

## **Portal de Eventos**

**Miguel Ângelo Pinto Borges**

**Dissertação para obtenção do Grau de Mestre em  
Engenharia Informática, Área de Especialização em  
Arquiteturas, Sistemas e Redes**

**Orientador: Dr. António Costa**

**Co-orientador: Dr. Ângelo Martins**

**Júri:**

Presidente:

Doutor Luís Miguel Moreira Lino Ferreira, DEI/ISEP

Vogais:

Doutor Paulo Alexandre Gandra de Sousa, DEI/ISEP

Doutor António Manuel Cardoso da Costa, DEI/ISEP

Doutor Ângelo Manuel Rego e Silva Martins, DEI/ISEP

Porto, novembro 2013



*Aos meus pais e amigos  
por todo o apoio que me deram*



# RESUMO

Na última década tem-se assistido ao aparecimento de várias redes sociais, no entanto, apesar de maior parte delas terem suporte para a promoção de eventos, estas não têm grandes funcionalidades úteis, relacionadas com o tema, como ferramentas de apoio aos promotores, procura de eventos baseados em geolocalização, integração com outros sistemas, gestão de entradas, bilheteiras...

Nesta dissertação é documentado o desenvolvimento de um sistema que tem como objetivo colmatar esses problemas.

**Palavras-chave (Tema):** Redes Sociais, Eventos, Espaços

**Palavras-chave (Tecnologias):** Laravel, PHP



# ABSTRACT

Over the last decade has seen the appearance of various social networks, even though, most them has support for the promotion of events, they don't have major useful features related to the issue, as tools in support of promoters, search for events based on geolocation, integration with other systems, the management of entries, ticket offices ...

This dissertation documented the development of a system which aims to solve these problems.

**Keywords (Business):** Social Networks, Events, Places

**Keywords (Technologies):** Laravel, PHP



# AGRADECIMENTOS

Em primeiro lugar agradeço aos meus orientadores, o Dr. António Manuel Cardoso da Costa e o Dr. Ângelo Manuel Rego e Silva Martins, pela disponibilidade que sempre demonstram, pela prontidão nas suas respostas às minhas dúvidas e pelo valioso contributo dado no desenvolvimento da presente dissertação e projeto.

Ao meu colega Sérgio Lapa, por me ter acompanhado durante as fases planeamento e desenho do sistema, e por ter disponibilizado o seu servidor caseiro, depois de ter desistido.

Ao meu colega Jacinto Barbosa, pela sua ajuda no desenvolvimento do plano de negócio deste projeto.

Aos meus colegas de curso, pela ajuda e apoio mútuo, pelas suas sugestões e críticas ao sistema, assim como pelo companheirismo demonstrado.

A todos os docentes do MEI-ISEP, que me ajudaram a melhorar as minhas competências.

A todos eles, o meu mais sincero agradecimento.



# ÍNDICE

<b>1. INTRODUÇÃO .....</b>	<b>1</b>
1.1. ENQUADRAMENTO.....	2
1.2. APRESENTAÇÃO DO PROJETO.....	2
1.3. MÉTODO DE DESENVOLVIMENTO .....	3
1.4. ORGANIZAÇÃO DA DISSERTAÇÃO .....	3
<b>2. ESTADO DA ARTE.....</b>	<b>5</b>
2.1. ENQUADRAMENTO.....	6
2.2. ANÁLISE DO MERCADO.....	7
2.2.1. <i>WeGoOut</i> .....	8
2.2.2. <i>VaiBater</i> .....	10
2.2.3. <i>TYMR</i> .....	12
2.2.4. <i>Outros Portais</i> .....	13
2.2.5. <i>Conclusão</i> .....	13
<b>3. ENGENHARIA DE REQUISITOS.....</b>	<b>15</b>
3.1. LEVANTAMENTO DE REQUISITOS .....	16
3.1.1. <i>Requisitos funcionais</i> .....	16
3.1.2. <i>Requisitos não-funcionais</i> .....	17
3.2. CASOS DE USO .....	18
3.2.1. <i>Diagrama de casos de uso – “Relação atores”</i> .....	19
3.2.2. <i>Diagrama de casos de uso – “Geral”</i> .....	20
3.2.3. <i>Diagrama de casos de uso – “Gestão de eventos”</i> .....	22
3.3. MODELO DE DOMÍNIO.....	23
3.4. ARQUITETURA LÓGICA .....	24
<b>4. ANÁLISE E MODELAÇÃO .....</b>	<b>25</b>
4.1. DIAGRAMA DE COMPONENTES.....	26
4.2. TECNOLOGIA.....	26
4.2.1. <i>Composer</i> .....	27
4.2.2. <i>Laravel</i> .....	29
4.3. DESENVOLVIMENTO ORIENTADO POR TESTES.....	34
4.3.1. <i>Importância dos testes</i> .....	34
4.3.2. <i>Configuração do ambiente</i> .....	36

4.3.3.	<i>Modelos</i> .....	37
4.3.4.	<i>Controladores</i> .....	38
<b>5.</b>	<b>IMPLEMENTAÇÃO</b> .....	<b>41</b>
5.1.	INTRODUÇÃO .....	42
5.2.	COMPONENTES CHAVE .....	42
5.2.1.	<i>Base do Sistema</i> .....	42
5.2.2.	<i>Integração com as Redes Sociais</i> .....	53
5.2.3.	<i>Relatórios/Estatísticas</i> .....	56
5.3.	SEGURANÇA.....	57
5.3.1.	<i>Codificação das passwords</i> .....	57
5.3.2.	<i>Cross-site request forgery</i> .....	58
5.3.3.	<i>Cross Site Scripting</i> .....	58
5.4.	USABILIDADE E ACESSIBILIDADE .....	59
<b>6.</b>	<b>CONCLUSÕES</b> .....	<b>63</b>
6.1.	RESUMO .....	64
6.2.	OBJETIVOS REALIZADOS.....	64
6.3.	LIMITAÇÕES E TRABALHO FUTURO .....	65
6.4.	APRECIÇÃO FINAL.....	65
<b>7.</b>	<b>REFERÊNCIAS</b> .....	<b>67</b>
<b>8.</b>	<b>ANEXOS</b> .....	<b>71</b>
	ANEXO 1 – PLANO DO PROJETO.....	73
	ANEXO 2 – MODELO DE DADOS .....	89
	ANEXO 3 – PLANO DE NEGÓCIO .....	99

# ÍNDICE DE FIGURAS

FIGURA 1 – DIAGRAMA DE CASOS DE USO – “RELAÇÃO ATORES” .....	19
FIGURA 2 – DIAGRAMA DE CASOS DE USO – “GERAL” .....	20
FIGURA 3 – DIAGRAMA DE CASOS DE USO – “GERAL (CONTINUAÇÃO)” .....	21
FIGURA 4 – DIAGRAMA DE CASOS DE USO – “GESTÃO DE EVENTOS” .....	22
FIGURA 5 – MODELO DE DOMÍNIO DO SISTEMA .....	23
FIGURA 6 – ARQUITETURA LÓGICA DO SISTEMA .....	24
FIGURA 7 – DIAGRAMA DE COMPONENTES DO SISTEMA FINAL .....	26
FIGURA 8 – CICLO DESENVOLVIMENTO TDD .....	35
FIGURA 9 – COMPARAÇÃO ENTRE O TDD E OS TESTES TRADICIONAIS .....	35
FIGURA 10 – DIAGRAMA DE PACOTES DO SISTEMA .....	43
FIGURA 11 – EXTENSÃO DO MODELO ELOQUENT .....	45
FIGURA 12 – CLASSE LANGUAGE .....	45
FIGURA 13 – DIAGRAMA DE CLASSES DO COMPONENTE DE GESTÃO DE UTILIZADORES, GRUPOS E PERMISSÕES .....	46
FIGURA 14 – DIAGRAMA DE CLASSES DO COMPONENTE DE GESTÃO DE ESPAÇOS .....	47
FIGURA 15 – ESTRUTURA DE UMA TABELA COM RELAÇÃO POLIMÓRFICA .....	48
FIGURA 16 – DIAGRAMA DE CLASSES DO COMPONENTE DE GESTÃO DE EVENTOS .....	49
FIGURA 17 – REPOSITORY PATTERN .....	52
FIGURA 18 – DIAGRAMA DE CLASSES DO COMPONENTE DE INTEGRAÇÃO COM SERVIÇOS EXTERNOS .....	54
FIGURA 19 – ESTRUTURA DO COMPONENTE SOCIAL .....	55
FIGURA 20 – ESTRUTURA DO COMPONENTE <i>STATISTICSREPORTS</i> .....	57
FIGURA 21 – TELA DE AUTENTICAÇÃO .....	59
FIGURA 22 – PÁGINA DE VISUALIZAÇÃO DE UM EVENTO NUMA TELA FULLHD .....	60
FIGURA 23 - PÁGINA DE VISUALIZAÇÃO DE UM EVENTO NUM DISPOSITIVO MÓVEL .....	60
FIGURA 24 – TELA DE ALTERAÇÃO DA LOCALIZAÇÃO DE UM UTILIZADOR .....	61
FIGURA 25 – MAPA DO PORTAL .....	61



# ÍNDICE DE TABELAS

TABELA 1 – NÚMERO DE ESPETÁCULOS E ESPAÇOS DE LAZER EM PORTUGAL, EM 2011 (FONTE: INE, PORDATA) .....	6
TABELA 2 – RESUMO DAS FUNCIONALIDADES DOS PORTAIS CONCORRENTES (S – SIM / N – NÃO) .....	14
TABELA 3 – PONTOS FORTES E PONTOS FRACOS DA CONCORRÊNCIA.....	14
TABELA 4 – FORMATO BREVE DOS CASOS DE USO DO DIAGRAMA – “RELAÇÃO ACTORES” .....	19
TABELA 5 – FORMATO BREVE DOS CASOS DE USO DO DIAGRAMA – “GERAL”.....	21
TABELA 6 – FORMATO BREVE DOS CASOS DE USO DO DIAGRAMA – “GERAL (CONTINUAÇÃO)” .....	22
TABELA 7 – FORMATO BREVE DOS CASOS DE USO DO DIAGRAMA – “GESTÃO DE EVENTOS” .....	22
TABELA 8 – DESCRIÇÃO DOS PACOTES BASE DO SISTEMA.....	44
TABELA 9 – DESCRIÇÃO BREVE DAS CLASSES BASE DO SISTEMA.....	50



# ÍNDICE DE CÓDIGO FONTE

CÓDIGO FONTE 1 – EXEMPLO DE GESTÃO DE DEPENDÊNCIAS COM COMPOSER.....	28
CÓDIGO FONTE 2 - EXEMPLOS BÁSICOS DO USO DO ORM ELOQUENT.....	30
CÓDIGO FONTE 3 - EXEMPLOS DE RELAÇÕES COM O ORM ELOQUENT .....	30
CÓDIGO FONTE 4 - CRIAÇÃO DA TABELA USERS COM <i>SCHEMA BUILDER</i> E <i>MIGRATIONS</i> .....	31
CÓDIGO FONTE 5 - POPULAÇÃO DA TABELA USERS .....	32
CÓDIGO FONTE 6 - EXEMPLO DE UMA ROTA COM LARAVEL .....	32
CÓDIGO FONTE 7 - EXEMPLO DE UM CONTROLLER EM LARAVEL .....	33
CÓDIGO FONTE 8 – MÉTODO PARA PREPARAR O AMBIENTE DE DESENVOLVIMENTO.....	36
CÓDIGO FONTE 9 – EXCERTO DE TESTES REALIZADOS AO MODELO “EVENT” .....	37
CÓDIGO FONTE 10 – OUTPUT DA EXECUÇÃO DOS TESTES .....	37
CÓDIGO FONTE 11 – EXCERTO DE TESTES FEITOS AO CONTROLADOR <i>EVENTSCONTROLLER</i> .....	39
CÓDIGO FONTE 12 – DEFINIÇÃO DE RELAÇÕES POLIMÓRFICAS COM LARAVEL .....	47
CÓDIGO FONTE 13 – EXEMPLO DE UTILIZAÇÃO DE UMA RELAÇÃO POLIMÓRFICA EM LARAVEL .....	48
CÓDIGO FONTE 14 – TESTE UNITÁRIO PARA VERIFICAR SE O CONTROLADOR INVOCA O REPOSITÓRIO CORRETAMENTE.....	53
CÓDIGO FONTE 15 – PROTECÇÃO CSRF .....	58
CÓDIGO FONTE 16 – VALIDAÇÃO AUTOMÁTICA DOS MODELOS QUANDO ESTES SÃO GUARDADOS.....	58
CÓDIGO FONTE 17 – EXEMPLO DO USO DE ATRIBUTOS DE ACESSIBILIDADE NUM DROPDOWN MENU .....	59



# NOTAÇÃO E GLOSSÁRIO

<b>Active Record</b>	É uma camada de mapeamento objecto-relacional responsável pela interoperabilidade entre a aplicação e a base de dados e pela abstração dos dados.
<b>Apache</b>	Servidor HTTP <i>open source</i> , mais utilizado no mundo.
<b>App</b>	Aplicação móvel ou social.
<b>CSS</b> Cascading Style Sheets	Linguagem de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como HTML ou XML.
<b>Framework</b>	Conjunto de classes ou código que colaboram para realizar uma responsabilidade para um domínio de um subsistema da aplicação. Ajuda os programadores a construir aplicações mais rapidamente.
<b>JavaScript</b>	Linguagem de programação <i>client-side</i> que tem como objetivo dar alguma dinâmica a páginas HTML.
<b>jQuery</b>	Framework JavaScript.
<b>MySQL</b>	SGBD <i>open source</i> .
<b>Open Source</b>	Termo que se refere genericamente a <i>software</i> que respeita as quatro liberdades definidas pela Free Software Foundation.
<b>PHP</b> Hypertext Preprocessor	Linguagem de programação web <i>server-side</i> .
<b>SGBD</b> Sistema de Gestão de Base de Dados	Conjunto de <i>software</i> responsável pela gestão de base de dados.
<b>UML</b> Unified Modeling Language	Linguagem de modelação que permite que os desenvolvedores de <i>software</i> visualizem os produtos de seus projetos em diagramas padronizados.
<b>URL</b> Uniform Resource Locator	Endereço de um recurso.
<b>Wireframe</b>	Guia visual básico para sugerir a estrutura de um <i>website</i> ou de uma App. Ilustração semelhante do <i>layout</i> de elementos fundamentais na interface. Normalmente são concluídos antes que qualquer trabalho artístico seja desenvolvido.



# 1. INTRODUÇÃO

*Este capítulo é uma breve apresentação do projeto e do âmbito onde se insere, bem como a metodologia de desenvolvimento do mesmo.*

### 1.1. Enquadramento

Esta dissertação visa documentar todo o processo de desenvolvimento do projeto desenvolvido no âmbito da unidade curricular “Tese/Dissertação/Estágio”, do ano letivo de 2012/2013, do Mestrado em Engenharia Informática, do Instituto Superior de Engenharia do Porto, para obtenção do Grau de Mestre em Engenharia Informática, Área de Especialização em Arquiteturas, Sistemas e Redes. Esta unidade curricular é o culminar do ciclo de estudos, devendo ser, num contexto normal, a última unidade curricular em que o aluno obtém aprovação. Nessa medida, este projeto permite aplicar e desenvolver não só os conhecimentos e competências adquiridas nas várias unidades curriculares, como também desenvolver várias competências, no domínio da pesquisa, da escrita e da comunicação

### 1.2. Apresentação do projeto

O turismo é considerado um sector fundamental da economia portuguesa visto que o país é visitado por milhões de turistas anualmente. Uma das principais dificuldades como que os turistas estrangeiros e nacionais se deparam é descobrir o que podem/têm para fazer para se divertirem, entreterem, passar o tempo ou simplesmente encontrarem lugares para visitar e/ou conhecer. Foi a pensar nestes problemas que surgiu a ideia de agregar este tipo de informações num único local.

O objetivo desta tese é criar um sistema que permita facilmente descobrir, promover e gerir eventos (e espaços), não só para Portugal, mas também para o resto do mundo.

Outro problema que pretendemos colmatar é o da criação de ferramentas de apoio aos promotores, como por exemplo a interligação com outros serviços, que lhes permita gerirem os seus eventos, nos mais diversos locais na rede global (Facebook, Google+, Twitter), a partir de um único local, poupando-lhes assim tempo e trabalho; a aproximação aos utilizadores (pessoas que atendem aos eventos), através da promoção direta ou indireta dos eventos a realizar.

O sistema final será constituído por um portal mais aplicações móveis, cujo principal objetivo é descobrir e localizar eventos e espaços, através das tecnologias de geolocalização existentes nos mesmos.

## 1.3. Método de desenvolvimento

Para o desenvolvimento deste projeto foi aplicado o processo de desenvolvimento de *software* OpenUp<sup>1</sup>, uma versão simplificada do processo unificado. Baseia-se numa estratégia iterativa e incremental que se traduz num ciclo de vida estruturado. No OpenUp, o esforço de cada integrante da equipa é medido em micro-incrementos, pequenas partes constituintes do projeto que serão as unidades de medida do progresso do projeto. De acordo com este método o desenvolvimento de *software* é composto por quatro fases: iniciação, elaboração, construção e transição. Além disso, o OpenUp também divide o projeto em iterações, que são intervalos de tempo planeados, tipicamente medidos em semanas. Cada fase engloba uma ou mais iterações. O objetivo de dividir a evolução do projeto em iterações é estabelecer metas para serem cumpridas e poder compartilhar os resultados parciais com todos os *stakeholders*.

Os membros da equipa utilizaram o Redmine<sup>2</sup> como ferramenta para gestão do projeto e gestão de tarefas e o *git* como tecnologia de versionamento do código-fonte.

Em suma, o projeto foi orientado por práticas definidas no OpenUP, ainda que tenham sido introduzidas algumas adaptações, dado a equipa ter apenas dois elementos. Os principais artefactos desenvolvidos são: Plano de Projeto, Lista de Tarefas, Lista de Riscos, Planos de Iteração e Avaliação do Estado Atual (*Status Assessment*).

No Anexo 1 encontra-se o planeamento do projeto mais pormenorizado, incluindo os artefactos enunciados anteriormente.

## 1.4. Organização da dissertação

Esta dissertação é composta por seis capítulos e dois anexos, cujo teor se descreve de seguida:

### Capítulo 1 - Introdução

Neste capítulo é feita uma breve apresentação do projeto e no âmbito onde se insere, bem como a metodologia de desenvolvimento do mesmo.

---

<sup>1</sup> <http://epf.eclipse.org/wikis/openup/>

<sup>2</sup> <http://www.redmine.org/>

### **Capítulo 2 – Estado da Arte**

Neste capítulo, na primeira parte é abordado mais pormenorizadamente o enquadramento da solução bem como o problema a que propõe resolver. Na segunda parte é apresentado um estudo que mostra as principais abordagens concorrentes da abordagem que se vai elaborar na tese para resolver o problema ou problemas próximos, mostrando as vantagens e deficiências dessas abordagens concorrentes, e identifica com clareza quais dessas deficiências serão colmatadas pela dissertação que se vai fazer ou que se fez.

### **Capítulo 3 – Engenharia de Requisitos**

Neste capítulo começa por salientar a importância da “Engenharia de Requisitos” no processo de desenvolvimento de *software*. É também elaborado e descrito o levantamento dos requisitos (funcionais e não funcionais), e a partir destes são descritos os casos de uso, o modelo de domínio e a arquitetura do sistema.

### **Capítulo 4 – Análise e Modelação**

Neste capítulo numa primeira parte é evidenciado os componentes do sistema. Numa segunda parte, é apresentado o estado atual da tecnologia usada para o desenvolvimento da solução e por fim, na terceira parte pormenoriza a preocupação para com a qualidade e correto funcionamento da solução final, ao descrever a metodologia de testes seguida, demonstrando como o desenvolvimento orientado a testes para escrever e testar o nosso código-fonte, se torna mais eficiente e descrevendo alguns dos testes efetuados.

### **Capítulo 5 – Implementação**

Neste capítulo descreve a implementação dos principais módulos da solução, através da *framework* utilizada. É detalhada a arquitetura de cada um, destacando as principais funções de cada módulo. São também apresentadas algumas das soluções ao nível da segurança, usabilidade e acessibilidade.

### **Capítulo 6 – Conclusões**

Neste capítulo é feito um balanço do projeto, enumerando quais os objetivos atingidos e quais os aspetos que eventualmente sejam possíveis de melhorar no futuro. São também apontadas as dificuldades encontradas, os conhecimentos adquiridos e a apreciação final do autor.

## 2. ESTADO DA ARTE

*Este capítulo, na primeira parte aborda mais pormenorizadamente o enquadramento da solução bem como o problema a que propõe resolver. Na segunda parte apresenta um estudo que mostra as principais abordagens concorrentes da abordagem que se vai usar na tese para resolver o problema ou problemas próximos, mostrando as vantagens e deficiências dessas abordagens concorrentes e identificando com clareza quais dessas deficiências serão colmatadas por esta dissertação.*

## 2.1. Enquadramento

O turismo tem uma importância verdadeiramente estratégica para a economia portuguesa em virtude da sua capacidade em criar riqueza e emprego. Trata-se de um sector em que Portugal tem vantagens competitivas claras como sucede com poucos outros. Está a ter lugar uma grande aposta no turismo por parte do Governo e dos empresários do sector. Com essa aposta, o turismo está a viver um bom momento. As receitas estão a aumentar. Existe capacidade instalada de boa qualidade em termos de infra-estruturas e de recursos humanos. Estão a ser lançados numerosos projetos de alta qualidade nas zonas tradicionais. Estão a surgir novos destinos de grande qualidade, fruto da iniciativa empresarial e da capacidade do Governo em desbloquear processos que se encontravam parados há anos, sendo que a aposta no turismo vai continuar.[1], [2]

No entanto, os produtos e as experiências procuradas pelos turistas têm vindo a evoluir. Neste ponto destaca-se a tendência para um aumento da diversificação das experiências. Neste contexto, é cada vez mais importante a oferta de um conjunto alargado de produtos que dê resposta a uma procura diversificada. Ainda para mais, existe uma tendência para uma redução do peso das viagens organizadas, por oposição ao crescimento do DIY (*do it yourself*). No entanto, uma das principais dificuldades destes turistas é descobrirem o que podem/têm para fazer para se divertirem, entreterem, passar o tempo ou simplesmente encontrarem lugares para visitar e/ou conhecer.

No período de 2007 a 2009, houve um reforço notório nas ações de promoção turística tendo sido investidos mais de 150 milhões de euros. No âmbito dos eventos, foi feita uma forte aposta num programa de âmbito nacional cujo investimento superou os 75 milhões de euros. Este programa cobriu desde eventos culturais, abarcando diversos tipos de música, teatro, festividades regionais, a eventos desportivos, que contribuiram para, por um lado completar e reforçar a experiência do turista e, por outro, aumentar a visibilidade nacional e internacional de Portugal enquanto destino turístico de referência (Tabela 1). [1], [3]

<b>Espetáculos</b>	<b>Número</b>	<b>Espaços</b>	<b>Número</b>
Música e Dança	11.926	Recintos Culturais	347
Teatro	12.174	Museus e Jardins	340
Ópera	155	Galerias	881
Desporto	0	Bares	1.987
<b>Total:</b>	<b>24.255</b>	Discotecas	759
		<b>Total:</b>	<b>4.314</b>

Tabela 1 – Número de espetáculos e espaços de lazer em Portugal, em 2011 (Fonte: INE, PORDATA)

Como as maiores organizadoras, as agências de viagens são na sua maioria planeadoras de atividades turísticas, sendo que a maioria dos eventos programados são excursões. No caso dos mais jovens e de muitos emigrantes estes tipos de eventos não são atrativos, pelo que a sua maioria prefere saídas à noite.

Embora exista este tipo de informação, esta encontra-se fracionada e dispersa. Os eventos culturais costumam estar divulgados nos *sites* das câmaras municipais, enquanto que os bares e discotecas optam pela divulgação dos seus eventos através das redes sociais ou pelo tradicional “passa a palavra”. É para colmatar estes problemas que surgiu a ideia de agregar este tipo de informações num único local, onde os utilizadores possam encontrar a informação que procuram facilmente.

Outro dos problemas que o serviço propõe colmatar será o da gestão dos eventos e espaços, por parte dos promotores, pelos diversos locais na rede global. Para isso o sistema estará interligado com outros serviços, o que permite aos promotores de eventos gerirem os seus eventos, nos mais diversos locais (Facebook, Google+, Twitter, ...), a partir de um único local, poupando-lhes assim tempo e trabalho.

Este serviço apesar de ser idealizado para Portugal, não há nada impeditivo para que este possa ser aplicado internacionalmente, até porque este tipo de problemas não é específico de Portugal.

Em suma, o portal de eventos surge da oportunidade identificada em atuar como um portal de procura de eventos. Criar e disponibilizar um sistema (portal e aplicações móveis) para procura de todo o tipo de eventos: desportivos, de lazer, de treino/*fitness*, concertos de música, encontros/*meetings*, etc. Deverá dar resposta à procura existente no mercado nacional português, englobando as necessidades da população na procura de entretenimento diário (diurno ou noturno), sazonal (férias entre outros) ou simplesmente satisfação de gostos musicais temporais.

## 2.2. Análise do mercado

Embora no último ano tenham aparecido muitos sistemas nesta área de negócio, neste estudo foi analisado o portal “VaiBater”, por ter sido o único portal a ir ao encontro do tema deste trabalho, aquando da submissão da proposta de tese, e os portais “WeGoOut” e

“TYMR” por serem os sistemas que mais se enquadram no tema deste trabalho, sendo o “TYMR” o mais recente.

Os critérios para a seleção dos concorrentes foram os sistemas que, de certa forma agregam a informação de eventos e a disponibilizam organizadamente; permitem a importação e a exportação de outros sistemas; tenham a vertente de rede social (amigos); tenham ferramentas de apoio aos promotores e que tenham soluções para dispositivos móveis. Os outros portais não foram descritos ao pormenor por serem muito limitativos e semelhantes entre si.

### **2.2.1. WeGoOut<sup>1</sup>**

Nesta secção, mostra-se como o portal “WeGoOut” permite que os utilizadores descubram os eventos (festas) mais relevantes e quais deles estão a ser frequentados pelos seus amigos, sendo possível visualizar fotografias da festa, por exemplo, para ver se está com “bom ambiente”. A plataforma, para além de ser um motor de pesquisa de eventos também permite descobrir pessoas. Também é evidenciado que este projeto funciona bastante em cima da rede social “Facebook” (estando totalmente dependente da mesma), enumerando algumas dessas funcionalidades e estando assim dependente desta. São também descritas algumas ferramentas de apoio aos promotores de eventos.

#### **2.2.1.1. Descrição geral**

Este é um portal destinado aos mercados de Portugal, Brasil e todo o mercado internacional, pois está equipado com três pacotes de idiomas: Português (Portugal), Português (Brasil) e Inglês (Internacional).

O WeGoOut permite a procura de eventos, associados unicamente a festas (o que possivelmente é a sua maior desvantagem), sem necessitar de autenticação, em que à partida são mostrados os eventos sugeridos da semana atual. Além destes permite a pesquisa de todos os eventos para “hoje”, “amanhã”, “próximo fim-de-semana”, para a semana atual ou mesmo para o mês corrente atual. Permite ainda a filtragem de eventos por localidade, com a facilidade de mostrar a localização dos mesmos num mapa do Google Maps incluído na página principal.

---

<sup>1</sup> <http://wegoout.com>

Com uma conta registada é-nos apresentado uma listagem de eventos perto de nós, os eventos que os nossos amigos planeiam ir, a opção de convidar amigos do Facebook para o WeGoOut ou mesmo uma lista de amigos sugeridos.

O sistema permite a pesquisa de eventos ou de pessoas. Para além da informação base do evento, é disponibilizado o número de pessoas que vão atender ao evento (individualizados por género), fotos do evento (enviadas pelos utilizadores) e os “meus amigos” e “amigos de amigos” que informaram que iriam ao evento.

Na pesquisa de pessoas existe a possibilidade de procura não só do nome mas também pelos seguintes critérios: sexo, idade, distância à minha cidade, quantidade de interesses em comum, um interesse em particular, com/sem foto. Permite através desta pesquisa a possibilidade de seguirmos uma pessoa em particular (através do Facebook regista a pessoa como favorita). Existe também a possibilidade de enviar uma mensagem privada a um utilizador, bloqueá-lo ou reportar um abuso por parte do mesmo.

Recentemente foram adicionadas novas funcionalidades dirigidas para os produtores de eventos, como a possibilidade de convidar utilizadores para o evento e respetiva gestão da “*guest list*”<sup>1</sup>, gestão dos “*check in*”<sup>2</sup> e a disponibilização de relatórios com as informações relevantes de para evento.

Na visualização de um perfil de uma pessoa, permite consultar as festas a que a mesma aderiu, visualizar os seus seguidores, as pessoas que o mesmo segue, ou mesmo os detalhes da mesma (estilo de vida, interesses) – informações retiradas do Facebook.

### 2.2.1.2. Aplicações móveis

Atualmente o WeGoOut dispõe de aplicações para dispositivos iOS e Android, o que é uma mais-valia para o utilizador. Entre as funcionalidades das *apps* destacam-se:

- A procura das melhores festas, eventos e lugares;
- Onde os teus amigos e amigos de amigos vão;
- Encontrar as festas com melhor relação rapazes/raparigas;
- Avaliação das festas;

---

<sup>1</sup> Listas de convidados

<sup>2</sup> Entradas, presenças...

- Fotos ao vivo da festa;
- Quem vai e quem já lá está;
- Possibilidade de te adicionares à “*guest list*” das festas.

### 2.2.2. VaiBater

Nesta secção, mostra-se que o portal “VaiBater”<sup>1</sup> é simplista, mas permite responder à pergunta fundamental “Onde posso sair?”. Também são descritos problemas de otimização, segurança, falta de integração com as redes sociais e a inexistência de soluções para sistemas móveis.

#### 2.2.2.1. Descrição Geral

Este projeto é um portal simplista, destinado aos mercados de Portugal, Angola, Moçambique e Brasil. Este portal tem como objetivo facilitar o trabalho de comunicação dos espaços e dos promotores de eventos.

Para usufruir das funcionalidades que o portal oferece, não é necessário efetuar um registo, nem de se autenticar de nenhuma forma. Aliás, só é permitido o registo de utilizadores que sejam produtores de eventos ou donos de espaços onde se organizam eventos.

Ao entrar no portal é exibida uma lista de eventos ordenada por tempo e pela distância entre o local do evento e localização do utilizador, sendo essa localização obtida pela tecnologia de georreferenciação dos *browsers* automaticamente ao entrar na página. A filtragem de eventos é um pouco pobre, pois só é possível a filtragem pelo tipo de evento, a partir de uma data e pela entrada livre ou não do mesmo.

Como já foi referido anteriormente o VaiBater, para além de fornecer uma lista de eventos também fornece uma lista de espaços. Esta listagem torna-se útil para os visitantes do portal, pois permite responder a perguntas do tipo “Que discotecas/bares tenho ao pé?”, que se enquadram no mesmo tema de divulgação de eventos. No entanto, esta secção conta com uma filtragem ainda mais pobre que a filtragem dos eventos, onde só é possível a filtragem por todos ou por só um tipo de espaço.

---

<sup>1</sup> <http://vaibater.com>

O sistema permite adicionar novos eventos e espaços, mas é necessário estar registado no sistema, sendo esse registo só acessível a produtores de eventos (produtor) e donos de espaços/estabelecimentos públicos (espaço).

Após a criação de um evento é possível repeti-lo várias vezes, indicando o horário para cada repetição. Cada repetição não passa de uma duplicação do original. No entanto existe uma opção que permite que ao editar uma das repetições, estas sejam replicadas nas restantes.

A integração do portal VaiBater com as redes sociais é quase inexistente. Este apenas está presente no Facebook e Google+. Quanto ao Facebook possui uma página, mas não é atualizada com os eventos. Apenas é utilizada para divulgar novas funcionalidades, com muita pouca frequência. Também possui, na página de cada evento, dois *widjets* do Facebook: o “*Like Button*” [4] e o “*Comments Box*” [5]. Quanto ao Google+ este encontra-se totalmente abandonado, apenas existe uma página/perfil, sem qualquer conteúdo.

Atualmente não existe qualquer otimização/compatibilidade/aplicação, para sistemas móveis. Alias, o portal é praticamente impossível de utilizar em janelas com menos de cerca 900 pixéis de largura.

#### **2.2.2.2. Tecnologias utilizadas**

Segundo o *Wappalyzer* [6], o portal VaiBater foi desenvolvido na linguagem de programação server-side PHP, corre num servidor Apache, sobre o sistema operativo UNIX, CentOS. Quanto às tecnologias *client-side* utilizadas, destacam-se a *framework Javascript* “jQuery” e a *framework CSS* “Twitter Bootstrap”.

O portal não apresenta cuidados ao nível da otimização. A primeira coisa que se nota é a inclusão de muitos ficheiros “.js” e “.css” (cerca de 27 + 4, respetivamente), o que origina um problema de performance. Em seguida notam-se que alguns destes ficheiros não estão minificados, e por conseguinte ocupam mais espaço do que precisam. Estes dois problemas levam a que a página demore mais tempo o carregar, o que pode levar a que utilizadores desistam da ferramenta, por a navegação não ser fluída. São utilizadas imagens já com tamanho considerável, o que atrasa ainda mais o carregamento do portal. Estes problemas de otimização levam a que o tempo que o portal demora carregar completamente, na primeira visita, seja em média de 12 segundos – valores muito elevados para os dias de hoje. Se tomamos em consideração que foi utilizada uma ligação de 100mb e que o portal é muito simples, ainda menos se justificam tais valores. Este problema pode ainda ser maior se

consideramos que o portal também é utilizado em Angola e Moçambique, onde a velocidade média de acesso à Rede Global é muito menor que à média nacional [7], [8]. A inclusão e utilização dos cerca de 27 ficheiros Javascript leva a que seja executado muito código no lado do cliente, e que sejam utilizados cerca de 13Mb de memória e alguma percentagem de processamento da máquina do utilizador.

Ao nível da segurança foram encontradas algumas falhas e más configurações. A primeira foi o servidor listar os ficheiros de subpastas do *webhost* que não tenham uma página. Também foram detetadas mensagens notoriamente de “*debug*”, na consola Javascript do *browser*. Desta duas formas um utilizador pode se aproveitar destas más configurações para aprender mais sobre o sistema, de forma a conseguir penetra-lo. Outra falha detetada foi ao nível de permissões de acesso, conseguindo acesso a páginas simplesmente reescrevendo as URLs.

### 2.2.3. TYMR<sup>1</sup>

O TYMR é uma jovem rede social 100% portuguesa direcionada a eventos (que surgiu também no decorrer da elaboração desta tese). O objetivo do TYMR é a descoberta, gestão e promoção de eventos.

A rede social conta já com algumas das funcionalidades que estavam previstas para o sistema a desenvolver, tais como:

- Geolocalização;
- Sistema de recomendação personalizado;
- Pesquisa por categorias;
- Partilha de eventos com amigos;
- Saber que eventos em que os amigos vão estar presentes;
- Soluções para organizadores de eventos;
- Compra de bilhetes para eventos;

Qualquer pessoa pode registar-se, criar uma página e um evento, começar a vender bilhetes para o mesmo e promovê-lo.

Quanto às soluções para organizadores de eventos, o TYMR oferece funcionalidades como:

- Gestão do nº de bilhetes vendidos;

---

<sup>1</sup> <https://tymr.com/>

- Validação de bilhetes em tempo real;
- Faturação;
- Estatísticas dos participantes;
- Questionários;

Atualmente o TYMR também oferece qualquer solução/aplicação, para sistemas móveis.

#### 2.2.4. Outros Portais

Existem ainda outros portais de menor dimensão onde é possível a pesquisa simples de eventos. Alguns desses casos são o Entrada Livre<sup>1</sup>, o Bon Place<sup>2</sup> e o Espaço Evento<sup>3</sup>. Estes têm funcionalidades muito limitadas, que se baseiam exclusivamente na pesquisa de eventos, em que o Bon Place permite a inserção de eventos no sistema mediante registo. É de notar que não existem soluções para equipamentos móveis, nem qualquer integração com redes sociais.

A nível internacional existe o Festicket<sup>4</sup>, o Eventbrite<sup>5</sup>, onde os principais destaques são o sistema a venda de bilhetes, onde o promotor pode definir os tipos de bilhete, quantidades e preços dos mesmos e a gestão de entradas nos eventos.

#### 2.2.5. Conclusão

Durante o último ano, surgiram diversos portais onde é possível a pesquisa de eventos. No entanto só dois é que se destacam, sendo estes o TYMR e o WeGoOut. No entanto, a maior lacuna neste tipo de portais é a falta de soluções móveis (com a exceção do WeGoOut) e o facto de serem centralizados, ou seja, não disponibilizam a informação a terceiros, como por exemplo para as redes sociais, limitando-se ao uso dos *widgets* de cada rede. Exceção é o WeGoOut pois funciona “em cima” do Facebook, o que facilita a sua utilização, mas que pode ser prejudicial, pois o sistema depende inteiramente dessa rede social. Por essa razão a informação é muito limitada.

No apoio aos produtores e organizadores de eventos estes dois portais também estão um passo à frente. Para além de ser possível a um promotor gerir as “*guest lists*” e os “*check ins*”

---

<sup>1</sup> <http://tmnentradalivre.sapo.pt/>

<sup>2</sup> <http://www.bonplace.com/>

<sup>3</sup> <http://www.espacoevento.net>

<sup>4</sup> <http://www.festicket.com>

<sup>5</sup> <https://www.eventbrite.pt/>

de um evento, os sistemas disponibilizam ferramentas de geração de relatórios, com as estatísticas relevantes de um evento.

	Eventos										Espaços											
	Procurar	Adicionar	Partilhar	Importar	Exportar	Distribuir/Difundir	Eventos Privados	Watchlist	Avaliar	Comentar	Inserir Fotos/Vídeos	Promover	Procurar	Adicionar	Partilhar	Importar	Avaliar	Comentar	Inserir Fotos/Vídeos	Amigos	Aplicações Móveis	Relatórios Técnicos
WeGoOut	S	N	N	S <sup>1</sup>	N	N	N	N	N	N	S <sup>1</sup>	N	N	N	N	N	N	N	N	S <sup>1</sup>	S <sup>2</sup>	S
VaiBater	S	S	S	N	N	N	N	N	S	S	N	N	S	S	N	N	S	S	N	N	N	N
Entrada Livre	S	S	S	N	S	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Bon Place	S	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Espaço Evento	S	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Tymr	S	S	N	N	N	N	N	S	N	S	N	N	N	N	N	N	N	N	N	S	N	N

Tabela 2 – Resumo das funcionalidades dos portais concorrentes (S – Sim / N – Não)

Outra falha encontrada nestes sistemas foi a falta de suporte a eventos avançados (p.e. festivais os festas populares/religiosas, que se prolongam por vários dias e possuem sub-eventos) e a sessões, pois existem eventos que são periódicos (p.e. espetáculos de teatro).

Concorrentes diretos	Pontos Fortes	Pontos Fracos
<b>WeGoOut</b>	<ul style="list-style-type: none"> <li>Interface bonita, simples e 14orna14ive</li> <li>Solução para iOS e Android</li> <li>Integração com o Facebook</li> </ul>	<ul style="list-style-type: none"> <li>Dependência do Facebook</li> <li>Não permite a inserção de eventos</li> <li>Não contém base de dados de espaços</li> </ul>
<b>VaiBater</b>	<ul style="list-style-type: none"> <li>Fundado por Figuras Públicas (Mónica e Rubim)</li> <li>Sistema muito minimalista</li> </ul>	<ul style="list-style-type: none"> <li>Sistema único centralizado</li> <li>Inexistência de Soluções Móveis</li> </ul>
<b>EntradaLivre</b>	<ul style="list-style-type: none"> <li>Associada a uma grande marca (TMN)</li> <li>Suporta a exportação dos eventos para ical e Google Calendar</li> </ul>	<ul style="list-style-type: none"> <li>Sistema centralizado</li> <li>Inexistência de Soluções Móveis</li> <li>Não contém base de dados de espaços</li> </ul>
<b>Tymr</b>	<ul style="list-style-type: none"> <li>Venda de bilhetes</li> <li>Relatórios personalizados</li> </ul>	<ul style="list-style-type: none"> <li>Sistema centralizado</li> <li>Inexistência de Soluções Móveis</li> <li>Não contém base de dados de espaços</li> </ul>

Tabela 3 – Pontos fortes e pontos fracos da concorrência

<sup>1</sup> Importa do Facebook

<sup>2</sup> iPhone e Android

# 3. ENGENHARIA DE REQUISITOS

*Este capítulo começa por salientar a importância da “Engenharia de Requisitos” no processo de desenvolvimento de software. É também elaborado e descrito o levantamento dos requisitos (funcionais e não funcionais), e a partir destes são descritos os casos de uso, o modelo de domínio e a arquitetura do sistema.*

## 3.1. Levantamento de requisitos

Após o estudo e análise do “Estado da Arte” referido no capítulo anterior, o passo seguinte foi definir as funcionalidades que o portal deveria ter, analisando as fraquezas e as forças que os sistemas atuais têm, visando transformar as fraquezas em oportunidades.

O projeto a desenvolver irá ser potencialmente utilizado por milhares de utilizadores e, como tal, é essencial que o portal a desenvolver tenha uma estrutura sólida em termos de robustez e desempenho, seguindo as boas práticas de engenharia de *software*.

Para atingir esses objetivos é vital um bom planeamento do projeto, focando principalmente o levantamento de requisitos, pois será a partir daí que o desenvolvimento da aplicação se irá iniciar. A especificação de requisitos é a tarefa mais importante na fase de análise de um sistema. Requisitos mal especificados podem produzir atrasos no projeto.

Os requisitos, de um modo geral, podem ser classificados em dois grandes grupos: os requisitos funcionais e os não funcionais.

### 3.1.1. Requisitos funcionais

Esta secção apresenta os requisitos funcionais do sistema. Os requisitos funcionais são aqueles que descrevem o comportamento do sistema, suas ações para cada entrada, ou seja, descrevem o que tem que ser feito pelo sistema.

#### **Cadastro**

Em primeiro lugar, foi necessário investigar qual a informação que deverá constar na conta do utilizador, de modo a torna-la o mais completa possível. Para além dos comuns campos para identificação do utilizador e do registo dos seus dados pessoais, outros tiveram de ser contemplados, tais como a data de nascimento, a localização, o sexo e até o número de telemóvel, de modo a que mais tarde seja possível agrupar perfis. Para que o sistema funcione eficazmente é necessário que o utilizador preencha o maior número de campos possíveis.

#### **Eventos e espaços**

Como a temática central do sistema se baseia em eventos, é obrigatório que os utilizadores os possam inserir manualmente no sistema. É também necessária uma boa categorização dos mesmos, para que mais tarde seja possível fazer uma filtragem eficiente. O mesmo se aplica aos espaços.

#### **Sistemas externos**

Deverá ser possível importar e exportar a informação para o máximo de sistemas/tecnologias possíveis, como por exemplo formato ical, Google Calendar, ou redes sociais.

#### **Difusão da Informação**

O sistema deverá ter a possibilidade de difusão dos espaços e eventos pelo menos nas maiores redes sociais, Facebook e Twitter, caso tal seja pretendido pelo utilizador que os criou.

#### **Estatísticas**

Deverá haver uma área onde os promotores possam visualizar estatísticas dos seus eventos.

#### **Amizade**

Os utilizadores devem poder fazer “amizades” com outros utilizadores, assim como enviar convites de amizades para outros utilizadores e convidar utilizadores por *email*.

### **3.1.2. Requisitos não-funcionais**

Os requisitos não funcionais são aqueles que expressam como devem ser executadas determinadas tarefas. Em geral relacionam-se com padrões de qualidade tais como confiabilidade, usabilidade, desempenho, etc. Nos requisitos não-funcionais também são apresentados restrições e especificações de uso para os requisitos funcionais.

#### **Desempenho**

Qualquer operação efetuada no portal deve, sempre que possível, ser processada com rapidez, pois o utilizador final pretenderá utilizar um serviço que responda rapidamente às suas solicitações.

#### **Usabilidade**

A interface do portal deverá ser uma interface de fácil utilização e que não seja pesada, seguindo as normas da W3C, incluindo a sua validação.

#### **Confiabilidade**

Tendo em conta que o portal irá ser utilizado a nível nacional, é essencial que o mesmo tenha garantias elevadas de disponibilidade e elevada tolerância a falhas, assim como um tempo reduzido de reinício após falhas e baixa probabilidade de corrupção de dados após falhas.

#### **Segurança da informação**

Cada utilizador do sistema só deverá ter acesso aos seus próprios dados, e aos dados que outros partilhem, à exceção de utilizadores devidamente autorizados. Todas as palavras-passe do sistema e dados bancários dos utilizadores deverão estar cifrados.

A criação de novos eventos e espaços deverá ser autorizada por um administrador, salvo se o utilizador tenha permissões para tal.

A base de dados deverá estar protegida de acessos não-autorizados.

Deverá ser feito o registo das operações mais “sensíveis” e que se achar mais convenientes, permitindo a rastreabilidade das mesmas.

#### **Acessibilidade**

A aplicação deverá ser concebida, construída e instalada para que todos os utilizadores possam ter igual acesso à informação e funcionalidades, independentemente de terem alguma deficiência ou não. Para tal deverá seguir as normas WCAG<sup>1</sup>. [9], [10]

#### **Multi-idioma**

O sistema deverá ter suporte para vários idiomas, inicialmente português e inglês, podendo ser posteriormente adicionados novos idiomas.

## **3.2. Casos de Uso**

Após o levantamento de requisitos é necessário uma representação das funcionalidades do sistema, bem como a explicação mais detalhada de cada uma delas, assim como as acções que o utilizador deverá ter com o mesmo.

A seguir são apresentados os diagramas de caso de usos das funcionalidades do sistema.

---

<sup>1</sup> Web Content Accessibility Guidelines

### 3.2.1. Diagrama de casos de uso - “Relação atores”

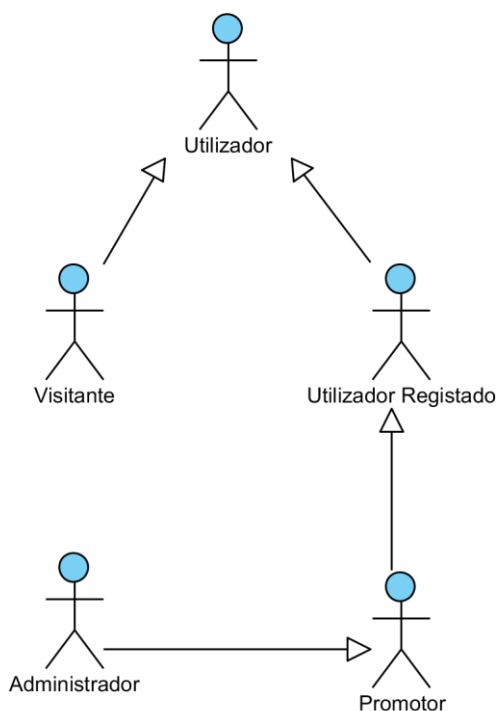


Figura 1 – Diagrama de Casos de Uso – “Relação Atores”






ID	Nome	Descrição
AC01	 Utilizador	Agente externo/pessoa ao sistema que interage com o mesmo.
AC02	 Visitante	Utilizador sem permissões no sistema. Apenas pode visualizar as páginas em que não tenha qualquer tipo de restrição.
AC03	 Utilizador Registrado	Utilizador que tem uma conta aprovada e que está autenticado no sistema. Pode aceder a um maior número de recursos do sistema.
AC04	 Promotor	Utilizador Registrado com permissões de gestão de eventos e espaços.
AC05	 Administrador	Utilizador Registrado com todas as funcionalidades de um promotor, com permissões de administração de todo o sistema.

Tabela 4 – Formato breve dos Casos de Uso do Diagrama – “Relação Actores”

## 3.2.2. Diagrama de casos de uso – “Geral”

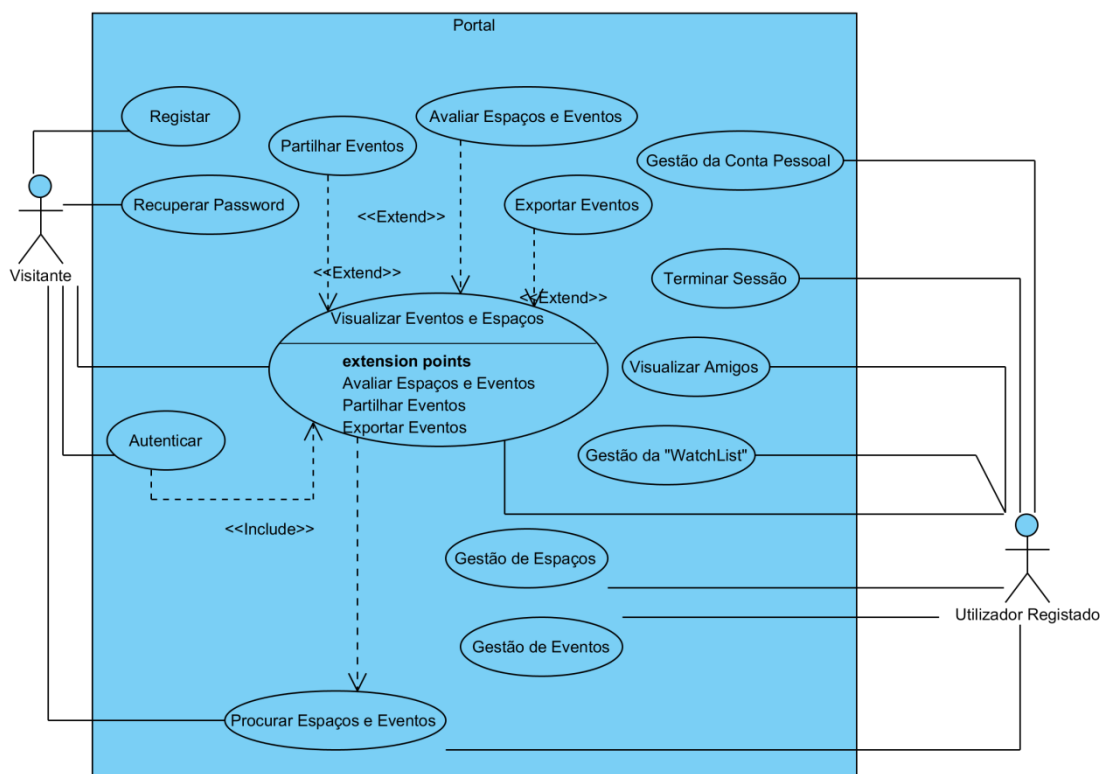


Figura 2 – Diagrama de Casos de Uso – “Geral”

ID	Nome	Descrição
UC08	Procurar Espaços e Eventos	O utilizador pesquisa eventos e/ou espaços, com possibilidade de usar vários filtros, como por exemplo a localização atual; outra localização; tipos de eventos/espaços; por data; De amigos/ que amigos participam.
UC11	Autenticar	O utilizador fornece as suas credenciais para ter acesso ao sistema. Poderá autenticar-se via Facebook e Google Oauth.
UC12	Visualizar Eventos e Espaços	O utilizador visualiza os detalhes de um evento ou espaço. Dentro dos detalhes, poderá visualizar fotografias e vídeos, podendo consultar informações como: Informações gerais (nome, descrição); Localização; Contactos; Bilheteiras; Programas (eventos); Estatísticas de quem vai (eventos);
UC13	Registrar	O utilizador deve registar-se no sistema para poder utilizar as principais funcionalidades do sistema. Deve ser possível o registo via Facebook e Google Oauth. O registo é principalmente constituído por inserir os dados pessoais do utilizador e a ativação da conta.
UC14	Recuperar Password	Permite a um utilizador registado no sistema que se tenha esquecido da sua password, que lhe seja atribuída uma nova.
UC15	Visualizar Amigos	O utilizador pode visualizar os seus amigos e as informações destes, como: Eventos a que pretendam ir; Eventos a que foi; etc.
UC17	Partilhar Eventos	O utilizador partilha um evento através do: Facebook; Google+; Twitter; LinkedIn; Email; ...

UC18	Exportar Eventos	O utilizador exporta um evento através de: iCal; .ics; Google Calendar; ...
UC19	Gestão da "WatchList"	O utilizador faz a gestão da sua "watchlist", organizando assim os eventos a que pretende assistir. Pode configurar alertas para: Eventos futuros; Eventos com alterações; etc.
UC21	Avaliar Espaços e Eventos	O utilizador avalia um espaço ou evento. No caso dos eventos só poderá avaliar após estes começarem.
UC27	Terminar Sessão	O utilizador termina a sua sessão, voltando a ser um simples visitante.
UC28	Gestão da Conta Pessoal	O utilizador faz a gestão da sua conta pessoal, podendo: editar o seu perfil; alterar o seu email e password, eliminar a sua conta, etc.
UC05	Gestão de Espaços	O utilizador faz a gestão dos seus espaços, podendo: Adicionar novos espaços; Editar espaços existentes; etc.
UC04	Gestão de Eventos	O utilizador faz a gestão dos seus eventos podendo entre outros, adicionar novos eventos; atualizar eventos existentes; importar eventos de outros sistemas; etc.

Tabela 5 – Formato breve dos Casos de Uso do Diagrama – "Geral"

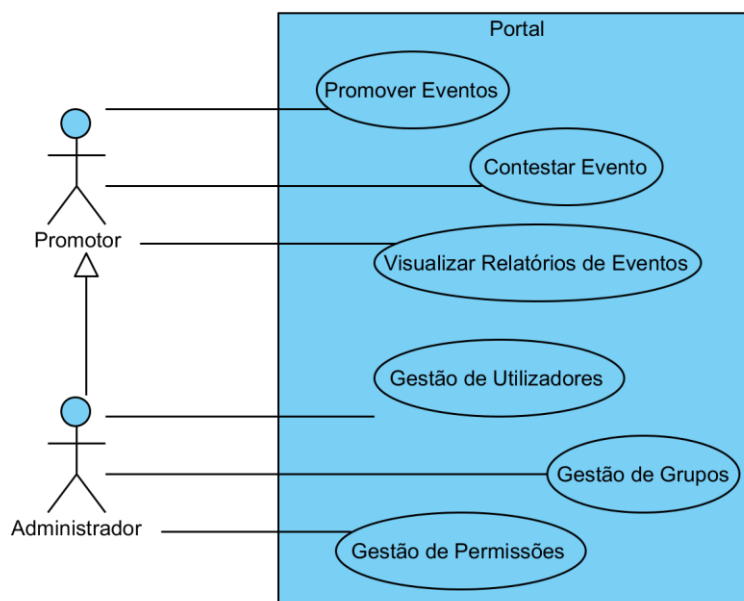


Figura 3 – Diagrama de Casos de Uso – "Geral (continuação)"

ID	Nome	Descrição
UC01	Gestão de Utilizadores	O administrador faz a gestão de utilizadores no sistema, podendo: <ul style="list-style-type: none"> <li>• Ativar/desativar utilizadores;</li> <li>• Editar todas as informações utilizadores;</li> <li>• Apagar utilizadores;</li> <li>• Ver dados estatísticos dos utilizadores;</li> </ul>
UC03	Gestão de Permissões	O administrador faz a gestão de permissões por grupo e/ou utilizador.

UC02	Gestão de Grupos	<p>O administrador faz a gestão de grupos de utilizador:</p> <ul style="list-style-type: none"> <li>• Criar grupos;</li> <li>• Adicionar utilizadores a grupos;</li> <li>• Editar grupos;</li> <li>• Apagar grupos;</li> </ul>
UC20	Contestar Evento	O utilizador faz um requerimento a requerer a gestão do evento.
UC07	Visualizar Relatórios de Eventos	<p>O utilizador visualiza os relatórios dos eventos já realizados podendo visualizar:</p> <ul style="list-style-type: none"> <li>• Quem esteve presente;</li> <li>• Dados demográficos de quem esteve presente;</li> <li>• outras estatísticas úteis.</li> </ul>
UC06	Promover Eventos	O utilizador envia sms ou emails a promover os seus eventos.

Tabela 6 – Formato breve dos Casos de Uso do Diagrama – “Geral (continuação)”

### 3.2.3. Diagrama de casos de uso – “Gestão de eventos”

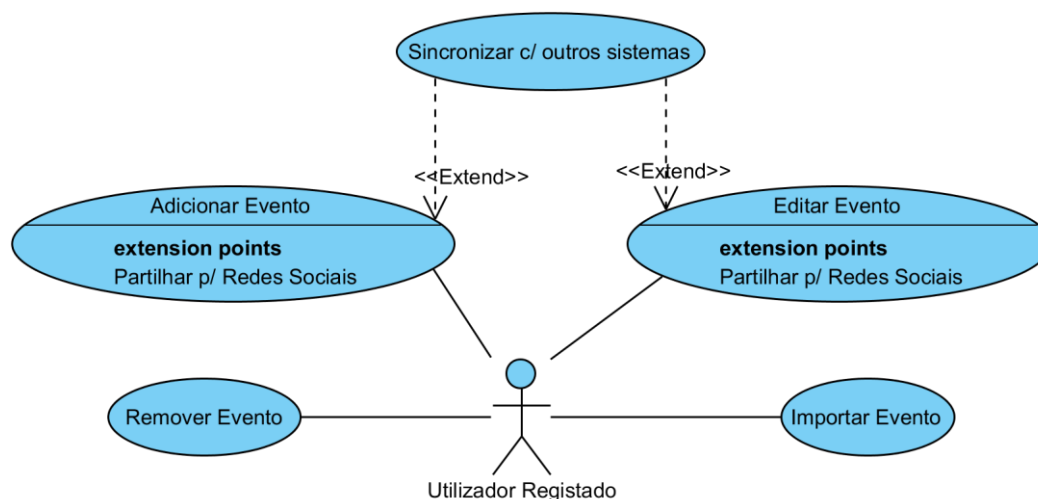


Figura 4 – Diagrama de Casos de Uso – “Gestão de Eventos”

ID	Nome	Descrição
UC04.UC01	Adicionar Evento	O utilizador adiciona um evento ao sistema, indicando variados detalhes e informações como informações gerais (nome, descrição, cartaz); localização; contactos; bilheteiras; programas; etc.
UC04.UC02	Editar Evento	O utilizador modifica os detalhes e informações de um evento.
UC04.UC03	Remover Evento	O utilizador elimina um evento do sistema.
UC04.UC04	Sincronizar c/ outros sistemas	O utilizador sincroniza os seus eventos com sistemas externos, nomeadamente com outras redes sociais.
UC04.UC05	Importar Evento	O utilizador importa um evento através de redes sociais ou através de formatos como iCal e .ics.

Tabela 7 – Formato breve dos Casos de Uso do Diagrama – “Gestão de Eventos”

### 3.3. Modelo de Domínio

Após a definição dos casos de uso é possível identificar as classes conceptuais do sistema:

- **User** – Classe representativa de um utilizador. A cada utilizador podem estar associadas várias permissões de acesso a funcionalidades do sistema.
- **Group** – Classe representativa de um grupo de utilizadores. A cada grupo podem estar associadas várias permissões de acesso a funcionalidades do sistema.
- **Country** – Classe representativa de um país.
- **Place** – Classe representativa de um espaço/local.
- **Event** – Classe representativa de um evento. Um evento pode ter vários sub-eventos.
- **Session** – Classe representativa de uma sessão/horário de um evento.
- **Media** – Classe representativa de uma media (imagem, vídeo).
- **Contact** – Classe representativa de um contacto.
- **Type** – Classe representativa de um tipo/tag.

Depois de identificadas as classes conceptuais candidatas do sistema, é necessário identificar as associações entre elas (denominação e multiplicidade) e os seus atributos, o que resulta no Modelo de Domínio (Figura 5). Optou-se por um modelo relativamente simples, que capturasse apenas os principais conceitos e ligações. Por exemplo, as redes sociais não estão aqui representadas.

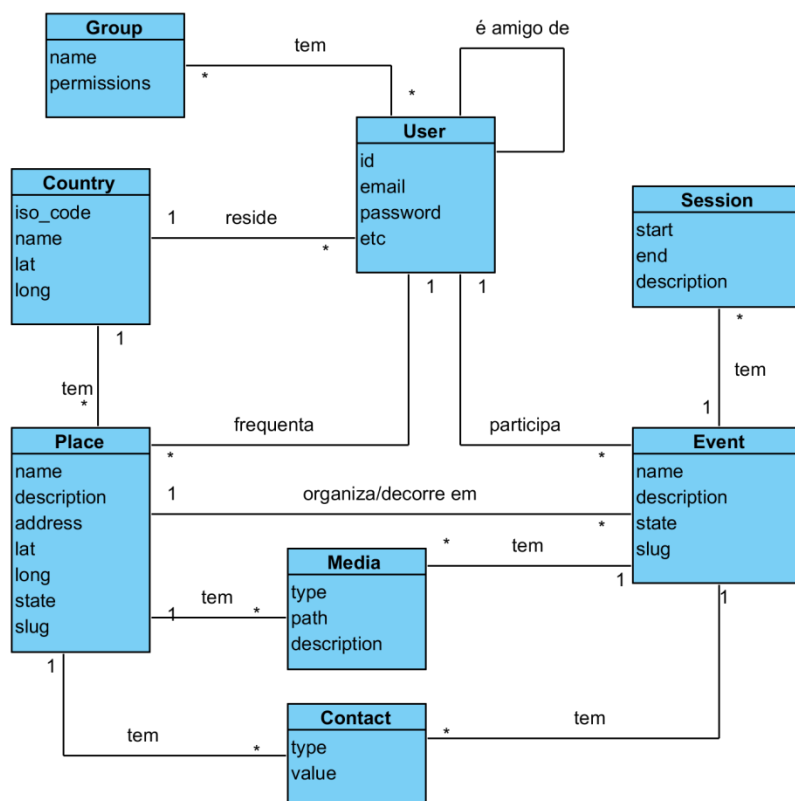


Figura 5 – Modelo de domínio do sistema

### 3.4. Arquitetura Lógica

O sistema utilizará uma arquitetura cliente-servidor baseada em *web services* (Figura 6), em que os *web browsers* e as aplicações móveis serão os clientes e o servidor *web* o servidor, que tem como obrigação responder aos pedidos dos clientes e efetuar pedidos às APIs dos sistemas externos (Facebook, Twitter, Google, ...).

Esta abordagem favorece a modularidade e escalabilidade do sistema, permitindo a fácil adição de novos serviços e a integração com mais fornecedores externos. Esta arquitetura presta-se a ser implementada numa infraestrutura do tipo *cloud* (Amazon EC2, Heroku, etc.), pelo que o aumento de carga pode ser facilmente resolvido com colocação em serviço de mais instâncias de *workers*. O principal risco de estrangulamento ao nível da escalabilidade será na base de dados, pelo que se verificar o aumento exponencial dos dados será necessário implementar *balancers*.

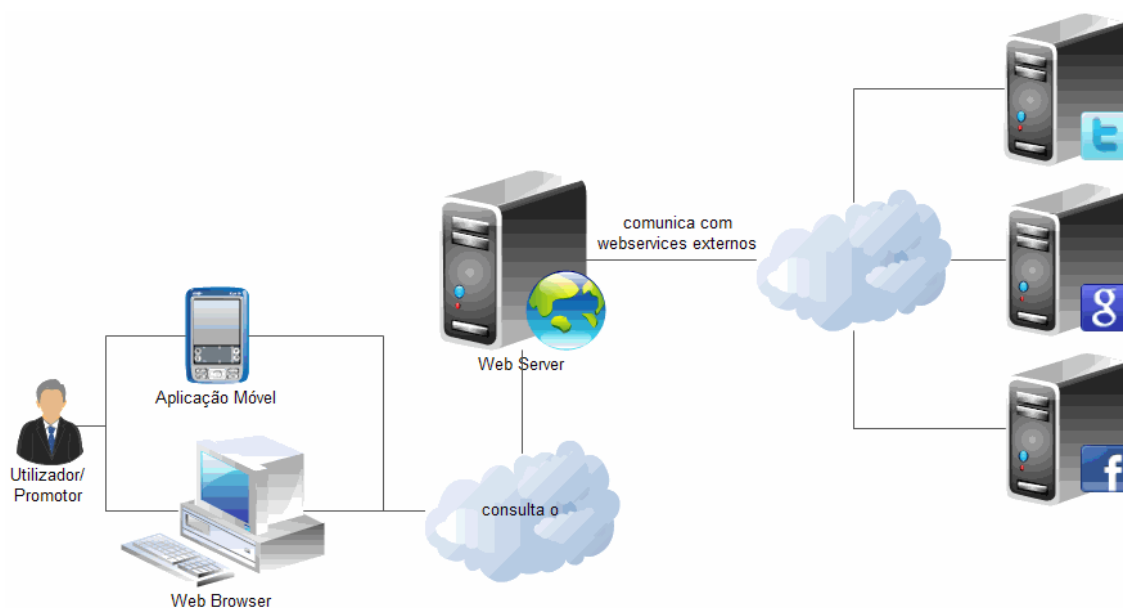


Figura 6 – Arquitetura lógica do sistema

# 4. ANÁLISE E MODELAÇÃO

*Este capítulo numa primeira parte evidencia os componentes do sistema. Numa segunda parte, é apresentado o estado atual da tecnologia usada para o desenvolvimento da solução e, por fim, na terceira parte pormenoriza a preocupação para com a qualidade e correto funcionamento da solução final, ao descrever a metodologia de testes seguida, demonstrando como o desenvolvimento orientado a testes para escrever e testar o código-fonte se torna mais eficiente e descrevendo alguns dos testes efetuados.*

## 4.1. Diagrama de componentes

O sistema final será constituído primeiramente por dois grandes componentes que serão o portal e as aplicações móveis. O Portal está dividido em três subcomponentes responsáveis pelo frontoffice, backoffice e pela API utilizada para comunicação entre aplicações móveis/sistema. Por último existirão componentes mais específicos em que cada um deles será responsável por um conjunto de funcionalidades distintas (Figura 7):

- **Import/Export:** Responsável pela importação e exportação de informação de/para outros formatos.
- **Social:** Responsável pela comunicação com serviços externos, como por exemplo pela autenticação ou a gestão da informação através destes.
- **Report:** Responsável por gerar relatórios de eventos, úteis aos promotores.
- **Payment:** Responsável pelos pagamentos do sistema.

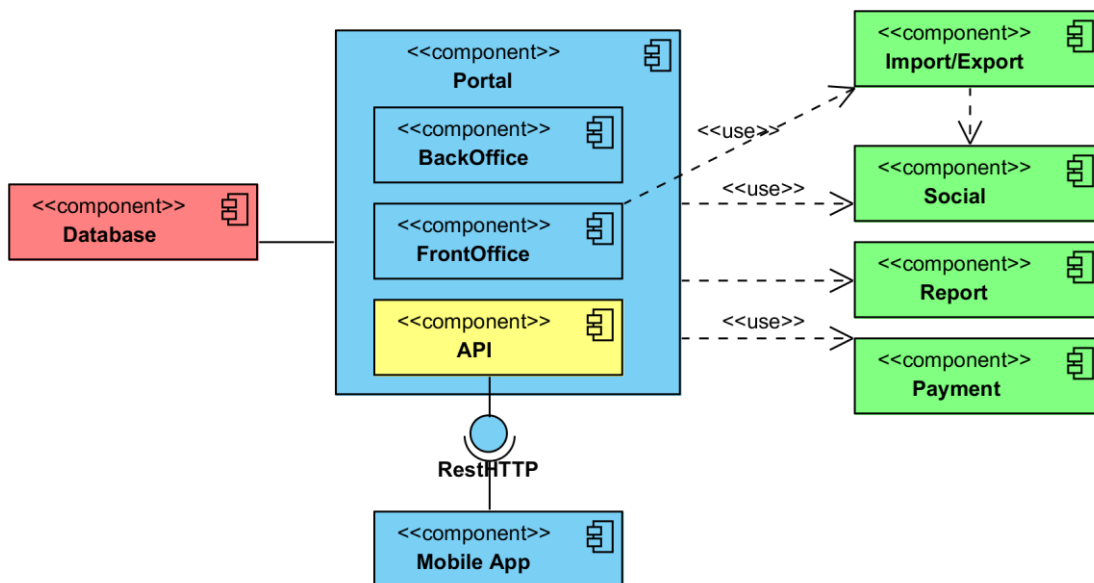


Figura 7 – Diagrama de componentes do sistema final

## 4.2. Tecnologia

O desenvolvimento deste projeto utilizará, sempre que possível, tecnologias *Open Source*, primeiramente para diminuir custos de desenvolvimento e manutenção, e por ser possível adaptar futuramente as tecnologias às necessidades do projeto, se tal for necessário.

Foi escolhido o PHP<sup>1</sup> como linguagem de programação do portal, primeiramente por ser uma linguagem que foi projetada para a criação de *scripts web* [11] e por ser a linguagem mais utilizada neste contexto, tendo assim uma imensa comunidade à sua volta, sendo muito fácil encontrar qualquer tipo de recurso. Em segundo lugar pela experiência que o programador tem no desenvolvimento nesta linguagem (cerca de 5 anos), poupando assim tempo ao evitar ter que se aprender outra linguagem.

A linguagem PHP esteve nos últimos anos quase estagnada, com as *frameworks* a tornarem-se obsoletas, estando a perder terreno para o Ruby on Rails, em termos de inovação e facilidade de desenvolvimento [12]. O último ano foi um excelente ano para a comunidade PHP, graças à adição de vários recursos úteis nas versões 5.3, 5.4 e 5.5, bem como os inúmeros projetos que contribuíram para que a linguagem avançasse para o próximo nível.

### 4.2.1. Composer<sup>2</sup>

Inspirado em ferramentas como Bundler<sup>3</sup> e NPM<sup>4</sup>, a comunidade PHP agora pode parar de reinventar a roda vezes sem conta graças ao *composer*. O *node.js*<sup>5</sup> foi a primeira tecnologia que tornou o uso de “*packages*” (pacotes) confortável. Os *packages* são instalados localmente para a pasta do projeto, sendo relativamente fácil encontrar documentação para a maioria dos *plugins* e de enviar os nossos próprios pacotes. [13], [14]

O PHP, há uns anos atrás, ofereceu o PEAR e o PECL como alternativa, mas estes não eram muito intuitivos e fáceis de usar. Além disso, instalavam todos os *packages* globalmente, estando as dependências ligadas ao PHP ao invés das próprias aplicações. Isto originava problemas caso houvesse várias aplicações exigindo diferentes versões do mesmo *package*.

O *composer* veio colmatar estas falhas, armazenando os *packages* localmente e tendo a capacidade de criar um ficheiro de dependências (*composer.json*) por projeto. Isto significa que se pode facilmente distribuir os projetos com este ficheiro de dependências. Por exemplo, caso se esteja a desenvolver uma aplicação que necessite de se ligar às redes sociais Facebook e Twitter, é necessário incluirmos os SDKs de cada uma. Através do *composer* isso é muito

---

<sup>1</sup> <http://php.net/>

<sup>2</sup> <http://getcomposer.org/>

<sup>3</sup> <http://bundler.io/>

<sup>4</sup> <http://npmjs.org/>

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

fácil. Basta criar o ficheiro seguinte nomeando-o de `composer.json` e correr o comando `composer update` (depois de ter o *composer* instalado) [15].

```
composer.json
{
    "require": {
        "facebook/php-sdk": "v3.2.*",
        "j7mbo/twitter-api-php": "dev-master"
    },
    "require-dev": {
        "phpunit/phpunit": "dev-master",
        "mockery/mockery": "dev-master"
    }
}
```

Código Fonte 1 – Exemplo de gestão de dependências com composer

Neste caso, o projeto requer a versão 3.2 mais recente do SDK do Facebook e a versão mais recente da API do Twitter. Ainda requer o phpUnit e o *mockery* caso a aplicação seja executada em modo de desenvolvimento.

O *composer* é uma aplicação escrita em PHP e vem com um recurso de *autoloader*, que irá incluir automaticamente as dependências estipuladas, deixando as aplicações o mais simples possível.

Todos estes recursos são uma melhoria definitiva, no entanto, sem a adoção da comunidade, isso nada significaria. Esta ferramenta foi muito bem aceite na comunidade PHP. Grandes projetos, como a *Symfony*<sup>1</sup> e o *Laravel*<sup>2</sup>, já utilizam o *composer* para fazer a gestão das suas dependências e dos seus componentes [16], [17]<sup>3</sup>. Ter uma *framework* dividida em componentes significa que se pode facilmente construir a sua própria *framework* personalizada de acordo com as nossas necessidades. Por outras palavras o programador pode escolher os componentes de cada *framework* que convém.

O *composer* tem um repositório de componentes (*Packagist*<sup>4</sup>), em que é possível encontrar mais de 15 000 componentes diferentes que a comunidade poderá usar nos seus projetos [18].

---

<sup>1</sup> <http://symfony.com/>

<sup>2</sup> <http://laravel.com/>

<sup>3</sup> Procederam ao envio dos seus componentes para a biblioteca do composer.

<sup>4</sup> <https://packagist.org/>

## 4.2.2. Laravel

Como já foi referido, a linguagem PHP esteve a perder o entusiasmo da comunidade de programadores web. Algumas das suas *frameworks*<sup>1</sup> mais emblemáticas, como o CodeIgniter<sup>1</sup>, estavam a tornar-se obsoletas rapidamente.

No entanto, no último tempo surgiram *frameworks* da nova geração (Symfony 2, FuelPHP<sup>2</sup>, Laravel, ...) que tornaram novamente o PHP “atraente”. Graças a estas, o PHP já não está condenado a ser utilizado para a maioria dos *sites* pessoais ou para os *blogs* WordPress. É graças a estas, ao composer e às últimas versões do PHP que, na minha opinião, estamos prestes a experienciar o próximo renascimento da mais popular linguagem *server-side* da web. [13]

Foi escolhida para a base do desenvolvimento do projeto a *framework* Laravel (versão 4), por ser uma *framework* da nova geração, por ter uma boa semântica e sintaxe, por ser criativa, o que lhe permite destacar-se entre um grande número de *frameworks*. Isto faz que se use o PHP facilmente, sem sacrificar a potência e a eficiência [15], [19]. A Laravel é uma ótima opção, tanto para projetos amadores como para soluções mais exigentes, como é o caso deste projeto. Embora a sua documentação seja algo fraca, o seu código-fonte está muito bem comentado, o que reduz esse problema. A comunidade também tem ajudado a colmatar esse problema criando vários artigos, livros, *screencasts*, diminuindo assim a sua curva de aprendizagem [13], [20]. Outro facto interessante desta *framework* é o de utilizar vários componentes da *framework* Symfony, graças ao composer.

Em seguida são apresentados alguns componentes/recursos úteis desta *framework*.

### 4.2.2.1. Eloquent<sup>3</sup>

O *eloquent* é um ORM incluído no Laravel que fornece uma implementação simples e elegante do padrão *ActiveRecord* para trabalhar com base de dados. Cada tabela de uma base de dados tem um *Model* (Modelo) correspondente que é usado para interagir com essa tabela. Podem-

---

<sup>1</sup> <http://ellislab.com/codeigniter>

<sup>2</sup> <http://fuelphp.com/>

<sup>3</sup> <http://laravel.com/docs/eloquent>

se fazer muitas coisas, como a combinação de resultados de várias tabelas, para economizar chamadas à base de dados (conhecido como *eager loading*<sup>1</sup>) e é simples de se usar.

```
// define a User Model
class User extends Eloquent {
    protected $table = 'users';
}

// get all users
$users = User::all();

// get users with age > 18
$users = User::where('age', '>', '18');

// get user with an id of 1
$user = User::find(1);

// prints user email
echo $user->email;

// update a user
$user->email = 'miguelborges@miguelborges.com';
$user->age = 23;
$user->save();

// delete a user
$user->delete();
```

Código Fonte 2 - Exemplos básicos do uso do ORM Eloquent

Considere a seguinte relação entre tabelas: um utilizador tem várias tarefas. Em Laravel, depois de estabelecer um método de pesquisa rápida para cada modelo, pode facilmente manipular qualquer tipo associação. Eis alguns exemplos para exemplificar a simplicidade de uso.

```
// Get all tasks by the author with an id of 1
$tasks = User::find(1)->tasks;

// Get the author of a task
$author = Task::find(5)->user->username;

// Insert a new task by author
$task = new Task([ 'title' => 'Go to store.' ]);
User::find(1)->tasks()->insert($task);
```

Código Fonte 3 - Exemplos de relações com o ORM Eloquent

### 4.2.2.2. Migrations<sup>2</sup>

As migrações podem ser vistas como um tipo de controlo de versões para as bases de dados. Estas foram ganhando popularidade através do *Ruby on Rails* e que algumas das novas *frameworks* trouxeram para o PHP.

---

<sup>1</sup> <http://laravel.com/docs/eloquent#eager-loading>

<sup>2</sup> <http://laravel.com/docs/migrations>

Estas permitem a uma equipa modificar o esquema da base de dados, mantendo todas as cópias da aplicação actualizadas. As migrações são utilizadas em conjunto com o *Schema Builder*<sup>1</sup> para fazer facilmente a gestão do esquema de dados das aplicações.

```
<?php
use Illuminate\Database\Migrations\Migration;

class CreateUsersTable extends Migration {
    /**
     * Run the migrations.
     */
    public function up() {
        Schema::create('users', function($table) {
            $table->increments('id');
            $table->string('email')->unique();
            $table->string('password');
            $table->string('firstname')->nullable();
            $table->string('lastname')->nullable();
            $table->boolean('activated')->default(0);
            $table->text('description')->nullable();
            $table->timestamps();
            $table->softDeletes();

            $table->engine = 'InnoDB';
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down() {
        Schema::drop('users');
    }
}
```

Código Fonte 4 - Criação da tabela users com *schema builder* e *migrations*

Depois dos *schemas* das tabelas definidos só é necessário correr o comando `php artisan migrate` para aplicar as novas alterações.

Através desta funcionalidade é possível:

- Criar novas migrações `php artisan migrate:make create_users_table`
- Aplicar todas as migrações pendentes `php artisan migrate`
- Fazer rollback à última migração aplicada `php artisan migrate:rollback`
- Fazer rollback a todas as migrações `php artisan migrate:reset`
- Fazer rollback a todas as migrações e aplica-las novamente
  - `php artisan migrate:refresh`
  - `php artisan migrate:refresh --seed`

O Laravel também oferece uma forma rápida e fácil de população da base de dados

<sup>1</sup> <http://laravel.com/docs/schema>

```

class UserTableSeeder extends Seeder {
    /**
     * Run the database seeds.
     * @return void
     */
    public function run() {
        Eloquent::unguard();

        // Flush all rows
        $users = [
            [
                'email' => 'miguelborges@miguelborges.com',
                'password' => 'zsdgfgghk',
                'firstname' => 'Miguel',
                'lastname' => 'Borges'
            ], [
                'email' => 'mapb.1990@gmail.com',
                'password' => 'dfgh',
                'firstname' => 'Angelo',
                'lastname' => 'Pinto'
            ]
        ];

        DB::table('users')->insert($users);
    }
}

```

Código Fonte 5 - População da tabela users

### 4.2.2.3. Routes<sup>1</sup>

A maioria dos programadores novatos de PHP não está familiarizada com qualquer coisa que não seja o mais natural dos sistemas de rotas. Criam uma árvore de pastas para combinar com URI desejado e mais nada. Por exemplo, adicionar um ficheiro `index.php` à pasta `blog/admin/` para depois acede-lo através da URL `localhost/blog/admin/index.php`. Para aplicações simples e pequenas pode até servir, mas provavelmente precisar-se-á de mais flexibilidade e controlo sobre qual a rota que é acionada na aplicação.

A Laravel tem uma abordagem extremamente simples e fácil de usar para roteamento. Como exemplo, vamos escrever o caminho necessário para a visualização de uma view (vista) para um perfil de utilizador.

```

Route::get('users/{id}', function($id) {
    // find the user
    $user = User::find($id);

    // display view, and pass user object
    return View::make('users.profile')
        ->with('user', $user);
});

```

Código Fonte 6 - Exemplo de uma rota com laravel

<sup>1</sup> <http://laravel.com/docs/routing>

Agora, sempre que alguém fizer um request à URI `exemplo.com/users/1`, a view `users/profile.php` irá ser exibida.

Alternativamente poderá também utilizar os tradicionais `controllers` (controladores) para lidar com a lógica.

```
Route::get('users/{id}', 'Users@show');
```

Agora, `controllers/user.php` será responsável de exibir a view.

```
class UsersController extends Controller {
    /**
     * Display the specified resource.
     */
    public function show($id)
    {
        // find the user
        $user = User::find($id);

        // display view, and pass user object
        return View::make('users.profile')
            ->with('user', $user);
    }
}
```

Código Fonte 7 - Exemplo de um `controller` em laravel.

#### 4.2.2.4. Outros recursos/funcionalidades

- Artisan<sup>1</sup> - É uma ferramenta de linha de comandos. Esta fornece uma série de comandos úteis para o uso durante o desenvolvimento de uma aplicação. É impulsionada pelo componente Symfony Console.
- Blade<sup>2</sup> - Simples motor de *template*. Suporta herança de templates e secções.
- Resourceful Controllers<sup>3</sup> - Permite facilmente construir serviços e APIs RESTful.

---

<sup>1</sup> <http://laravel.com/docs/artisan>

<sup>2</sup> <http://laravel.com/docs/templates#blade-templating>

<sup>3</sup> <http://laravel.com/docs/controllers#resource-controllers>

### 4.3. Desenvolvimento orientado por testes

Testar o código-fonte pode se tornar aborrecido, mas o impacto de não o fazer pode originar situações ainda mais aborrecidas!

#### 4.3.1. Importância dos testes

Apesar de os computadores serem máquinas, os programadores são humanos e desenvolvem *software* com erros, normalmente denominados de *bugs*. É uma ocorrência praticamente inevitável em sistemas de média ou grande dimensão, mesmo quando desenvolvidos pelos melhores profissionais. A especificação e validação formal de *software* é extremamente complexa e exigente em recursos computacionais, sendo raramente aplicável em sistemas minimamente complexos. Não sendo viável a validação formal, a alternativa é o teste do *software*.

O desenvolvimento orientado por testes (*Test-Driven Development* ou TDD, em inglês) é uma técnica de programação que exige que se escreva o código e os testes automatizados simultaneamente. Isso garante que se teste o código e permite testar novamente o código de forma rápida e fácil, tantas vezes quando forem necessárias, uma vez que é automatizado [21], [22].

O desenvolvimento orientado por testes ou TDD anda à volta de um curto ciclo de desenvolvimento iterativo:

1. Antes de se escrever qualquer código, primeiro deve-se escrever um teste automatizado para o código. Ao escrever os testes automatizados, deve-se levar em conta todas as entradas possíveis, erros e saídas.
2. A primeira vez que se executar o teste automatizado, o teste deve falhar, indicando que o código ainda não está pronto.
3. Começa-se a programação. Uma vez que já existe um teste automatizado, enquanto o código falhar no teste, significa que ele ainda não está pronto. O código pode ser corrigido até que ele passe em todas as afirmações.
4. Uma vez que o código passe no teste, pode-se começar a limpá-lo. Enquanto o código passar no teste, significa que ele continua a funcionar.
5. Passa-se para o método/função seguinte e recomeça-se o processo.

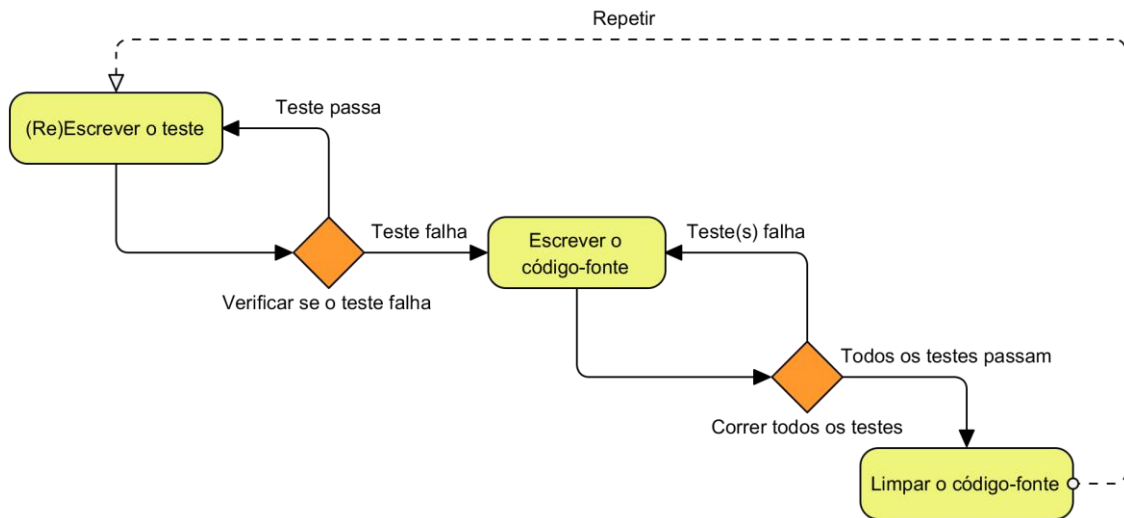


Figura 8 – Ciclo desenvolvimento TDD

As principais vantagens do TDD são [23], [24]:

- Baixa drasticamente a probabilidade de ocorrência de *bugs* num *software* lançado para produção;
- Facilita a criação de código com alva coesão e baixo acoplamento;
- Reduz o tempo gasto em depuração e em correção de bugs;
- Serve de suporte para testes de regressão;
- Encoraja o *refactoring*;
- Serve como documentação.

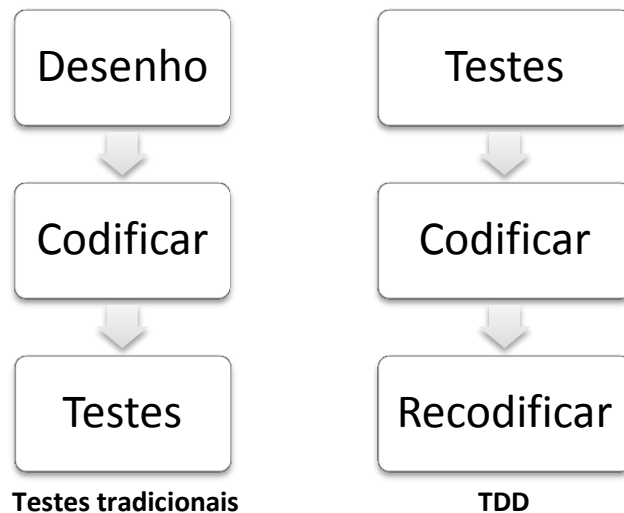


Figura 9 – Comparação entre o TDD e os testes tradicionais

### 4.3.2. Configuração do ambiente

A versão 4 da Laravel oferece grandes melhorias em relação aos testes, quando comparado à sua versão anterior. É possível configurar um ambiente de testes, distinto do ambiente de desenvolvimento. Isso aplica-se à base de dados. Ao colocar as configurações da DB (`database.php`) dentro da pasta de configuração de testes (`app/config/testing`) significa que essas configurações só serão utilizadas num ambiente de testes (que a própria Laravel define automaticamente). Como tal, quando a aplicação é acedida normalmente, as configurações de teste não serão aplicadas.

O ideal para o ambiente de teste seria que a BD estivesse sempre limpa, e com os dados originais. Para isso é necessário migrar os dados antes de cada teste. Para isso, adiciona-se o seguinte método à superclasse `app/tests/TestCase.php`, já disponibilizada com a *framework*.

```
/**
 * Migrates the database and set the mailer to 'pretend'.
 * This will cause the tests to run quickly.
 */
private function prepareForTests()
{
    Artisan::call('app:install');
    Mail::pretend(true);
}
```

Código Fonte 8 – Método para preparar o ambiente de desenvolvimento

Este método irá instalar BD e alterar o estado da classe `Mailer` da `Laravel` classe para `pretend`. Desta forma, o `Mailer` não enviará nenhuma *mail* real, na execução dos testes. Em vez disso, ele irá registar as mensagens "enviadas". Para finalizar é necessário invocar o método anterior no método `setUp()`, que é executado antes de cada teste, pelo `PHPUnit`.

```
/**
 * Default preparation for each test
 */
public function setUp()
{
    parent::setUp(); // Don't forget this!
    $this->prepareForTests();
}
```

A partir daqui, ao escrever os testes, pode-se simplesmente estender a classe `TestCase` e a BD será iniciada e migrada antes de cada teste.

### 4.3.3. Modelos

A expressão conhecida "modelo gordo, controlador magro" ("fat model, skinny controller", em inglês), é simplesmente um padrão que indica que é melhor manter os controladores o mais minimalistas possíveis: lidar com as respostas, enviar mensagens para o domínio, carregar vistas (*views*). Os cálculos ou as operações de qualquer complexidade devem ser colocados dentro de um modelo ou serviço. Por esta razão, é fundamental que os modelos sejam totalmente testados. [25]–[28]

A regra básica para saber o que é necessário testar num modelo é que tudo o que tem uma oportunidade de falhar deve ser testado. Mais especificamente o mínimo que se deve testar é:

- Validações;
- Scopes;
- Assessores e mutuadores;
- Associações/Relações;
- Métodos personalizados;

Segue um conjunto de testes realizado a um dos modelos da aplicação.

```
public function testIsValidwithoutName() {
    $this->assertNotValid($this->event);
}

public function testUserHavePermission() {
    $event = Event::find(1);

    should::beTrue($event->can('edit-contacts', false, 1));
}

public function testHaveSubEvent() {
    $this->assertHasMany('subevents', 'Event');
}

public function testIsUserHavePermissionOnSubevent() {
    $subevent = \App\Models\Event::find(8);
    $event->givePermissions(1, static::$permissions);
    should::beTrue($subevent->can('edit-contacts', false, 2));
}
```

Código Fonte 9 – Excerto de testes realizados ao modelo "Event"

```
PS \Tese\portal> .\vendor\bin\phpunit
PHPUnit 3.8-dev by Sebastian Bergmann.

Configuration read from D:\Users\Miguel
Borges\Documents\Trabalhos\Tese\portal\phpunit.xml

.....

Time: 928 ms, Memory: 14.50Mb

OK (12 tests, 18 assertions)
```

Código Fonte 10 – Output da execução dos testes

### 4.3.4. Controladores

Testar controladores não é a coisa mais fácil do mundo. O problema não é testa-los, mas sim saber o que testar. Os testes dos controladores devem verificar as respostas, garantir que os métodos de acesso aos dados são invocados, e se as variáveis são enviadas corretamente para as *views*.

A Laravel já fornece um conjunto de *Assertions* na superclasse `Illuminate\Foundation\Testing\TestCase`, que ajudam e reduzem o tempo necessário para executar as afirmações básicas:

- `assertViewHas`
- `assertResponseOk`
- `assertRedirectedTo`
- `assertRedirectedToRoute`
- `assertRedirectedToAction`
- `assertSessionHas`
- `assertSessionHasErrors`.

No Código Fonte 11 são apresentados alguns excertos de teste feito ao controlador `EventsController`, onde é demonstrado o uso de alguns dos *helpers* enunciados acima.

```
public function testIndex()
{
    $this->call('GET', 'events');
    $this->assertViewHas('events');

    $events = $response->original->getData()['events'];
    $this->assertInstanceOf('Illuminate\Database\Eloquent\Collection',
$events);
}

public function testStore()
{
    $this->mock
        ->shouldReceive('create')
        ->once()
        ->with($input);

    $this->app->instance('App\Models\Event', $this->mock);
    $this->call('POST', 'events');
    $this->assertRedirectedToRoute('events.index');
}

public function testStoreFails()
{
    // Set stage for a failed validation
    Input::replace(['title' => '']);

    $this->app->instance('App\Models\Event', $this->mock);
    $this->call('POST', 'events');

    // Failed validation should reload the create form
    $this->assertRedirectedToRoute('events.create');

    // The errors should be sent to the view
    $this->assertSessionHasErrors(['name']);
}
```

```
public function testStoreSuccess()
{
    // Set stage for successful validation
    Input::replace(['name' => 'Event Title', 'type' => 1]);

    $this->mock
        ->shouldReceive('create')
        ->once();

    $this->app->instance('App\Models\Event', $this->mock);
    $this->call('POST', 'events');

    // Should redirect to collection, with a success flash message
    $this->assertRedirectedToRoute('events.index', ['flash']);
}
```

Código Fonte 11 – Excerto de testes feitos ao controlador EventsController



# 5. IMPLEMENTAÇÃO

*Neste capítulo descreve a implementação dos principais módulos da solução, através da framework utilizada. É detalhado a arquitetura de cada um, destacando as principais funções de cada módulo. São também apresentadas algumas das soluções ao nível da segurança, usabilidade e acessibilidade.*

### 5.1. Introdução

Para a construção de um sistema robusto, mas ao mesmo tempo adaptável, extensível e facilmente entendido é necessário o uso de boas práticas de *software* como o uso de padrões GRASP e GoF. O uso de padrões de *software* é vantajoso pois facilita a comunicação entre programadores, descrevem uma solução tipificada e “provada”, facilita a reutilização dos componentes em larga-escala e capturam o conhecimento dos peritos e *trade-offs* efetuados. [29]

Robert Martin identificou os princípios SOLID, como os princípios *standard* da programação orientada a objetos, os quais fornecem uma boa base para o *design* de aplicações, sendo estes: [30]

- **Single Responsibility Principle:** Define que uma classe só deve ter uma, e somente uma razão para ser modificada;
- **Open Closed Principle:** Define que o código deve estar aberto para extensões, mas fechado para modificações;
- **Liskov Substitution Principle:** Estabelece que os objetos devem ser substituíveis com instâncias dos seus subtipos, sem alterar o correto funcionamento da aplicação;
- **Interface Segregation Principle:** Este princípio afirma que a implementação de uma interface não deve ser forçada a depender de métodos que não usa;
- **Dependency Inversion Principle:** Define que o código de alto nível não deve depender de código de baixo nível, e que as abstrações não devem depender de detalhes.

A construção do sistema seguiu estes mesmos princípios utilizando sempre que necessário padrões de *software*.

### 5.2. Componentes Chave

#### 5.2.1. Base do Sistema

Como foi referido anteriormente o portal foi desenvolvido utilizando a *framework* Laravel (Pág. 29) e por conseguinte baseada no padrão MVC de arquitetura de *software*. Este padrão

possibilita a fácil manutenção e expansão do sistema, podendo ser definidos componentes independentes.

A Laravel já disponibiliza uma estrutura e organização para os componentes base que definem o sistema (modelos, controladores e visualizações). No entanto foi necessário adapta-las para uma melhor organização e manutenção do mesmo (Figura 10).

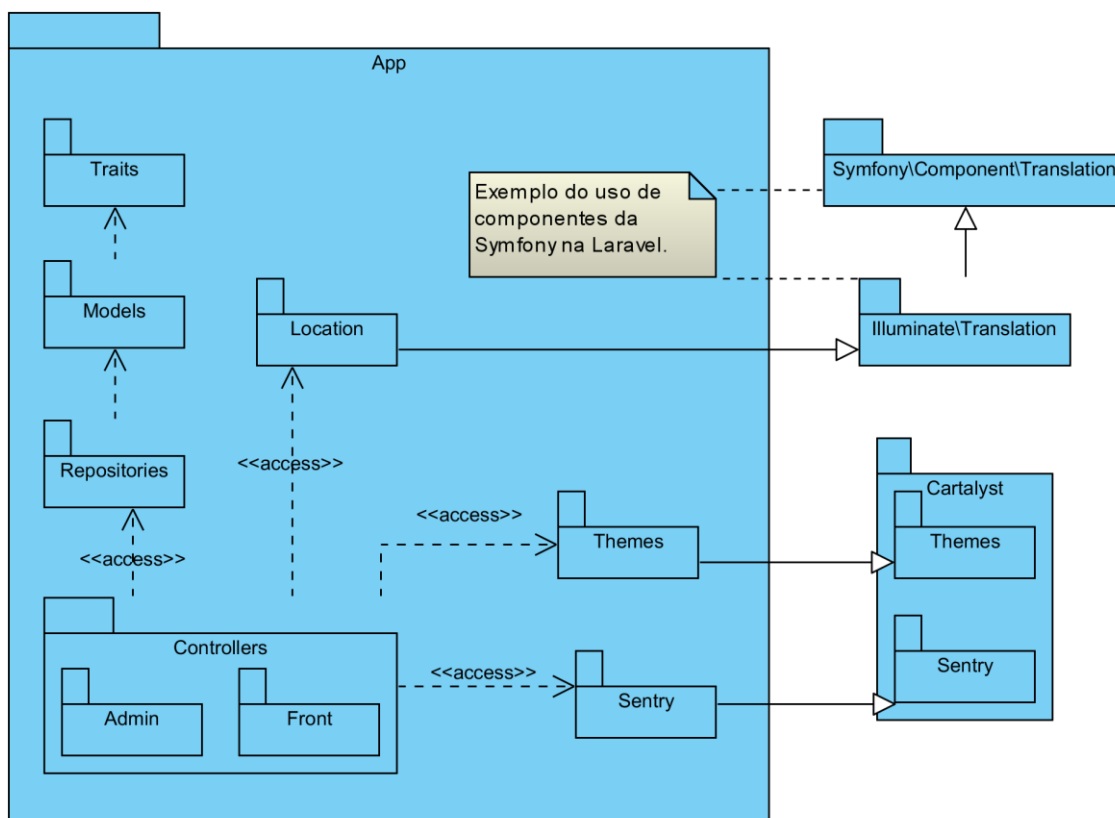


Figura 10 – Diagrama de pacotes do sistema

Nome	Descrição
App	Pacote base do sistema. É neste pacote que se encontra os componentes base do sistema.
Traits	Pacote onde se encontram algumas traits base do sistema.
Symfony\Component\Translation	Componente de localização da Symfony.
Models	Pacote onde se encontram os modelos do sistema.
Location	Extensão do componente de localização da Laravel.
Illuminate\Translation	Componente de localização da Laravel.









Nome	Descrição
 Repositories	Conjunto de repositórios de acesso aos dados (ver secção 5.2.1.2 Repositórios).
 Themes	Extensão do componente de temas da Cartalyst.
 Cartalyst\Themes	Componente de temas da Cartalyst.
 Cartalyst\Controllers	Pacote onde se encontram os controladores do sistema. Encontram-se divididos por Front e Admin.
 Sentry	Componente de gestão de utilizadores, grupos e permissões da Cartalyst.
 Sentry	Extensão do componente de gestão de utilizadores, grupos e permissões da Cartalyst.
 Admin	Controladores do backend.
 Front	Controladores do frontend.

Tabela 8 – Descrição dos pacotes base do sistema

Para além da utilização dos componentes fornecidos com a *framework*, foi necessário estender alguns, como é do componente de localização, onde foi necessário adicionar a funcionalidade de idioma padrão que caso uma tradução não exista no idioma em uso, este apresenta a tradução do idioma padrão; do componente de gestão de utilizadores, grupos e permissões da Cartalyst<sup>1</sup>, onde foi adicionado a funcionalidade de utilizador anónimo quando não há nenhum autenticado; e do componente de gestão de temas, também da Cartalyst, que foi adicionado a funções de funções individuais de tema (cada tema pode ter as suas próprias funções).

### 5.2.1.1. Modelos

Os modelos, na arquitetura MVC, são as classes que representam os dados da aplicação, as regras de negócios e a lógica. Foi utilizado o ORM da Laravel para a definição dos modelos e respetivo mapeamento para base de dados. Todos os modelos à exceção das classes `User`, `Group` e `Throttle`, herdam da classe `BaseModel`, sendo esta uma herança do Eloquent. Desta forma é possível adicionar várias funcionalidades a todos os modelos facilmente.

<sup>1</sup> <https://cartalyst.com/>

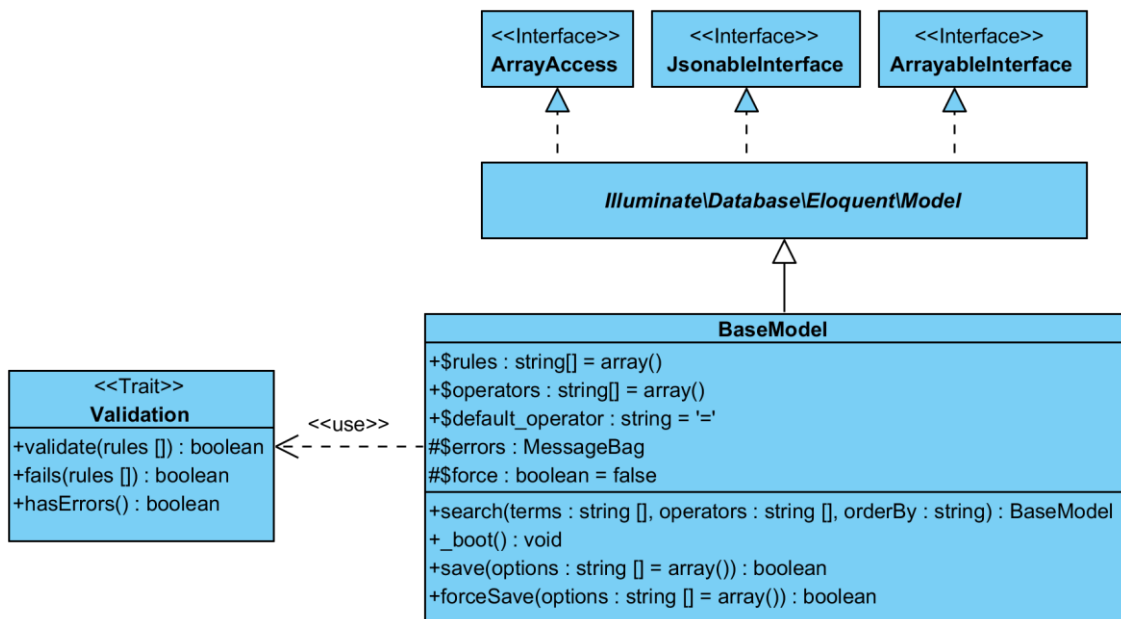


Figura 11 – Extensão do modelo Eloquent

Foi criada uma classe para representar os idiomas do sistema, a tratar da formatação dos dados relacionados com o mesmo, a data, a hora e números.

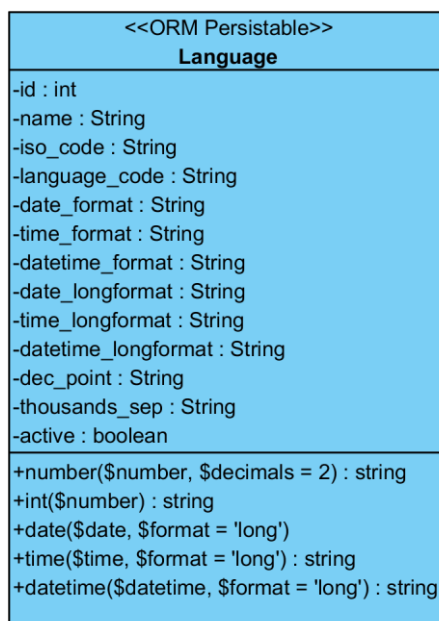


Figura 12 – Classe Language

Encontra-se na Tabela 9 a descrição breve das classes base do sistema.

Para o módulo de gestão de utilizadores, grupos e permissões foram definidas as classes ilustradas no diagrama de classes da Figura 13.

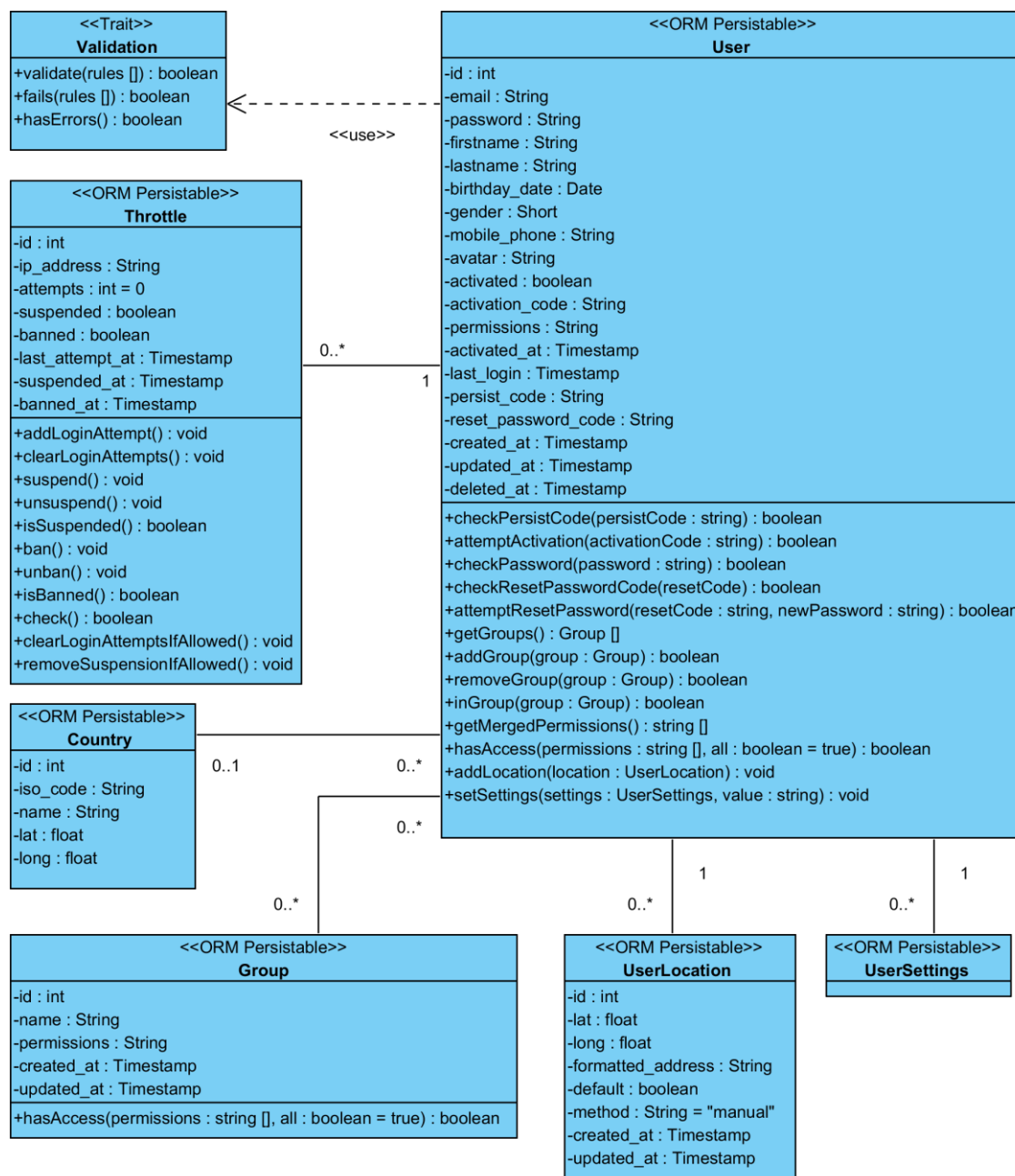


Figura 13 – Diagrama de classes do componente de gestão de utilizadores, grupos e permissões

A classe UserLocation representa as localizações de um utilizador, bem como a sua morada. Desta forma é possível saber onde o utilizador esteve e as suas zonas habituais. As opções da conta de utilizador foram transferidas para a classe UserSettings por uma questão de performance e estabilidade.

Para o módulo de gestão de espaços foram definidas as classes ilustradas no diagrama de classes da Figura 14.

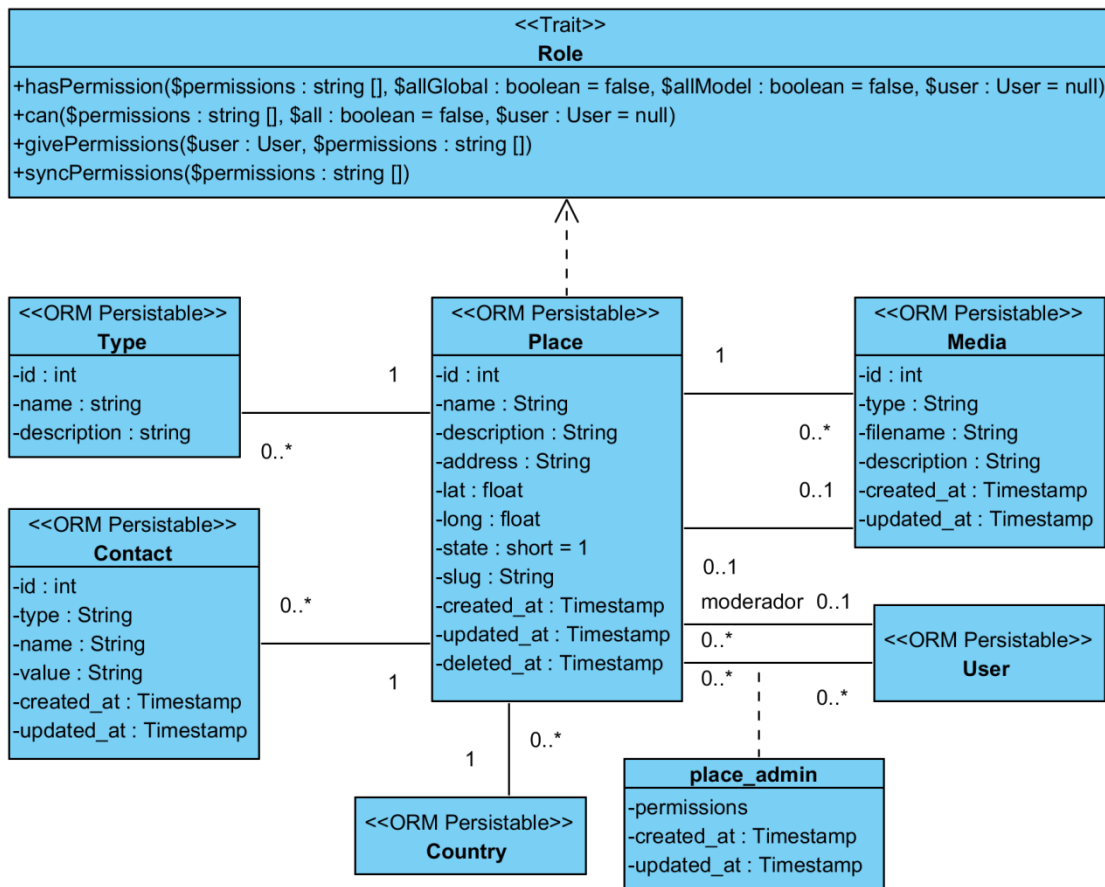


Figura 14 – Diagrama de classes do componente de gestão de espaços

De notar que as relações entre a classe `Place` e as classes `Type`, `Contact` e `Media`, são relações polimórficas. As relações polimórficas permitem que um modelo possa pertencer a mais do que um modelo, numa associação única. Por exemplo, o modelo `Contact` pode pertencer a qualquer modelo `Place` ou `Event` (Figura 16). Essa relação define-se da seguinte maneira:

```

class Contact extends Eloquent {
    public function contactable() {
        return $this->morphTo();
    }
}

class Place extends Eloquent {
    public function contacts() {
        return $this->morphMany('App\Models>Contact', 'contactable');
    }
}

class Event extends Eloquent {
    public function contacts() {
        return $this->morphMany('App\Models>Contact', 'contactable');
    }
}
  
```

Código Fonte 12 – Definição de relações polimórficas com Laravel

Estando definidas as relações, pode-se obter os contactos de um evento ou espaço, ou melhor ainda aceder o espaço ou o evento a que um contacto pertence:

```
// get contacts of place
$place = Place::find(1);

foreach ($place->contacts as $contact)
{
    //
}

// get model of contact
$contact = Contact::find(1);

$contactable = $contact->contactable;
```

Código Fonte 13 – Exemplo de utilização de uma relação polimórfica em Laravel

A relação `contactable` no modelo `Contact` irá retornar uma instância de `Place` ou `Event`, dependendo de qual é o tipo de modelo que possui o contacto.

Para ajudar a entender como isso funciona, segue a estrutura da base de dados para uma relação polimórfica:

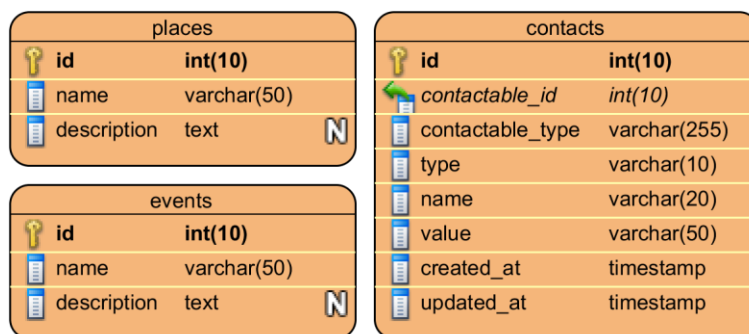


Figura 15 – Estrutura de uma tabela com relação polimórfica

Os campos-chave para salientar na tabela `contacts` são os `contactable_id` e `contactable_type`. O campo `id` conterà a chave de identificação de, neste exemplo, `places` ou `events`, enquanto que o campo `type` conterà o nome da classe que possui o modelo. Esta é a estrutura que permite que o ORM determinar que tipo de modelo possui para retornar ao aceder a relação `contactable`.

Para o módulo de gestão de eventos foram definidas as classes ilustradas no diagrama de classes da Figura 16.

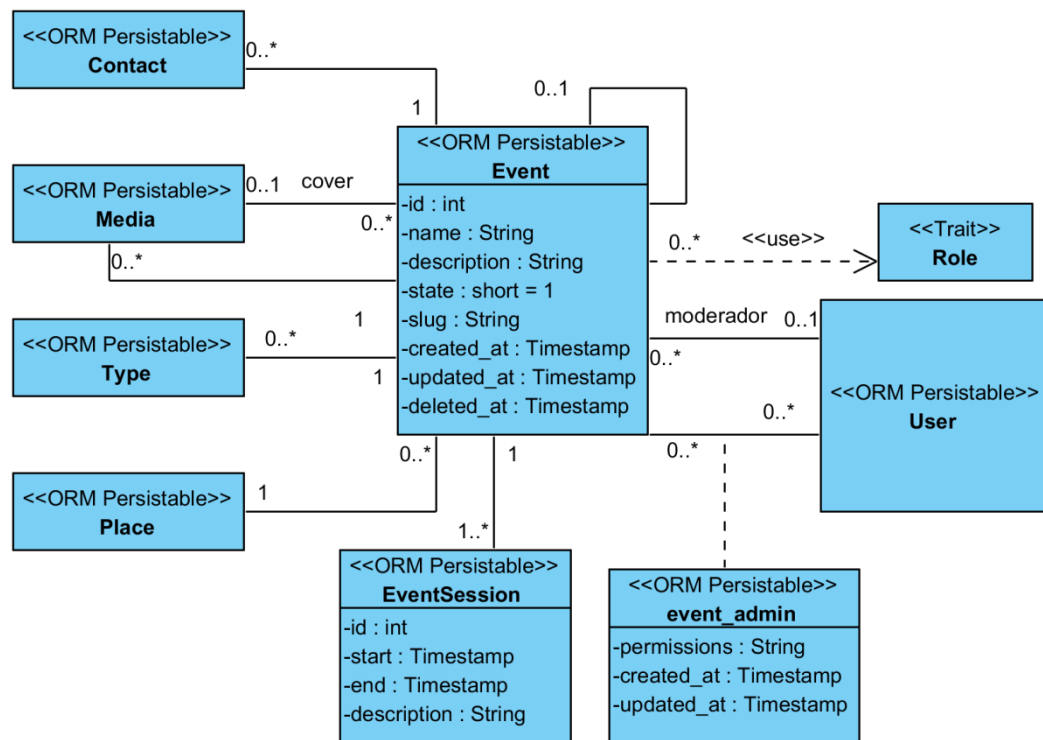


Figura 16 – Diagrama de classes do componente de gestão de eventos

Nos diagramas da Figura 14 e da Figura 16 existe duas classes associativas (place\_admin e event\_admin) que são responsáveis pelas permissões de moderação de um evento/espço por um utilizador, ou seja, quando um utilizador cria um espaço/evento, pode dar a outros utilizadores, permissões para, por exemplo, editar as informações base, adicionar remover contactos, sessões (no caso dos eventos), ... São utilizadas em conjunto com a *trait* Role

Uma *trait* é um conjunto de métodos, que são estruturalmente semelhantes a uma classe (mas não pode ser instanciada), que permite a reutilização de métodos em várias classes independentes. [31] Como o PHP é uma linguagem de herança única, uma subclasse pode herdar apenas uma superclasse; e é aqui que as *traits* são úteis. [32]

O melhor uso de *traits* é demonstrado quando várias classes partilham a mesma funcionalidade, que é o caso das classes Event e Place, que necessitam de

*“A melhor demonstração das traits é quando múltiplas classes partilham a mesma funcionalidade.”*

testar o acesso a opções de moderação de utilizadores. Em vez de fazer o clássico copiar/colar dos métodos, usa-se *Traits*. Desta forma, torna-se o código reutilizável, seguindo o princípio DRY (Don't Repeat Yourself).












Nome	Descrição
 <b>User</b>	Classe representativa de um utilizador. A cada utilizador podem estar associadas várias permissões de acesso a funcionalidades do sistema.
 <b>UserLocation</b>	Classe representativa de uma localização de um utilizador. Permite fazer o percurso de um utilizador.
 <b>UserSettings</b>	Classe representativa das opções um utilizador.
 <b>Group</b>	Classe representativa de um grupo de utilizadores. Um utilizador pode pertencer a vários grupos. A cada grupo podem estar associadas várias permissões de acesso a funcionalidades do sistema.
 <b>Throttle</b>	Classe representativa de tentativas de acesso e estado da conta de um utilizador.
 <b>Country</b>	Classe representativa de um país.
 <b>Place</b>	Classe representativa de um espaço/local.
 <b>place_admin</b>	Classe de associação entre espaços e utilizadores. Permite dar permissões de moderação de um espaço a um utilizador.
 <b>Event</b>	Classe representativa de um evento. Um evento pode ter vários sub-eventos.
 <b>EventSession</b>	Classe representativa de uma sessão/horário de um evento.
 <b>event_admin</b>	Classe de associação entre eventos e utilizadores. Permite dar permissões de moderação de um evento a um utilizador.
 <b>Media</b>	Classe representativa de uma media (imagem, vídeo).
 <b>Contact</b>	Classe representativa de um contacto.
 <b>Type</b>	Classe representativa de um tipo/tag.
 <b>Validation</b>	Trait de validação para modelos Eloquent.
 <b>Role</b>	Trait de permissões por utilizador para objectos Eloquent.

Tabela 9 – Descrição breve das classes base do sistema

Encontra-se no Anexo 2 a descrição das tabelas mapeadas (modelo de dados).

### 5.2.1.2. Repositórios (Dependency Injection)

A fundação da *framework* Laravel é um poderoso contentor de inversão de controlo (inversion of control ou IoC, em inglês). Para compreender verdadeiramente a *framework*, é necessário ter um conhecimento profundo sobre o funcionamento do seu contentor. Entretanto, deve-se notar que o contentor IoC é simplesmente um mecanismo conveniente para se alcançar o padrão de desenvolvimento de *software* injeção de dependência (*dependency injection*, em inglês). O uso do contentor não é obrigatório para se injetar dependências, ele apenas facilita essa tarefa. [33], [34]

Em primeiro lugar, é exemplificado porque é que a injeção de dependência é vantajosa. Considere a seguinte classe e método:

```
class EventController extends BaseController {
    public function getIndex() {
        $events = Event::all();
        return view::make('events.index', compact('events'));
    }
}
```

Mesmo sendo um código conciso, não se pode testá-lo sem aceder a uma base de dados. Em outras palavras, o ORM Eloquent está fortemente acoplado ao controlador. Não se pode, de alguma forma, usar ou testar o controlador sem usar também todo o Eloquent, incluindo a necessidade de uma base de dados. Este código também viola um princípio de desenvolvimento de *software* conhecido como separação de conceitos (*separation of concerns* ou SoC, em inglês). Em outras palavras: o controlador sabe mais do que deveria. Os controladores não precisam saber de onde vêm os dados, mas somente como acede-los. O controlador não precisa saber que os dados estão disponíveis no MySQL, apenas que eles existem em algum lugar. [34]

#### Separação de conceitos

Toda classe deve ter apenas uma responsabilidade e esta responsabilidade deve ser completamente encapsulada pela classe.

Posto isto, é melhor desacoplar completamente a camada web (controlador) da nossa camada de acesso aos dados. Isso permite migrar mais facilmente a implementação de armazenamento de dados e tornará o código mais fácil de testar. [19]

Por estas razões, ao invés de acoplar o controlador ao Eloquent, injetar-se uma classe repositória.

Em primeiro lugar, é necessário definir uma interface e uma implementação correspondente:

```
interface EventRepository {
    public function all();
    // ...
}

class EloquentEventRepository implements EventRepository {
    public function all()
    {
        return Event::all();
    }
    // ...
}
```

Em seguida, injeta-se uma implementação da interface no controlador:

```
class EventController extends BaseController {
    public function __construct(EventRepository $events)
    {
        $this->events = $events;
    }

    public function getIndex() {
        $events = $this->events->all();
        return View::make('events.index', compact('events'));
    }
}
```

Desta forma o controlador é completamente ignorante quanto ao local onde os dados estão armazenados. Neste caso, esta ignorância é benéfica! Os dados podem estar em uma base de dados MySQL, MongoDB ou Redis. O controlador não reconhece a diferença, e isso não é da sua responsabilidade. Fazendo esta pequena mudança, pode-se testar a nossa camada *web* separadamente da camada de dados, além de se poder facilmente alternar a implementação de armazenamento de dados (Figura 17).

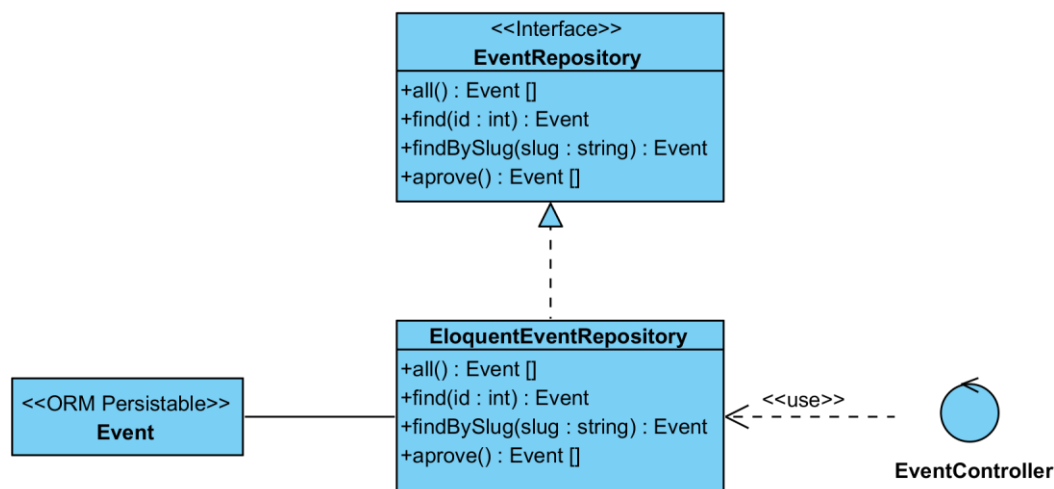


Figura 17 – Repository Pattern

Para solidificar a compreensão, segue um teste rápido. Em primeiro lugar, simula-se (*mock*, em inglês) o repositório vinculando-o ao contentor IoC da aplicação. Em seguida, certifica-se que o controlador invoca o repositório devidamente:

```
public function testIndexActionBindsEventsFromRepository()
{
    // Preparar...
    $repository = Mockery::mock('EventRepository');
    $repository->shouldReceive('all')->once()->andReturn(array('foo'));
    App::instance('EventRepository', $repository);

    // Agir...
    $response = $this->action('GET', 'EventController@getIndex');

    // Conferir...
    $this->assertResponseOk();
    $this->assertViewHas('events', array('foo'));
}
```

Código Fonte 14 – Teste unitário para verificar se o controlador invoca o repositório corretamente

## 5.2.2. Integração com as Redes Sociais

A integração com as redes sociais é um dos pontos forte do modelo de negocio deste portal, e por isso este componente deverá ser forte e robusto, mas ao mesmo tempo facilmente estendível e adaptável. Foi utilizado o pacote **PHPoAuthLib**<sup>1</sup> desenvolvido por David Desberg[35] como base deste pacote. O PHPoAuthLib oferece uma base OAuth (versões 1 e 2) para os principais sistemas existentes: Twitter, Flickr, Tumblr, ... (OAuth 1) e Facebook, LinkedIn, Google, Microsoft, ... (OAuth 2). Para além disso, o pacote está perfeitamente bem construído ao nível da engenharia de *software*, pois utiliza os padrões **Strategy** e **Adapter**, nos seus componentes, o que facilita a sua extensibilidade.

A integração do sistema com as redes sociais tem duas áreas: **Autenticação** e **Partilha de Informação**. Para a autenticação foram escolhidos os dois serviços mais utilizados para o efeito Facebook e Google. Para implementar o registo e a autenticação para cada uma deles, foi estendida as classes respetivas de cada serviço do pacote PHPoAuthLib, e criado uma nova interface (**AuthInterface**), que define um conjunto de métodos relativos à área, e que é implementada pelas classes criadas, aplicando assim o padrão de *software* “**Adapter**”. A mesma coisa foi feita para a partilha de informação, criando-se a interface **EventInterface**, só que no entanto só implementada pela classe Facebook.

<sup>1</sup> <https://github.com/Lusitanian/PHPoAuthLib/tree/master/src/OAuth>

Também foi criado um novo **cliente http** para um maior controlo das comunicações entre o sistema e os sistemas externos. Para tal foi criada uma nova classe (**StreamClient**) que implementa a interface **ClientInterface**, disponibilizada também pela biblioteca utilizada.

Por fim, foi criada uma fábrica para a criação dos objetos dos serviços, aplicando assim o padrão de *software* “**FactoryMethod**”.

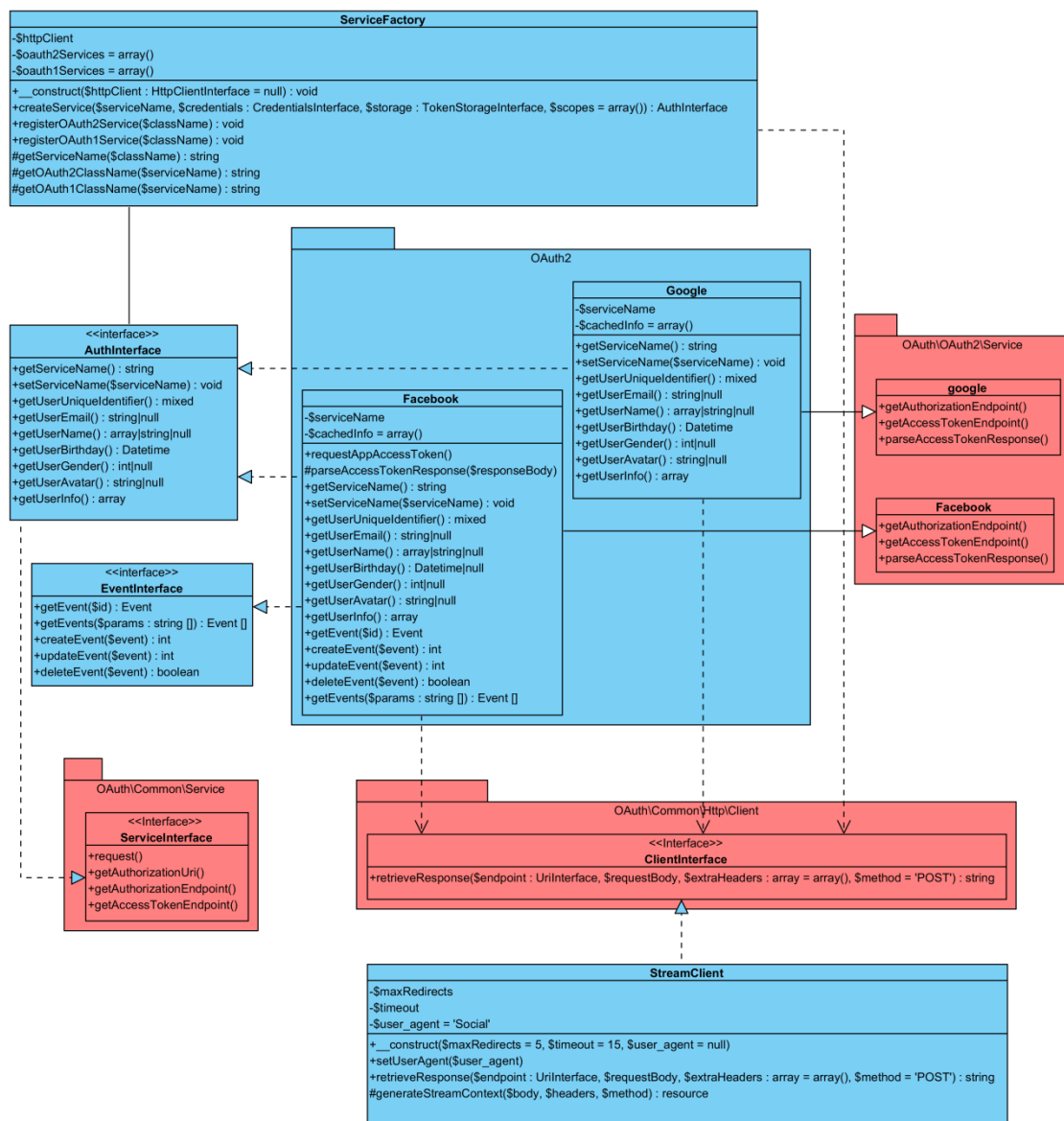


Figura 18 – Diagrama de classes do componente de integração com serviços externos

Após da definição da base do componente social, foi criado uma classe de fachada (padrão Facade) para esconder os detalhe da criação de serviços, de autenticação via serviços, requerimentos de tokens de acesso, .... A Laravel disponibiliza *Facades* para todos os seus componentes, e encoraja a criação de novas *facades* para cada novo componente criado. [19],



de duas classes que implementassem respetivamente as interfaces `ProviderInterface` (*repository pattern*) e `LinkInterface`.

Resumidamente temos uma fábrica de serviços, que centraliza a criação dos serviços, permitindo assim isolar a criação da classe da sua utilização e permitir um controlo fácil do processo de criação; e uma fachada que encapsula as maiores funcionalidades, do componente, permitindo assim a sua rápida e fácil utilização. Assim é demonstrado que com a utilização de alguns padrões de desenvolvimento de *software* é possível construir um sistema robusto, mas facilmente extensível e adaptável.

### 5.2.3. Relatórios/Estatísticas

A geração de relatórios é outro ponto forte do modelo de negócio proposto por este sistema, que pode favorecer a sua rápida aceitação pelos promotores de eventos. Este permite que os promotores de eventos visualizar algumas estatísticas sobre os seus eventos, como por exemplo o número de pessoas que foram ao evento em relação às que disseram que iam ou o número de pessoas que vão ao evento por género ou idade. A visualização deste tipo de estatísticas pode auxiliar o promotor a planear melhor os seus eventos ou saber mais pormenorizadamente de como correu. A visualização das estatísticas é feita através de gráficos, para uma melhor e mais rápida análise dos dados. Foi utilizada a biblioteca `phpChart`<sup>1</sup> para auxiliar o desenho dos mesmos.

Este componente independente é composto por um adaptador para desenhar os gráficos (`phpChartAdapter`). A classe `phpChartAdapter` implementa a interface `GraphStatisticsInterface`, aplicando assim o padrão GoF **Adapter**. Assim é possível adicionar ou até substituir facilmente a criação dos gráficos, sem se ter que alterar a arquitetura do sistema. Segundo o Princípio da Responsabilidade Única (Single Responsibility Principle, em inglês), que afirma que uma classe deve ter um, e somente um, motivo para mudar. Em outras palavras, o escopo e responsabilidade de uma classe devem ser concentradas. Como já foi referido anteriormente, a ignorância é boa quando se trata das responsabilidades de uma classe. A classe deve fazer o seu trabalho, e não deve ser afetada por alterações em qualquer das suas dependências [19].

---

<sup>1</sup> <http://phpchart.net/>

Aplicando esse princípio ao componente as instâncias da interface `GraphStatisticsInterface`, só devem ter a responsabilidade de desenhar os gráficos. A responsabilidade da obtenção e cálculo dos dados deverá ser atribuída a outra classe (mesmo princípio aplicado aos controladores [ver 5.2.1.2. Repositórios (Dependency Injection)]). Daí foi necessário criar uma classe abstrata (`EventStatistics`) em que as suas instâncias, é que têm a responsabilidade da obtenção e cálculo dos dados. A classe `EventStatistics` é uma classe abstrata, pois mais uma vez, permite mais uma vez desacoplar completamente o acesso aos dados. Por último, aplicando o padrão *Factory Method*, foi criada a classe `Statistics`, que permite centralizar num único ponto a criação de objetos do tipo `GraphStatisticsInterface` e `EventStatistics`.

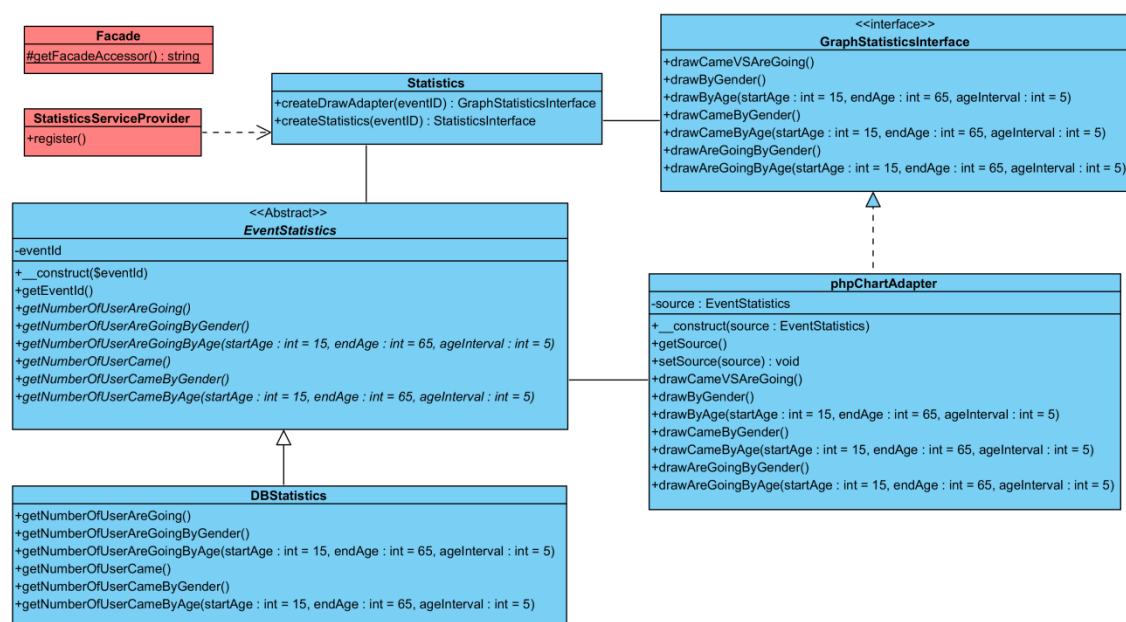


Figura 20 – Estrutura do componente *StatisticsReports*

## 5.3. Segurança

A segurança é um dos pontos fundamentais na construção de uma aplicação. Por isso foi tomado um especial cuidado nesta área.

### 5.3.1. Codificação das passwords

Sendo as *passwords* de acesso um dado extremamente e valioso e pessoal, é necessário codificá-las para que se por alguma falha de segurança, alguma pessoa estranha à equipa tenha acesso à base de dados, esta não as possa descobrir.

Para isso foi utilizado o sistema de *hashing* Bcrypt que a própria framework oferece.

### 5.3.2. Cross-site request forgery

A Laravel fornece um método simples de proteção contra ataques CSRFs. Fornece um filtro (csrf) que pode ser aplicado a qualquer rota, que compara o *token* enviado pelo formulário pelo que foi gerado anteriormente para uma determinada página.

```
// CSRF Filter
Route::filter('csrf', function() {
    if (Session::token() != Input::get('_token')) {
        throw new Illuminate\Session\TokenMismatchException;
    }
});

// Enable CSRF Protection
$this->beforeFilter('csrf', array('on' => array('post', 'put')));

// CSRF Protection in form

```

Código Fonte 15 – Protecção CSRF

Nesta aplicação foi aplicada essa proteção a todos os pedidos do tipo POST e PUT.

### 5.3.3. Cross Site Scripting

Para este tipo de ataques primeiramente são validados todos os dados recebidos e que são guardados e que tenham algum tipo de persistência. Para adicionar outra camada de proteção para este tipo de ataques, todos os dados provenientes da base de dados são escapados, impedindo assim que caso haja código HTML este não seja interpretado como tal.

Os modelos são validados automaticamente quando são guardados, graças à implementação do padrão **Publish/Subscriber** (Código Fonte 16).

```
BaseBode1.php

/**
 * Register event bindings.
 *
 * @return void
 */
public static function boot()
{
    parent::boot();

    static::saving(function($model)
    {
        if (!$model->force) return $model->validate();
    });
}
```

Código Fonte 16 – Validação automática dos modelos quando estes são guardados

## 5.4. Usabilidade e Acessibilidade

Para a construção da interface do *frontend* foi utilizado a versão 3 da *framework* css Twitter Bootstrap<sup>1</sup>, que segue os *web standards*, e com um mínimo de trabalho, pode ser utilizado para criar sites que possam ser acedidos por tecnologias de acessibilidade.

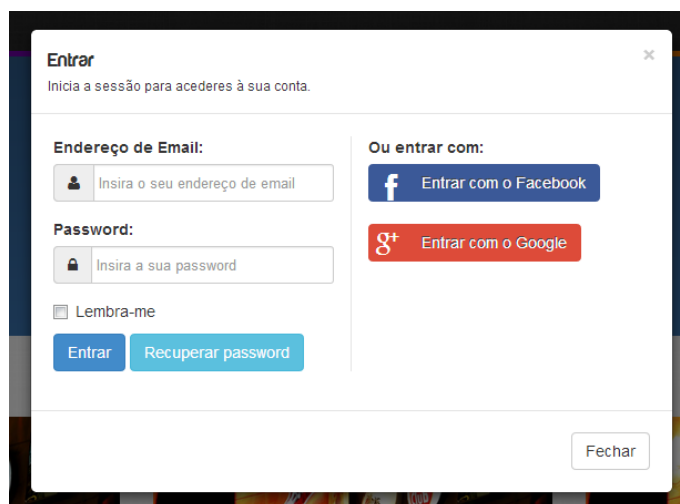


Figura 21 – Tela de autenticação

Ainda na área da acessibilidade, foram utilizadas as práticas comuns do HTML, como o uso dos atributos `title` e `alt` nos elementos correspondentes; o uso dos novos elementos introduzidos pelo HTML5, para definir a estrutura das páginas; o uso de atributos ARIA<sup>2</sup> para tornar a aplicação ainda mais acessível a pessoas com mobilidade reduzida (Código Fonte 17). [38], [39], [40, p. 5], [41], [42]

```
<ul class="pull-right nav navbar-nav">
  <li class="dropdown">
    <a id="userMenu" ... title="Perfil" role="button">Miguel Borges</a>
    <ul class="dropdown-menu pull-right" role="menu" aria-labelledby="userMenu">
      <li role="presentation"><a ... tabindex="-1" role="menuitem"
title="Perfil">Perfil</a></li>
      <li role="presentation"><a ... tabindex="-1" role="menuitem"
title="Definições da Conta">Definições da Conta</a></li>
      <li role="presentation" class="divider"></li>
      <li role="presentation"><a ... tabindex="-1" role="menuitem" title="Terminar
Sessão">Terminar Sessão</a></li>
    </ul>
  </li>
</ul>
```

Código Fonte 17 – Exemplo do uso de atributos de acessibilidade num dropdron menu

Outra técnica utilizada foi utilizar uma ligação de avanço da navegação, pois pode ser frustrante e cansativo para pessoas com mobilidade reduzida ter que ter passar

<sup>1</sup> <http://getbootstrap.com>

<sup>2</sup> Accessible Rich Internet Application

repetidamente pelas ligações de navegação do *site* para chegar ao conteúdo principal da página. As pessoas que utilizam leitores de tela enfrentam um problema semelhante quando o guia de interação da página não é bem definido. [43]

O portal por ter sido construído com o Twitter Bootstrap 3, este é desde logo compatível com os vários tipos de dispositivos, desde os dispositivos móveis (Figura 23) até aos dispositivos com grandes resoluções (FullHD e *retina ready*) (Figura 22), adaptando-se a cada tipo para uma melhor experiência do utilizador.

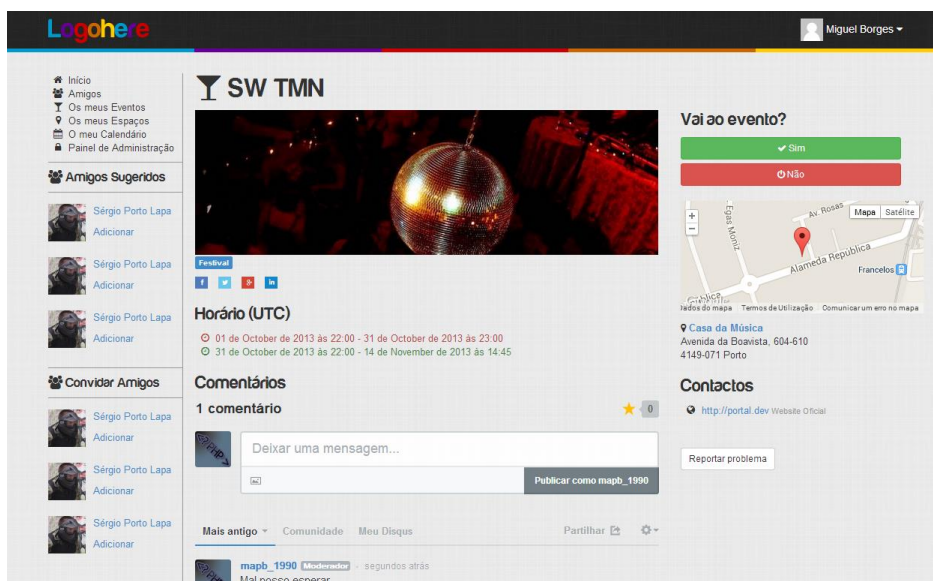


Figura 22 – Página de visualização de um evento numa tela FullHD

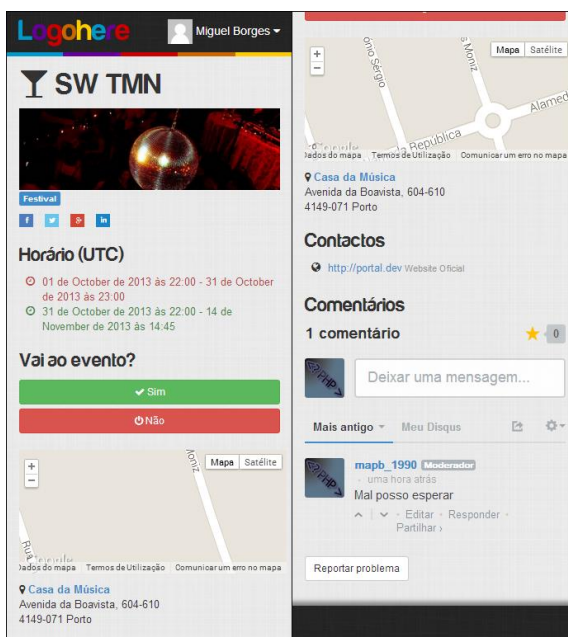


Figura 23 - Página de visualização de um evento num dispositivo móvel

Foi criada uma interface simples e intuitiva para uma maior usabilidade pelos utilizadores.

Foram também utilizados os novos tipos de entrada de dados do HTML5, que embora muitos ainda não sejam suportados pelos *browsers*, este são suportados pelos dispositivos móveis, sendo uma mais-valia, pois assim, estes adaptam o teclado de acordo com o tipo do campo. Por exemplo, num campo do tipo `number` o dispositivo irá mostrar um teclado numérico, enquanto se o campo for do tipo `email`, este irá disponibilizar um teclado alfanumérico com o carácter arroba (@), e algumas extensões de domínios em destaque.

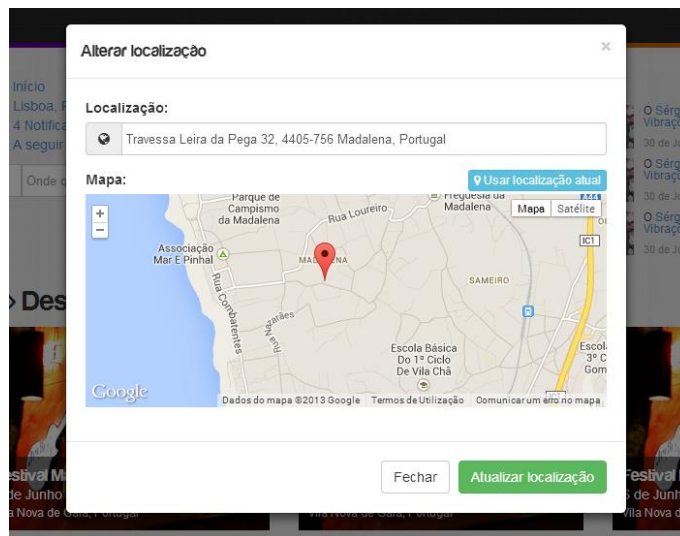


Figura 24 – Tela de alteração da localização de um utilizador

Na área da usabilidade ainda houve uma preocupação com que as funcionalidades fossem acedidas com apenas algumas interações/clicques.

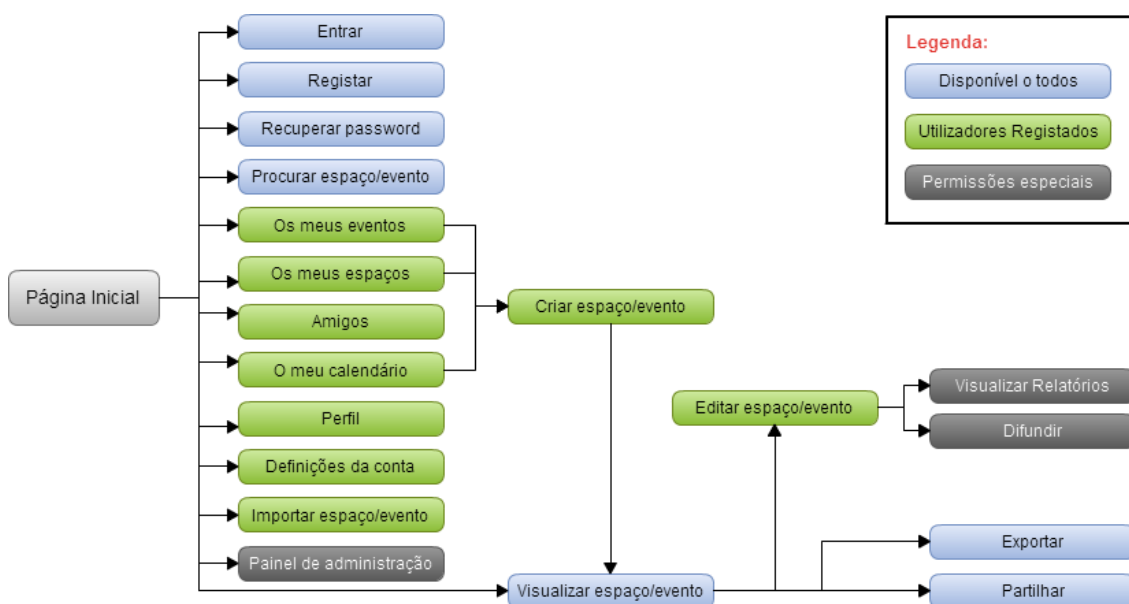


Figura 25 – Mapa do portal



# 6. CONCLUSÕES

*Este capítulo reitera de forma sumária e em jeito de desfecho as principais conclusões e a apreciação geral do trabalho desenvolvido ao longo do estágio.*

### 6.1. Resumo

Embora a quando a submissão da proposta desta dissertação não houvesse soluções para os problemas que se propõe resolver (informações sobre eventos dispersa, inexistência de ferramentas de apoio aos promotores), durante o tempo de desenvolvimento deste projeto surgiam algumas soluções para alguns destes problemas.

O objetivo desta dissertação foi criar um protótipo avançado de um portal que oferece-se soluções para os problemas existentes na área da promoção e gestão de eventos.

Para além da criação deste protótipo, outro objetivo foi aplicar as boas práticas de desenvolvimento de *software* desde a aplicação de um processo de desenvolvimento (neste caso o OpenUP), um processo de testes (desenvolvimento orientado por testes [TDD]), a aplicabilidade dos princípios SOLID e de padrões GRASP e GoF.

### 6.2. Objetivos realizados

O objetivo principal do projeto era a criação de um sistema que permitir-se facilmente descobrir, promover e gerir eventos e espaços, não só para Portugal, mas também para o resto do mundo. Esse objetivo foi cumprido. Primeiramente a descoberta de eventos e espaços e eventos pode ser obtida facilmente graças à geolocalização, e por ser facilmente filtrada. A promoção dos eventos por parte dos promotores é facilitada graças à integração com os principais sistemas externos (Facebook). Os eventos adicionados ou atualizados no sistema podem, caso o promotor deseje, serem inseridos ou atualizados nos outros sistemas. Assim os promotores podem gerir os seus eventos, nos mais diversos locais na rede global, a partir de um único local, poupando-lhes assim tempo e trabalho. Também foi desenvolvido um módulo de relatórios que permite os promotores visualizarem de facilmente algumas estatísticas dos seus eventos.

Para além das funcionalidades terem sido implementadas, também era importante que o sistema fosse bem desenhado, para que fosse facilmente entendido e escalável. Esse objetivo também foi conseguido.

### 6.3. Limitações e trabalho futuro

Nas primeiras versões do plano de projeto (página 63) existiam funcionalidades implementadas, a possibilidade de avaliação dos espaços e dos eventos, de os utilizadores fazerem a sua própria *watchlist*, e algumas funcionalidades de apoio aos promotores, como a possibilidade destes enviarem *email* e sms aos utilizadores (podendo-os filtrar, por exemplo, por idade, sexo e localização) ou a possibilidade destes venderem os bilhetes a partir do sistema. Essas funcionalidades foram descartadas do plano de desenvolvimento (no âmbito da tese) devido à desistência tardia de um membro da equipa. Essa desistência tornou-se um problema, pois para além de levar à não implementação das funcionalidades já referidas, levou a que me fossem atribuídas algumas funcionalidades cruciais para o correto funcionamento sistema, já numa fase tardia (Setembro de 2013).

Também estava previsto o desenvolvimento de um protótipo de aplicação móvel, de complemento ao sistema, que no entanto, por acumulação de tarefas e falta de tempo não foi possível desenvolver.

No entanto, estas funcionalidades como outras previstas no plano de negócios irão ser implementadas futuramente, bem como o melhoramento do módulo de estatísticas, ou como o suporte de outros sistemas externos.

### 6.4. Apreciação final

Ao longo do desenvolvimento do projeto, foi notório o potencial que este projeto tem, para se tornar numa ideia de negócio (sendo que foi elaborado um plano de negócio). A prova disso foi o aparecimento repentino de alguns portais semelhantes ao longo do desenvolvimento.

Esta ideia de negócio foi submetida ao concurso de inovação e empreendedorismo “InovPortugal<sup>1</sup>” – 1ª edição, tendo atingindo a penúltima fase do concurso.

Na medida do possível este projeto será continuado e transformado numa ideia de negócio.

---

<sup>1</sup> <http://www.acreditaportugal.pt/>



## **7. REFERÊNCIAS**

- [1] Turismo de Portugal, «Plano Estratégico Nacional para o Turismo para o Desenvolvimento do Turismo em Portugal». 2007 [Online]. Disponível em: <http://www.turismodeportugal.pt/Portugu%C3%AAs/turismodeportugal/publicacoes/Documents/PENT%202007.pdf>. [Acedido: 08-Nov-2012]
- [2] *Estatísticas do Turismo - 2012*. Lisboa: Instituto Nacional de Estatística, I.P. [Online]. Disponível em: [http://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine\\_publicacoes&PUBLICACOESpub\\_boui=143016657&PUBLICACOESmodo=2](http://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_publicacoes&PUBLICACOESpub_boui=143016657&PUBLICACOESmodo=2). [Acedido: 08-Nov-2012]
- [3] Miguel Borges, Sérgio Lapa, e Jacinto Barbosa, «Portal de Eventos - Plano de Negócio». Jan-2013.
- [4] «Like Button», *Facebook Developers*. [Online]. Disponível em: <https://developers.facebook.com/docs/reference/plugins/like/>. [Acedido: 17-Jul-2013]
- [5] «Comments Box», *Facebook Developers*. [Online]. Disponível em: <https://developers.facebook.com/docs/reference/plugins/comments/>. [Acedido: 17-Jul-2013]
- [6] Elbert Alias, *Wappalyzer*. [Online]. Disponível em: <http://wappalyzer.com/>. [Acedido: 01-Out-2012]
- [7] Sapo Notícias, «Moçambique é dos países africanos com internet mais rápida», *Sapo Notícias*, 03-Jul-2012 [Online]. Disponível em: <http://noticias.sapo.mz/tecnologia/noticias/artigo/1254414.html>. [Acedido: 15-Out-2012]
- [8] Net Index, «Household Download Index». [Online]. Disponível em: <http://www.netindex.com/download/allcountries/>. [Acedido: 15-Out-2012]
- [9] «WCAG Overview». [Online]. Disponível em: <http://www.w3.org/WAI/intro/wcag.php>. [Acedido: 18-Set-2013]
- [10] Ben Caldwell, Michael Cooper, Loretta Guarino Reid, e Gregg Vanderheiden, «Web Content Accessibility Guidelines (WCAG) 2.0», *REC-WCAG20-20081211*, Dez. 2008 [Online]. Disponível em: <http://www.w3.org/TR/WCAG20/>. [Acedido: 18-Set-2013]
- [11] PHP Team, «O PHP e as outras linguagens», *O PHP e as outras linguagens*, 23-Jun-2013. [Online]. Disponível em: [http://www.php.net/manual/pt\\_BR/faq.languages.php](http://www.php.net/manual/pt_BR/faq.languages.php)
- [12] Jeffrey Way, «Why Laravel is Taking the PHP Community by Storm», *Tuts+ Premium*, 29-Nov-2012. [Online]. Disponível em: <https://tutsplus.com/tutorial/why-laravel-is-taking-the-php-community-by-storm/>. [Acedido: 16-Set-2013]
- [13] Gabriel Manricks, «Why 2013 is the Year of PHP», 15-Jan-2013. [Online]. Disponível em: <http://net.tutsplus.com/articles/editorials/why-2013-is-the-year-of-php/>. [Acedido: 16-Set-2013]
- [14] Thiago Belem, «Gerenciando dependências com o Composer | Thiago Belem / Blog», 21-Out-2012. [Online]. Disponível em: <http://blog.thiagobelem.net/gerenciando-dependencias-com-o-composer/>. [Acedido: 21-Out-2013]
- [15] D. Rees, *Laravel: Code Bright*. Leanpub, 2012 [Online]. Disponível em: <https://leanpub.com/codebright>. [Acedido: 18-Out-2013]
- [16] Laravel Team, «Documentation - Laravel PHP Framework». [Online]. Disponível em: <http://laravel.com/docs/quick>. [Acedido: 17-Set-2013]

- 
- [17] Symfony Team, «Download - Symfony». [Online]. Disponível em: <http://symfony.com/download>. [Acedido: 17-Set-2013]
- [18] «Packagist Statistics». [Online]. Disponível em: <https://packagist.org/statistics>. [Acedido: 17-Set-2013]
- [19] T. Otwell, *Laravel: From Apprentice To Artisan*. Leanpub, 2013 [Online]. Disponível em: <https://leanpub.com/laravel>. [Acedido: 18-Out-2013]
- [20] Kenny Meyers, «Laravel 4 Primer». [Online]. Disponível em: <http://www.thenerdary.net/post/52067531360/laravel-4-primer>. [Acedido: 18-Jul-2013]
- [21] K. Beck, *Test-driven development: by example*. Boston: Addison-Wesley, 2003.
- [22] D. Astels, *Test-driven development a practical guide*. Upper Saddle River, N.J: Prentice Hall PTR, 2003.
- [23] S. Kumar e S. Bansal, «Comparative Study of Test Driven Development with Traditional Techniques», vol. 3, Mar. 2013 [Online]. Disponível em: <http://www.doaj.org/doaj?func=fulltext&ald=1378465>. [Acedido: 22-Out-2013]
- [24] «Test Driven Development». [Online]. Disponível em: <http://c2.com/cgi/wiki/TestDrivenDevelopment>. [Acedido: 22-Out-2013]
- [25] Jamis Buck, «the { buckblogs :here }: Skinny Controller, Fat Model». [Online]. Disponível em: <http://weblog.jamisbuck.org/2006/10/18/skinny-controller-fat-model>. [Acedido: 23-Out-2013]
- [26] J. Way, *Laravel Testing Decoded*. Leanpub, 2013 [Online]. Disponível em: <https://leanpub.com/laravel-testing-decoded>. [Acedido: 18-Out-2013]
- [27] Zizaco Zizuini, «Testing Like a Boss in Laravel: Models | Nettuts+», 18-Fev-2013. [Online]. Disponível em: <http://net.tutsplus.com/tutorials/php/testing-like-a-boss-in-laravel-models/>. [Acedido: 22-Set-2013]
- [28] «Getting started with testing Laravel 4 Models», *Culttt*. [Online]. Disponível em: <http://culttt.com/2013/05/20/getting-started-with-testing-laravel-4-models/>. [Acedido: 19-Out-2013]
- [29] Nuno Silva, «Design OO e Padrões», MoodleISEP, 2012 [Online]. Disponível em: <http://moodle.isep.ipp.pt>
- [30] Robert C. Martin, «Getting a SOLID start» [Online]. Disponível em: <https://sites.google.com/site/unclebobconsultingllc/getting-a-solid-start>. [Acedido: 09-Nov-2013]
- [31] Dejan Marjanovic, «PHP 5.4 is Here! What You Must Know | Nettuts+», 05-Mar-2012. [Online]. Disponível em: <http://net.tutsplus.com/tutorials/php/php-5-4-is-here-what-you-must-know/>. [Acedido: 21-Out-2013]
- [32] PHP Team, «PHP: Traits - Manual». [Online]. Disponível em: <http://www.php.net/manual/en/language.oop5.traits.php>. [Acedido: 21-Out-2013]
- [33] Philip Brown, «Creating flexible Controllers in Laravel 4 using Repositories | Culttt», 08-Jul-2013. [Online]. Disponível em: <http://culttt.com/2013/07/08/creating-flexible-controllers-in-laravel-4-using-repositories/>. [Acedido: 21-Out-2013]
- [34] C. Fido, *Implementing Laravel*. Leanpub, 2013 [Online]. Disponível em: <https://leanpub.com/implementinglaravel>. [Acedido: 18-Out-2013]
-

- [35] D. D. L. Clevel e 2012 23 Followers 38 Starred 5 Following, «Lusitanian (David Desberg)». [Online]. Disponível em: <https://github.com/Lusitanian>. [Acedido: 04-Nov-2013]
- [36] Laravel Team, «Facades - Laravel PHP Framework». [Online]. Disponível em: <http://laravel.com/docs/facades>. [Acedido: 04-Nov-2013]
- [37] Laravel Team, «Package Development - Laravel PHP Framework». [Online]. Disponível em: <http://laravel.com/docs/packages>. [Acedido: 04-Nov-2013]
- [38] James Craig e Michael Cooper, «Accessible Rich Internet Applications (WAI-ARIA) 1.0», [Online]. Disponível em: <http://www.w3.org/TR/wai-aria/>. [Acedido: 09-Jul-2013]
- [39] Steve Faulkner, Hans Hillen, Gez Lemon, e David MacDonald, «Using WAI-ARIA in HTML» [Online]. Disponível em: <http://rawgithub.com/w3c/aria-in-html/master/index.html>. [Acedido: 09-Nov-2013]
- [40] «HTML5 accessibility». [Online]. Disponível em: <http://www.html5accessibility.com/>. [Acedido: 09-Nov-2013]
- [41] «Accessibility | MDN». [Online]. Disponível em: <https://developer.mozilla.org/en-US/docs/Accessibility>. [Acedido: 09-Jun-2013]
- [42] «The Accessibility Project». [Online]. Disponível em: <http://a11yproject.com/>. [Acedido: 09-Nov-2013]
- [43] Cameron Cundiff, «How-to: Use Skip Navigation links - The Accessibility Project», 11-Mai-2013. [Online]. Disponível em: <http://a11yproject.com/posts/skip-nav-links/>. [Acedido: 09-Jun-2013]

## **8. ANEXOS**



# **Anexo 1 - Plano do Projeto**

*Especificação do plano de projeto e dos planos de iteração*

## 1. PLANO DE PROJETO

### *Histórico de Revisão*

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>
<b>20-02-2013</b>	1.0	Versão Inicial	Miguel Borges
<b>10-09-2013</b>	2.0	Reconfiguração dos requisitos devido à desistência de um membro da equipe	Miguel Borges
<b>25-09-2013</b>	2.1	Ajustamento das datas estimadas de conclusão das iterações abertas	Miguel Borges

### 1.1. INTRODUÇÃO

Esse documento visa a especificação das principais características do projeto "Portal de Eventos", assim como a definição de todas as actividades e principais iterações que serão realizadas. Para desenvolver o portal, a equipa seguirá o modelo de processo de desenvolvimento de *software* "OpenUp", uma versão simplificada de processo unificado. A sua estratégia é iterativa e incremental se traduz num ciclo de vida estruturado. No OpenUp, o esforço de cada integrante da equipa é medido em micro-incrementos, pequenas partes constituintes do projeto que serão as unidades de medida do progresso do projeto. De acordo com este método o desenvolvimento de *software* é composto por quatro fases: iniciação, elaboração, construção e transição. Além disso, o OpenUp também divide o projeto em iterações, que são intervalos de tempo planeados, tipicamente medidos em semanas. Cada fase engloba uma ou mais iterações. O objetivo de dividir a evolução do projeto em iterações é estabelecer metas para serem cumpridas e poder compartilhar os resultados parciais com todos os *stakeholders*.

### 1.2. ORGANIZAÇÃO DO PROJETO

O projeto está dividido em basicamente duas secções: *backend* e *frontend*. Todos os membros da equipa trabalharão em ambas as secções, exercendo as suas funções. O **gestor do projeto** (Miguel Borges) será responsável por coordenar o trabalho da equipa e garantir que o projeto ocorra de acordo com o modelo de processo, respeitando as metas de tempo e definição da documentação necessária, mantendo todos os membros da equipa informados sobre o andamento do projeto e cumprimento dos requisitos e prazos. Também determinará formalmente a estrutura das tarefas a serem realizadas considerando o cronograma, além de

propor mudanças de plano, caso seja necessário, baseando-se nos impactos causados pelos riscos.

Os **arquitectos** serão responsáveis por descrever toda a estrutura do projeto, esquematizar possíveis módulos, componentes e suas interconexões a fim de facilitar o entendimento por todos e não deixar ambiguidades. Isto será importante para determinar em que bases técnicas toda a produção será baseada, de maneira que a não haver razões para modificações futuras, pois tudo o que é determinado pelos arquitectos deve ou deverá ser o mais correto possível.

Os **analistas** serão responsáveis por captar o que as necessidades do mercado e expressar isso através dos casos de uso, que servirão de guia para os engenheiros de testes. Os **engenheiros de software** serão responsáveis por implementar os casos de uso especificados pelos analistas e testá-los e realizar testes unitários aos componentes quando necessário.

Os **designers criativos** serão responsáveis por criar o protótipo e o design da interface da Web, capturando os requisitos dessa interface, inclusive os requisitos de usabilidade e de fazer a revisões, fornecendo o feedback adequado, na implementação final da interface da Web, pelos engenheiros de software. Os **engenheiros de testes**, serão responsáveis por executar os testes, o que inclui a execução e configuração dos testes, a avaliação da execução dos testes e a recuperação dos erros, por avaliar os resultados de teste e por registrar os defeitos identificados.

Estes papéis não serão restritos durante todo o processo de *software*, até porque sendo uma equipa pequena, os papéis descritos anteriormente serão compartilhados pelos dois membros da equipa. No entanto, cada um apresenta as responsabilidades e estas devem ser cumpridas escrupulosamente. O engenheiro de teste de uma implementação não poderá ser a mesma pessoa que desempenhou o papel de engenheiro de *software* por uma questão de imparcialidade.

Os membros da equipa utilizarão o Redmine (<http://www.redmine.org/>) como ferramenta para gestão do projeto e gestão de tarefas. O projeto será versionado através do git e será utilizado Bitbucket ([https://bitbucket.org/mapb\\_1990/tmdei-portal](https://bitbucket.org/mapb_1990/tmdei-portal)) como repositório. As ferramentas utilizadas pelos desenvolvedores também serão padronizadas para evitar problemas de incompatibilidade de versões, e manter todos os integrantes do projeto sob as mesmas soluções.

### 1.3. PRÁTICAS E MEDIDAS UTILIZADAS NO PROJETO

Este projeto será orientado por práticas definidas no OpenUP com algumas adaptações. Os principais artefactos que serão desenvolvidos serão: Plano de Projeto, Lista de Tarefas, Lista de Riscos, Planos de Iteração e Avaliação do Estado Actual (*Status Assessment*).

### 1.4. METAS E OBJETIVOS DO PROJETO

Como o projeto será baseado em interações constantes, há a possibilidade de alteração das datas dependendo da velocidade Desenvolver. Como base para todo o processo, segue o cronograma abaixo.

Fase	Iteração	Principais Objetivos	Data de Estimada de Conclusão	Estimativa (dias)
Iniciação	1I1	<ul style="list-style-type: none"> <li>• Analisar o estado da arte</li> <li>• Desenvolver o plano de negócio</li> <li>• Analisar os requisitos</li> <li>• Desenvolver plano de testes</li> </ul>	28-02-2013	90
Elaboração	2E1	<ul style="list-style-type: none"> <li>• Desenvolver modelo de domínio</li> <li>• Desenvolver diagrama de classes</li> <li>• Desenvolver diagrama de componentes</li> <li>• Desenvolver diagrama de instalação</li> <li>• Construir wireframes</li> </ul>	20-04-2013	30
Elaboração	3E2	<ul style="list-style-type: none"> <li>• Construir plano de projeto</li> <li>• Instalar ferramentas de apoio: <ul style="list-style-type: none"> <li>○ Dashboard</li> <li>○ Repositório</li> </ul> </li> <li>• Definir práticas de desenvolvimento a utilizar</li> <li>• Instalar e configurar framework</li> </ul>	30-04-2013	15
Construção	4C1	<ul style="list-style-type: none"> <li>• Implementar o layout do backend</li> <li>• Desenvolver a base do backend</li> <li>• Implementar os UC01, 02 e 03</li> </ul>	15-06-2013	60
Construção	5C2	<ul style="list-style-type: none"> <li>• Implementar do layout do frontend</li> <li>• Desenvolver da base do frontend</li> <li>• Implementar os UC11, 13, 14, 27 e 28</li> </ul>	14-09-2013	20
Construção	6C3	<ul style="list-style-type: none"> <li>• Refinar o modelo de dados</li> <li>• Implementar os UC04.[01-03], 05, 08, 12 e 20</li> </ul>	07-10-2013	15
Construção	7C4	<ul style="list-style-type: none"> <li>• Implementar os UC18 e UC4.5</li> </ul>	06-11-2013	5
Construção	8C5	<ul style="list-style-type: none"> <li>• Implementar o UC21</li> </ul>	Adiada	4
Construção	9C6	<ul style="list-style-type: none"> <li>• Refinar os casos de uso e modelo de dados</li> <li>• Implementar o UC15</li> </ul>	Adiada	10
Construção	10C7	<ul style="list-style-type: none"> <li>• Refinar o modelo de dados</li> <li>• Implementar o registo e autenticação através das redes sociais, a partilha de informação nas redes social (share/UC17)</li> <li>• Implementar o UC04.04</li> </ul>	26-10-2013	21

<b>Construção</b>	11C8	<ul style="list-style-type: none"> <li>Refinar o modelo de dados</li> <li>Implementar o UC19</li> </ul>	Adiada	3
<b>Construção</b>	12C9	<ul style="list-style-type: none"> <li>Implementar serviço de newsletters</li> </ul>	Adiada	4
<b>Construção</b>	13C10	<ul style="list-style-type: none"> <li>Refinar o modelo de dados</li> <li>Implementar o UC06</li> </ul>	Adiada	10
<b>Construção</b>	14C11	<ul style="list-style-type: none"> <li>Refinar os casos de uso e modelo de dados</li> <li>Implementar sistema de venda de bilhetes online</li> </ul>	Adiada	10
<b>Construção</b>	15C12	<ul style="list-style-type: none"> <li>Refinar o modelo de dados</li> <li>Implementar o UC07</li> </ul>	31-10-2012	15

<b>Código da Iteração</b>	<b>Nome</b>
<b>1I1</b>	Análise de Negócio
<b>2E1</b>	Análise Arquitetural do Sistema
<b>3E2</b>	Apoio ao Desenvolver
<b>4C1</b>	Painel de Administração
<b>5C2</b>	Base do Frontend
<b>6C3</b>	Espaços e Eventos
<b>7C4</b>	Importação e Exportação
<b>8C5</b>	Avaliação
<b>9C6</b>	Amizade
<b>10C7</b>	Social
<b>11C8</b>	Whatchlist
<b>12C9</b>	Newsletter
<b>13C10</b>	Anunciar
<b>14C11</b>	Bilheteira
<b>15C12</b>	Estatísticas

### 3. PLANOS DE ITERAÇÃO

## Plano De Iteração 1 - Iniciação 1

### Análise de Negócio

#### 1. Marcos Chave

<b>Marco</b>	<b>Data</b>
Início da iteração	01-10-2012
1ª reunião com os orientadores	09-10-2012
Estado da Arte	04-12-2012
2ª reunião com os orientadores	17-12-2012
Plano de Negócios	18-01-2013
Análise de Requisitos	23-02-2013
Validação dos artefactos pela equipa	27-02-2013
Fim da Iteração	28-02-2013

#### 2. Objetivos de alto nível

- Estudo do Estado da Arte
- Elaborar documento de Plano de Negócios
- Obter requisitos iniciais, desenvolvendo os respectivos casos de uso
- Análise inicial dos riscos
- Primeiros casos de testes
- Estudos das tecnologias envolvidas (PHP, MySQL, Laravel, etc.)

#### 3. Critério de avaliação

- Documentos do Estado da Arte, Plano de Negócios e Requisitos validados pela equipa.

## Plano De Iteração 2 - Elaboração 1

### Análise Arquitetural do Sistema

#### 1. Marcos Chave

<b>Marco</b>	<b>Data</b>
Início da iteração	04-03-2013
3ª reunião com os orientadores	06-03-2013
Validação dos artefactos pela equipa	17-03-2013
Validação das wireframes pela equipa	20-04-2013
Fim da Iteração	16-04-2013

#### 2. Objetivos de alto nível

- Modelo de Domínio
- Diagrama de Classes
- Diagrama de Componentes
- Diagrama de Instalação
- Construção de Wireframes

#### 3. Critério de avaliação

- Projeto de arquitetura completo e validado
- 80% dos casos de uso detalhados

## Plano De Iteração 3 - Elaboração 2

### Apoio ao Desenvolvimento

#### 1. Marcos Chave

Marco	Data
Início da iteração	20-03-2013
Fim da Iteração	06-05-2013

#### 2. Objetivos de alto nível

- Definição, configuração e distribuição dos ambientes de desenvolvimento (Git + Redmine)
- Domínio das tecnologias envolvidas (PHP, MySQL, Laravel)...
- Definição do [Plano de Projeto](#)
- Definição das [Práticas de Desenvolvimento](#)
- Instalação e configuração da framework base do sistema

#### 3. Critério de avaliação

- [Plano de Projeto](#) aprovado pela equipa
- [Práticas de Desenvolvimento](#) aprovadas pela equipa
- 90% dos casos de uso detalhados

# Plano De Iteração 4 - Construção 1

## Painel de Administração

### 1. Marcos Chave

Marco	Data
Início da iteração	07-05-2013
Fim da Iteração	15-06-2013

### 2. Objetivos de alto nível

O objetivo da primeira construção será construir o núcleo do sistema e a o painel de administração, agregando as suas funcionalidades mais importantes. No final dessa iteração, será possível Gerir Utilizadores, Grupos e Permissões. Implementação os casos de uso relativos ao Painel de Administração, nomeadamente:

- UC01 - Gestão de Utilizadores
- UC02 - Gestão de Grupos
- UC03 - Gestão de Permissões
- UC11 - Autenticar
- UC14 - Recuperar Password
- UC27 - Terminar Sessão

### 3. Atribuições de itens de trabalho

N. da Tarefa	Tarefa	Prioridade	Responsável
27	Autenticação	Normal	Miguel Borges
25	Gestão de Grupos	Normal	Miguel Borges
24	Gestão de Permissões	Normal	Miguel Borges
26	Gestão de Utilizadores	Normal	Miguel Borges
23	Implementação do Tema	Normal	Miguel Borges
37	Localização	Normal	Miguel Borges

### 4. Critério de avaliação

- Casos de uso determinados na lista de actividades para a iteração 4 implementados, passando correctamente pelos testes.
- Sistema a funcionar de forma integrada.

## Plano De Iteração 5 - Construção 2

### Base do Frontend

#### 1. Marcos Chave

Marco	Data
Início da iteração	21-07-2013
Validação da layout	26-07-2013
Implementação da layout	03-08-2013
Autenticação	30-08-2013
Registo de utilizadores	12-09-2013
Gestão da conta pessoal	14-09-2013
Fim da Iteração	14-09-2013

#### 2. Objetivos de alto nível

O objetivo principal desta iteração é construir a base do frontend do sistema, nomeadamente a construção e implementação do layout e a implementação dos casos de uso relativos à gestão básica da conta do utilizador, nomeadamente:

- UC11 - Autenticar
- UC13 - Registrar
- UC14 - Recuperar Password
- UC27 - Terminar Sessão
- UC28 - Gestão da Conta Pessoal

#### 3. Atribuições de itens de trabalho

N. da Tarefa	Tarefa	Prioridade	Responsável	Tempo Estimado (horas)
54	Criação da layout	Urgente	Miguel Borges	48
38	Codificação da Layout	Imediata	Miguel Borges	48
39	Implementação da Layout	Imediata	Miguel Borges	12
40	Autenticação	Alta	Miguel Borges	6
43	Gestão da Conta Pessoal	Alta	Miguel Borges	4
41	Registo de utilizadores	Normal	Miguel Borges	8
42	Recuperar password	Normal	Miguel Borges	2

#### 4. Critério de avaliação

- Casos de uso determinados na lista de actividades para a iteração 5 implementados, passando correctamente pelos testes.
- Sistema a funcionar de forma integrada.

## Plano De Iteração 6 - Construção 3

### Gestão dos Espaços e eventos

#### 1. Marcos Chave

Marco	Data
Início da iteração	17-06-2013
Gestão de espaços (backend)	28-06-2013
Gestão de eventos (backend)	20-07-2013
Gestão de espaços (frontend)	21-09-2013
Gestão de eventos (frontend)	07-10-2013
Fim da Iteração	07-10-2013

#### 2. Objetivos de alto nível

O objetivo principal desta iteração é adicionar as funcionalidades de gestão de espaços e eventos ao backend e do frontend, nomeadamente a procura e visualização, a criação, a edição e remoção de eventos. Em suma, a implementação dos seguintes casos de uso:

- UC08 - Procurar espaços e eventos
- UC12 - Visualizar espaços e eventos
- UC04.UC1 - Adicionar espaço/evento
- UC04.UC2 - Editar espaço/evento
- UC04.UC3 - Remover espaço/evento

#### 3. Critério de avaliação

- Casos de uso determinados na lista de actividades para a iteração 6 implementados, passando correctamente pelos testes.
- Sistema funcionando de forma integrada.

## Plano De Iteração 7 - Construção 4

### Importação e Exportação¶

#### 1. Marcos Chave

<b>Marco</b>	<b>Data</b>
Início da iteração	04-11-2013
Exportação	04-11-2013
Importação	06-11-2013
Fim da Iteração	06-11-2013

#### 2. Objetivos de alto nível

O objetivo principal desta iteração é adicionar ferramentas que permitam a importação e exportação de eventos, nomeadamente a implementação dos seguintes casos de uso:

- UC04.UC5 - Importar evento
- UC18 - Exportar eventos

#### 3. Critério de avaliação

- Casos de uso determinados na lista de actividades para a iteração 7 implementados, passando correctamente pelos testes.
- Importação de pelo menos do Facebook.
- Exportação para pelo menos formato .ics.
- Sistema funcionando de forma integrada.

## Plano De Iteração 10 - Construção 7

### Social

#### 1. Marcos Chave

<b>Marco</b>	<b>Data</b>
Início da iteração	07-10-2013
Registo e autenticação	15-10-2013
Sincronização de eventos	24-10-2013
Partilhar eventos	26-10-2013
Fim da Iteração	26-10-2013

#### 2. Objetivos de alto nível

O objetivo principal desta iteração é adicionar a possibilidade de autenticação e registo via redes sociais, a partilha e sincronização de eventos nas redes sociais, nomeadamente a implementação dos seguintes casos de uso:

- UC04.UC4 - Sincronizar evento c/ outros sistemas
- UC17 - Partilhar eventos

#### 3. Critério de avaliação

- Casos de uso determinados na lista de actividades para a iteração 10 implementados, passando correctamente pelos testes.
- Autenticação e registo por pelo menos 2 redes.
- Sincronização pelo menos com o facebook.

## Plano De Iteração 15 - Construção 12

### Estatísticas

#### 1. Marcos Chave

Marco	Data
-------	------

Início da iteração	27-10-2013
--------------------	------------

Fim da Iteração	04-11-2013
-----------------	------------

#### 2. Objetivos de alto nível

O objetivo principal desta iteração é adicionar ferramentas que permitam visualizar graficamente algumas estatísticas de um evento, nomeadamente a implementação do casos de uso UC07 - Visualizar relatórios de eventos.

#### 3. Critério de avaliação

- Casos de uso determinados na lista de actividades para a iteração 15 implementados, passando correctamente pelos testes.
- Sistema funcionando de forma integrada.

# Riscos

## Identificação dos riscos correspondentes ao projeto.

### Detalhes

ID	Data	Nome	Descrição	Tipo <sup>1</sup>	Impacto <sup>2</sup>	Probabilidade <sup>3</sup>	Responsável	Estratégia
1	13-04-2013	Falta de Gestor	Gestor indisponível à mais de 1 semana	P	A	B	Equipa	Renomear outro gestor
2	13-04-2013	Falta de domínio da tecnologia	Falta de proficiência da equipe com a tecnologia escolhida	T	A	M	Equipa	Organização de reuniões para preparação da equipe
3	13-04-2013	Riscos de requisitos	Riscos procedentes de mudanças nos requisitos do sistema	R	M	B	Gestor e Analistas	Debates para decidir as melhores opções
4	13-04-2013	Não cumprimento do cronograma	Pessoas descritas do projeto	E	M	M	Gestor	Adaptar o esforço da equipe para atender às demandas do projeto
5	13-04-2013	Problemas com o Dashboard	Problemas com o repositório podem nos impedir de aceder à wiki e ao issue-tracker	T	M	B	Gestor	Ter uma plataforma de backup dos documentos
6	13-04-2013	Problemas com o Repositório	Problemas com o repositório podem nos impedir de aceder ao código-fonte	T	A	B	Gestor	Ter uma plataforma de backup do código-fonte
7	13-04-2013	Abandono da disciplina	Abandono da disciplina por um membro da equipe	P	A	B	Gestor	Reconfigurar os requisitos de forma à suprir a função abandonada
8	13-04-2013	Quedas de energia e/ou internet durante as reuniões de trabalho		O	M	M	Gestor	Ir para a casa de outro membro da equipa
9	13-04-2013	Viagens de membros da equipe		P	B	M	Gestor	Realocar tarefas mais importantes para as pessoas que não forem viajar

<sup>1</sup> E-Estimativa/O-Organização/P-Pessoas/R-Requisitos/T-Tecnologia

<sup>2</sup> B-Baixo/M-Moderado/A-Alto

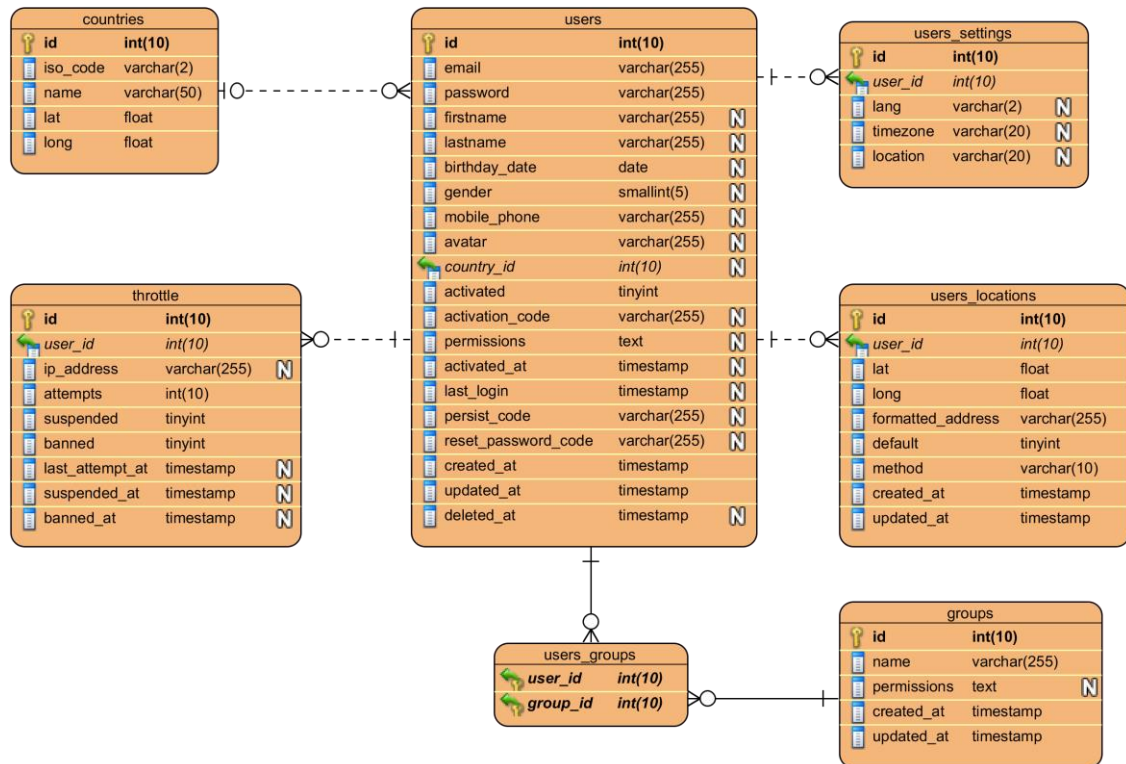
<sup>3</sup> B-Baixa/M-Moderada/A-Alta



# **Anexo 2 – Modelo de Dados**

*Especificação do Modelo de Dados do sistema*

## 4. GESTÃO DE UTILIZADORES, GRUPOS E PERMISSÕES



### 4.1. SUMÁRIO

Nome	Descrição
users_groups	Tabela pivot entre utilizadores e grupos de utilizadores
users	Tabela para guardar os dados básicos dos utilizadores
groups	Tabela para guardar os dados dos grupos de utilizadores
users_settings	Tabela para guardar as opções dos utilizadores
users_locations	Tabela para guardar as localizações dos utilizadores
countries	Tabela para guardar países

### 4.2. DICIONÁRIO DE DADOS

#### 4.2.1. USERS

Nome	Descrição	Tipo de Dados	de	N	PK	U	Restrições
------	-----------	---------------	----	---	----	---	------------

Id	Identificador do utilizador	int	X
Email	Email do utilizador	varchar	
password	Password encriptada do utilizador	varchar	
firstName	Nome próprio do utilizador	varchar	X
lastName	Apelido do utilizador	varchar	X
birthday_date	Data de nascimento do utilizador	date	X
gender	Género do utilizador (0-ND;1-M;2-F)	smallint	X
mobile_phone	Telemóvel do do utilizador	varchar	X
Avatar	Endereço da imagem de perfil do utilizador	varchar	X
country_id		int	X
activated	Conta de utilizador activa (0/1)	tinyint	
activation_code	Código de activação da conta de utilizador	varchar	X
permissions	Permissões do utilizador (json)	text	X
activated_at	Data de activação da conta	timestamp	X
last_login	Data da última autenticação na conta	timestamp	X
persist_code	Código identificador do utilizado em cookies ou sessões.	varchar	X
reset_password_code	Código de recuperação de password	varchar	X
created_at	Data de criação do utilizador	timestamp	
updated_at	Data da última modificação dos dados do utilizador	timestamp	
deleted_at	Data de eliminação da conta do utilizador	timestamp	X

#### 4.2.2. USERS\_SETTINGS

Nome	Descrição	Tipo de Dados	N	PK	U	Restrições
Id	Identificador das opções do utilizador	int		X		
user_id	Identificação do utilizador a que pertencem as opções	int				
Lang	Idioma do utilizador	varchar		X		
Timezone	Fuso horário do utilizador	varchar		X		
Location	Método de localização por defeito do utilizador (address-Morada do utilizador;last-última localização)	varchar		X		

### 4.2.3. USERS\_LOCATIONS

Nome	Descrição	Tipo de Dados	N	PK	U	Restrições
Id	Identificação da localização do utilizador	int		X		
user_id	Identificação do utilizador a que pertence a localização	int				
Lat	Latitude da localização (formato DDD)	float				
Long	Longitude da localização (formato DDD)	float				
formatted_address	Morada formatada da localização	varchar				
Default	Sinalizador para morada do utilizador	tinyint				
Method	Método de obtenção da localização (geo-localização; manual-manual)	varchar				
created_at	Data de criação da localização	timestamp				
updated_at	Data de actualização dos dados da localização	timestamp				

### 4.2.4. COUNTRIES

Nome	Descrição	Tipo de Dados	N	PK	U	Restrições
Id	Identificador do país	int		X		
iso_code	ISO Code do país	varchar				
Name	Nome do país	varchar				
Lat	Latitude do país (formato DDD)	float				
Long	Longitude do país (formato DDD)	float				

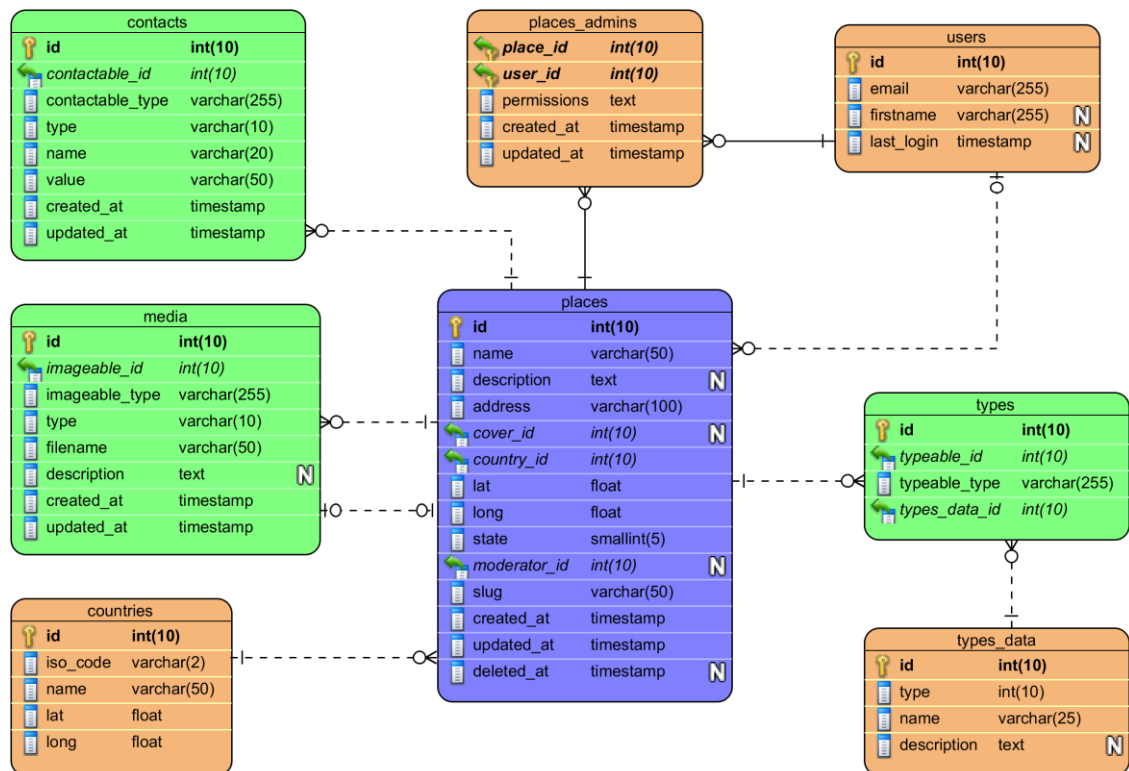
### 4.2.5. GROUPS

Nome	Descrição	Tipo de Dados	N	PK	U	Restrições
Id	Identificação do grupo	int		X		
Name	Nome do grupo	varchar				
permissions	Permissões do grupo (json)	text			X	
created_at	Data de criação do grupo	timestamp				
updated_at	Data de actualização dos dados do grupo	timestamp				

### 4.2.6. USERS\_GROUPS



Nome	Descrição	Tipo de Dados	N	PK	U	Restrições
user_id	Identificação do utilizador que pertence ao grupo	int			X	
group_id	Identificação do grupo a que pertence o utilizador	int			X	

## 5. GESTAO DE LOCAIS



### 5.1. SUMÁRIO

Nome	Descrição
users	Tabela para guardar os dados básicos dos utilizadores
types_data	Tabela para guardar tags
types	Tabela pivot entre locais e tags
places_admins	Tabela pivot entre utilizadores com acesso a administração a locais
places	Tabela para guardar os dados de locais
media	Tabela para guardar medias (imagens, vídeos, ...)

 countries	Tabela para guardar países
 contacts	Tabela para guardar contactos

## 5.2. DICIONÁRIO DE DADOS

### 5.2.1. PLACES

Nome	Descrição	Tipo de Dados	N	PK	U	Restrições
id	Identificador do local	int		X		
Nome	Nome do local	varchar				
description	Descrição do local	text		X		
address	Morada do local	varchar				
cover_id	Identificador da imagem de rosto do local	int		X		
country_id	Identificador do país a que pertence o local	int				
lat	Latitude do local (formato DDD)	float				
long	Longitude do local (formato DDD)	float				
state	Estado do local (0 - Pendente; 1 - Aprovado; 2- Rejeitado)	smallint				
moderator_id	Identificador do utilizador que moderou o local	int		X		
slug	Slug do local	varchar				
created_at	Data de criação do local	timestamp				

### 5.2.2. PLACES\_ADMINS

Nome	Descrição	Tipo de Dados	N	PK	U	Restrições
place_id	Identificador do local que utilizador tem permissões de administração	int		X		
user_id	Identificador do utilizador que tem permissões de administração no local	int		X		
permissions	Permissões do utilizador, no local (json)	text				
created_at	Data de criação das permissões de acesso do utilizador, ao local	timestamp				
updated_at	Data da última modificação dos das permissões do utilizador para o local	timestamp				

updated_at	Data da última modificação dos dados do local	timestamp	
deleted_at	Data de eliminação do local	timestamp	X

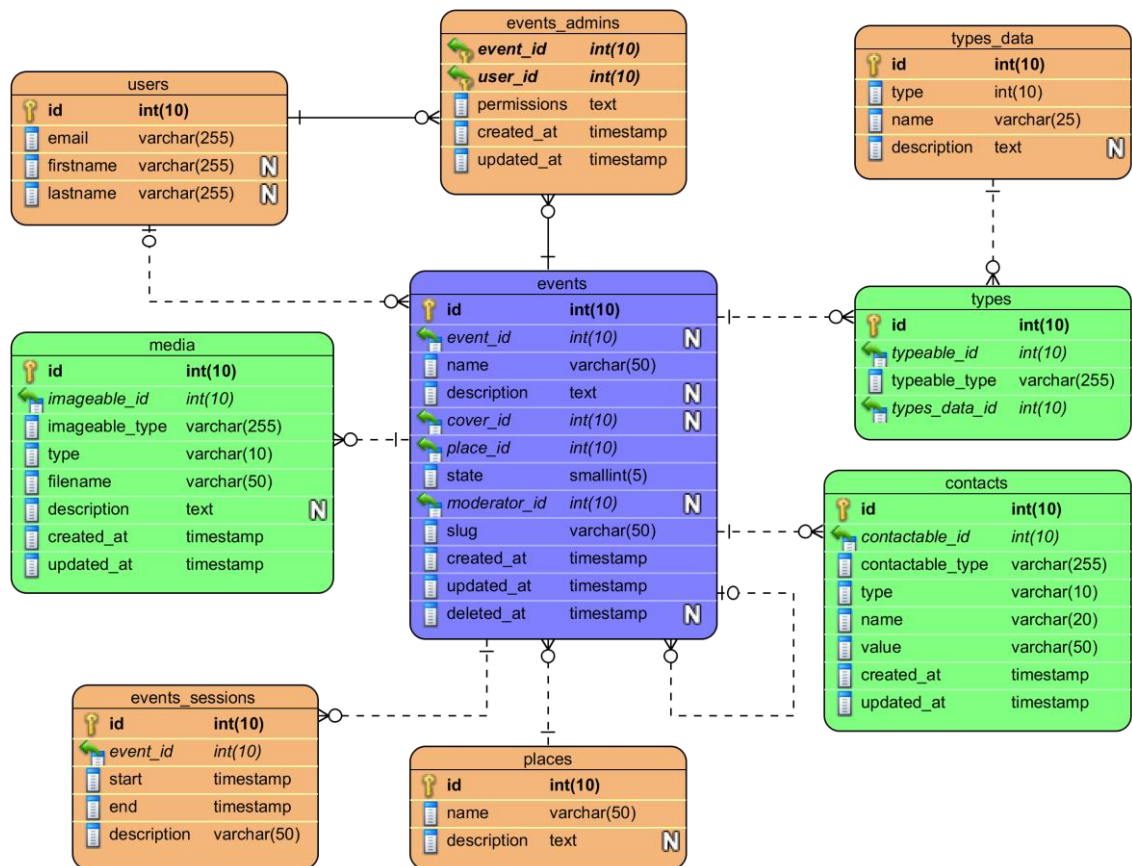
### 5.2.3. MEDIA

Nome	Descrição	Tipo de Dados	N PK U Restrições
id	Identificador da media	int	X
imageable_id	Identificador da entidade a que a media pertence	int	
imageable_type	Tipo da entidade a que a media pertence	varchar	
type	Tipo de media (image/video/...)	varchar	
fileName	Endereço da media	varchar	
description	Descrição da media	text	X
created_at		timestamp	
updated_at		timestamp	









### 5.2.4. CONTACTS


Nome	Descrição	Tipo de Dados	N PK U Restrições
Id	Identificador do contacto	int	X
contactable_id		int	
contactable_type	Tipo da entidade a que o contacto pertence	varchar	
Type	Tipo do contacto (phone, mobile_phone, fax, email, site)	varchar	
Name	Nome do do contacto	varchar	
Value	Valor do contacto	varchar	
created_at		timestamp	
updated_at		timestamp	

## 6. GESTÃO DE EVENTOS



### 6.1. SUMÁRIO

Nome	Descrição
 types	Tabela pivot entre locais e tags
 types_data	Tabela para guardar tags
 users	Tabela para guardar os dados básicos dos utilizadores
 media	Tabela para guardar medias (imagens, vídeos, ...)
 events	Tabela para guardar eventos
 events_admins	Tabela pivot entre utilizadores com acesso a administração a eventos
 events_sessions	Tabela para guardar sessões de eventos
 contacts	Tabela para guardar contactos

 places	Tabela para guardar os dados de locais
--	--

## 6.2. DICIONÁRIO DE DADOS

### 6.2.1. EVENTS

Nome	Descrição	Tipo Dados	de N PK U Restrições
id	Identificador do evento	int	X
event_id	Identificador do evento pai a que pertence o evento	int	X
name	Nome do evento	varchar	
description	Descrição do evento	text	X
cover_id	Identificador da imagem de rosto do evento	int	X
place_id	Identificador do local onde o evento decorre	int	
state	Estado do evento (0 - Pendente; 1 - Aprovado; 2-Rejeitado)	smallint	
moderator_id	Identificador do utilizador que moderou o evento	int	X
Slug	Slug do evento	varchar	
created_at		timestamp	
updated_at		timestamp	
deleted_at		timestamp	X

### 6.2.2. EVENTS\_ADMINS

Nome	Descrição	Tipo Dados	de N PK U Restrições
event_id	Identificador do evento que utilizador tem permissões de administração	int	X
user_id	Identificador do utilizador que tem permissões de administração no evento	int	X
permissions	Permissões do utilizador, no evento (json)	text	
created_at		timestamp	
updated_at		timestamp	

**6.2.3. EVENTS\_SESSIONS**

<b>Nome</b>	<b>Descrição</b>	<b>Tipo de Dados</b>	<b>N</b>	<b>PK</b>	<b>U</b>	<b>Restrições</b>
id	Identificador da sessão	int		X		
event_id	Identificador do evento a que pertence a sessão	int				
start	Data de início da sessão	timestamp				
end	Data de término da sessão	timestamp				
description	Descrição da sessão	varchar				

# **Anexo 3 – Plano de Negócio**

*Plano de negócio do projeto, elaborado em conjunto com outros colegas*

# Portal de Eventos

## Plano de Negócios

1070985 – Jacinto Barbosa

1080708 – Miguel Borges

1081326 – Sérgio Porto Lapa



# Portal de Eventos

## Plano de Negócios

### Índice

<b>CAPÍTULO 1 - INTRODUÇÃO</b> .....	<b>3</b>
<b>CAPÍTULO 2 - SUMÁRIO EXECUTIVO</b> .....	<b>4</b>
2.1. O CONCEITO DO NEGÓCIO.....	4
2.2. EQUIPA DE GESTÃO .....	4
2.3. MERCADO E CONCORRÊNCIA.....	4
2.4. MARKETING E VENDAS.....	4
2.5. ESTRATÉGIA DE CRESCIMENTO.....	5
2.6. VOLUME DE NEGÓCIOS E INVESTIMENTOS .....	5
<b>CAPÍTULO 3 - CARACTERIZAÇÃO DO NEGÓCIO</b> .....	<b>6</b>
3.1. APRESENTAÇÃO DO NEGÓCIO.....	6
3.2. MISSÃO .....	6
3.3. MOTIVAÇÃO .....	6
3.4. PONTOS CRÍTICOS PARA O DESENVOLVIMENTO DO PROJECTO.....	7
<b>CAPÍTULO 4 - PLANO DE MARKETING</b> .....	<b>8</b>
4.1. ANÁLISE DO MERCADO .....	8
4.2. ANÁLISE DA CONCORRÊNCIA .....	9
4.3. ANÁLISE PEST .....	9
4.3.1. <i>Factores Políticos</i> .....	9
4.3.2. <i>Factores Económicos</i> .....	10
4.3.3. <i>Factores Socioculturais</i> .....	10
4.3.4. <i>Factores Tecnológicos</i> .....	10
4.4. ANÁLISE SWOT .....	10
4.5. OBJECTIVOS ESTRATÉGICOS.....	11
4.6. MARKETING MIX .....	11
4.6.1. <i>Produto/Serviço</i> .....	11
4.6.1.1. Distribuição dos Eventos .....	12
4.6.1.2. Envio de SMS's e Newsletters .....	12
4.6.1.3. Conteúdos Destacados.....	12
4.6.1.4. Novos Serviços e Funcionalidades.....	12
4.6.2. <i>Preço</i> .....	13
4.6.3. <i>Distribuição</i> .....	13
4.6.4. <i>Comunicação</i> .....	13
4.6.4.1. Utilizadores .....	13
4.6.4.2. Produtores .....	13
4.6.4.3. Manutenção mensal .....	14

4.7. PREVISÃO DAS VENDAS .....	14
<b>CAPÍTULO 5 - PLANO ORGANIZACIONAL .....</b>	<b>16</b>
<b>CAPÍTULO 6 - PLANO DE PRODUÇÃO .....</b>	<b>17</b>
<b>CAPÍTULO 7 - PLANO E FINANCEIRO .....</b>	<b>18</b>
7.1. PRESSUPOSTOS DO PROJECTO .....	18
7.2. PLANO DE EXPLORAÇÃO PREVISIONAL.....	18
7.2.1. <i>Volume de Negócio</i> .....	18
7.2.2. <i>Custo das Mercadorias Vendidas e Consumidas</i> .....	19
7.2.3. <i>Fornecimentos e Serviços Externos</i> .....	20
7.2.4. <i>Custos com o Pessoal</i> .....	20
7.3. PLANO DE INVESTIMENTO.....	20
7.4. NECESSIDADE DE FUNDO DE MANEIO .....	22
7.5. VIABILIDADE ECONÓMICA E FINANCEIRA .....	22
<b>CAPÍTULO 8 - CONCLUSÃO .....</b>	<b>24</b>
<b>CAPÍTULO 9 - REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>25</b>
<b>CAPÍTULO 10 - ANEXOS .....</b>	<b>26</b>
10.1. PROJEÇÕES DOS DADOS DE ACESSO AO PORTAL/SERVIÇO.....	26
10.2. RENDABILIDADE DA PUBLICIDADE.....	26
10.3. OUTRAS PREVISÕES.....	27

## Capítulo 1 - Introdução

Com o presente plano de negócio pretende-se fazer uma análise da viabilidade de um projecto que está a ser desenvolvido no âmbito de uma Tese de Mestrado, pelos alunos Miguel Borges e Sérgio Lapa.

Numa primeira fase, pretendemos como objectivos uma análise o mais profunda possível da área do negócio (devido à sua curta existência no mercado português) iniciando-se na análise Interna e Externa (PEST e SWOT), definição de uma Missão, Visão, Valores e Objectivos estratégicos, uma análise do mercado já existente e da possível concorrência, assim como análise do mercado alvo (segmentar e posicionar este mediante os objectivos da empresa) e a definição de um plano de Marketing que permita atingir os objectivos propostos.

Numa segunda fase, o objectivo é verificar se a constituição de uma empresa para comercializar o serviço deve ou não ser feita. Desta forma partimos de uma análise financeira da empresa através de um Projecto de Investimento. A análise vai ser feita através de indicadores apropriados como o VAL, a TIR, o Prazo de Recuperação do Capital, o IR que indicam a possível viabilidade da empresa a curto e longo prazo.

## Enquadramento

Este projecto foi desenvolvido do âmbito do trabalho curricular da cadeira de Inovação e Empreendedorismo, do Mestrado de Engenharia Informática, do Instituto Superior de Engenharia do Porto, do ano lectivo 2012/2013, Leccionada pelo Professor João Pedro Andrade.

## Capítulo 2 - Sumário Executivo

### 2.1. O Conceito do Negócio

O portal de eventos surgiu da oportunidade identificada em actuar como um portal de procura de eventos. Criar e disponibilizar um sistema (portal e aplicações móveis) para procura de todo o tipo de eventos: desportivos, de lazer, de treino/*fitness*, concertos de música, encontros/meetings, etc. Deverá dar resposta à procura existente no mercado nacional português, englobando as necessidades da população na procura de entretenimento diário (diurno ou nocturno), sazonal (férias entre outros) ou simplesmente satisfação de gostos musicais temporais

### 2.2. Equipa de Gestão

A equipa de gestão do portal de eventos é um dos pontos fortes do negócio, sendo composta por dois profissionais, que possuem sólida experiência em negócios e tecnologia, actuando há mais de cinco anos na prestação de serviços via internet. Possuem formação ao nível de mestrado em engenharia informática e, sobretudo, motivação para enfrentar e superar os desafios de administrar, gerar resultados positivos e conquistar uma significativa quota de mercado para este negócio.

### 2.3. Mercado e Concorrência

Trata-se de um mercado bastante recente (menor que 8 meses) e por tal não é conhecida a sua dimensão real.

No que respeita à concorrência existem apenas três concorrentes directos, mas que não contemplam a existência de aplicações móveis e a falta de apoio aos produtores ao divulgar os seus eventos.

A proposta deste portal procura precisamente colmatar esta falta, uma vez que existe uma expressiva utilização de dispositivos móveis.

### 2.4. Marketing e Vendas

A estratégia de marketing do portal de eventos visa conquistar mercado de forma rápida, focando-se numa primeira fase ao mercado nacional e posteriormente ao mercado internacional. Os principais fundamentos de marketing foram considerados num plano que pretende alcançar tanto consumidores, como promotores de eventos que se associarão a este portal. As estimativas apontam para uma publicação nacional de 1000 eventos por ano, num prazo de 2 anos, uma quota de mercado de 50% no prazo máximo de 3 anos, uma expansão ao mercado internacional com 20% quota de mercado em pelo menos 5 países e colaborar com os maiores/principais promotores de eventos.

A previsão das vendas teve como mercado-alvo, o mercado português uma vez que o seu lançamento será inicialmente para Portugal e prevê-se que em 2013 contaremos com cerca de

780 utilizadores registados com um crescimento exponencial que ultrapassará um milhão em 2016.

## 2.5. Estratégia de Crescimento

O lançamento do portal será efectuado em Portugal para o mercado nacional numa fase inicial e prevê-se a sua expansão para o mercado internacional numa segunda fase, principalmente para os países de língua oficial portuguesa. Estando previsto também para a segunda fase a disponibilização do portal em outras línguas.

## 2.6. Volume de Negócios e Investimentos

O volume de negócios não terá expressão em 2013, mas em 2014 já se prevê um VN superior a 16.500,00€ e que crescerá com uma previsão de em 2016 ultrapassar já a fasquia dos 150.000,00€/ano.

A actividade da empresa será sustentada em serviços Web pelo que o investimento inicial irá se apoiar em capitais próprios no valor de 5.000€.

## Capítulo 3 - Caracterização do Negócio

### 3.1. Apresentação do Negócio

O turismo é considerado um sector fundamental da economia portuguesa visto que o nosso país é visitado por milhares de turistas anualmente, uma das principais dificuldades que estes turistas encontram ao chegarem ao nosso país é descobrirem o que podem/têm para fazer para se divertirem, entreterem, passar o tempo ou simplesmente encontrarem lugares para visitar e/ou conhecer.

Como as maiores organizadoras, as agências de viagens são na sua maioria planeadoras de actividades turísticas, sendo que a maioria dos eventos programados são excursões. No caso dos mais jovens e de muitos emigrantes estes tipos de eventos não são atractivos, pelo que a sua maioria prefere saídas à noite. Foi para colmatar estes problemas que surgiu a ideia de agregar este tipo de informações num único local.

O projecto consiste num serviço que permita facilmente descobrir, promover e gerir evento. Os eventos podem ser dos mais variados tipos, como por exemplo concertos, eventos desportivos, festas populares, “workshops”, meetings, etc... Outro aspecto interessante deste serviço será a aproximação dos utilizadores (pessoas que atendem aos eventos) aos promotores ou até aos seus artistas preferidos, através da promoção directa ou indirecta dos eventos a realizar.

Outro dos problemas que o serviço bem colmatar será o da gestão dos eventos e espaços, por parte dos promotores, pelos diversos locais na rede global. Para isso o sistema estará interligado com outros serviços, o que permite aos promotores de eventos gerirem os seus eventos, nos mais diversos locais (facebook, google+, twitter, ...), a partir de um único local, poupando-lhes assim tempo e trabalho.

Este serviço apesar de ser lançado em Portugal, não há nada impeditivo para que este possa ser usado internacionalmente, principalmente por países de Língua Portuguesa.

### 3.2. Missão

*“Facilitar o acesso à informação para descoberta, promoção, divulgação e gestão de eventos.*

*Ser um portal de referência a nível nacional cativando promotores e público em geral.”*

### 3.3. Motivação

Acreditamos ser possuidores de formação e de conhecimento acumulado, que nos dão o “know-how” necessário para o desenvolvimento, promoção e gestão deste projecto.

Sentimo-nos fortemente motivados e empenhados em dar toda a dedicação necessária para o sucesso deste negócio.

A motivação para a criação deste negócio surge do levantamento do seu potencial, associado à ideia de ter o nosso próprio emprego e a oportunidade de gerar valor económico que dificilmente será possível se continuarmos como trabalhadores por conta de outrem. Tendo em conta a actual situação do mercado de trabalho, os baixos salários normalmente propostos e ainda em situações em que se assiste a uma redução salarial. Pretendemos assim fazer face ao desejo de uma mudança a esta situação.

Além disso ter o nosso próprio negócio permite-nos de uma melhor forma dar expressão às nossas qualidades profissionais.

### 3.4. Pontos Críticos para o Desenvolvimento do Projecto

Os pontos críticos para este negócio são:

- A aceitação do nosso serviço junto dos promotores comerciais é fundamental para a captação de fontes de receita que assegurem a continuidade do negócio.
- Sucesso da utilização das nossas aplicações para dispositivos móveis (*smartphones* e *tablets*), pois caso as aplicações móveis não sejam suficientemente atractivas e práticas de utilizar, isso irá certamente condicionar a distribuição da informação sobre os eventos em promoção.
- A internacionalização do produto será um factor exponencial para o sucesso do volume de negócios.

## Capítulo 4 - Plano de Marketing

### 4.1. Análise do Mercado

O sistema terá como público-alvo, que serão os clientes do serviço, três grupos distintos (Tabela 1):

- Promotores de eventos – Produtores de eventos que queiram difundir de uma forma fácil e rápida os seus eventos, pelo máximo de clientes possíveis.
- Gerentes de estabelecimentos de lazer – Bares, discotecas, teatros, espaços culturais, entre outros. Gerentes que queiram difundir e dar a conhecer o seu espaço.
- População em geral – comum cidadão nacional ou turista internacional que pretenda consultar/descobrir eventos e espaços.

Número de Espectáculos		Número de Espaços de Lazer	
Espectáculos	Número	Espaços	Número
Música e Dança	11.926	Recintos Culturais	347
Teatro	12.174	Museus e Jardins	340
Ópera	155	Galerias	881
Desporto	0	Bares	1.987
<b>Total:</b>	<b>24.255</b>	Discotecas	759
		<b>Total:</b>	<b>4.314</b>

TABELA 1 E 2 - NÚMERO DE ESPECTÁCULOS/EVENTOS E ESPAÇOS DE LAZER EM PORTUGAL, EM 2011

- **Nota 1:** Dados obtidos através dos portais do Instituto Nacional de Estatística, PORDATA e Loja do Software. [1–6].
- **Nota 2:** Não nos foi possível verificar o número de eventos desportivos, mas especulamos que essa parcela seja maior que as restantes.

## 4.2. Análise da Concorrência

A concorrência a nível nacional neste ramo é muito recente (última metade do 2012) e só cobre o último ponto do público-alvo do nosso serviço. Seguem uma tabela com a listagem dos principais concorrentes directos e os respectivos pontos fortes/fracos de cada um deles.

Concorrentes directos	Pontos Fortes	Pontos Fracos
WeGoOut	<ul style="list-style-type: none"><li>• Interface bonita, simples e intuitiva</li><li>• Solução para iOS</li><li>• Integração com o Facebook</li></ul>	<ul style="list-style-type: none"><li>• Inexistência de Apps para Android</li><li>• Dependência do Facebook</li><li>• Não permite a inserção de eventos</li><li>• Não contém base de dados de espaços</li></ul>
VaiBater	<ul style="list-style-type: none"><li>• Fundado por Figuras Públicas (Mónica e Rubim)</li></ul>	<ul style="list-style-type: none"><li>• Sistema único centralizado</li><li>• Inexistência de Soluções Móveis</li></ul>
EntradaLivre	<ul style="list-style-type: none"><li>• Associada a uma grande marca (TMN)</li></ul>	<ul style="list-style-type: none"><li>• Só suporta eventos pagos</li><li>• Sistema centralizado</li><li>• Inexistência de Soluções Móveis</li><li>• Não contém base de dados de espaços</li></ul>

Pode-se considerar um concorrente indirecto as Páginas Amarelas<sup>1</sup>. Estas apesar de não serem especializadas na área de negócio do nosso serviço contêm algumas informações de espaços de lazer.

## 4.3. Análise PEST

A análise externa revela que Portugal é um bom mercado-alvo para a introdução deste negócio, graças a este ser um destino turístico, mas no entanto devido a factores económicos no território nacional, pode haver uma diminuição dos eventos realizados o que leva à necessidade de o negócio ser exportado para outros países.

### 4.3.1. Factores Políticos

- Livre circulação de Pessoas e Bens na União Europeia facilita a promoção de eventos ou do serviço noutros países.
- A subida dos impostos, prevista no novo orçamento de Estado, que pode implicar uma quebra de vendas em Portugal.

<sup>1</sup> <http://www.pai.pt/>

#### 4.3.2. Factores Económicos

- A dificuldade no acesso aos recursos financeiros. Os bancos sobem as taxas de juros ou dificultam a aprovação dos créditos atrasando processos e diminuindo o número de eventos realizados pela população.

#### 4.3.3. Factores Socioculturais

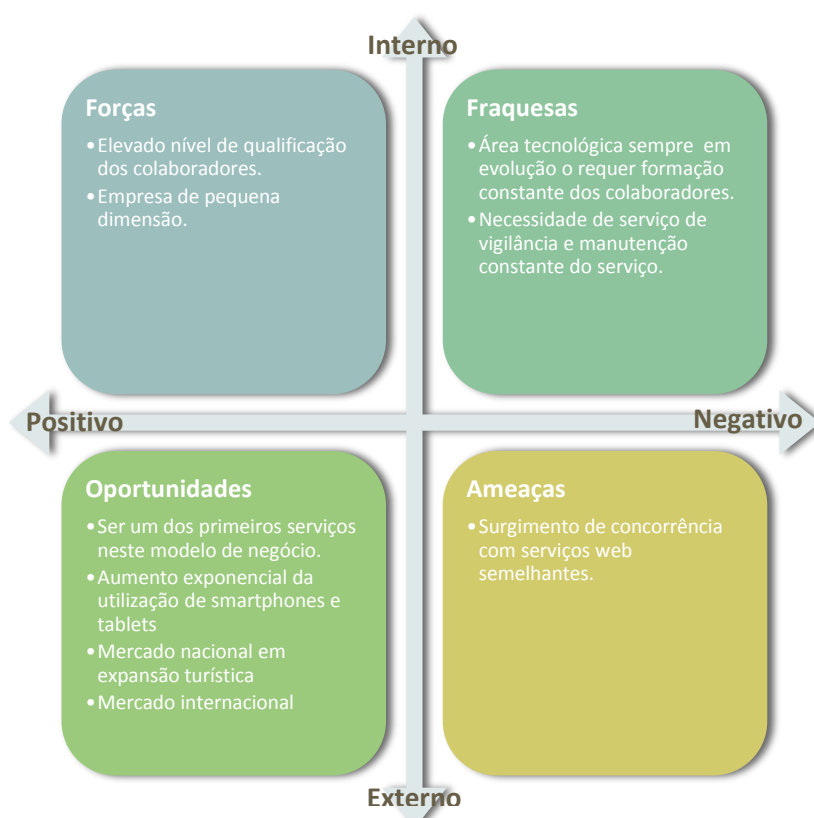
- O programa do Governo das “Novas Oportunidades” lançou milhares novos utilizadores de Internet em Portugal e o aumento do desemprego leva as pessoas a um constante uso doméstico da Internet o que potencia as visualizações dos eventos e por conseguinte a rentabilidade do negócio através da publicidade.
- Utilização do Inglês como linguagem de programação facilita a expansão da empresa para o estrangeiro.
- Portugal como um destino turístico. Leva a que os produtores de eventos tentem “captar” os turistas para os seus eventos.

#### 4.3.4. Factores Tecnológicos

- Surgimento de novas tecnologias e ferramentas no âmbito do desenvolvimento web.
- Melhoramento das capacidades de processamento e armazenamento dos dispositivos móveis, o que leva ao desenvolvimento de aplicações móveis cada vez mais sofisticadas e completas.

### 4.4. Análise SWOT

Analisando os ambientes interno e externo do negócio, obteve-se a seguinte tabela SWOT:



## 4.5. Objectivos Estratégicos

Os objectivos estratégicos do negócio passam por:

1. Atingir uma publicação nacional de 1000 eventos por ano, num prazo de 2 anos, pois quanto mais conteúdo o portal tiver, mais clientes e visitantes atrai.
2. Conseguir uma quota de mercado de 50% no prazo máximo de 3 anos, visto que é dos primeiros negócios neste ramo e o único que oferece soluções de promoção aos produtores de eventos.
3. Expandir o negócio ao mercado internacional com 20% quota de mercado em pelo menos 5 países, no prazo de 5 anos. Isto leva um aumento exponencial do volume de negócios.
4. Colaborar com os maiores/principais promotores de eventos, o que leva a uma maior visibilidade e transpassa uma maior confiança aos produtores locais/pequena dimensão e aos próprios utilizadores.
5. Conseguir 5000 visualizações diárias, o que deixa uma almofada no volume de negócios, através da rentabilidade da publicidade por *pageviews*.

As estratégias usadas para a concretização desses objectivos são:

1. Publicação dos eventos nas redes sociais. (Aumenta a visibilidade do serviço)
2. Envio de newsletters semanais aos utilizadores registados. (Retém os utilizadores no serviço)
3. Efectuar parcerias com operadoras de redes móveis (via sms/convites a custo zero). (Retém os utilizadores no serviço)
4. Desenvolver contactos e acções comerciais junto dos promotores de eventos. (credibiliza o serviço e aumenta o conteúdo).
5. Desenvolver aplicações móveis para facilitar o acesso à informação em qualquer lugar. (Alarga o número de utilizações do serviço)
6. Acréscimo de outros idiomas (que não o Português) ao portal para facilitar consulta. (Atrai novos utilizadores e por conseguinte novos mercados)
7. Efectuar campanhas publicitárias no mercado. (Aumenta a visibilidade do serviço)  
[Ver secção 3.6 - Marketing Mix]

## 4.6. Marketing Mix

O Marketing Mix explícita a estratégia a ser adoptada nos seus diversos aspectos.

### 4.6.1. Produto/Serviço

O tratamento do serviço será o mesmo para todos os clientes, sem distinção. O mesmo ocorre para os consumidores interessados em adquirir algum serviço, garantindo assim um serviço uniforme e sem variações, mais adequado à estratégia inicial de expansão rápida e ganho de “*market share*”.

#### **4.6.1.1. Distribuição dos Eventos**

Serviço inovador que permite aos produtores de eventos “espalharem” e gerirem as informações dos seus eventos, por serviços externos, a partir de um único local.

No lançamento do portal estarão disponíveis os seguintes sistemas externos, para a partilha e gestão da informação:

- Facebook
- Twitter
- Google+

#### **4.6.1.2. Envio de SMS's e Newsletters**

Serviço inovador que permite aos produtores de eventos enviarem SMS's e/ou E-mails, de forma personalizada, aos utilizadores registados no portal, para promoverem os seus eventos de forma directa e eficaz.

Os produtores poderão segmentar e filtrar os utilizadores que desejam que sejam informados (p.e. área geográfica, idade, gostos, ...).

#### **4.6.1.3. Conteúdos Destacados**

Este serviço proporciona maior destaque e visibilidade aos eventos e/ou aos espaço. Os eventos e/ou espaço estarão visíveis de modo rotativo, na montra de anúncios colocada num lugar de destaque, de acordo com a categoria e região em que o anúncio foi inserido.

Desta forma os seus anúncios serão mostrados num lugar de destaque a quem está interessado e efectuou uma pesquisa relacionada com o conteúdo.

Nas listagens de conteúdos, estes também aparecerão de forma destacada dos restantes.

Dependendo do resultado, será analisada a viabilidade de venda por palavra-chave e região.

#### **4.6.1.4. Novos Serviços e Funcionalidades**

Está programada uma série de melhorias no sistema, com a inserção de novos serviços e funcionalidades, o que agregará maior valor ao serviço junto ao público-alvo. As melhorias planeadas são:

- Criação de um “tutorial” *online* para ensinar produtores a retirarem o maior proveito do sistema;
- Inserção de novos idiomas, o que possibilita o uso do serviço por um maior número de pessoas;
- Criação de um serviço que permita aos produtores e gerentes de espaços gerirem reservas e venda de bilhetes;
- Criação de um serviço de venda de bilhetes de grandes eventos, a um preço mais baixo que a concorrência.

- Criação de um serviço de notificações, que notifica os utilizadores de eventos na sua zona e/ou de acordo com os seus gostos.

#### 4.6.2. Preço

A estratégia de ganho de mercado implica uma política de preços acessíveis ao público-alvo. Com base na nossa experiência e na percepção do mercado da Internet, definiu-se a seguinte estratégia de preços:

Serviços	Preço
Publicidade p/ pageview	0,025 €
Sms	0,045 €
Email	0,005 €
Conteúdos destacados	10,00 €
Distribuição dos Eventos	Grátis

TABELA 3 – PREÇOS DOS SERVIÇOS

#### 4.6.3. Distribuição

Como o serviço a ser desenvolvido é um serviço *online*, naturalmente que a sua distribuição do serviço será feita através da Internet, em que com base na análise do mercado e em consonância com a estratégia de marketing estipulada, o mercado-alvo será, inicialmente Portugal, com ampliação gradativa em Países de Língua Portugueses por fim no resto do Mundo.

#### 4.6.4. Comunicação

Serão utilizados vários canais de publicidade para promover o serviço, junto dos utilizadores e dos produtores de eventos, com políticas de publicidade e promoções específicas para cada caso.

##### 4.6.4.1. Utilizadores

- **Publicidade inicial:** 30 dias
- **Público-alvo:** homens e mulheres, faixa etária entre 14 e 40 anos
- **Acções:**

Acção	Impacto Pretendido
Internet	100.000
E-mail	50.000

##### 4.6.4.2. Produtores

- **Publicidade inicial:** 30 dias
- **Público-alvo:** Produtores de eventos e gerentes de espaços de lazer com eventos regulares
- **Acções:**

Acção	Impacto Pretendido
Internet	100
E-mail	40

<b>Telemarketing</b>	20 gerentes de espaços de lazer 10 grandes produtores
----------------------	--

#### 4.6.4.3. Manutenção mensal

- **Ações:**

<b>Ação</b>	<b>Impacto Pretendido</b>
Internet	100.000
E-mail	20.000
Panfletos	3.000

### 4.7. Previsão das Vendas

A previsão das vendas teve como mercado-alvo, o mercado português. Sendo o projecto apresentado neste plano, um serviço *online*, este irá certamente ser utilizado por outros países, desde logo os países de língua oficial portuguesa, e a quando a implementação de novos idiomas no sistema, por os restantes mercados.

Para se poder fazer uma previsão das vendas dos serviços anteriormente descritos, foi necessário fazer primeiro uma previsão dos dados de acesso ao portal, visto que esta é a base do negócio.

	2013	2014	2015	2016
<b>Utilizadores Registados</b>	579	6 444	8 834	15 774
<b>Pageviews</b>	7 525	86 600	364 750	1 059 600

TABELA 4 - PREVISÕES DOS DADOS DE ACESSO AO PORTAL

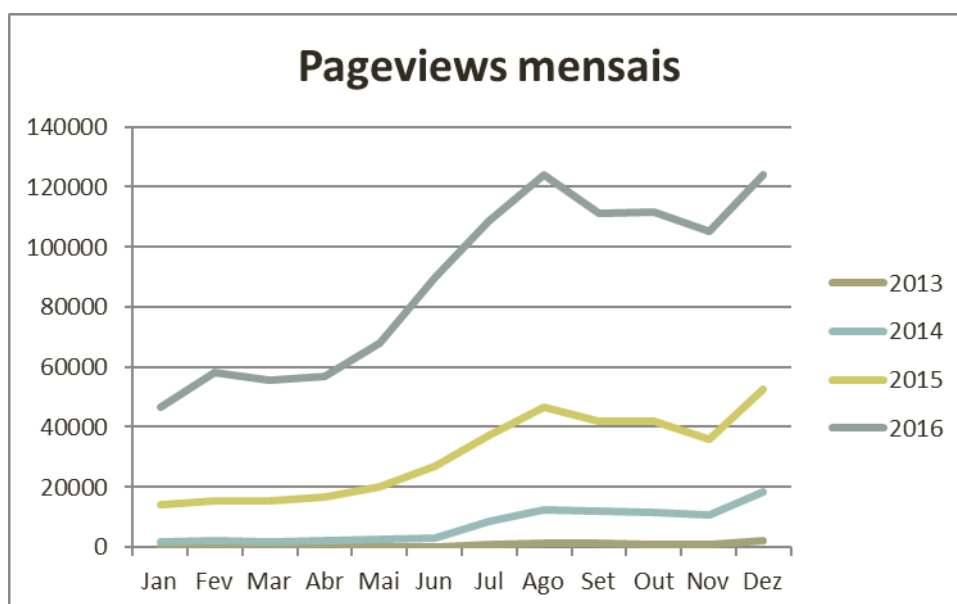
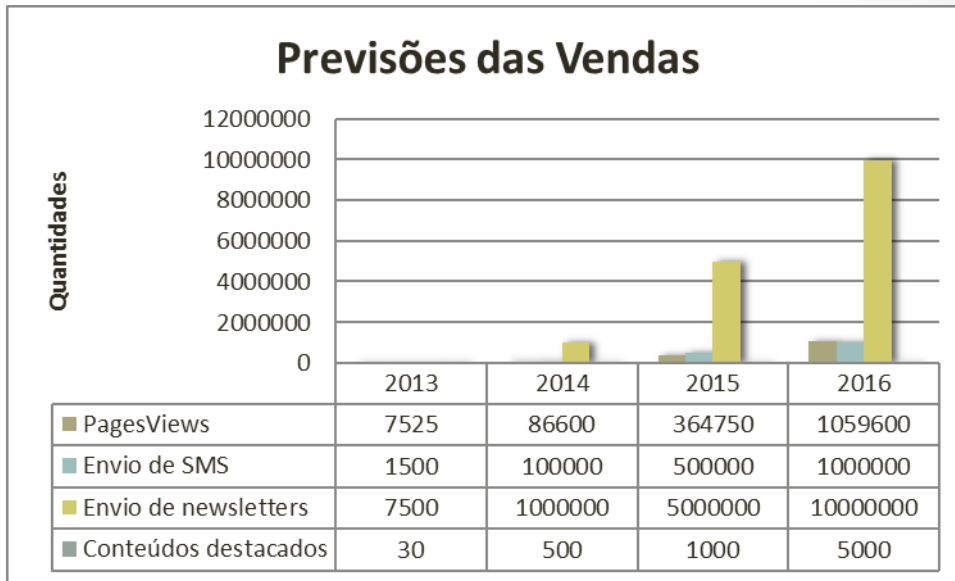


GRÁFICO 1 - PREVISÃO DAS PAGEVIEWS MENS AIS



**GRÁFICO 2 - PREVISÕES DAS VENDAS DOS SERVIÇOS**

## Capítulo 5 - Plano Organizacional

Atendendo a que este negócio, nos primeiros anos, incluirá apenas dois postos de trabalho que serão os seus sócios gerentes, não existe a necessidade da criação de um organograma, nem a definição de um instrumento de coordenação de actividades e de controlo da actuação dos seus elementos.

A distribuição dos seus recursos humanos é simples e não existe a necessidade da definição de canais de comunicação oficial.

## Capítulo 6 - Plano de Produção

Será alugado um servidor em regime de *cloud computing* para alojamento do portal de eventos. A manutenção e o desenvolvimento de novos serviços do portal serão assegurados pelos dois sócios-gerentes. A criação e gestão de conteúdos dos perfis das redes sociais também serão asseguradas pelos mesmos.

Dado tratar-se da prestação de um serviço, não se aplica aqui a necessidade de prever as quantidades a produzir para fazer face ao previsto no plano de vendas.

## Capítulo 7 - Plano e Financeiro

### 7.1. Pressupostos do Projecto

Como o projecto irá ser desenvolvido no âmbito de uma Tese de Mestrado, os custos de desenvolvimento, não entram no estudo económico e financeiro. O início da actividade propriamente dita, será no último trimestre de 2013 (1 de Outubro), mas prevê-se que o sistema já esteja disponível ao público em Agosto, de forma a ir ganhando notoriedade. Por isso, o número de meses de exploração em 2013 será de 3 meses, sendo 12 meses, nos anos seguintes.

Devido ao projecto ser um serviço *online* os recebimentos dos serviços são pagos a pronto e os rendimentos da publicidade são recebidos no final do mês. No entanto, está a ser estudado um sistema de crédito (parecido com o do Facebook), onde os utilizadores terão que carregar as suas contas para à posteriori poderem utilizar os serviços pagos. Os pagamentos também são efectuados a pronto.

### 7.2. Plano de Exploração Previsional

#### 7.2.1. Volume de Negócio

Dado que o *start-up* do portal é Portugal, estima-se que 80% dos clientes/utilizadores virão deste país e os demais de outros países de língua oficial portuguesa ou com uma grande percentagem de emigrantes portugueses. As projecções de *pageviews* foram feitas tendo em conta dados estatísticos da utilização de serviços *online* e das redes sociais em Portugal (Ver Anexos - página 26).

Vol. Negócios/Receb.	2013	2014	2015	2016
<b>Publicidade:</b>	<b>186,17 €</b>	<b>2.142,48 €</b>	<b>9.023,92 €</b>	<b>26.214,50 €</b>
Qtd. PagesViews:	7.525	86.600	364.750	1.059.600
Preço unitário:	0,0247 €	0,0247 €	0,0247 €	0,0247 €
<b>Envio de SMS:</b>	<b>67,50 €</b>	<b>4.500,00 €</b>	<b>22.500,00 €</b>	<b>45.000,00 €</b>
Quantidade:	1.500	100.000	500.000	1.000.000
Preço unitário:	0,05 €	0,05 €	0,05 €	0,05 €
<b>Envio de newsletters:</b>	<b>37,50 €</b>	<b>5.000,00 €</b>	<b>25.000,00 €</b>	<b>50.000,00 €</b>
Quantidade:	7.500	1.000.000	5.000.000	10.000.000
Preço unitário:	0,01 €	0,01 €	0,01 €	0,01 €
<b>Conteúdos destacados:</b>	<b>300,00 €</b>	<b>5.000,00 €</b>	<b>10.000,00 €</b>	<b>30.000,00 €</b>
Quantidade:	30	500	1.000	3.000
Preço unitário:	10,00 €	10,00 €	10,00 €	10,00 €
<b>Total</b>	<b>591,17 €</b>	<b>16.642,48 €</b>	<b>66.523,92 €</b>	<b>151.214,50 €</b>

TABELA 5 - VOLUME DE NEGÓCIO / RECEBIMENTOS

A maior parcela dos rendimentos do Volume de Negócio são obtidos através dos rendimentos do envio de Newsletters (~39%) e SMSs (~34%), seguindo-se os conteúdos destacados (~23%) e por último, os rendimentos com a publicidade (~13%) (Gráfico 3 e 2).

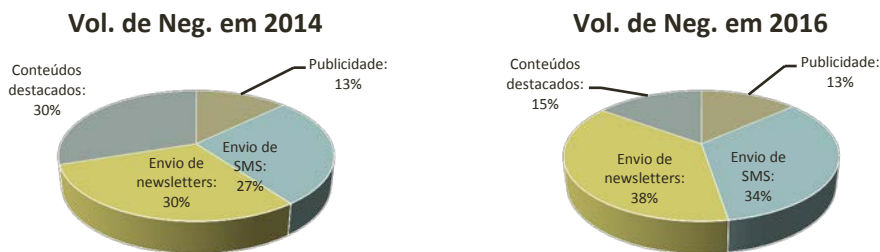


GRÁFICO 3 E 4 - PARCELAS DO VOL. DE NEG. EM 2014 E 2016

No entanto para o cálculo dessa parcela, só foi contabilizada a rentabilidade da publicidade exposta no portal. Ficaram de lado a rentabilidade da publicidade nas aplicações móveis, por esta ainda ter uma variação muito grande ao longo do tempo, e por conseguinte impossível de estimar. No entanto, actualmente esta rende 3 vezes mais que a publicidade nos *websites*. [7]

### 7.2.2. Custo das Mercadorias Vendidas e Consumidas

O valor do custo das mercadorias vendidas e consumidas deve-se simplesmente ao custo de envio das SMS's enviadas, através do fornecedor, pois os restantes serviços não requerem o uso de produtos nem de serviços.

Produtos	Preço
Publicidade p/ pageview	- €
SMS	0,02 €
Email	- €
Conteúdos destacados	- €

TABELA 6 - CUSTO UNITÁRIO DOS SERVIÇOS

Custo das Merc. Vend.	2013	2014	2015	2016
<b>SMS:</b>	<b>31,50 €</b>	<b>2.100,00 €</b>	<b>10.500,00 €</b>	<b>21.000,00 €</b>
Quantidade:	1.500	100.000	500.000	1.000.000
Custo unitário:	0,02 €	0,02 €	0,02 €	0,02 €
<b>Servidores:</b>	<b>295,16 €</b>	<b>1.180,65 €</b>	<b>1.180,65 €</b>	<b>1.180,65 €</b>
Custo unitário:	98,39 €	98,39 €	98,39 €	98,39 €
<b>Domínios:</b>	<b>80,00 €</b>	<b>80,00 €</b>	<b>80,00 €</b>	<b>80,00 €</b>
Custo unitário:	20,00 €	20,00 €	20,00 €	20,00 €
<b>Total</b>	<b>406,66 €</b>	<b>3.360,65 €</b>	<b>11.760,65 €</b>	<b>22.260,65 €</b>

TABELA 7 – CMVMC

### 7.2.3. Fornecimentos e Serviços Externos

Para o efeito considerámos serviços como Electricidade, água, telefone e acesso à internet do escritório, assim como rendas e alugueres ou serviços de publicidade e contabilidade exterior.

Fonerc. e Serv. Externos	2013	2014	2015	2016
<b>Serviços Especializados</b>	<b>63,00 €</b>	<b>492,00 €</b>	<b>492,00 €</b>	<b>492,00 €</b>
Telefone+Internet	63,00 €	492,00 €	492,00 €	492,00 €
<b>Materiais</b>	<b>142,28 €</b>	<b>480,54 €</b>	<b>480,54 €</b>	<b>480,54 €</b>
Mat. e utenc. de desg. ráp.	142,28 €	480,54 €	480,54 €	480,54 €
<b>Energia e Fluídos</b>	<b>177,00 €</b>	<b>708,00 €</b>	<b>708,00 €</b>	<b>708,00 €</b>
Electricidade	135,00 €	540,00 €	540,00 €	540,00 €
Água	42,00 €	168,00 €	168,00 €	168,00 €
<b>Serviços Diversos</b>	<b>2.220,00 €</b>	<b>8.580,00 €</b>	<b>8.580,00 €</b>	<b>8.580,00 €</b>
<u>Rendas e Alugueres</u>	1.080,00 €	4.320,00 €	4.320,00 €	4.320,00 €
Comunicações Móveis	90,00 €	360,00 €	360,00 €	360,00 €
<u>Serv. de Publicidade</u>	300,00 €	900,00 €	900,00 €	900,00 €
Serviços de contabilidade	750,00 €	3.000,00 €	3.000,00 €	3.000,00 €
<b>Total</b>	<b>2.602,28 €</b>	<b>10.260,54 €</b>	<b>10.260,54 €</b>	<b>10.260,54 €</b>

TABELA 8 - FSES

### 7.2.4. Custos com o Pessoal

A empresa ontem dois trabalhadores permanentes com uma remuneração base mensal de 1000€, com uma taxa de crescimento a 5%, e de um subsidio de alimentação de 100€/mês. A taxa de contribuição para a Segurança Social utilizada é der cerca de 18%.

Gastos com Pessoal	2013	2014	2015	2016
<b>Base de incidência</b>	<b>6.000,00 €</b>	<b>29.400,00 €</b>	<b>30.870,00 €</b>	<b>32.413,50 €</b>
Segurança Social	1.080,00 €	5.292,00 €	5.556,60 €	5.834,43 €
Subsidio de Aliment.	600,00 €	2.400,00 €	2.400,00 €	2.400,00 €
<b>Total</b>	<b>7.680,00 €</b>	<b>37.092,00 €</b>	<b>38.826,60 €</b>	<b>40.647,93 €</b>

TABELA 9 - CUSTOS COM O PESSOAL

## 7.3. Plano de Investimento

A actividade da empresa será sustentada em serviços Web pelo que o investimento inicial irá se apoiar em capitais próprios no valor de 10.000€ conforme indicado nas tabelas que se seguem nos pontos posteriores.

Equipamento	Valor Aquisição	Valor Residual	Vida útil (anos)
<u>2 Mesas</u>	178,00 €	17,80 €	6
<u>2 Cadeiras</u>	78,00 €	7,80 €	3
<u>Móvel Arrumação - Arquivo</u>	349,00 €	34,90 €	6
<u>2 Computadores</u>	1.198,00 €	119,80 €	4
<u>4 Monitores</u>	476,00 €	47,60 €	5
<u>Kit central+telefones</u>	280,00 €	28,00 €	7

TABELA 10 – EQUIPAMENTOS

Investimento/Depreciações	2013	2014	2015	2016
<b>Investimento capital fixo:</b>				
2 Mesas	178,00 €	- €	- €	- €
2 Cadeiras	78,00 €	- €	- €	- €
Móvel Arrumação - Arquivo	- €	349,00 €	- €	- €
2 Computadores	1.198,00 €	- €	- €	- €
4 Monitores	476,00 €	- €	- €	- €
Kit central+telefones	280,00 €	- €	- €	- €
<b>Total de Investimentos Cap. Fixo</b>	<b>2.210,00 €</b>	<b>349,00 €</b>	<b>- €</b>	<b>- €</b>
<b>Depreciações Exercício:</b>				
2 Mesas	26,70 €	26,70 €	26,70 €	26,70 €
2 Cadeiras	23,40 €	23,40 €	23,40 €	23,40 €
Móvel Arrumação - Arquivo		52,35 €	52,35 €	52,35 €
2 Computadores	269,55 €	269,55 €	269,55 €	269,55 €
4 Monitores	85,68 €	85,68 €	85,68 €	85,68 €
Kit central+telefones	36,00 €	36,00 €	36,00 €	36,00 €
<b>Total Depreciações Exercício</b>	<b>441,33 €</b>	<b>493,68 €</b>	<b>493,68 €</b>	<b>493,68 €</b>
<b>Depreciações Acumuladas:</b>				
2 Mesas	26,70 €	53,40 €	80,10 €	106,80 €
2 Cadeiras	23,40 €	46,80 €	70,20 €	93,60 €
Móvel Arrumação - Arquivo	- €	52,35 €	104,70 €	157,05 €
2 Computadores	269,55 €	539,10 €	808,65 €	1.078,20 €
4 Monitores	85,68 €	171,36 €	257,04 €	342,72 €
Kit central+telefones	36,00 €	72,00 €	108,00 €	144,00 €
<b>Total Depreciações Acumuladas</b>	<b>441,33 €</b>	<b>935,01 €</b>	<b>1.428,69 €</b>	<b>1.922,37 €</b>
<b>Investimento Líquido:</b>				
2 Mesas	151,30 €	124,60 €	97,90 €	71,20 €
2 Cadeiras	54,60 €	31,20 €	7,80 €	- 15,60 €
Móvel Arrumação - Arquivo		296,65 €	244,30 €	191,95 €
2 Computadores	928,45 €	658,90 €	389,35 €	119,80 €
4 Monitores	390,32 €	304,64 €	218,96 €	133,28 €
Kit central+telefones	244,00 €	208,00 €	172,00 €	136,00 €
<b>Total Investimento Líquido</b>	<b>1.768,67 €</b>	<b>1.623,99 €</b>	<b>1.130,31 €</b>	<b>636,63 €</b>

TABELA 11 - INVESTIMENTO/DEPRECIACÕES

## 7.4. Necessidade de Fundo de Maneio

A actividade da empresa nos primeiros meses deverá ser sustentada pelos capitais próprios mas rapidamente será recuperada pela previsão inicial abaixo descrita.

Investimento e NFM	2013	2014	2015	2016
<b>Necessidades Cíclicas:</b>				
Clientes (VN)	591,17 €	16.642,48 €	66.523,92 €	151.214,50 €
<b>Total NC</b>	<b>591,17 €</b>	<b>16.642,48 €</b>	<b>66.523,92 €</b>	<b>151.214,50 €</b>
<b>Recursos Cíclicos:</b>				
Fornecedores (CMVMC)	406,66 €	3.360,65 €	11.760,65 €	22.260,65 €
<b>Total RC</b>	<b>406,66 €</b>	<b>3.360,65 €</b>	<b>11.760,65 €</b>	<b>22.260,65 €</b>
<b>NFM=NC-RC</b>	<b>184,51 €</b>	<b>13.281,83 €</b>	<b>54.763,26 €</b>	<b>128.953,85 €</b>
Investimento em NFM	184,51 €	13.097,33 €	41.481,43 €	74.190,59 €

TABELA 12 - INVESTIMENTO E NFM

## 7.5. Viabilidade Económica e Financeira

De acordo com os valores estimados anteriormente e apurado os *cash flows* de cada ano, o VAL obtido para este investimento é de 8.231,33 € , a um custo de oportunidade de 4,13% (valor médio oferecido pelos bancos<sup>1</sup> por depósitos a prazo de 4 anos) [7]. Isto é, o valor gerado pela aplicação de fundos neste investimento, ao fim de 3 anos e 3 meses.

Free Cash Flow	Ano 0	2013	2014	2015	2016
<b>Entrada de fluxos</b>					
EBIT		-10.097,77 €	-34.070,71 €	5.676,12 €	78.045,38 €
IRC		0,00 €	0,00 €	0,00 €	19.511,35 €
EBIT-IRC=EBIT (1-T)		-10.097,77 €	-34.070,71 €	5.676,12 €	58.534,04 €
Depreciações		441,33 €	493,68 €	493,68 €	493,68 €
Imparidades/Provisões		0,00 €	0,00 €	0,00 €	0,00 €
Cash Flow Operacional		-9.656,44 €	-33.577,03 €	6.169,80 €	59.027,72 €
Valor Res. de Continuidade					255,90 €
Valor Res. de Continuidade NFM					128.953,85 €
<b>Total da entrada de fluxos</b>		<b>-9.656,44 €</b>	<b>-33.577,03 €</b>	<b>6.169,80 €</b>	<b>188.237,47 €</b>
<b>Saídas de fluxos</b>					
Investimentos Capital Fixo	5.000,00 €				
Investimento NFM		184,51 €	13.097,33 €	41.481,43 €	74.190,59 €
<b>Total Saídas de fluxos</b>	<b>5.000,00 €</b>	<b>184,51 €</b>	<b>13.097,33 €</b>	<b>41.481,43 €</b>	<b>74.190,59 €</b>
<b>Free cash flow</b>	<b>-5.000,00 €</b>	<b>-9.840,95 €</b>	<b>-46.674,35 €</b>	<b>-35.311,63 €</b>	<b>114.046,88 €</b>
<b>Cash Flow Actualizado</b>	<b>-5.000,00 €</b>	<b>-9.450,64 €</b>	<b>-43.045,38 €</b>	<b>-31.274,48 €</b>	<b>97.001,83 €</b>

TABELA 13 – PREVISÃO DOS CASH FLOWS

A taxa de rentabilidade interna oferecida neste investimento será de 9,15% , ou seja os capitais investidos neste projecto serão valorizados a essa mesma taxa, em detrimento dos

<sup>1</sup> Dados obtidos em 5 de Dezembro de 2012

4,13% oferecidos em investimentos sem risco. O investimento inicial será recuperado ao fim de 3 anos, 10 meses e 13 dias . O índice de rentabilidade indica-nos que, por cada euro investido neste projecto, o mesmo gerará 1,6463 €.

Indicadores (k=4,13%)	Valor
VAL	8.231,33 €
TIR	9,15%
IR	1,6463
PRI	3 anos, 10 meses e 13 dias

TABELA 14 - INDICADORES DE AVALIAÇÃO FINANCEIROS

## Capítulo 8 - Conclusão

Considerando as bases do projecto e os valores nele designados, através dos critérios de análise de investimento, é possível afirmar que este tem viabilidade económica e financeira, com rentabilidades bastante atractivas.

Importa mencionar que a concretização deste projecto de investimento teve como mercado-alvo, o mercado português. Sendo o projecto apresentado neste documento, um serviço *online*, este irá certamente ser utilizado por outros países, desde logo os países de língua oficial portuguesa, e a quando a implementação de novos idiomas no sistema, por os restantes mercados. Isto leva a que as previsões apresentadas, nomeadamente o Volume de Negócios e consequentemente o CMVMC, estejam previstas em baixo. Esta situação pode levar a um aumento dos lucros, reduzindo assim o PRI, e aumentando ainda mais o VAL, TIR e IR, tornando este projecto ainda mais atractivo.

Não nos foi possível fazer estimativas a nível mundial, pela notável complexidade na obtenção dos dados e com a limitação do tempo disponível para o desenvolvimento deste projecto de investimento.

## Capítulo 9 - Referências Bibliográficas

- [1] Loja do Software, «Base de Dados de Portugal - Estatísticas por Categoria» Disponível em: <http://www.lojadosoftware.pt/bases-de-dados-em-excel/dados-de-portugal/estatisticas-por-categoria.php>.
- [2] Instituto Nacional de Estatística, «Estatísticas da Cultura 2011».
- [3] «PORDATA - Música, dança e variedades: sessões e espectadores - Portugal» Disponível em: <http://www.pordata.pt/Portugal/Musica++danca+e+variedades+sessoes+e+espectadores-181>.
- [4] «PORDATA - Espectáculos ao vivo: sessões e espectadores - Portugal» Disponível em: <http://www.pordata.pt/Portugal/Espectaculos+ao+vivo+sessoes+e+espectadores-583>.
- [5] «PORDATA - Ópera: sessões e espectadores - Portugal» Disponível em: <http://www.pordata.pt/Portugal/Opera+sessoes+e+espectadores-182>.
- [6] «PORDATA - Teatro: sessões e espectadores - Portugal» Disponível em: <http://www.pordata.pt/Portugal/Teatro+sessoes+e+espectadores-183>.
- [7] «Melhores depósitos a prazo | Pedro e o Blog» Disponível em: <http://www.pedropais.com/depositos-e-investimentos/melhores-depositos-a-prazo>.

## Capítulo 10 - Anexos

### 10.1. Projecções dos dados de acesso ao portal/serviço

As projecções dos dados de acesso ao portal foram projectadas tendo em conta diversos factores:

1. Número de pesquisas mensais no motor de busca Google (7 480 000) (Figura 1);
2. Resultado das acções promocionais do Plano de Marketing (Página 11);
3. Considerando que cada *user session* equivale a 5 (cinco) *pageviews*;
4. Aumento natural de visitas conseqüente à visibilidade do portal;

	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez	Total
2013	0	0	0	0	0	0	930	1240	1200	930	900	2325	7525
2014	1550	1960	1860	2100	2480	3000	8525	12400	12000	11625	10500	18600	86600
2015	13950	15400	15500	16500	20150	27000	37200	46500	42000	41850	36000	52700	364750
2016	46500	58000	55800	57000	68200	90000	108500	124000	111000	111600	105000	124000	1059600

TABELA 15 - PREVISÃO DAS PAGEVIEWS POR MÊS

	2013	2014	2015	2016
Utilizadores	579	6444	8834	15774
Pageviews	7525	86600	364750	1059600

TABELA 16 - PREVISÃO DAS PAGEVIEWS E UTILIZADORES REGISTRADOS POR ANO

### 10.2. Rendabilidade da Publicidade

Para determinar a rendibilidade da publicidade *online*, foram tentados contactos com várias empresas de afiliados (Google AdSense<sup>1</sup>, NetAffiliation<sup>2</sup>, NetLucro<sup>3</sup>), de forma a obter algumas projecções neste ramo. Infelizmente não foram obtidas quaisquer respostas.

No entanto a Google, possui a uma ferramenta<sup>4</sup> que permite obter uma estimativa do CPM (Custo Por Mil)<sup>5</sup> para *keywords* fornecidas, no qual obtivemos o valor aproximado de 24,74€.

Palavra-chave	Concorrência	Pesquisas globais mensais	Pesquisas locais mensais	CPM Aproximado (Pesquisa)
eventos	Baixa	7 480 000	165 000	24,74 €

FIGURA 1 - RESULTADOS DA KEYWORD "EVENTOS", NO GOOGLE KEYWORD TOOL

<sup>1</sup> <https://www.google.com/adsense>

<sup>2</sup> <http://www.netaffiliation.com/>

<sup>3</sup> <http://www.netlucro.com/>

<sup>4</sup> [https://adwords.google.com/o/Targeting/Explorer?\\_c=1000000000&\\_u=1000000000&\\_o=cues&ideaRequestType=KEYWORD\\_IDEAS](https://adwords.google.com/o/Targeting/Explorer?_c=1000000000&_u=1000000000&_o=cues&ideaRequestType=KEYWORD_IDEAS)

<sup>5</sup> [http://en.wikipedia.org/wiki/Cost\\_per\\_mille](http://en.wikipedia.org/wiki/Cost_per_mille)

### 10.3. Outras Previsões

Material Escritório	2013	2014	2015	2016
<u>Papel (resmas)</u>	2,99 €	11,96 €	11,96 €	11,96 €
<u>Canetas Bic</u>	4,29 €	8,58 €	8,58 €	8,58 €
<u>Tinteiros</u>	65,00 €	390,00 €	390,00 €	390,00 €
<u>Capas</u>	20,00 €	20,00 €	20,00 €	20,00 €
Outros Materiais	50,00 €	50,00 €	50,00 €	50,00 €
<b>Total</b>	<b>142,28 €</b>	<b>480,54 €</b>	<b>480,54 €</b>	<b>480,54 €</b>

TABELA 17 - MATERIAL DE ESCRITÓRIO