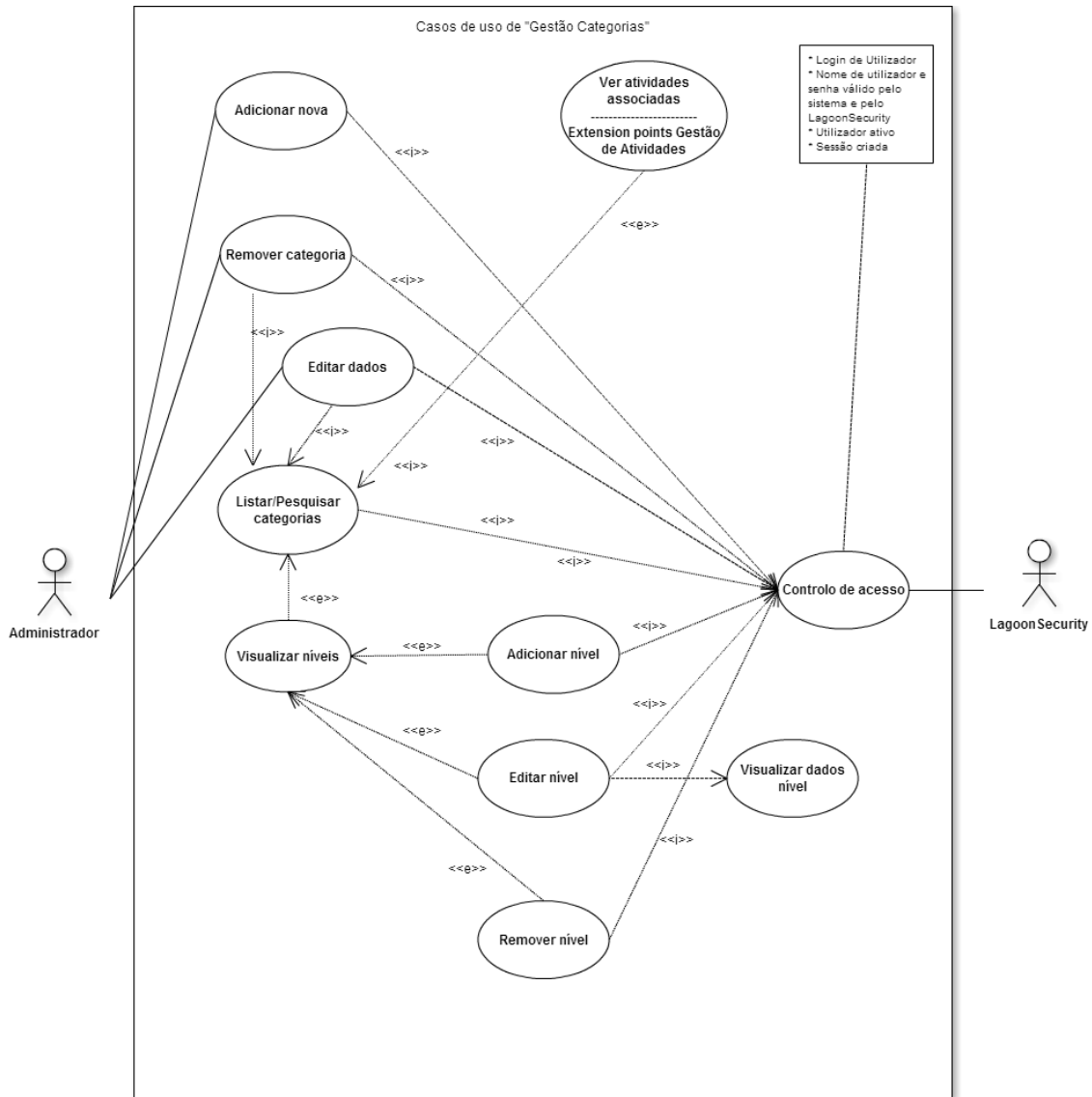


Fig. 74 - Diagrama caso de uso módulo "Gestão Categorias"

8. Diagrama de casos de uso do módulo "Gestão Atividades"

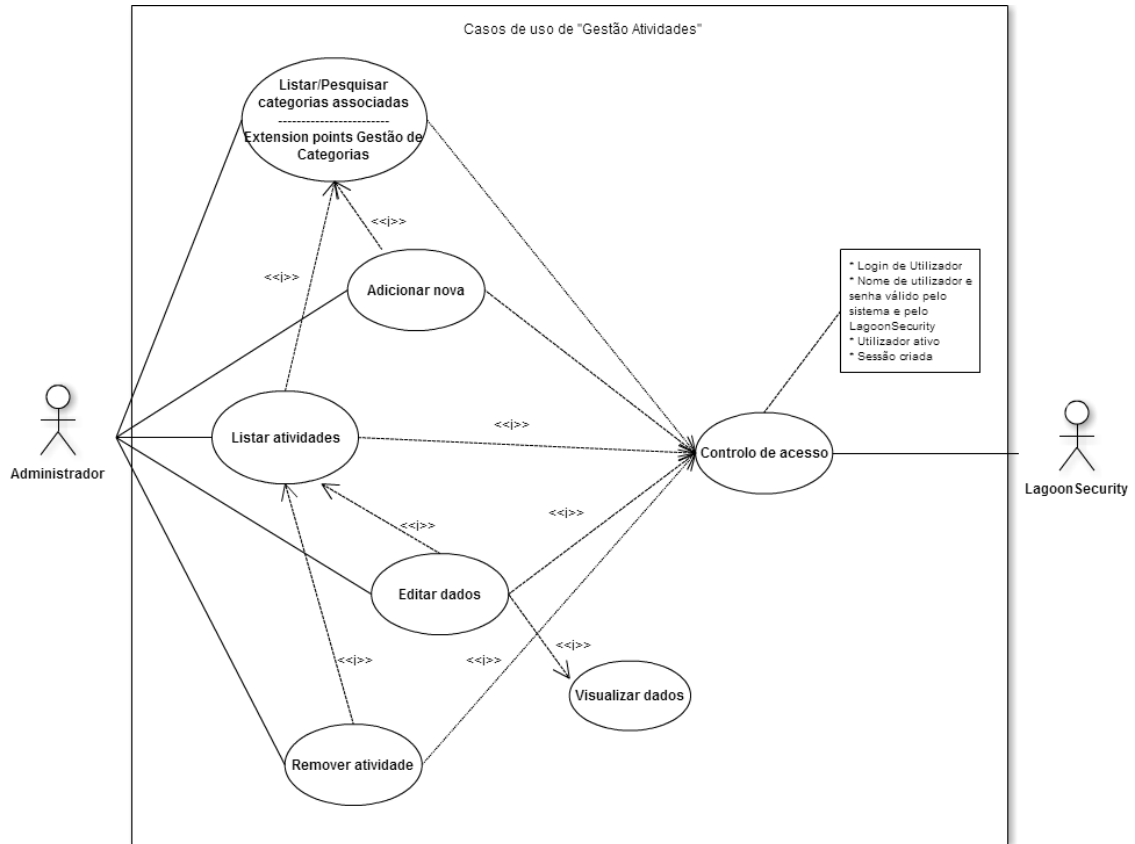


Fig. 75 - Diagrama caso de uso módulo "Gestão Atividades"

9. Diagrama de casos de uso do módulo “Gestão Eventos”

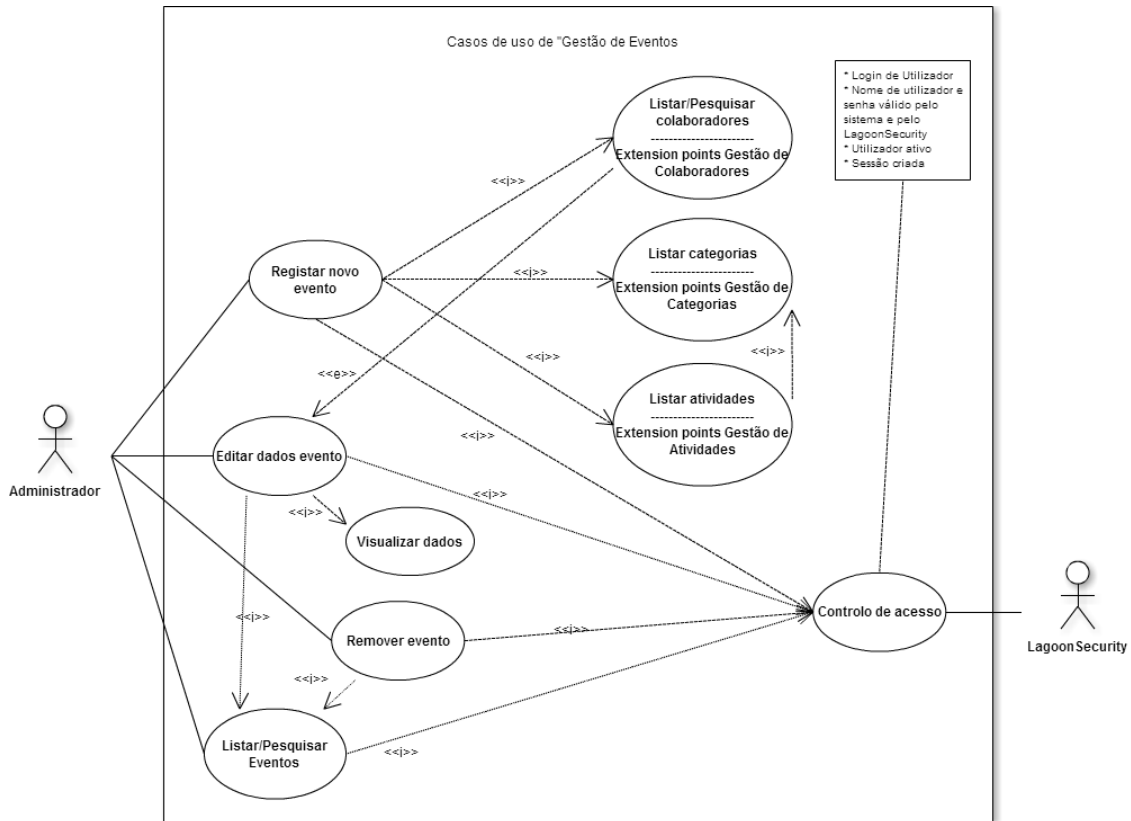


Fig. 76 - Diagrama caso de uso módulo "Gestão Eventos"

10. Diagrama de casos de uso do módulo “Gestão Badges”

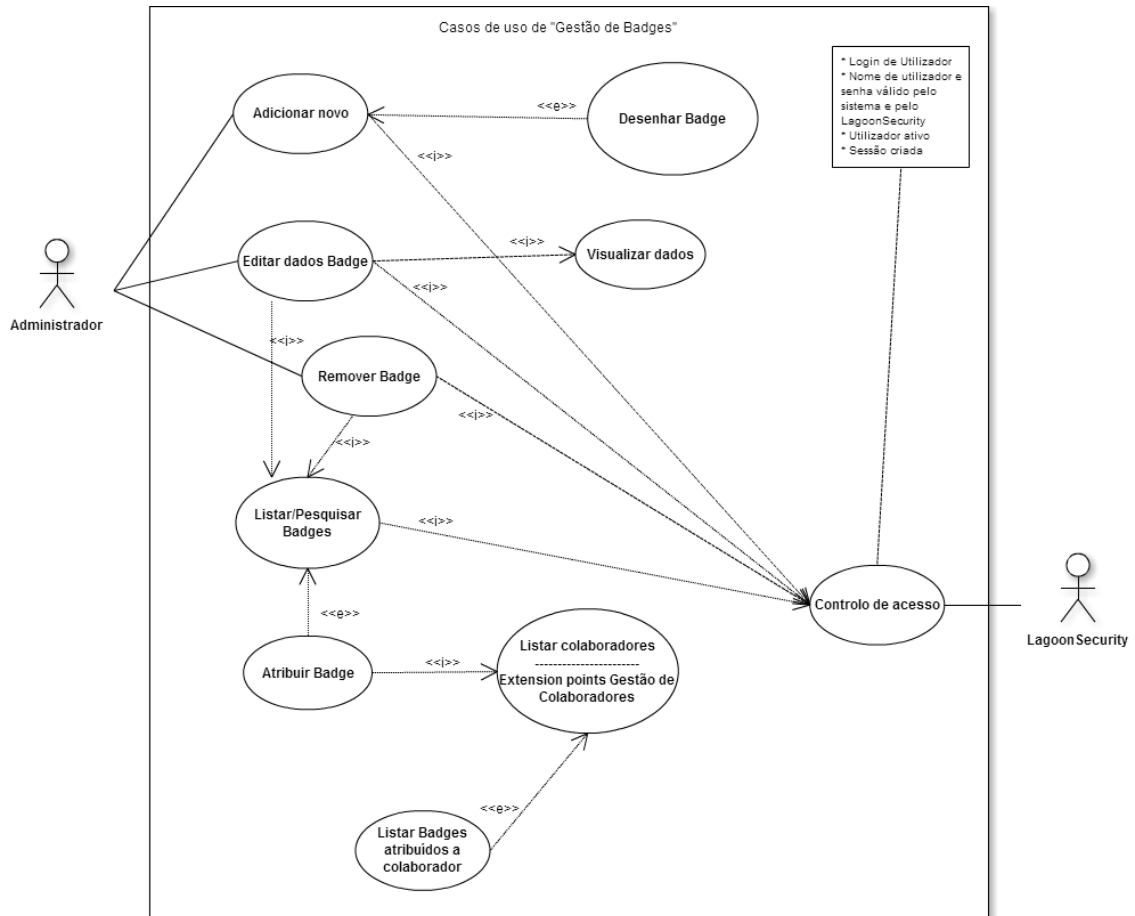


Fig. 77 - Diagrama caso de uso módulo "Gestão Badges"

11. Diagrama de casos de uso do módulo "Estatísticas"

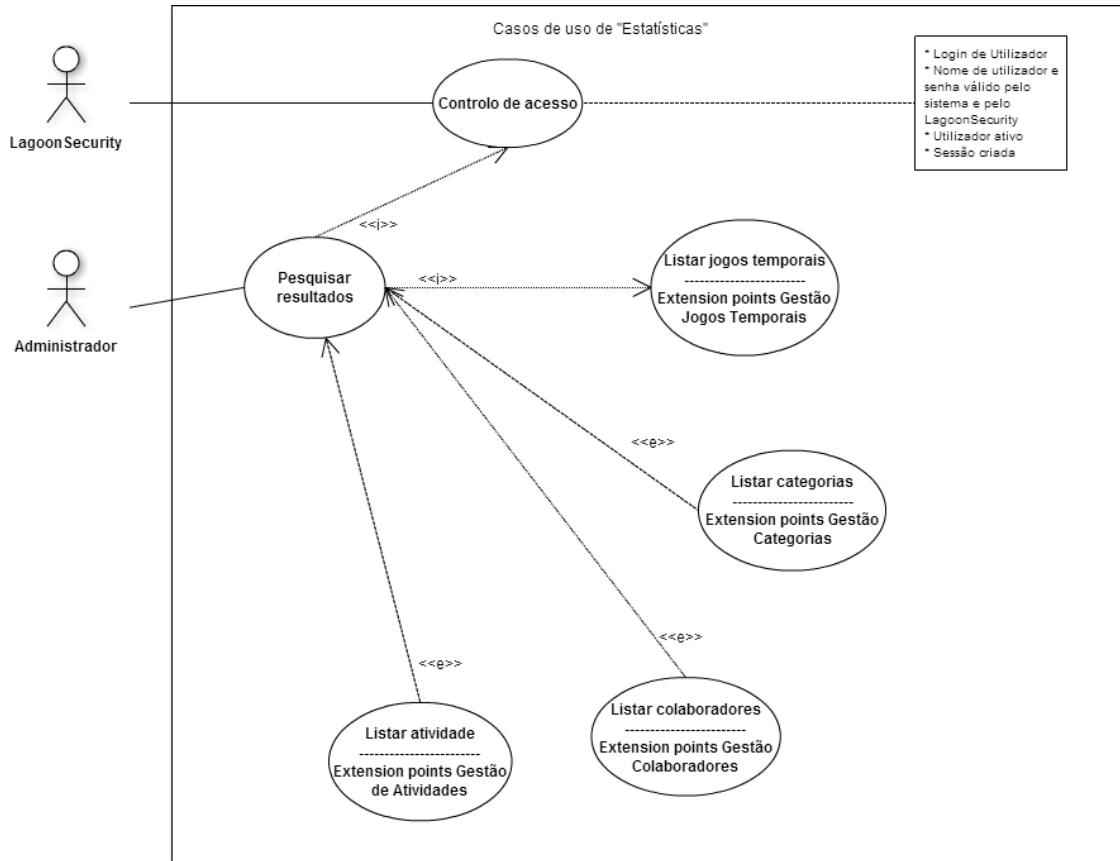


Fig. 78 - Diagrama caso de uso módulo "Estatísticas"

12. Diagrama de casos de uso do módulo "Gestão Aplicações Externas"

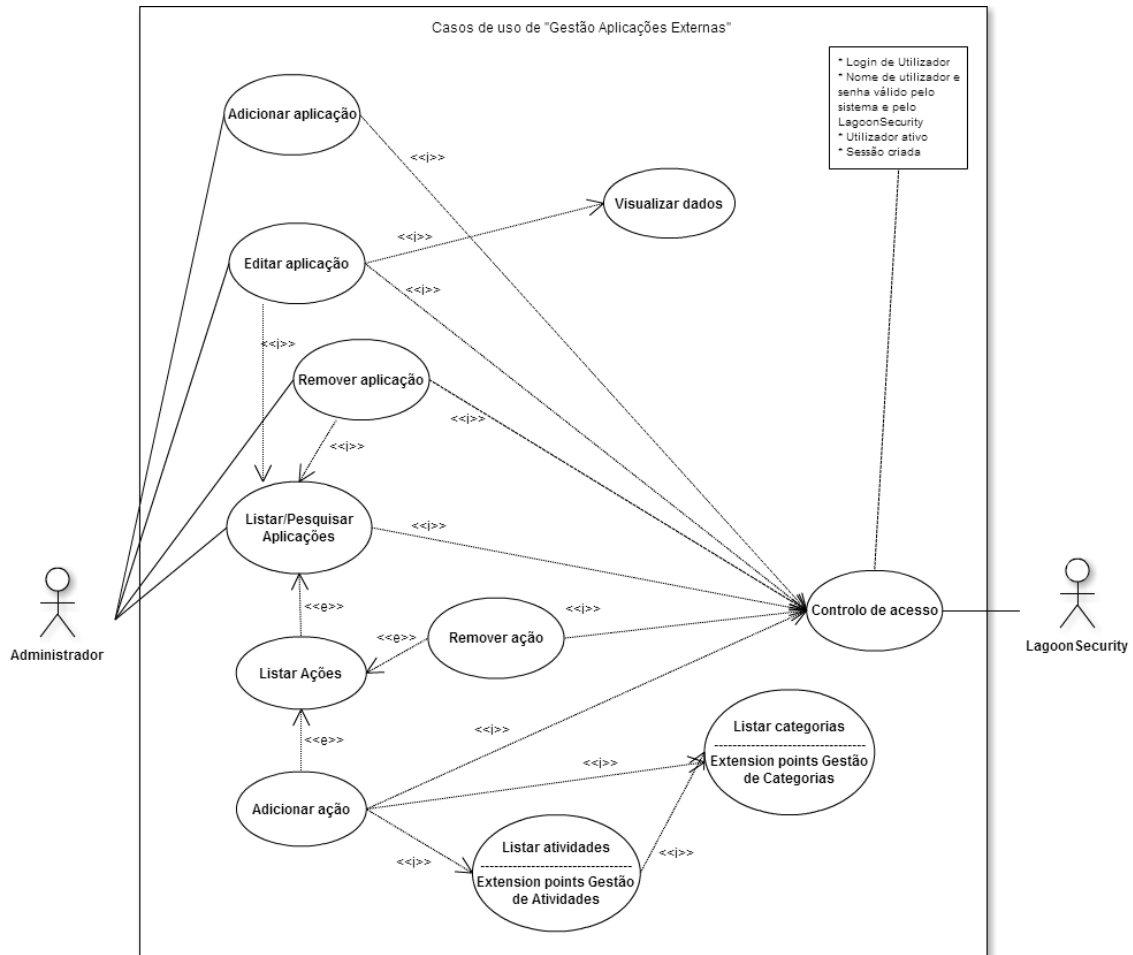


Fig. 79 - Diagrama caso de uso módulo "Gestão Aplicações Externas"

Anexo B – Modelo Físico de Datos

