



Universidade do Minho

Mestrado em Engenharia Informática – 1º Ano

## Análise e Concepção de Software

### Engenharia de Aplicações Web

#### Relatório do Projeto

Elementos do Grupo:

Luís André Mota Araújo - PG19835  
Rafael Eduardo de Carvalho Pereira - PG19753 (Gestor)  
Rui Alexandre Afonso Pereira - PG20201

Braga, Fevereiro de 2012

# Índice

Introdução .....	7
Contextualização.....	7
Apresentação do Caso de Estudo .....	8
Motivação e Objectivos.....	9
Estrutura do Relatório.....	10
Levantamento e Análise de Requisitos .....	11
Personas .....	11
Estudante.....	11
Adulto .....	12
Cliente.....	14
Análise de Requisitos .....	16
Especificação de Grupos de Utilizadores.....	16
Especificação dos Casos de Uso .....	20
Especificação do Dicionário de Dados.....	36
Especificação Textual e Visual da Camada de Visualização .....	42
Especificação Textual das Vistas da Aplicação Web .....	53
Notas Finais – Levantamento e Análise de Requisitos .....	61
Modelação dos Dados (Entidades e Relacionamentos) .....	63
Metodologia .....	64
Identificação das Entidades .....	64
Identificação dos Atributos.....	65
Identificação dos Relacionamentos .....	66
Construção do Modelo de Dados .....	67
Notas Finais – Entidades e Relacionamentos .....	68
Tecnologias.....	69
Tecnologias de Cliente .....	70
Lógica de Cliente .....	70
AJAX .....	70
Design da Interface de Utilizador.....	71
Tecnologias de Servidor .....	72
Linguagem e Servidor Web.....	72
Servidor de Base de Dados .....	72
Framework.....	73
Desenho da Aplicação Web (Coarse Design).....	74

Vista de Utilizadores Registados.....	77
Vista de Visitantes.....	80
Vista de Gestão de Negócio.....	82
Notas Finais - Coarse Design.....	84
Implementação.....	85
Implementação Webratio .....	85
Componente Funcional .....	86
Componente Não Funcional .....	100
Trabalho Futuro .....	101
Notas Finais - Implementação <i>Webratio</i> .....	103
Implementação PHP.....	104
Framework <i>Yii</i> .....	105
IDE – Netbeans.....	108
Componente Funcional .....	109
Componente Não Funcional .....	125
Publicação da Página Web.....	128
Notas Finais – Implementação PHP .....	130
Conclusão e Trabalho Futuro.....	131
Bibliografia .....	132
Referências Web.....	133
Lista de Siglas e Acrónimos .....	134
Anexos.....	135
Relação entre os Grupos de Utilizadores .....	136
Diagramas de Atividade dos Casos de Uso.....	137
Sign-Up .....	137
Login .....	137
Logout.....	138
Ver Eventos Por Mapa .....	139
Ver Eventos Favoritos.....	139
Pesquisar Multicritérios.....	139
Confirmar Endereço de Email .....	140
Comentar Evento .....	140
Marcar Evento Como Favorito.....	141
Desmarcar Favorito .....	141
Criar Evento .....	141

Editar Evento.....	142
Solicitar Aparecimento Na Próxima Newsletter .....	142
Enviar Newsletter.....	143
Ver Evento Rápido.....	143
Ver Evento Detalhado .....	143
Navegar No Mapa .....	144
Modelo Entidades e Relacionamentos - CHEN .....	146
Modelo de Dados – Webratio.....	147
Diagrama de Entidades Relacionamentos – Mysql.....	148
Código SQL – Base de Dados Eventhis .....	149
Código Javascript – Google Maps – Webratio .....	156
Código do Template Power Index de Comentários – Webratio .....	158
Screenshots da Aplicação Desenvolvida com o Webratio.....	163
Manual de Utilizador da Aplicação Final .....	168
Nota Introdutória ao Manual.....	168
Alterar Perfil.....	169
Criar Evento .....	171
Gerir Eventos .....	173
Alterar Evento .....	174
Gerir Comentários .....	177
Gerir Newsletter.....	178
Gerir Utilizadores .....	179
Visualizar Página de Evento .....	181
Visualizar Página Inicial .....	183
Página Inicial vista de um Utilizador Visitante.....	184
Fazer Pesquisa.....	185
Ver Eventos Favoritos.....	186
Ver os Meus Eventos .....	187

## Índice de Figuras

Figura 1 Representação gráfica da Página principal para Guests.....	43
Figura 2 Representação gráfica da Página principal para Utilizadores registados. ....	45
Figura 3 Página que permitirá aos utilizadores registados inserirem eventos no sistema.....	47
Figura 4 Página que permitirá aceder todas as informações que o sistema disponibiliza relativamente a um evento. ....	49
Figura 5 Página que permitirá a um Guest registar-se no sistema.....	51
Figura 6 Logótipo da ferramenta de desenvolvimento de Aplicações Web orientadas aos dados usada para o desenvolvimento da aplicação Web <i>Eventhis</i> .....	85
Figura 7 Página de Pesquisa de Eventos. ....	86
Figura 8 Página de Autenticação. ....	87
Figura 9 Área de Registo de Utilizadores. ....	88
Figura 10 Área de Visualização de Eventos. ....	89
Figura 11 Página de (Eventos) Favoritos. ....	90
Figura 12 Área de Edição de Registo.....	91
Figura 13 Área de Visualização de Eventos. ....	93
Figura 14 Área de Gestão de Eventos. ....	94
Figura 15 Páginas da Vista de Administrador.....	95
Figura 16 Serviço Web (REST) que permite retornar os eventos para uma área indicada. ....	98
Figura 17 XML <i>Out Unit</i> que extrai os eventos de uma determinada área usando uma <i>Seletor Unit</i> e converte para XML.....	98
Figura 18 Exemplo de pedido e output XML do serviço Web. ....	99
Figura 19 Resultados que advém do serviço Web. ....	99
Figura 20 Alteração do Estilo para a introdução de imagens nos comentários. ....	100
Figura 21 Exemplo de seleção de coordenada geográfica para a realização de um evento....	101
Figura 22 Arquitetura do Framework Yii.....	106
Figura 23 IDE - NetBeans com a biblioteca <i>Yii</i> incluída em Include Path. ....	108
Figura 24 Componente de autenticação. ....	109
Figura 25 Código necessário para a utilização do componente de autenticação. ....	109
Figura 26 Menu da aplicação na vista de utilizadores autenticados. ....	111
Figura 27 Menu da aplicação para a vista de Administradores.....	112
Figura 28 Código do componente googlemap. ....	113
Figura 29 Código necessário para o pedir um mapa estático (imgmap). ....	115
Figura 30 Código necessário para realizar um pedido de morada. ....	116
Figura 31 Resultado da aplicação do UtilGeocode e mapas estáticos. ....	116
Figura 32 Código que permite desenhar o mapa na página principal, sendo especificadas todas as opções, desde o tamanho mapa aos eventos que se pretendem registar. ....	118
Figura 33 Método do controller <i>EventoController</i> que realiza a pesquisa dos eventos não terminados da área provida e devolve o resultado em JSON. ....	118
Figura 34 Função Javascript que é chamada sempre que as fronteiras do mapa mudem. ....	119
Figura 35 Resultado da combinação do mapa com serviço Web, Javascript e AJAX.....	120
Figura 36 Configuração final das marcas, eventos e janela de informação. ....	120

Figura 37 Definição dos componentes do formulário de pesquisa como active ( <i>TextField</i> e <i>DropDownList</i> ). ....	121
Figura 38 Código Javascript que realiza pesquisa AJAX. ....	121
Figura 39 Formulário de pesquisa AJAX. ....	122
Figura 40 Utilização do widget timepicker para suportar a inserção da data de início de um evento. ....	123
Figura 41 Resultado final da utilização do widget timepicker. ....	123
Figura 42 Página de gestão de comentários. ....	124
Figura 43 Código que permite a apresentação de um comentário. ....	126
Figura 44 Código que permite listar os comentários. ....	126
Figura 45 Exemplo de um comentário. ....	127
Figura 46 Configuração da base de dados remota. ....	128
Figura 47 Estrutura remota após upload para o endereço remoto. ....	128
Figura 48 Página de Autenticação. ....	163
Figura 49 Página que permite ver um evento detalhado para utilizadores autenticados. ....	164
Figura 50 Página de Criação de Evento. ....	165
Figura 51 Página de Edição de Evento. ....	165
Figura 52 Página Favoritos. ....	166
Figura 53 Página de Registo na Aplicação Web. ....	166
Figura 54 Página principal para utilizadores registados. ....	167
Figura 55 Página principal para Visitantes. ....	167

## **Introdução**

### **Contextualização**

A internet possibilitou-nos, desde o seu início, uma forma única de exposição de informação visível para todo o mundo. Neste contexto começaram a surgir várias formas de publicidade, nas quais se inserem publicidade a restaurantes, publicidade a museus, publicidade a vários eventos como festas noturnas, feiras, encontros, entre outros.

Apesar de todas as informações acima descritas já se encontrarem disponibilizadas no mundo da internet, estas surgem de uma forma muito dispersa e, quando surgem visíveis aos navegadores, aparecem muitas vezes descontextualizadas com a localização normal do mesmo. É deste modo que surge a necessidade de agregar vários locais/eventos num só sistema para, desta forma, combater a dispersão de informação que existe por toda a internet. Além de tal, a informação será preferencialmente mostrada ao utilizador conforme a sua zona predefinida, para assim mostrarmos sempre informações que melhor se identifiquem com o utilizador do nosso sistema.

## Apresentação do Caso de Estudo

O nosso sistema procurará então expor informação aos seus utilizadores de uma forma contextualizada e, além de tal, permitirá ainda que possam ser feitas várias pesquisas multicritério de forma a, assim, proporcionar ao nosso utilizador procurar os locais/eventos que mais lhe interessem.

Desta forma, a nossa aplicação Web agregará vários eventos que poderão ser criados pelos utilizadores registados e que deverão ser sempre rotulados por um título, uma descrição, uma localização GPS, uma data de início e uma data de fim. Um dado evento será ainda etiquetado por TAG's (tipos) que deverão descrever do que se trata tal evento inequivocamente. A consulta de tais eventos e sua visualização descritiva poderá ser feita por todos os utilizadores quer registados, quer não registados.

Além de tal, todos os eventos poderão ser comentados pelos utilizadores registados que, assim, poderão deixar o seu feedback para que todos os outros utilizadores possam ler. Poderão ainda, sempre que queiram, confirmar ou não que irão a tal evento ou local ou ainda escolher tal evento para a sua lista preferencial.

O nosso sistema permitirá ainda que sejam feitas pesquisas multicritérios de forma a puder disponibilizar informações ainda mais adequadas ao que o utilizador procura.

É objectivo expandir os nossos serviços a uma larga área geográfica, procurando assim aumentar o espectro do nosso público-alvo tendo, assim, como objectivo primordial ao nível do modelo de negócio, aumentar os nossos potenciais ganhos.

## Motivação e Objectivos

O sistema idealizado parece inovador e aliciante para a população geral e que, assim, consegue encontrar espaço no mercado atual das aplicações Web.

O produto distingue-se por agregar vários locais/eventos de interesse num só portal e, assim, permitir aos utilizadores do mesmo encontrar toda a informação que precisam num só domínio. Além de tal, algo muito característico do sistema e que o diferencia de todos os outros, é o facto de mostrar preferencialmente informação adequada à localização do utilizador e permitir que este possa fazer várias pesquisas consoante os critérios que mais lhe interessem.

## Estrutura do Relatório

Este documento é suporte ao projeto de engenharia web desenvolvido ao longo de um semestre para o módulo de engenharia Web, em que foi desenvolvida uma aplicação relativa à gestão de divulgação de eventos.

Em primeiro lugar é documentado o processo de levantamento de requisitos, embora as tecnologias tenham sido escolhidas primeiro, achamos que tal é um erro, pois as tecnologias só devem ser escolhidas depois de se conhecer o produto que se vai construir. Aqui, são identificadas as características pretendidas para o produto. A secção Modelação dos dados, também se pode considerar análise dos requisitos, pois essa modelação inclui um estudo das entidades envolvidas e os seus relacionamentos.

De seguida, aborda-se e documentam-se as escolhas das tecnologias utilizadas para o desenvolvimento do projeto, tendo sido escolhido a linguagem, o framework, as tecnologias de cliente e de servidor (apache, mysql, etc).

O desenho da aplicação Web vem fechar a componente de análise da aplicação a desenvolver-se, sendo a última anterior à implementação da aplicação.

Por fim, são apresentadas e documentadas as duas aplicações Web desenvolvidas, sendo explicados as decisões mais importantes e as alterações mais determinantes.

Para além das secções relativas ao desenvolvimento do produto existem as secções de conclusão, bibliografia, referências Web, lista de siglas e acrónimos e os anexos.

Algo que transversal, é a existência de uma subsecção de Notas Finais em jeito de conclusão por cada secção deste documento, o que é muito relevante, pois permite obter o sumo da secção em causa numa leitura rápida.

# Levantamento e Análise de Requisitos

## Personas

Existem muitas técnicas de levantamento de requisitos, como as entrevistas, objectivos, observação directa dos sistemas, personas e muitas outras, no entanto a técnica que melhor se adequa ao projecto que está ser desenvolvido é a personas, uma vez que não há utilizadores alvo disponíveis e é uma aplicação para as massas.

A técnica personas consiste em inventar personagens potencialmente interessadas no produto e posteriormente tentar extrair requisitos através da análise das mesmas. Foi esta a estrutura seguida, para cada uma das personas fizemos uma descrição, usando inclusive, uma imagem, posteriormente fizemos uma análise da persona do ponto de vista do sistema.

Conhecendo os objectivos do sistema que está a ser desenvolvido, é fácil perceber que é um sistema sem barreiras etárias, tendo como único requisito para a sua normal utilização conhecimentos básicos ao nível das novas tecnologias. Por exemplo, um idoso pode não estar interessado em festas nocturnas, mas pode usar o sistema para procurar feiras ou outros eventos mais adequados às suas preferências.

Assim, foi decidido escolher três personas, um estudante, um pai de família, e um cliente, esta última para tentar explorar a componente de negócio do sistema.

### Estudante

- Tiago Martins é um jovem de 21 anos e é solteiro;
- O Tiago estuda História na Universidade de Aveiro, mas a sua residência é em Vila Real;
- Participa frequentemente na praxe;
- Gosta de actividades radicais;
- Sempre que pode, sai com os amigos a noite;
- Adora conhecer novas pessoas e conviver com toda gente;



- A informática faz parte do seu dia-a-dia;
- Gosta de ser informado sobre novos eventos e festas na sua zona, para poder participar;

### O Tiago da perspectiva do Sistema

**O Tiago, como a maioria dos jovens, prefere eventos e festas virados para a sua idade, como festas Universitárias, discotecas, concertos, e desportos/actividades radicais. Por isso, este grupo de utilizadores tem uma necessidade de poder pesquisar eventos por tipos como os referidos anteriormente.**

**Estas razões levaram o sistema incluir tipos para catalogar os eventos, permitindo que os utilizadores encontrem eventos que se enquadrem mais no seu perfil com uma maior facilidade.**

**Como se verifica, o Tiago, é um estudante deslocado, pelo que quando está na universidade pretende ter acesso a informações acerca de eventos que se realizem nas proximidades de Aveiro, no entanto quando está de fim-de-semana ou de férias os eventos que mais lhe poderão interessar estão nas proximidades de Vila Real.**

**Para colmatar a necessidade identificada no parágrafo anterior, o sistema permitirá que o Tiago, ou qualquer outro utilizador em condições semelhantes, possam alterar a sua localização pré-definida com facilidade, podendo os utilizadores ter acesso mais facilitado a informação relativa aos eventos que mais lhe interessam.**

### Adulto

- António João é um chefe de família com 42 anos;
- A sua família é constituída pela sua mulher e pelos seus dois filhos, a Lara de 17 anos e o Diogo de 16;
- Esta família, com o ordenado do Sr. António e da sua mulher, enquadrar-se na classe económica média-alta;
- É trabalhador na indústria da restauração e



costuma dispensar muito do seu tempo livre a fazer atividades com a sua família;

- Algo que lhe dá muito prazer fazer é, pelo menos uma vez por semana, ir jantar fora com os seus filhos e a sua mulher, procurando sempre conhecer novos restaurantes;
- Os seus filhos começam agora a interessar-se por eventos sociais onde possam desfrutar de bons momentos com os seus amigos;
- A família do Sr. João procura, sempre que pode, passar férias todos juntos, viajando e conhecendo, cada vez melhor, o seu país.

#### O Sr. António da perspectiva do Sistema

O Sr. António é o típico homem português que gosta de comer bem e, como trabalhador da indústria de restauração que é, gosta de conhecer novos espaços destinados para tal. É aqui que se enquadra o motivo pelo qual o nosso sistema seria muito útil para este chefe de família. O Sr. António poderia assim utilizar o motor de busca por critérios do nosso sistema onde, assim, poderia encontrar restaurantes perto da sua zona. Além de tal, poderá ainda ponderar a sua decisão de escolha de restaurante lendo a descrição do mesmo e, ainda, lendo os comentários, isto é, os feedback's deixados por outros utilizadores do sistema.

O nosso sistema seria ainda útil para esta família quando, a mesma, está a passar férias e procura quer sítios onde comer, quer eventos que possam visitar. É aqui, também, que surge a importância do nosso sistema ser algo que abranja uma boa área de localizações.

Uma vez que os filhos deste casal estão agora a atravessar o início da sua juventude começam, também eles, a interessar-se por locais/eventos onde possam passar alguns momentos de diversão com os seus amigos. Desta forma, destaca-se a possibilidade que o nosso sistema oferece de possibilitar a pesquisa e consulta de eventos (temporários) que estejam marcados perto da sua zona preferencial (provavelmente a sua zona de residência). Além de tal, durante as suas férias com os pais, poderão também eles pesquisar eventos perto da zona onde estejam para, assim, também eles, terem algum tempo de diversão específico para os mesmos.

É de salientar, por fim, em relação ao que foi dito acima, que a grande diversidade de tipos de eventos que serão disponibilizados pelo nosso sistema permitirão, desta forma,

**alcançar uma maior gama de público-alvo e, portanto, um maior número de potenciais clientes para utilizar o nosso sistema.**

### **Cliente**

- Júlio Vieira é um homem de 41 anos e é viúvo.
- Trabalha no departamento de marketing de uma empresa que comercializa bebidas, sendo que neste momento ocupa o cargo de diretor;
- É muito trabalhador e brioso.
- Gosta de usar as novas tecnologias profissionalmente e para lazer, não tendo dificuldade em fazê-lo;
- Gosta de saber qual é o efeito que as suas campanhas estão a ter em tempo real, pois tal permite-lhe planear o trabalho futuro com mais facilidade e fiabilidade;
- De momento está interessado em organizar eventos abertos ao público, para divulgar os novos produtos da sua empresa;



### **O Cliente da perspectiva do Sistema**

**O Sr. Júlio é um cliente tipo do sistema, pois precisa de divulgar eventos ao público em geral e gosta de usar tecnologias, pelo que se sentirá à vontade usando uma aplicação Web para o auxiliar a divulgar os eventos que fazem parte da estratégia de marketing escolhida.**

**A funcionalidade que permite solicitar, no ato da criação de um evento, que o evento surja na próxima *newsletter* do sistema é útil para um marketer, pois permite publicitar de uma forma simples os eventos que criar, pelo que tal é útil para o Sr. Júlio.**

**Este cliente tem uma necessidade que obriga a estender um pouco as funcionalidades identificadas para o sistema, por exemplo, daria jeito ao Sr. Júlio saber quantas vezes que o seu evento já foi visto pelos utilizadores da aplicação, pois tal possibilitaria que o mesmo tivesse um feedback do seu trabalho muito mais rápido.**

Desta persona infere-se por isso que devemos manter visível um contador por evento do sistema, com o objectivo de contar o número de visitas, essa informação já seria útil, mas a verdade é que tal não seria totalmente eficaz, pois mesmo assim o Sr. Júlio não tem como saber quantas pessoas se esperam no evento, pelo que para tal achamos melhor inserir mais um contador, que terá a cada momento número de utilizadores que estão a pensar ir ao evento.

Nestas condições o sistema já seria bastante útil ao Sr. Júlio, pois as suas necessidades profissionais, como a divulgação e controlo do resultado da estratégia escolhida, já estariam devidamente complementadas.

## Análise de Requisitos

A análise de requisitos para uma aplicação Web não é igual à análise de requisitos para uma aplicação monolítica, sendo que, nas aplicações Web é necessário ter uma preocupação especial em identificar grupos de utilizadores e ter um cuidado especial ao nível da interface gráfica, pois nos dias que correm é crítico para o sucesso de uma aplicação Web que a mesma tenha um aspecto, interatividade e usabilidade bastante evoluídos.

Para além das necessidades especiais identificadas, as aplicações Web tem algo de semelhante relativamente com as aplicações monolíticas, como por exemplo a necessidade de se identificarem as funcionalidades a proporcionar e a necessidade de se desenvolver um dicionário de dados (modelo de domínio), mas mesmo nestas necessidades transversais a quase todos os tipos de aplicações, é necessária uma abordagem adaptada à Web. Por exemplo, depois de identificar um caso de uso é necessário referir os grupos de utilizadores que o poderão usar.

Surgindo como a primeira fase de desenvolvimento de Software, a análise de requisitos permitiu estruturar e identificar as funcionalidades que o sistema proporcionará aos utilizadores. Abaixo serão especificados os grupos de utilizadores que o sistema terá, estando implícita uma análise comparativa entre os diferentes grupos.

## Especificação de Grupos de Utilizadores

Neste ponto, como acima já foi indicado, serão expostas as principais características e diferenças entre os diferentes grupos de utilizadores do sistema. Estes grupos são constituídos pelo Guest, Utilizador Registado, Cliente e Administrador. Ao longo desta estruturação e especificação, foram diferenciados tais grupos consoante as suas funcionalidades disponíveis, os seus direitos de acesso e, ainda, será feita uma descrição acerca do que cada grupo representa para o nosso sistema.

Grupo	
<b>Nome:</b>	Guest
<b>Descrição:</b>	Um utilizador não registado e pretende utilizar os serviços públicos
<b>Dados do perfil:</b>	
<b>Super Grupos:</b>	
<b>Sub Grupos:</b>	Utilizador Registado
<b>Funcionalidades acessíveis:</b>	Sign-Up, Ver Eventos no Mapa, Ver lista de Eventos, Navegar Mapa, Ver Evento Rápido, Ver Evento Detalhado, Pesquisar com multi-critérios
<b>Direitos de Acesso:</b>	Acede em modo de leitura aos eventos e comentários
- <b>Modo de Leitura</b>	Evento, Tipo e Comentário
- <b>Modo de Escrita</b>	

Grupo	
<b>Nome:</b>	Utilizador Registado
<b>Descrição:</b>	É um utilizador que já se registou no sistema e pode guardar preferências e outros registo de actividade no sistema.
<b>Dados do perfil:</b>	Nome completo, E-Mail, localização GPS pré-definida, foto (opcional) e password
<b>Super Grupos:</b>	Guest
<b>Subgrupos:</b>	Cliente, Administrador
<b>Funcionalidades acessíveis:</b>	Logout, Confirmar E-Mail, Criar Evento, Marcar como Favorito, Desmarcar Favorito, Ver eventos marcados como Favoritos, Comentar Evento
<b>Direitos de Acesso:</b>	
-Modo de Leitura	Evento, Tipo e Comentário
-Modo de Escrita	Utilizador

Grupo	
<b>Nome:</b>	Cliente
<b>Descrição:</b>	É um utilizador que já se registou no sistema e pode guardar preferências e outros registo de actividade no sistema.
<b>Dados do perfil:</b>	Nome completo, E-Mail, localização GPS pré-definida, foto (opcional) e password
<b>Super Grupos:</b>	Utilizador Registado
<b>Subgrupos:</b>	
<b>Funcionalidades acessíveis:</b>	Solicitar aparecimento na próxima newsletter
<b>Direitos de Acesso:</b>	
-Modo de Leitura	Evento, Tipo e Comentário
-Modo de Escrita	Utilizador

Grupo	
<b>Nome:</b>	Administrador
<b>Descrição:</b>	É um utilizador do sistema com direitos especiais, podendo gerir toda a aplicação
<b>Dados do perfil:</b>	Nome completo, E-Mail, localização GPS pré-definida, foto (opcional) e password
<b>Supergrupos:</b>	Utilizador Registado
<b>Subgrupos:</b>	
<b>Funcionalidades acessíveis:</b>	Enviar newsletter
<b>Direitos de Acesso:</b>	
- <b>Modo de Leitura</b>	Utilizador, Evento, Tipo, Comentário, Utilizador e Newsletter
- <b>Modo de Escrita</b>	Utilizador, Evento, Tipo, Comentário, Utilizador e Newsletter

No anexo 1 presente um esquema que ajuda a compreender a relação entre os grupos de utilizadores descritos acima.

## Especificação dos Casos de Uso

Enquadrando-se na especificação de requisitos, os casos de uso textuais surgem para manifestar sequências de mensagens intercambiáveis que existirão entre o nosso sistema e os vários atores/clientes do mesmo.

Desta forma, descreveremos vários casos de uso relativos às funcionalidades principais que o nosso sistema procurará disponibilizar. Destas, descreveremos abaixo o respectivo caso de uso referente às ações de sign-up, ver eventos no mapa, ver lista de eventos, navegar no mapa, ver evento rápido, ver evento detalhado, pesquisar por multicritério, logout, confirmar endereço de email, criar evento, marcar como favorito, desmarcar favorito, ver eventos marcados como favorito, comentar evento, ver comentários de um dado evento, solicitar aparecimento na próxima newsletter e enviar newsletter.

Como não poderia deixar de ser, foi registado o propósito e pré-condição, quando existente, para cada um dos casos de uso descritos.

Caso de Uso	
Nome:	Sign-Up
Propósito:	Para permitir que um utilizador possa registar-se no nosso sistema.
Pré-Condição:	
Cenário Principal:	
1	O utilizador escolhe a opção Sign-Up
2	É disponilizado o formulário a preencher pelo utilizador
3	Preenche os dados do formulário
4	Envia string de confirmação para o email
5	Confirma o registo com a string de confirmação

Caso de Uso	
Nome:	Ver eventos no Mapa
Propósito:	Permitir ao utilizador ver todos os locais e eventos do nosso sistema num mapa geográfico
Pré-Condição:	
Cenário Principal:	
1	Utilizador seleciona a "Vista por Mapa".
2	Sistema apresenta mapa com representação dos eventos/locais referentes disponíveis na área envolvente ao ponto definido como predefinido pelo utilizador.

Caso de Uso	
Nome:	Navegar Mapa
Propósito:	Para permitir ao utilizador navegar no mapa podendo, assim, visualizar todos os pontos de interesse na zona do mapa onde está a navegar
Pré-Condição:	
Cenário Principal:	
1	Utilizador selecciona a "Vista por Mapa"
2	Sistema apresenta mapa com representação dos eventos/locais próximos do local pré-definido pelo utilizador, quando este se trata de um utilizador registado
3	Utilizador vai navegando pelo mapa geográfico
4	Sistema actualiza mapa com representação dos seus eventos/locais para a zona que está no momento a ser visualizada pelo utilizador

Caso de Uso	
Nome:	Ver evento rápido
Propósito:	Para um utilizador poder ver alguns detalhes de um evento
Pré-Condição:	
Cenário Principal:	
1	O Utilizador clica num evento do mapa
2	O sistema identifica o evento, reúne informação acerca do evento e actualiza página, mostrando mais alguma informação acerca do roteiro

Caso de Uso	
Nome:	Ver Evento Detalhado
Propósito:	Para permitir que um utilizador possa visualizar a informação total de um dado evento.
Pré-Condição:	Um utilizador precisa de ter seleccionado um dado evento no mapa.
Cenário Principal:	
1	O utilizador solicita ver detalhes do evento seleccionado
2	É apresentado ao utilizador uma nova página, com os detalhes completos de um evento (i.e Descrição, Local, Data, Comentários)

Caso de Uso	
Nome:	Pesquisar com multi-critérios
Propósito:	Para um utilizador poder encontrar eventos ao seu gosto utilizando vários filtros de pesquisa
Pré-Condição:	
Cenário Principal:	
1	O utilizador recebe uma caixa de pesquisa e as opções possíveis
2	O utilizador escreve na caixa de pesquisa o que pretende pesquisar e escolhe um ou mais critérios de pesquisa
3	Se existir eventos com esses critérios, é actualizado o mapa e lista com estes novos eventos

Caso de Uso	
Nome:	Log-Out
Propósito:	Para um utilizador poder encerrar a sua sessão como utilizador registado
Pré-Condição:	O Utilizador deve estar autenticado na aplicação
Cenário Principal:	
1	O Utilizador escolhe efectuar o Log-Out
2	Após do sistema encerrar a sua sessão de utilizador registado, é apresentada a página para os guests

Caso de Uso	
Nome:	Confirmar endereço de email
Propósito:	Para permitir ao utilizador confirmar que forneceu um endereço de email correcto no acto de registo
Pré-Condição:	O utilizador deve ter realizado o pré-registo na aplicação Web à menos de 48 horas e o endereço de email não deve ter sido ainda confirmado
Cenário Principal:	
1	O utilizador abre a página de verificação de email, usando os parâmetros correctos
2	Se é identificado o pré-registo através dos parâmetros fornecidos pelo utilizador, o processo de registo está concluído com sucesso
3	É apresentada a página de boas-vindas para novos utilizadores

Caso de Uso	
Nome:	Criar Evento
Propósito:	Para um utilizador poder criar o seu evento
Pré-Condição:	O Utilizador precisa de ter efectuado o Log-In anteriormente
Cenário Principal:	
1	O Utilizador escolhe criar um novo Evento
2	É apresentado ao Utilizador um formulário
3	O Utilizador após preencher o formulário com os dados necessários para criar um evento
4	O utilizador submete o formulário
5	Se o formulário é válido, é apresentada ao utilizador a página do evento que acabou de criar

Caso de Uso	
Nome:	Marcar como Favorito
Propósito:	Permitir ao utilizador marcar um evento como Favorito
Pré-Condição:	O utilizador deve estar autenticado e a ver os detalhes de um evento
Cenário Principal:	
1	O utilizador identifica o evento que pretende marcar como favorito
2	O utilizador solicita ao sistema que o evento identificado seja marcado como favorito
3	O sistema identifica o utilizador e o evento e o utilizador, procedendo à marcação
4	O sistema actualiza a página do cliente, demonstrando o sucesso da operação

Caso de Uso	
Nome:	Desmarcar Favorito
Propósito:	Permitir ao utilizador retirar marca de favorito de um evento que tinha marcado previamente
Pré-Condição:	O utilizador deve estar autenticado e capaz de identificar o evento marcado como favorito
Cenário Principal:	
1	O Utilizador identifica o evento que pretende desmarcar como favorito
2	O utilizador solicita ao sistema que o evento identificado seja desmarcado
3	O sistema identifica o evento e o utilizador, procedendo à desmarcação
4	O sistema actualiza a página deixando demonstrado o sucesso da operação

Caso de Uso	
Nome:	Ver eventos marcados como Favoritos
Propósito:	Permitir ao utilizador aceder aos eventos que já marcou como favoritos com maior facilidade
Pré-Condição:	O utilizador deve estar autenticado, permitindo ao sistema que o identifique
Cenário Principal:	<p>1 O utilizador solicita lista dos seus eventos favoritos</p> <p>2 O sistema identifica o utilizador e os seus eventos favoritos</p> <p>3 O sistema actualiza a página deixando demonstrado o sucesso da operação</p>

Caso de Uso	
Nome:	Comentar Evento
Propósito:	Possibilitar aos utilizadores registados que tornem públicas as duas opiniões acerca dos eventos disponíveis no sistema
Pré-Condição:	O utilizador deve estar autenticado e na página detalhada de um evento
Cenário Principal:	
1	O utilizador recebe um formulário solicitando-lhe um título e um corpo para o comentário
2	O utilizador preenche o formulário e submete-o
3	O sistema identifica o evento e o utilizador, fazendo o registo do comentário
4	A página da vista detalhada do evento é actualizada, sendo verificável o comentário

Caso de Uso	
Nome:	Ver comentarios
Propósito:	Permitir ao utilizador ver os comentários de um dado evento
Pré-Condição:	O utilizador deve estar autenticado, permitindo ao sistema que o identifique; O utilizador deve estar na página do evento
Cenário Principal:	
1	O sistema identifica evento a ser visualizado
2	O sistema reúne comentários do evento
3	O sistema apresenta comentários dos eventos

Caso de Uso	
Nome:	Solicitar aparecimento na próxima newsletter
Propósito:	Possibilitar aos utilizadores que publicitem um evento recém-criado na newsletter do sistema
Pré-Condição:	O utilizador deve ter acabado de criar um evento e estar autenticado
Cenário Principal:	
1	O utilizador recebe um formulário perguntando-lhe se pretende que o evento que criou saia na próxima newsletter do sistema
2	O utilizador pretende publicitar o seu evento
3	O utilizador recebe uma página de pagamento
4	O utilizador paga
5	O sistema regista dados da operação e reencaminha o utilizador para a página do evento que acabou de publicitar

Caso de Uso	
Nome:	Enviar newsletter
Propósito:	Para o sistema poder enviar a newsletter semanal para os utilizadores
Pré-Condição:	
Cenário Principal:	
1	O sistema reúne todos os eventos que serão publicitados na próxima newsletter
2	O sistema constrói correio electrónico para enviar newsletter
3	O sistema envia correio electrónico com a newsletter para todos os utilizadores que o permitem

Caso de Uso	
Nome:	Remover Evento
Propósito:	Um utilizador pode pretender eliminar um evento criado porque o mesmo foi cancelado;
Pré-Condição:	O utilizador deve estar autenticado, permitindo ao sistema que o identifique; O utilizador deve estar na página de um evento que tenha sido criada por si
Cenário Principal:	
1	O sistema identifica o evento a remover.
2	O sistema remove registo do sistema

Caso de Uso	
Nome:	Editar Evento
Propósito:	Um utilizador pode pretender editar um evento que tenha criado;
Pré-Condição:	O utilizador deve estar autenticado, permitindo ao sistema que o identifique; O utilizador deve estar na página de um evento que tenha sido criado por si
Cenário Principal:	<ol style="list-style-type: none"> <li>1 O sistema identifica o evento a editar.</li> <li>2 O utilizador recebe uma página de edição de evento</li> <li>3 O utilizador altera o evento</li> <li>4 O sistema regista alterações ao evento</li> </ol>

Caso de Uso	
Nome:	Alterar Registo
Propósito:	Um utilizador pretende alterar a informação de registo
Pré-Condição:	O utilizador deve estar autenticado, permitindo ao sistema que o identifique
Cenário Principal:	<ol style="list-style-type: none"> <li>1 O sistema identifica o utilizador que pretende alterar a informação de registo</li> <li>2 O utilizador recebe um formulário de edição de registo</li> <li>3 O utilizador altera a informação do registo</li> <li>4 O sistema regista alterações à informação de registo do utilizador em causa</li> </ol>

## Especificação do Dicionário de Dados

Esta secção é muito importante, pois é aqui que são identificados os conceitos que serão necessários para o desenvolvimento do sistema, identificando os atributos dos mesmos e as suas relações, para além disso através da definição dos sinónimos dos conceitos estabelecemos o vocabulário permitido ao longo do projeto de forma bastante simples.

Foi elaborado um modelo conceptual (*Chen*) dos dados que vamos usar no desenvolvimento do sistema, tal foi muito útil, pois identificamos de forma simples as entidades (objetos), atributos e relações entre entidades. Para além do referido, os modelos conceptuais ainda podem ser usados para gerar de forma automática a base de dados, o que será muito útil numa fase posterior do projeto. O modelo conceptual é o anexo 3 deste relatório.

Objecto	
Nome:	Evento
Sinónimos:	Nenhum.
Descrição:	Um acontecimento físico bem definido espacial e temporalmente.
Exemplos:	Rock In Rio, Magusto do Cesium etc.
Propriedades	
Título:	Frase de resumo de um evento
Descrição:	Explicação completa acerca do conteúdo do evento
Tipo:	Palavra ou frase que transmite uma ideia do conteúdo do evento, como por exemplo desportivo, festivo ou noite.
Início:	Data e hora de início do evento

	Fim: Data e hora de final do evento
LocalGPS:	Coordenadas latitude longitude que identificam o local onde o evento se realizará.
Relações	
EventoComentárioTem	Relaciona o evento com os comentários realizados pelos utilizadores relativamente aos mesmos
EventoUtilizadorMarca	Um utilizador pode ser marcado como favorito por um utilizador registado, no entanto tal é opcional
EventoUtilizadorCria	Um utilizador pode criar um ou mais eventos
EventoNewsletterTem	Uma newsletter é composta por vários eventos
Componentes	
Comentários:	Lista de comentários escritos pelos utilizadores relativos ao evento.

Objecto	
Nome:	Comentário
Sinónimos:	Nenhum.
Descrição:	O registo de uma opinião de um utilizador registado da aplicação
Exemplos:	Praticamente todos os jornais permitem que os utilizadores escrevam comentários acerca das notícias, este objecto é algo muito semelhante
Propriedades	
Título:	Frase de resumo de um comentário
Texto:	Opinião de um utilizador registado relativamente a um evento disponível no sistema
Autor:	O utilizador registado que escreveu o comentário
DateTime:	Data e hora em que o comentário foi realizado
Relações	
EventoComentárioTem	Relaciona o evento com os comentários realizados pelos utilizadores relativamente aos mesmos

Objecto	
Nome:	Utilizador
Sinónimos:	Nenhum.
Descrição:	O registo de um utilizador registado da aplicação
Propriedades	
Local GPS:	Local escolhido pelo utilizador sendo como o seu local de defeito. Usado como "home page" para o mapa
Foto:	Foto escolhido pelo utilizador como forma de o representar
Nome:	Nome do utilizador registado
Hpassword:	Password do utilizador registado
E-Mail:	E-Mail do utilizador registado
Relações	
UtilizadorComentárioCria	Um utilizador cria um ou mais comentários relativos a um evento
EventoUtilizadorCria	Um utilizador pode criar um ou mais eventos
UtilizadorEventoMarca	Um utilizador pode marcar um ou mais eventos como favoritos
Componentes	
Comentários:	Lista de comentários escritos pelos utilizadores relativos a um evento.
Eventos:	Lista de eventos criados pelos utilizadores relativos ao evento.

Objecto	
Nome:	Tipo
Sinónimos:	Nenhum
Descrição:	Forma de melhor identificar o que um dado evento oferece ao seu potencial utilizador
Exemplos:	Desporto, Festa, Feira, ...
Propriedades	
Nome:	Titulo descritivo do que representa um dado evento (exemplo: Noite, Comida, ...)
Relações	
TipoEventoCataloga	Um tipo deverá ser capaz de catalogar inequivocamente um dado evento, procurando assim melhor explicar o que este tem para oferecer aos seus convidados
Componentes	
Eventos	Acontecimentos físicos bem definidos espaciais e temporalmente.

Objecto	
Nome:	Newsletter
Sinónimos:	Nenhum
Descrição:	Uma "revista" promocional enviada via email semanalmente para os utilizadores do sistema que aceitam receber publicidade
Exemplos:	Eventos da Semana, Newsletter de Natal, ...
Propriedades	
Título:	Título descritivo da promoção (exemplo: "Promoção de Natal", "Tem de Visitar Neste Verão", ...)
Data e hora de envio	Segunda-Feira (assume-se que vai ser enviado semanalmente à Segunda-Feira)
Relações	
NewsLetterEventoTem	Uma newsletter apresenta um ou mais eventos em destaque ao utilizador
Componentes	
Eventos	Acontecimentos físicos espacialmente e temporalmente bem definidos.

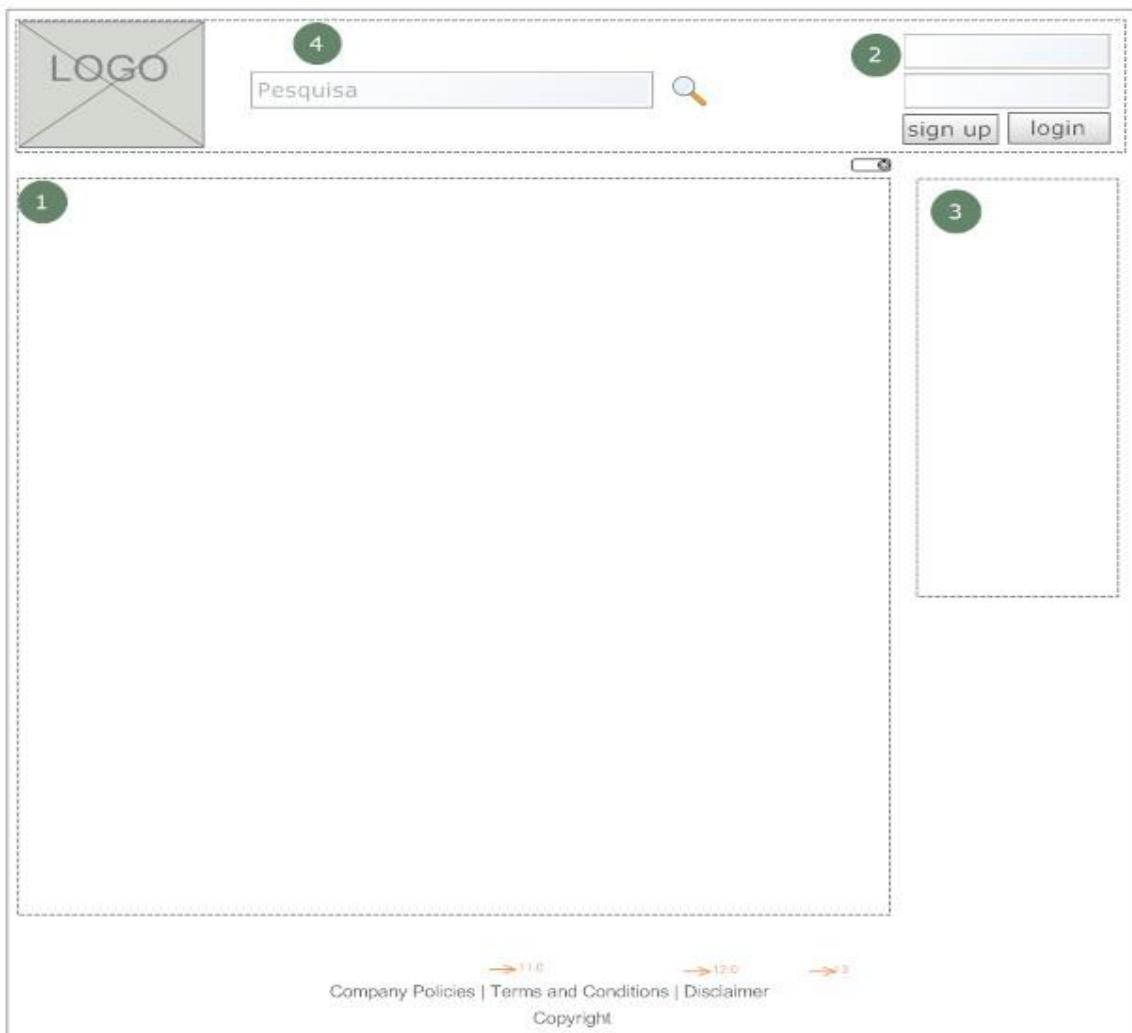
## Especificação Textual e Visual da Camada de Visualização

Neste ponto de especificação do projeto, foi representada a camada de visualização referente às várias páginas que constituirão a aplicação Web. Esta especificação visual procurou sucintamente apresentar as várias áreas (representam o que serão as “div” da aplicação), onde estarão acessíveis funcionalidades bem definidas do sistema.

As representações abaixo mostradas não evidenciam o aspecto da aplicação mas sim, e apenas, como será a estrutura de cada uma das suas páginas, páginas que despoletam várias funcionalidades do sistema.

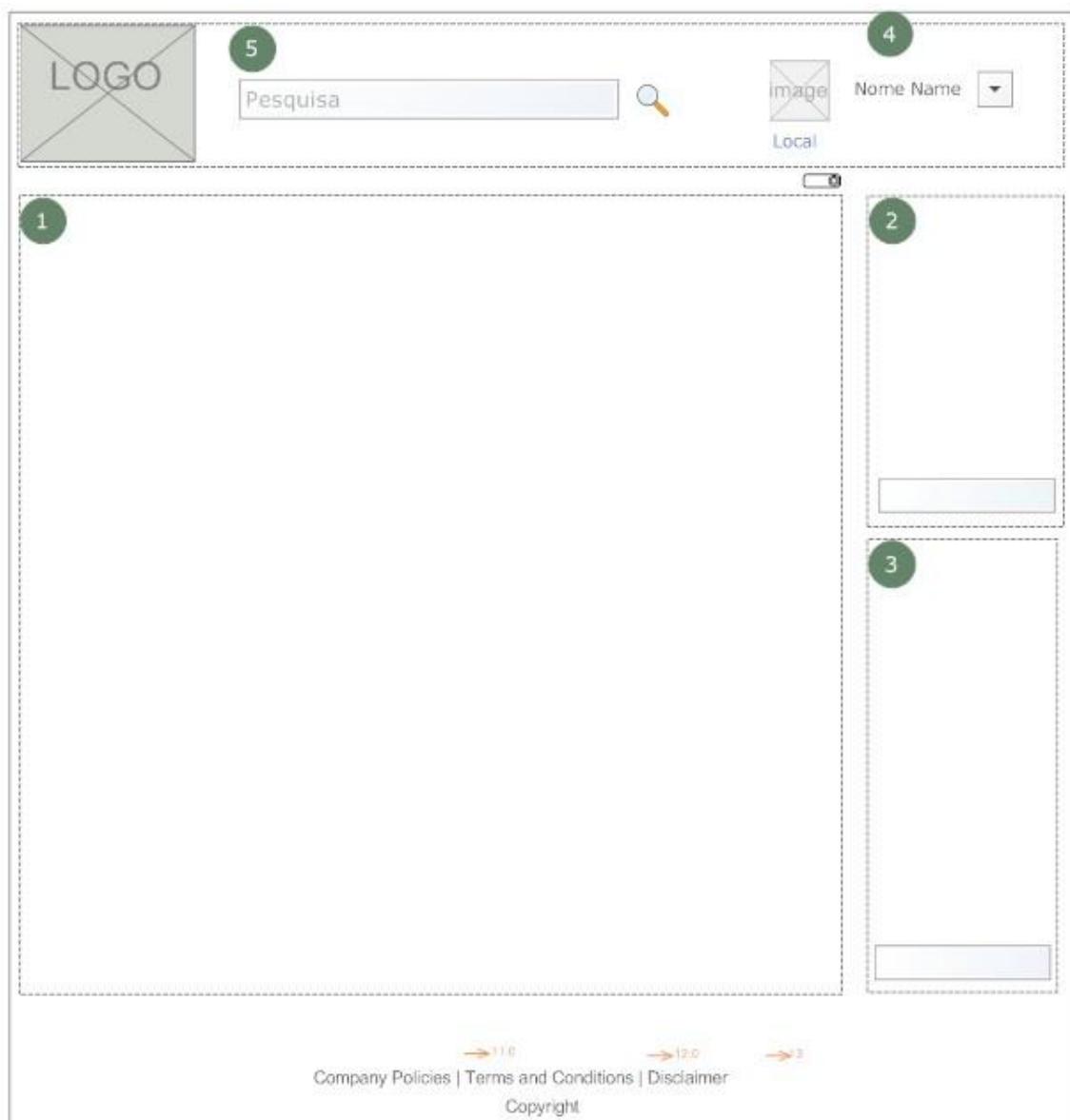
A especificação textual é uma descrição da página e das suas partes, onde também são identificados os grupos de utilizadores que podem aceder cada uma das páginas e as funcionalidades disponíveis para os mesmos.

São apresentadas algumas páginas com informação duplicada, devido à incompatibilidade entre funcionalidades e grupos de utilizadores. Só assim é possível ser coerente relativamente às funcionalidades e aos grupos de utilizadores.



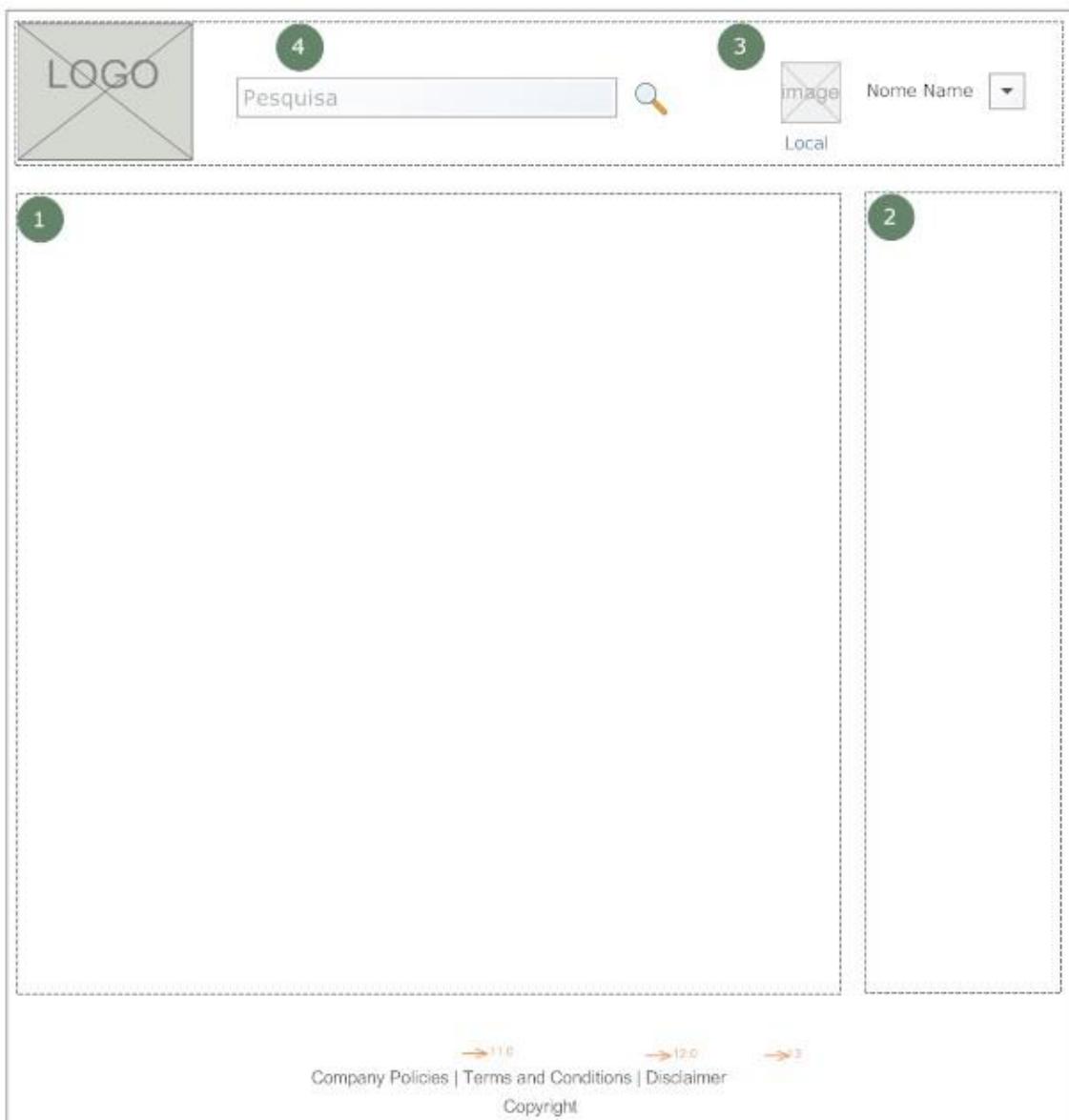
**Figura 1** Representação gráfica da Página principal para Guests.

Página da Aplicação	
Nome:	Página Principal sem Autenticação
Descrição:	Página que permite aceder aos eventos e as suas informações, estando apenas barradas as funcionalidades que requerem autenticação
Grupos de Utilizadores:	Guest
Casos de Uso:	Login, Sign-Up, Ver eventos no Mapa, Ver lista de Eventos, Navegar Mapa, Ver evento rápido, Ver Evento Detalhado e Pesquisar com multicritérios
Mapa da Página	
1- Mapa	Mapa com a representação dos eventos disponíveis no sistema para a área representada, o utilizador poderá aqui navegar sobre o mapa, causando a actualização dos eventos representados no mesmo.
2- Componente de Autenticação e Registo	Nesta área o utilizador pode autenticar-se na aplicação ou solicitar o registo na aplicação, o que o reencaminhará para uma página de registo.
3- Pequena Vista de Evento	Área que é preenchida com informações básicas acerca de um evento, quando o mesmo é clicado no mapa. A partir desta área o utilizador poderá seguir para a página que lhe permite ver todas as informações acerca do evento.
4- Formulário de Pesquisa	Área que permitirá ao utilizador pesquisar com vários critérios na informação que o sistema agrupa relativamente a eventos.



**Figura 2 Representação gráfica da Página principal para Utilizadores registados.**

Página da Aplicação	
Nome:	Página Principal com Autenticação
Descrição:	Página que permite aceder aos eventos e as suas informações, estando apenas barradas as funcionalidades que requerem autenticação
Grupos de Utilizadores:	Utilizador registado, Cliente
Casos de Uso:	Logout, Ver eventos no Mapa, Ver lista de Eventos, Ver evento rápido, Ver Evento Detalhado, Pesquisar com multi-critérios, Criar Evento, Navegar no Mapa, Marcar como Favorito e Desmarcar Favorito
Mapa da Página	
1- Formulário de Criação de Evento	Nesta área o utilizador poderá preencher um formulário representativo de um novo evento e submetê-lo para o sistema.
2- Favoritos	Área que permite ao utilizador aceder e ver alguma informação genérica relativa aos eventos que já marcou como favoritos durante a sua interacção com o sistema.
3- Pequena Vista de Evento	Área que é preenchida com informações básicas acerca de um evento, quando o mesmo é clicado no mapa. A partir desta área o utilizador poderá seguir para a página que lhe permite ver todas as informações acerca do evento.
4- Componente de Logout e Criar Evento	Nesta área o utilizador poderá fazer Logout da aplicação e solicitar a criação de um evento.
5- Formulário de Pesquisa	Área que permitirá ao utilizador pesquisar com vários critérios na informação que o sistema agrupa relativamente a eventos.



**Figura 3** Página que permitirá aos utilizadores registados inserirem eventos no sistema.

Página da Aplicação	
Nome:	Página de Criação de Evento
Descrição:	Página que permite que um utilizador registado crie um evento para adicionar ao sistema.
Grupos de Utilizadores:	Utilizador registado
Casos de Uso:	Logout, Pesquisar com multi-critérios, Criar Evento, Desmarcar Favorito e Solicitar o aparecimento na próxima newsletter.
Mapa da Página	
1- Formulário de Criação de Evento	Nesta área o utilizador registado poderá preencher o formulário de um evento e solicitar o aparecimento do mesmo na próxima newsletter do sistema, no final o mesmo poderá ser submetido.
2- Favoritos	Área que permite ao utilizador registado aceder e ver alguma informação genérica relativa aos eventos que já marcou como favoritos durante toda a sua interacção com o sistema.
3- Componente de Logout e Criar Evento	Nesta área o utilizador poderá fazer Logout da aplicação e solicitar a criação de um evento.
4- Formulário de Pesquisa	Área que permitirá ao utilizador pesquisar com vários critérios na informação que o sistema agrupa relativamente a eventos.

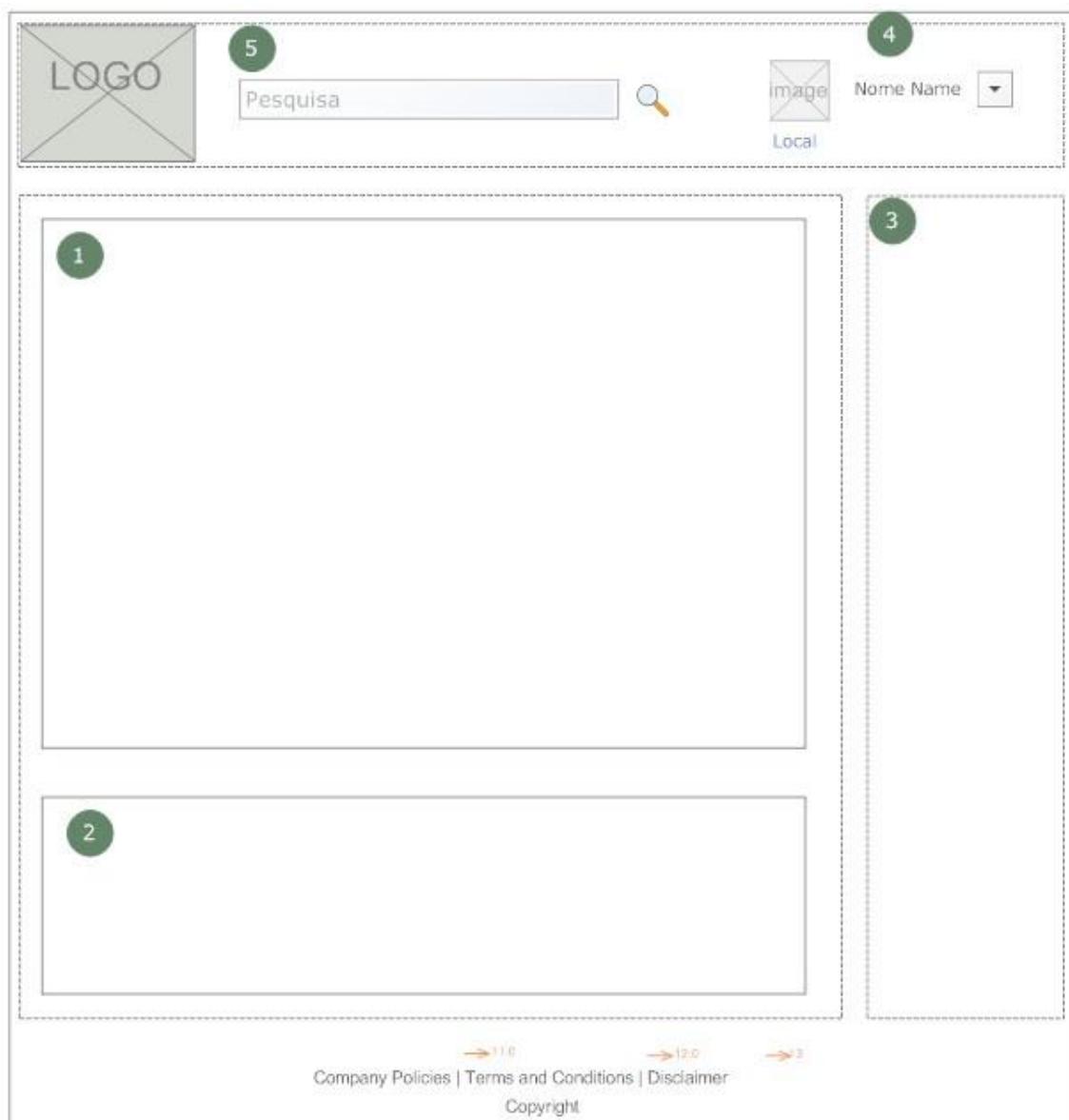
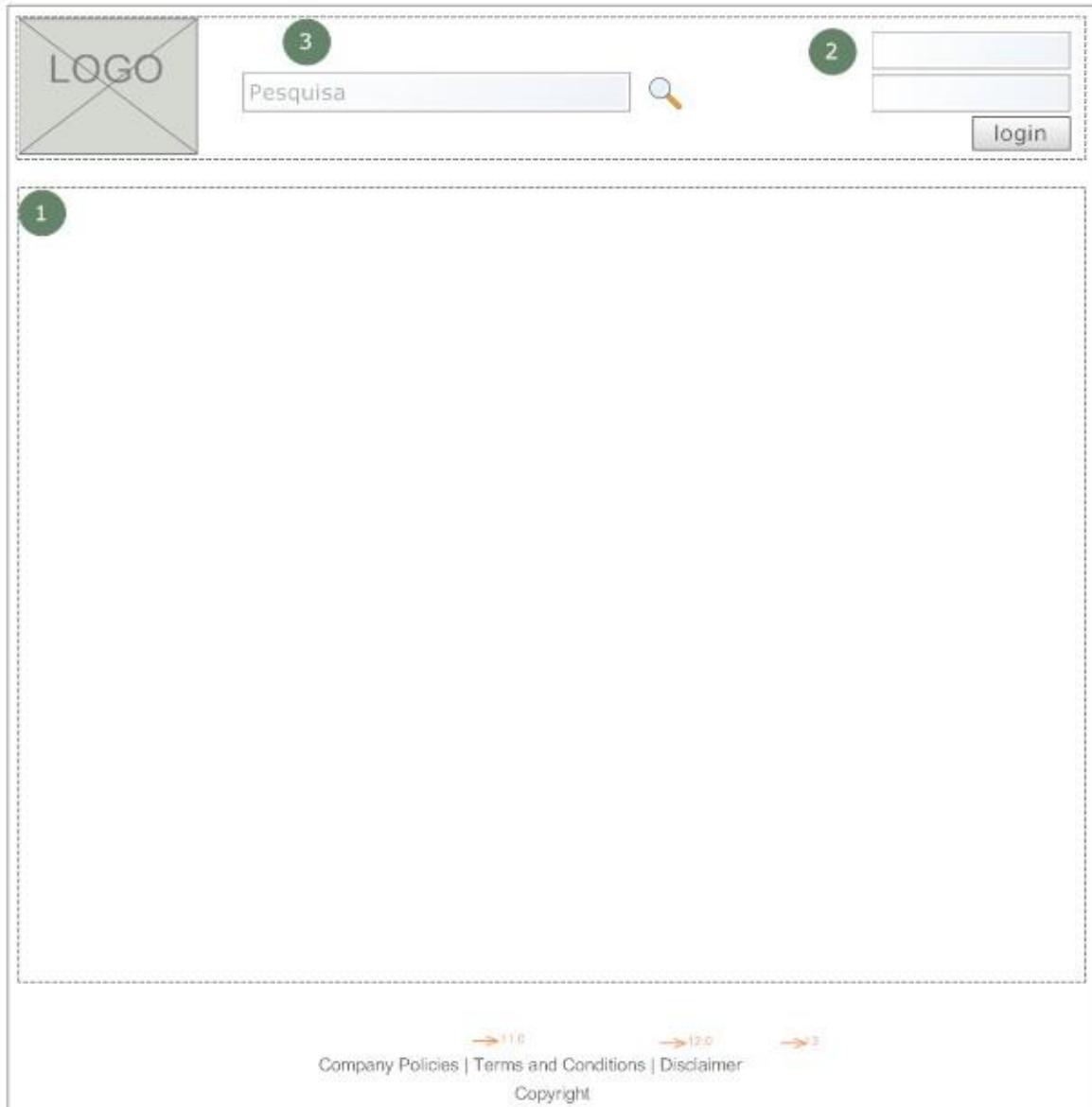


Figura 4 Página que permitirá aceder todas as informações que o sistema disponibiliza relativamente a um evento.

Página da Aplicação	
Nome:	Página de Visão Detalhada de um Evento
Descrição:	Página que permite a um utilizador registado do sistema ver os detalhes de um evento e comentá-lo
Grupos de Utilizadores:	Utilizador registado, Guest
Casos de Uso:	Logout, Login, Sign-Up, Pesquisar com multicritérios, Comentar, Ver Comentários, Criar Evento e Desmarcar Favorito
Mapa da Página	
1- Evento	Área com a descrição detalhada de um evento, onde poderá aceder-se a todos os dados que o sistema guarda acerca do mesmo
2- Zona de Comentários	Área onde estarão disponíveis os comentários realizados relativamente ao evento a ser visualizado e o formulário para submissão de mais comentários (apenas para utilizadores registados).
3- Favoritos	Área que permite ao utilizador registado aceder a alguma informação genérica relativa aos eventos que já marcou como favoritos durante toda a interacção com o sistema.
4- Componente Autenticação e Opções	Nesta área o utilizador registado poderá fazer Logout da aplicação e solicitar a criação de um evento. O Guest poderá proceder ao Login o Sign-Up.
5- Formulário de Pesquisa	Área que permitirá ao utilizador pesquisar com vários critérios na informação que o sistema agrupa relativamente a eventos.



**Figura 5** Página que permitirá a um Guest registrar-se no sistema.

Página da Aplicação	
Nome:	Página de registo no Sistema
Descrição:	Página que permite a um Guest registar-se no sistema
Grupos de Utilizadores:	Guest
Casos de Uso:	Login, Pesquisar com multi-critérios e Sign-Up
Mapa da Página	
1- Formulário para Sign-Up	Formulário que permitirá ao utilizador registar-se no sistema.
2- Componente de Autenticação e Registo	Nesta área o utilizador pode autenticar-se na aplicação ou solicitar o registo na aplicação, o que o reencaminhará para uma página de registo.
5- Formulário de Pesquisa	Área que permitirá ao utilizador pesquisar com vários critérios na informação que o sistema agrupa relativamente a eventos.

## **Especificação Textual das Vistas da Aplicação Web**

As vistas de uma aplicação Web são grupos de funcionalidades que estão atribuídas a um ou mais grupos de utilizadores, estas funcionalidades estão, por norma, divididas áreas.

As áreas de uma vista disponibilizam parte das funcionalidades a um ou mais grupos, sendo as funcionalidades ou grupos parte dos que a aplicação Web tem por input, em uma ou mais páginas.

Durante o desenvolvimento da fase de requisitos foram identificados três grupos de funcionalidades distintos, existindo algumas funcionalidades que são transversais a todos os grupos de funcionalidades, como é o Login e Logout, estes grupos de funcionalidades são a gestão de utilizadores, gestão de eventos e a gestão de negócio.

## **Vista de Utilizadores Registados**

Na vista utilizadores registados estão incluídas todas as funcionalidades relativas à criação, remoção, visualização e publicitação de um evento (negócio), participando na mesma todos os grupos de utilizadores da aplicação, exceptuando-se o dos administradores.

Esta vista tem duas áreas, a primeira delas relativa à criação, remoção, edição e publicitação de eventos a segunda relativa à visualização de eventos, que está disponível para todos os grupos de utilizadores de input da vista gestão de eventos.

Vista para a Aplicação Web			
Nome da Vista	Utilizadores Registados		
Descrição	Nesta vista estão incluídas todas as páginas que permitem aceder funcionalidades relacionadas com a gestão de eventos, nomeadamente de criação, actualização, pesquisa, visualização, crítica, avaliação de eventos e publicitação de eventos.		
Grupos de Utilizadores	Utilizador Registado, Cliente		
Casos de Uso	Ver eventos no mapa, Ver lista de eventos, Navegar mapa, Ver evento rápido, Ver evento detalhado, Pesquisar eventos com multicritérios, Criar evento, Remover Evento, Marcar evento como favorito, Ver eventos marcados como favoritos, Comentar evento, Solicitar aparecimento na próxima newsletter, Ver Comentários e Log-out		
Mapa da Vista			
Nome da área	Descrição da área	Objectos acedidos/geridos	Prioridade

Gestão de Eventos	<p>Na página de criação de eventos os utilizadores registados poderão criar eventos e solicitar o seu aparecimento na próxima newsletter, sendo que esta última implica um pagamento.</p>	Utilizador/Evento/Tipo/ Newsletter	Alta
Visualização de Eventos	<p>Na página principal para e na página ver evento detalhado para utilizadores registados será possível a qualquer utilizador aceder às funcionalidades relativas à visualização de informação, avaliação e comentários de um evento. Nas referidas páginas será ainda possível aceder a um formulário de pesquisa de eventos, onde está, como é óbvio, disponível a funcionalidade Pesquisar Eventos.</p> <p>Na página de ver evento detalhado para utilizadores registados estará disponível, para o utilizador que criou o evento, a funcionalidade remover evento. Em todas as páginas desta área está disponível a funcionalidade Log-out para todos os utilizadores autenticados.</p>	Utilizador/Evento/Tipo/ Comentário/Favoritos	Alta

Edição de Registo	Na página de edição de registo poderá ser actualizada a informação de registo de um utilizador autenticado.	Utilizador	Alta
-------------------	---	------------	------

## **Vista de Visitantes**

A vista de visualização sem autenticação acaba por ter muitas funcionalidades que também estão presentes na vista de gestão de utilizadores, mas como as páginas que as mesmas incluem são diferentes e os grupos de utilizadores de input das mesmas também, estas são claramente duas vistas diferentes.

A separação deve-se ao facto de não termos optado por ter uma vista por grupo de utilizadores, não existindo misturas funcionais.

Vista para a Aplicação Web			
Nome da Vista	Vista de Visitantes		
Descrição	Nesta vista estão incluídas todas as páginas que permitem aceder funcionalidades relacionadas com a visualização de eventos e os seus conteúdos, desde comentários, tipo, localização, descrição, título, etc.		
Grupos de Utilizadores	Guest		
Casos de Uso	Ver eventos no mapa, Ver lista de eventos, Navegar mapa, Ver evento rápido, Ver evento detalhado, Pesquisar eventos com multicritérios, Ver comentários, Login e Sign-Up		
Mapa da Vista			
Nome da área	Descrição da área	Objectos acedidos/geridos	Prioridade

Visualização de Eventos	<p>Na página principal de utilizadores não registados (guests) e na página ver evento detalhado para guests será possível a qualquer utilizador aceder às funcionalidades relativas à visualização de informação de um evento, desde o seu nome, data de inicio, comentários, etc. Nas referidas páginas será ainda possível aceder existentes na aplicação Web será possível aceder a um formulário de pesquisa de eventos, onde está, como é óbvio, disponível a funcionalidade Pesquisar Eventos.</p> <p>Em todas as páginas desta área estão disponíveis a funcionalidade de Login e Sign-Up para todos os utilizadores não autenticados.</p>	Utilizador/Evento/Tipo/Comentário/Favoritos	Alta
Registo	Na página de registo de um Guest na aplicação os guests poderão criar	Utilizador/Grupo/Login	Alta

	uma nova conta usando o caso de uso Sign-Up e Login.		
--	--	--	--

## **Vista de Gestão de Negócio**

A vista de negócio é a mais simples, tendo apenas uma funcionalidade para o Administrador do sistema. Com uma periodicidade a determinar o administrador vai poder enviar a newsletter do sistema, algo fulcral para o negócio do sistema, uma vez que os eventos que tiveram, no acto de registo, a si associada a solicitação de aparecimento na próxima newsletter do sistema necessitam de ser enviados na mesma.

Vista para a Aplicação Web			
Nome da Vista	Gestão de Negócio		
Descrição	A esta vista pertence a página de administrador		
Grupos de Utilizadores	Administrador		
Casos de Uso:	Enviar newsletter		
Mapa da Vista			
Nome da área	Descrição da área	Objectos acedidos/geridos	Prioridade
Gestão de Negócio	Na página principal para administrador estará disponível a funcionalidade que permite o envio da newsletter do sistema.	Administrador, Newsletter e Evento	Baixa

## Notas Finais – Levantamento e Análise de Requisitos

Desenvolver aplicações Web é algo cada vez mais importante, dado que boa parte das corporações estão completamente disponíveis na Web, não existindo praticamente nenhuma operação realizável numa aplicação monolítica, que não o seja na Web.

Nesta componente (levantamento de requisitos) do projecto foi muito importante usar uma abordagem mais direcionada para o desenvolvimento Web, uma vez que a abordagem usada no levantamento de requisitos para aplicações Web é um pouco diferente do usado para as aplicações tradicionais, tendo sido muito relevante compreender isso.

Identificar os grupos de utilizadores, prototipar a interface gráfica e estabelecer uma relação entre funcionalidades, páginas e grupos de utilizadores é essencial para a compreensão de uma aplicação Web, fazendo com que fique muito mais claro o que será o produto final.

Foi detectada uma lacuna no processo de levantamento de requisitos, pois não ficavam bem esclarecidas as relações entre as diferentes páginas da aplicação, tendo sido por isso elaborado um mapa da aplicação Web como complemento à secção de especificação da camada de visualização. Contudo, seria mais interessante desenvolver um protótipo interativo da aplicação, o que é possível construir sem escrever qualquer linha de código, usando ferramentas como Fore UI e Flair Builder, faltou o tempo para ser perfeccionista neste domínio.

Esta fase não serviu única e exclusivamente para identificar e especificar objectivamente a aplicação final, pois a elaboração de um dicionário de dados vai um pouco mais longe, permitindo estabelecer um dicionário de dados que normalizará a comunicação entre os intervenientes do projecto até que este se finde, possibilitando que todos os envolvidos no mesmo se entendam com maior facilidade.

A modelação e especificação das aplicações é cada vez mais importante, pois com o software que temos hoje disponível, principalmente as ferramentas CASE, permitem poupar muito trabalho gerando código a partir das modelações. Por exemplo, com o desenvolvimento do modelo Chen paralelamente à especificação do dicionário de dados, será possível gerar o código SQL para a implementação da base de dados de suporte à aplicação. Posteriormente,

usando o framework (Yii), que foi o escolhido, será possível gerar automaticamente todos os métodos CRUD e as classes dos objectos presentes no dicionário de dados.

Esta fase correu como esperado, sendo que foi executado com sucesso o que se esperava para esta fase, tendo sido ultrapassadas algumas limitações temporais.

## **Modelação dos Dados (Entidades e Relacionamentos)**

Construir aplicações Web implica muitas vezes a modelação e implementação da camada de dados, a não ser que a mesma seja estática ou utilize o sistema de ficheiros para serializar dados de que necessita.

Neste relatório procuramos explicar o processo usado para identificar a informação que terá que ser serializada pela aplicação Web em desenvolvimento, identificando as duas entidades, os atributos das entidades e os relacionamentos entre as entidades.

O facto de ter sido realizada uma análise de requisitos de forma exaustiva facilitou muito esta fase, pois o dicionário de dados desenvolvidos já continha algumas entidades, atributos e relações evidenciadas.

Assim, nesta fase, modelamos conceptualmente, usando duas notações diferentes, os dados que a aplicação Web utilizará.

## Metodologia

Uma abordagem bem estruturada é essencial para poupar tempo de desenvolvimento e obter melhores resultados.

Neste caso aproveitar o trabalho realizado na fase de análise de requisitos foi crucial e era algo lógico, pois a verdade é que no dicionário de dados desenvolvido estavam identificadas as entidades, os atributos e os relacionamentos, embora não explicitamente.

Assim, a metodologia seguida foi a partir dos objectos, atributos e relacionamentos detectados na fase anterior, transformando num modelo conceptual usando a notação Peter Chen, uma linguagem gráfica especializada na modelação *ER*. O modelo conceptual foi posteriormente inserido no *Web Ratio*, algo exigido no módulo em causa.

## Identificação das Entidades

Partindo do Dicionário de Dados existem entidades que são identificadas automaticamente, como Utilizador, Evento, Comentário, uma vez que foram os objetos detectados.

Existem algumas entidades que não são detectadas com tanta facilidade, pois as mesmas estão implícitas em relações, como é o caso do Evento Favorito, ou dizem respeito ao facto de ser requisito devido a ser uma aplicação Web, como é o caso da entidade Login que, como é óbvio, auxiliará a autenticação.

Na tabela introduzida abaixo são descritas as entidades do sistema individualmente.

Entidades	
Nome	Descrição
Evento	Entidade principal do sistema que representa todas as atividades do mundo físico que podem ser divulgadas no sistema.
Utilizador	Entidade que representa todos os utilizadores registados do sistema.
Comentário	Entidade que representa todas as opiniões que os utilizadores podem deixar registadas relativamente aos eventos divulgados no sistema.

Newsletter	Entidade que representa as edições de Email's semanais em que podem ser divulgados eventos presentes no sistema, mediante pré-pagamento. Esta é a entidade principal do modelo de negócio.
Tipo	A entidade tipo é auxiliar permitindo identificar os eventos com um nome mais simples, tornando as pesquisas em potência mais eficientes.
Login	Entidade que representa a informação que o sistema precisa de guardar na autenticação de um utilizador registado.

## Identificação dos Atributos

A identificação dos atributos foi bastante simples, pois estes estavam parcialmente identificados no dicionário de dados. No entanto, a este nível, não faltavam apenas os atributos, que advêm do facto de se tratar do desenvolvimento de uma aplicação Web, também foram identificados alguns requisitos que derivam das exigências do modelo relacional, como são os casos das chaves primárias.

Atributos			
Nome das Entidades	Nome dos Atributos	Tipo de Dados	Descrição
Evento	<u>NR</u>	Inteiro	Identificador único por cada Evento
	Título	String	Frase sugestiva relativa a um Evento
	Descrição	String	Texto que descreve o conteúdo de um Evento
	Latitude	Float	Longitude geográfica do Evento
	Longitude	Float	Latitude geográfica do Evento
	DataInício	DateTim e	Início do Evento
	DataFim	DateTim e	Final do Evento
Utilizador	<u>NR</u>	Inteiro	Identificador único por utilizador
	Nome	String	Nome pelo qual o utilizador gosta de ser identificado
	Email	String	Endereço de correio electrónico
	Hpassword	String	Password encriptada
	Latitude	Float	Latitude predefinida
	Longitude	Float	Longitude predefinida
	Foto	String	Endereço de uma fotografia no servidor
Comentário	<u>NR</u>	Integer	Identificador do Comentário no sistema
	Título	String	Título atribuído pelo utilizador ao comentário
	Texto	String	Corpo do comentário do autor

	<b>DateTime</b>	<b>DateTim e</b>	Data e hora em que o comentário entrou no sistema
Newsletter	<u>NR</u>	Integer	Identificador único de uma Newsletter
	Título	String	Título sugestivo da Newsletter
	DateTimeEnvio	DateTim e	Tuplo data-hora de envio da newsletter
Tipo	<u>NR</u>	Integer	Identificador único por cada tipo de evento existente
	Nome	String	Frase ou palavra que cataloga eventos
Login	<u>NR</u>	Integer	Identificador de uma autenticação no sistema
	StringSessão	String	String de sessão que identifica o login e o utilizador
	DateTimeLogin	DateTim e	Data e hora da autenticação
	DateTimeExpira	DateTim e	Data e hora em que a autenticação expira no caso de haver cookie
	Cookie	Boolean	Booleano que indica se utilizador pretende manter a sessão iniciada

Foram identificados e descritos os atributos, sendo o seu domínio parcialmente definido a partir dos tipos de dados, que ficaram desde já identificados. Os atributos estão na tabela acima organizados por entidade, estando a chave primária de cada entidade identificada com o sublinhado. A explicação adicional dos atributos segue na última coluna da tabela.

Os tipos de dados definidos foram cruciais para a construção do Modelo de Dados, daí terem sido utilizados os tipos acima. Se fosse para criar uma base de dados, os atributos teriam sempre a eles associados os seus tamanhos.

### Identificação dos Relacionamentos

Depois de identificadas as entidades e os atributos as relações foram algo simples de identificar, uma vez que as mesmas foram descritas no dicionário de dados e já tínhamos inclusive introduzido alguma semântica para as mesmas.

Foi decidido documentar os relacionamentos referindo para cada relação as duas entidades envolvidas, as multiplicidades da relação junto de cada entidade e a semântica da relação, tornando o relacionamento comprehensível e legível. Posteriormente os relacionamentos foram alvo de uma descrição textual adicional, estando estes referenciados a partir do seu ID.

Relações					
ID	Entidade	Multiplicidade	Semântica	Multiplicidade	Entidade
1	Utilizador	1	cria	N	Evento
2	Utilizador	N	marca como favoritos	M	Evento
3	Utilizador	1	escreve	N	Comentário
4	Utilizador	1	tem	N	Login
5	Evento	1	tem	N	Comentário
6	Tipo	1	cataloga	N	Evento
7	Newsletter	1	tem	N	Evento

Descrição das Relações	
ID	Descrição
1	Um utilizador registado pode criar muitos eventos.
2	Um utilizador pode marcar como favoritos os eventos do sistema.
3	Um utilizador pode escrever comentários acerca dos eventos do sistema.
4	Um utilizador pode fazer várias autenticações.
5	Um utilizador tem comentários.
6	Um evento tem um tipo que o resume.
7	As newsletters publicitam vários eventos.

## Construção do Modelo de Dados

A construção do Modelo de Dados no *WebRatio* foi muito simples, pois toda a informação estava já devidamente identificada, sendo por isso simples utilização da interface gráfica da aplicação.

A partir deste modelo de dados será possível criar classes *Java* que permitirão interagir com as entidades do projeto, realizando várias operações de forma simples, sem ser necessário escrever qualquer linha de código. É inclusive possível estabelecer uma ligação com um sistema de gestão de base de dados, criando a camada de persistência de forma automática.

Nota: A imagem do modelo de dados desenvolvido segue no anexo 3, não tendo sido aqui inserida para manter o relatório o mais breve possível.

## Notas Finais – Entidades e Relacionamentos

Ao longo da realização desta etapa do projeto, utilizando o dicionário de dados, dicionário que foi especificado na fase de requisitos do projeto, identificar as entidades, os atributos e os relacionamentos entre as entidades.

Posteriormente, utilizando ferramentas como o Web Ratio, foi desenhado o Modelo de Dados. Com o uso do *WebRatio* foi possível gerar o código SQL que nos permitirá, posteriormente, construir as tabelas relacionais da nossa base de dados. Tal é necessário porque terá que se desenvolver o projeto selecionado em *PHP e MySql*, conforme foi acordado com os docentes, ora o *WebRatio* é um *software* que gera uma aplicação em *Java*, pelo que é melhor desenvolver paralelamente o projeto em *PHP*.

A construção do Modelo de Dados e a documentação do mesmo foram atividades muito interessantes, principalmente porque este trabalho de modelação e especificação permitiram-nos realmente caminhar na implementação da aplicação, uma vez que foram utilizadas ferramentas que a partir do modelo geram realmente partes do produto. Tal é algo que se enquadra num módulo de Engenharia Web e faz com que a atividade de modelar não pareça um desperdício.

## Tecnologias

No desenvolvimento da nossa aplicação Web precisaremos de usar algumas tecnologias, uma vez que uma aplicação Web é sempre constituída por pelo menos duas partes, uma de cliente e outra de servidor.

Com o presente documento visamos apresentar as nossas escolhas ao nível das tecnologias escolhidas para o desenvolvimento da aplicação, tanto ao nível do servidor como do cliente, fundamentando-as sempre que possível com as características da aplicação que nos propomos desenvolver.

Por fim identificamos um framework que pretendemos usar para começarmos desde já estruturar a nossa aplicação, pois a verdade é que já não se justifica ter que desenvolver uma aplicação totalmente de raiz, uma vez que existem coisas tão repetitivas que nada acrescenta à aplicação final o facto de sermos nós a implementá-las.

## Tecnologias de Cliente

No desenvolvimento da nossa aplicação Web precisaremos de usar algumas tecnologias, uma vez que uma aplicação Web é sempre constituída por pelo menos duas partes, uma de cliente e outra de servidor.

Com o presente documento visamos apresentar as nossas escolhas ao nível das tecnologias escolhidas para o desenvolvimento da aplicação, tanto ao nível do servidor como do cliente, fundamentando-as sempre que possível com as características da aplicação que nos propomos desenvolver.

Por fim identificamos um framework que pretendemos usar para começarmos desde já estruturar a nossa aplicação, pois a verdade é que já não se justifica ter que desenvolver uma aplicação totalmente de raiz, uma vez que existem coisas tão repetitivas que nada acrescenta à aplicação final o facto de sermos nós a implementá-las.

### Lógica de Cliente

Como ficou bastante evidente na descrição da aplicação que nos propomos desenvolver, iremos precisar de usar algum serviço externo para representar localizações em mapas, pelo que nos parece que a aplicação terá que usar Javascript para que seja possível implementar algumas das funcionalidades já identificadas.

O Javascript será por isso a linguagem que utilizaremos no caso de necessitarmos de implementar lógica do lado do cliente.

### AJAX

Como é óbvio páginas que tenham mapas, listas e outros componentes que podem ser vistos como pesados, isto é, que aumentam consideravelmente o tempo de carregamento das páginas, não devem estar sempre a ser refrescadas, pois isso é danoso para a fluidez da aplicação e para a rede (sobrecarga desnecessária). Para evitar este tipo de problemas e para tornarmos a nossa aplicação Web tão dinâmica quanto possível é nosso objectivo fazer uso de técnicas com o AJAX (chamadas assíncronas ao servidor).

## **Design da Interface de Utilizador**

Como é óbvio as aplicações atualmente precisam de ser minimamente apelativas graficamente, caso contrário não têm grandes hipóteses de vir a conquistar visitantes, para conseguir que a nossa aplicação seja minimamente apelativa usaremos HTML combinado com CSS (*Cascade Style Sheets*) para o desenho da nossa página Web.

Poderá ser necessário recorrer a uma combinação de CSS com Javascript para conseguir alguma formatação mais dinâmica de algumas páginas (views) da nossa aplicação. Para além do referido, recorreremos ao Javascript sempre que precisemos de alterar a estrutura e conteúdos das páginas dinamicamente, isto é, aceder ou modificar o DOM (*Document Object Model*) ou o BOM (*Browser Object Model*).

Como nenhum dos elementos deste grupo tem grandes competências ao nível Design de interfaces de utilizador, principalmente no que diz respeito a interfaces direcionadas para a Web, é provável que teremos que recorrer a alguns templates já existentes.

## Tecnologias de Servidor

### Linguagem e Servidor Web

A aplicação que pretendemos desenvolver não apresenta requisitos que nos façam pender para esta ou aquela linguagem, sendo que a nossa escolha acabou por ser maioritariamente baseada nos custos que cada tecnologia tem associados.

Das tecnologias gratuitas, pensamos utilizar Java Server Pages ou PHP, no entanto o facto de não ser necessário portabilidade no que diz respeito ao sistema operativo do servidor da aplicação, acabamos por excluir o JSP, pois correr a aplicação na JVM não nos pareceu necessário. A existência de uma comunidade vasta de programadores que usam PHP também foi uma característica que nos levou à sua escolha.

Depois da escolha da linguagem de script, foi bastante fácil escolher o servidor, pois gratuito e para funcionar com PHP, nada melhor que o Apache *HTTP*, sendo que o mesmo também é gratuito.

### Servidor de Base de Dados

Ficou bastante explícito pela descrição da nossa aplicação que precisaríamos de guardar alguns dados relativamente a eventos, utilizadores e anunciantes, mas neste domínio não há grandes necessidades de armazenamento, sendo que não vale a pena nenhum investimento para guardar os dados da aplicação.

Depois de analisarmos alguns servidores de bases de dados gratuitos (*MySql*, *Postgres* etc.), a nossa escolha acabou por recair pelo *MySql*, uma vez que a sua integração com o PHP é muito boa e existe uma vasta comunidade que nos pode dar suporte na sua utilização.

## Framework

O *framework* foi a última componente tecnológica a ser escolhida, pois varia com a linguagem. De acordo com as ideias que temos para o desenvolvimento da aplicação, resolvemos escolher um *framework* desenvolvido em PHP que tivessem algumas características bem definidas, das quais destacamos:

- Suporte para MVC (*Model-View-Control Pattern*);
- Suporte ao uso AJAX;
- Templates;
- Suporte para Autenticação;
- Suporte para validação automática do input dos utilizadores;
- *Object-relational mapping*, isto é, que permite que as tabelas da base de dados sejam representadas e tratadas como classes dentro da aplicação.

Depois de algumas pesquisas, descobrimos que existiam alguns frameworks nestas condições (*Yii*, *Zend* e outros). A nossa escolha acabou por recair pelo *Yii*, pois o *Zend* não é gratuito, já o *Yii*, para além de ser gratuito tem uma comunidade muito ativa, existindo muita documentação e tutoriais em vídeo que auxiliam a aprendizagem e o desenvolvimento.

## Desenho da Aplicação Web (Coarse Design)

Nesta fase de especificação/modelação no desenvolvimento do projeto referente à cadeira de Engenharia Web, tivemos de recorrer às etapas anteriores para, desta forma, conseguirmos construir o *Coarse Design* da nossa aplicação web.

Antes de avançarmos na explicação do procedimento e na apresentação de tal construção, é importante fazermos uma breve abordagem ao que é o conceito *Coarse Design*. Como tal, *Coarse Design* tem como objectivo principal esboçar de cada *site view* da aplicação web, procurando sempre estabelecer um mapeamento entre os elementos relativos ao esquema de dados e a respectiva área onde eles serão apresentados e/ou geridos.

Deste modo, para fazer tal representação tivemos de recorrer às *site views*, que foram identificadas na fase de análise de requisitos e, a partir daí, fazer o desenho das mesmas identificando as operações que terão que ser feitas às entidades por *site view*. Neste ponto, como parte de um procedimento habitual na construção do *Coarse Design* de uma aplicação web, tivemos de identificar os objetos acedidos, listados, publicados, criados e ainda os conectados.

Esta fase do projeto enquadra-se na preparação para a futura modelação do mesmo, usando a ferramenta Webratio, onde, com tal ferramenta, será possível gerar a aplicação pretendida a partir da especificação desenvolvida e explicada. Torna-se, desta forma, uma ferramenta verdadeiramente útil possibilitando ainda que se poupe bastante tempo na futura implementação do projeto.

Surgindo numa fase de especificação do projeto, o desenho da aplicação permite que seja estruturada, em forma de rascunho, qual será o aspecto estrutural da aplicação web, procurando sempre enquadrar-se com as áreas de visualização da mesma as suas especificidades funcionais.

Surge, desta forma, como um passo bastante importante na planificação do que será e de como será a aplicação web que se está a desenvolver. A notação usada para fazer tal tipo de especificação passa pela identificação, em cada área do *site view*, quais as operações que terão de ser feitas e em relação a quais entidades. Surge desta forma a necessidade de identificar, com base numa notação específica, os vários tipos de funcionalidades relativas a dadas entidades. Isto é, têm de ser identificados inequivocamente os objectos acedidos, listados, publicados, criados e os conectados.

A notação específica, acima abordada, será representada abaixo para cada tipo de fragmento e, após tal enumeração, será feita uma breve abordagem ao que cada uma pretende representar:

- *Core(CoreEntity, Component1, ..., ComponentN);*
- *Access(Core Entity);*
- *Interconnection(Role1,...,RoleN);*
- *Create & Connect(Entity1,Role1,...,RoleN);*
- *Modify(Entity1);*
- *Delete(Entity1);*
- *Connect(Role1), Disconnect(Role1);*
- *Set(ContextInformation);*

Como é possível abordar, para denotar cada tipo de informação ou especificidade funcional, deverá ser usada uma notação informal como a apresentada.

O *Core*, que tem como “argumentos” uma entidade principal e vários componentes relativos à mesma, denota que tais componentes deverão estar publicados, isto é, deverão ser um conteúdo visível na respectiva área da aplicação.

Por sua vez, para representarmos o acesso a uma ou mais instâncias da uma entidade do projecto, deveremos representar como *Access(CoreEntity)*, em que *CoreEntity* é a entidade em causa.

Quando expomos a existência de uma interconexão, que denota a navegação de instâncias de uma dada entidade para instâncias de uma outra e, em que tais estão conectadas, isto é, têm uma relação entre elas, deveremos especificar escrevendo *Interconnection(Role1,...RoleN)*, tendo em conta que os “argumentos recebidos” pela interconexão representam as relações entre as entidades.

O fragmento *Create&Connect(Entity1,Role1,...,RoleN)*, denota a criação do que será uma nova instância da entidade passada por argumento e, permite ainda, que seja feita uma nova relação, ou várias, da instância criada através da passagem da mesma relação como 2º argumento, ou mais, em *Create&Connect*.

Por sua vez, quando é pretendido denotar a modificação ou apagamento dumha instância de uma dada entidade, podemos recorrer ao uso da notação *Modify(Entity1)* e/ou *Delete(Entity1)* em que *Entity1* representa a dada entidade cuja instância será modificada ou apagada.

Para se representar a criação ou apagamento de relações, referentes a uma dada instância, deverá ser usada a notação *Connect(Role1)* e *Disconnect(Role1)*, respectivamente.

Por fim, relativamente ao fragmento *Set(ContextInformation)*, tal deve ser escrito quando surge a necessidade de denotar o ajustamento de algum pedaço de informação que se encontra visível/disponível em todas as páginas que constituem o site view da aplicação web em desenvolvimento.

## Vista de Utilizadores Registados

Na vista de Utilizadores Registados estão disponíveis todas as funcionalidades relativas à gestão da conta de utilizador e gestão de eventos.

Esta vista é de utilizadores registados/autenticados, sendo que no ato de registo de um evento os utilizadores autenticados podem solicitar que o evento em criação surja na próxima *newsletter* do sistema, é que claro que mediante pagamento, neste caso os utilizadores autenticados assumem o papel de clientes.

A área de Visualização de Eventos presente nesta vista é muito semelhante à área de Visualização de Eventos da vista de Visitantes, no entanto existem diferenças importantes relativamente às funcionalidades a disponibilizar, uma vez que algumas delas só são possíveis para utilizadores registados.

As áreas de Edição de Registo e Gestão de Eventos só podem ser disponibilizadas para utilizadores registados e autenticados, uma vez que só é possível criar e editar eventos tendo uma conta, assim como só a existência de uma conta possibilita a edição de registo.

Visualização de Eventos	Explicação
Core(Utilizador,Username)	Em todas as páginas desta site view terão que ser mostradas informações sobre o utilizador autenticado, caso não seja um guest e para além disso, na página ver evento detalhado será mostrada informação sobre o seu criador.
Core(Evento,Titulo,Descrição,Latitude,Longitude,DataInicio,DataFim,Tipo)	Nas páginas principais guests e utilizadores autenticados e na de ver evento detalhado terá que ser publicada informação acerca de eventos.
Core(Comentário,Titulo,Texto,Datal)	Na página de ver eventos detalhados terá que estar disponível informação relativa aos comentários no mesmo.

Access(Evento,Comentário,Tipo,Utilizador)	Na página de ver detalhes de um evento será possível aceder a informação relativa a um evento, o seu tipo, comentários e autor dos comentários.
Interconnection(Evento,Comentário)	Na página de ver detalhes de um evento será possível navegar de um evento para os seus comentários, pelo que é necessária a interconexão.
Interconnection(Evento,Tipo)	Nas páginas de pesquisa será possível dividir os resultados de uma pesquisa por tipos e restringir o resultado aos mesmos, pelo que é necessária esta interconexão.
Create&Connect(Comentário, Utilizador,Evento)	A partir da página ver evento detalhado será possível a qualquer utilizador autenticado realizar comentários que serão associados a quem o escreveu (utilizador) e ao evento alvo.
Get(UserCtxParam)	Na página do Evento é necessário aceder ao ID do utilizador autenticado, uma vez que tal é necessário para operações como registo de um comentário e de marcar/desmarcar favorito.
Connect(Evento,Utilizador),Disconnect(Evento, Utilizador)	Na página do Evento será possível a um utilizador registado marcar um evento como favorito ou desmarca-lo, caso já o tenha marcado.

Gestão de Eventos	Explicação
Delete(Evento)	Na página de Edição de Evento será possível remover o evento do sistema.

Modify(Evento)	Na página de ver detalhes de um evento, no caso de esta estar a ser visualizada pelo utilizador que a criou e estando o mesmo autenticado, este poderá remover o evento do sistema.
Get(UserCtxParam)	Na página de Criar Evento é necessário aceder ao ID do utilizador que está a criar o evento, para que se possa associa-lo ao evento em causa no ato de criação do evento.
Create&Connect(Evento,Utilizador,Tipo,Newsletter)	A partir da página Criar Evento será possível a qualquer utilizador autenticado criar um evento, ficando o mesmo associado ao autor, a um tipo e opcionalmente, no caso de ser solicitado que o evento em causa apareça na próxima <i>newsletter</i> do sistema, a uma <i>newsletter</i> .

Utilizadores Registados	Explicação
Connect(Evento,Utilizador),Disconnect(Evento, Utilizador)	Na página de Favoritos será possível a um utilizador registado desmarcar um evento previamente marcado como favorito.

## Vista de Visitantes

Visualização de Eventos	Explicação
Core(Evento,Titulo,Descrição,Latitude,Longitude,DataInicio,DataFim,Tipo)	Na página principal de guest de ver evento detalhado de guests terá que ser publicada informação acerca de eventos.
Core(Comentário,Titulo,Texto,Data)	Na página de ver eventos de detalhados de guests terá que estar disponível informação relativa aos comentários do evento.
Access(Evento,Comentário,Tipo,Utilizador)	Nas páginas principal para guests e ver eventos detalhados para guests será possível aceder a informação relativa a um evento, o seu tipo, comentários e autor dos comentários.
Interconnection(Evento_Comentário)	Na página de ver detalhes de um evento para guests será possível navegar de um evento para os seus comentários, pelo que é necessária a interconexão.
Interconnection(Evento_Tipo)	Nas páginas de pesquisa será possível dividir os resultados de uma pesquisa por tipos e restringir o resultado aos mesmos, pelo que é necessária esta interconexão.
Create&Connect(Login,Utilizador)	Como a partir de qualquer página desta site view é possível realizar Login, quando tal acontece é necessário guardar um novo Login, que estará ligado ao utilizador que o realizou.

Registo	Explicação
Create&Connect(Utilizador,Grupo)	A partir da página de registo de um utilizador terá que ser possível gravar/criar um novo utilizador e associa-lo a um grupo.

## Vista de Gestão de Negócio

Relativamente à Vista de Gestão de Negócio, esta têm a funcionalidade fulcral do modelo de negócio da aplicação. Tal modelo assenta principalmente no envio de newsletter aos utilizadores do sistema, onde se publicitem os eventos que ainda não começaram e cujo criador tenha realizado um pagamento para que o seu evento constasse na newsletter do sistema. As funcionalidades presentes nesta vista passam pela própria criação da newsletter e da interligação que terá de existir entre o que publicita e os eventos que constituem a mesma publicidade.

É importante salientar que respectivamente à vista de gestão de negócios, esta estará apenas disponível para administradores da aplicação, pelos motivos óbvios de que só eles poderão compor a newsletter e só eles poderão enviar para os emails dos utilizadores a newsletter.

Por fim, é importante ainda salientar quais serão as componentes que irão identificar inequivocamente a entidade newsletter que são, como o próprio *Core(Newsletter, Nome, Data)* informalmente indica, um Nome descritivo e ainda uma Data, que será a data de criação da newsletter. Como já foi referido e facilmente se entende, a relação que irá ter de existir entre a newsletter e os eventos, mostrada pela notação *Interconnection(Newsletter\_Evento)*, é imprescindível uma vez que é sempre necessário identificar quais são os eventos que compõe uma instância da newsletter.

Para enviar uma newsletter com o conteúdo dos eventos a serem publicitados terá que se aceder aos eventos, para além disso, para personalizar a newsletter será preciso os nomes dos tipos, daí o *Access(Evento, Tipo)*.

Gestão de Negócio	Explicação
Core(Newsletter, Nome, Data)	Na página de envio de newsletter terão que ser publicados dados relativamente à mesma.
Interconnection(Newsletter_Evento)	As newsletters terão que estar ligadas aos eventos que a compõem, ou seja, o administrador deve poder navegar das

	newsletters para os eventos das mesmas.
Access(Evento,Tipo)	Para enviar a newsletter será necessário aceder aos eventos a enviar e ao seu tipo, para personalizar a newsletter.

## Notas Finais - Coarse Design

Nesta fase de especificação/modelação no desenvolvimento do projeto referente à cadeira de Engenharia Web, tivemos de recorrer às etapas anteriores para, desta forma, conseguirmos construir o *Coarse Design* da nossa aplicação web.

Antes de avançarmos na explicação do procedimento e na apresentação de tal construção, é importante fazermos uma breve abordagem ao que é o conceito *Coarse Design*. Como tal, *Coarse Design* tem como objectivo principal esboçar de cada *site view* da aplicação web, procurando sempre estabelecer um mapeamento entre os elementos relativos ao esquema de dados e a respectiva área onde eles serão apresentados e/ou geridos.

Deste modo, para fazer tal representação tivemos de recorrer às site views, que foram identificadas na fase de análise de requisitos e, a partir daí, fazer o desenho das mesmas identificando as operações que terão que ser feitas às entidades por *site view*. Neste ponto, como parte de um procedimento habitual na construção do *Coarse Design* de uma aplicação web, tivemos de identificar os objetos acedidos, listados, publicados, criados e ainda os conectados.

Esta fase do projeto enquadra-se na preparação para a futura modelação do mesmo, usando a ferramenta *Webratio*, onde, com tal ferramenta, será possível gerar a aplicação pretendida a partir da especificação desenvolvida e explicada. Torna-se, desta forma, uma ferramenta verdadeiramente útil possibilitando ainda que se poupe bastante tempo na futura implementação do projeto.

## Implementação

### Implementação Webratio

Este documento é suporte à fase de implementação da aplicação Web Eventhis, que está relacionada com publicação e publicitação de eventos na Web, facilitando aos utilizadores encontrarem atividades para ocuparem o seu tempo livre.

Durante as todas as fases anteriores foi sendo pensada e desenhada uma aplicação Web seguindo uma abordagem de engenharia. O processo usado foi muito cerimonioso mas teve como output documentação muito útil para a implementação da aplicação Web, pelo que serão feitas variadas referências à mesma, para justificar decisões da implementação.

O objetivo principal desta fase foi a implementação da aplicação Web pensada e desenhada em todas as fases ulteriores usando a aplicação *Webratio*, tal implicou alguma disciplina de desenvolvimento.



Figura 6 Logótipo da ferramenta de desenvolvimento de Aplicações Web orientadas aos dados usada para o desenvolvimento da aplicação Web *Eventhis*.

Este documento serve de suporte a todo o desenvolvimento, sendo a documentação para futuras alterações/iterações do projeto, uma vez que este é o primeiro protótipo da aplicação desenvolvida.

Foram identificadas duas componentes distintas na implementação da aplicação *Eventhis*, uma claramente funcional, que diz respeito à implementação de todas as funcionalidades (use cases) previstas e outra, não funcional, que visou alterar a representação visual da aplicação, permitindo tornar a aplicação mais apelativa em duas páginas da aplicação, como será explicado.

## Componente Funcional

### Vista de Visitantes

Na vista de visitantes (Guests) são disponibilizadas todas as funcionalidades relacionadas com a pesquisa e visualização de eventos e com a autenticação e criação de mais registo na aplicação.

Como foi explicado em etapas anteriores a pesquisa de eventos disponível para os utilizadores da aplicação é multicritérios, sendo possível pesquisar no título ou descrição de um evento e ainda restringir os resultados a um dos tipos existentes.

Abaixo temos a representação da página de pesquisa em *Webratio*, podemos verificar que são usadas duas *Units*, a Tipos e a Resultados, e o formulário Pesquisar Eventos. A *Unit* Tipos é um seletor que permite extrair da base de dados todos os tipos de eventos possíveis, sendo os mesmos usados para construir o formulário Pesquisar Eventos, mais especificamente o um campo de seleção (TipoEvento) que permitirá ao utilizador especificar um tipo para pesquisa.

Quando um utilizador da aplicação Web submete o formulário, a *Unit* Resultados apresenta os resultados da pesquisa, disponibilizando para todos os eventos resultado da pesquisa um link para a página do respetivo evento, a *Unit* Resultados é um *PowerIndex*.

Pode-se verificar que a *Unit* Resultados pode receber o identificador de um tipo e texto que é procurado na descrição ou título dos eventos do tipo indicado, se indicado, pois caso não o seja a pesquisa não é para todos os eventos disponíveis.



Figura 7 Página de Pesquisa de Eventos.

É na vista de visitantes que está disponível a página de autenticação, que está disponível para acesso a partir de qualquer página desta vista, uma vez que a página autenticação é uma *Landmark*, como se pode verificar a partir pela Figura 8.

Na página de autenticação é disponibilizado um formulário (Login) no qual o utilizador pode inserir o seu username e password, podendo realizar o seu login, para tal o utilizador tem que submeter o formulário que será encaminhado para uma *Login Unit*, onde serão validados os dados inseridos.

Caso o login seja bem-sucedido, o utilizador é encaminhado para a vista de utilizadores registados, no caso de este ser um utilizador registrado, ou para vista de administrador, no caso de o mesmo ser um administrador. Tal foi especificado a partir dos group\_id atribuído aos diferentes utilizadores, sendo que o módulo associado a um utilizador simples e a um administrador é diferente, tal pode ser verificado pela tabela 1, aos módulos foram por sua vez atribuídos os identificadores das vistas respetivas (tabela 2).

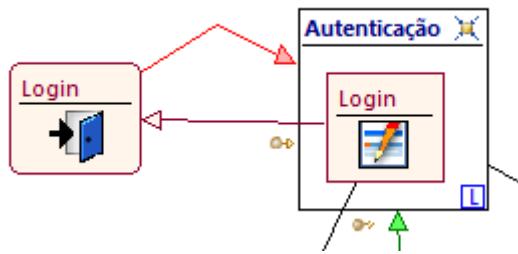


Figura 8 Página de Autenticação.

oid	groupname	module_oid
1	registados	1
2	administradores	2

Tabela 1 Tabela grupo.

oid	moduleid	modulename
1	sv4	registados
2	sv1	administradores

Tabela 2 Tabela module.

Um utilizador não registado pode seguir um link na página de autenticação que o levará para uma página de registo que está contida na área de registo, nesta página existe um formulário de registo que poderá ser submetido a uma *Operation Unit* que tratará de introduzir o novo utilizador no sistema, a referida *Operation Unit* foi denominada AddUtilizador e associada à entidade Utilizador, tal é verificável na imagem abaixo.

No caso de o registo ser bem-sucedido, o novo utilizador é encaminhado para a página de autenticação onde poderá realizar login.

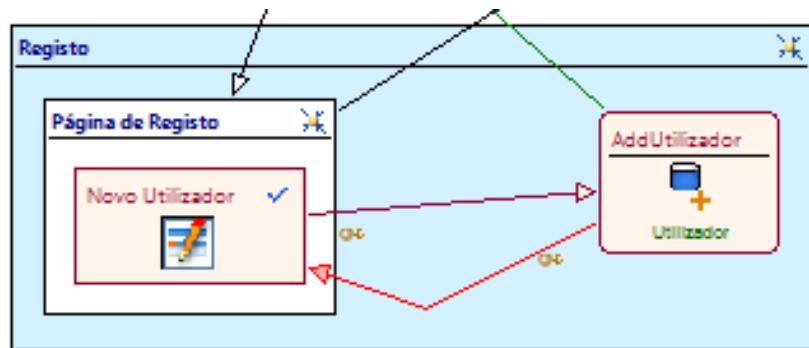


Figura 9 Área de Registo de Utilizadores.

A área de visualização de eventos tem duas páginas, a página principal para visitantes e a Página do Evento. A página principal para Visitantes tem um formulário e um *Power Index*, no entanto, tal não é usado na aplicação final, foi apenas para realizar testes, estas duas *Units* serviriam para o utilizador indicar uma zona geográfica, usando o formulário Zona, e ser-lhe-iam mostrados, usando o *Power Index* (MostrarEventos), os eventos da zona indicada. Na aplicação final acabamos por substituir estas duas *Units* por um mapa, provido pela API do *Google Maps* e por um *Werb Service* que criámos para extrair da base os eventos da zona do Mapa em visualização, tal será explicado totalmente na secção 2.1.4.

Partindo da página principal é possível ir para a página de um evento escolhido, aí é disponibilizada toda a informação existente relativa ao evento, usando a *Data Unit Evento*, e aos seus comentários com a *Power Index Unit Comentários*. Como só os utilizadores registados é que podem comentar eventos não foi dada essa opção na Página do Evento da vista de visitantes.

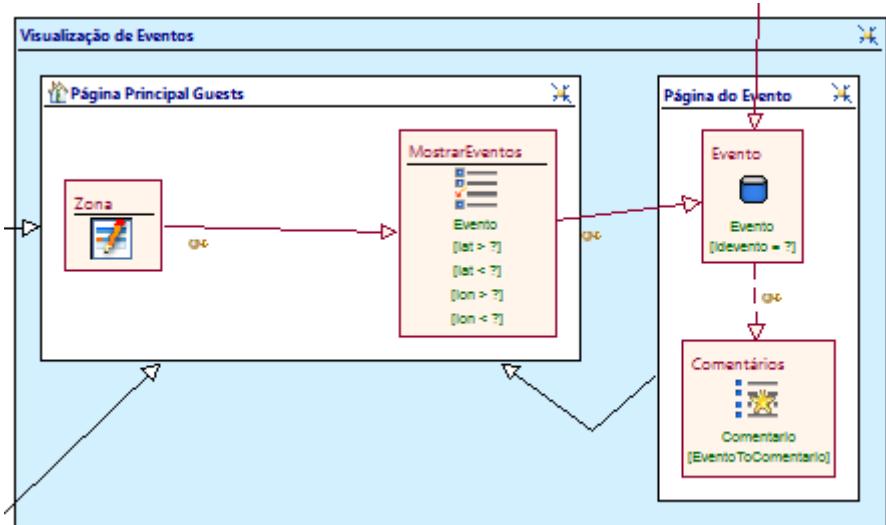


Figura 10 Área de Visualização de Eventos.

Foi criado um projeto de estilo para alterar *Power Index Unit* comentários, pois a representação gerada pelo *Webratio* não era muito agradável, essa componente da implementação é claramente não funcional, pelo que será explicada com maior detalhe na secção 2.2.

### Vista de Utilizadores Registados

Todos os utilizadores que acedem à vista de utilizadores registados já estão devidamente autenticados, pelo que já é possível aceder aos seus registo. Os eventos marcados como favoritos pelos utilizadores são guardados na base de dados como uma associação entre o utilizador e o evento em causa, é para permitir ao utilizador a consulta dos eventos que marcou como favoritos que serve a página Favoritos, cuja implementação/modelação usando o *Webratio* pode ser apreciada na Figura 11.

Para apresentar a um determinado utilizador os eventos que marcou como favoritos temos que identificar o utilizador autenticado, para tal foi usada uma *Get Unit* que foi denominada GetLoggedUser que devolve o identificador do utilizador autenticado na aplicação. Como é óbvio os eventos que já terminaram não têm interesse para os utilizadores, logo torna-se necessário só mostrar aos utilizadores os eventos marcados como favoritos que ainda não terminaram. Daí ter sido necessária a *Time Unit* DataHoraActual para filtrar na *Index Unit* Favoritos os eventos a mostrar.

A *Index Unit* Favoritos mostrará todos os eventos marcados pelo utilizador autenticado que ainda não terminaram, a partir dos eventos mostrados será possível seguir um link para a página do evento em que é possível aceder a toda a informação disponível relativa ao evento.

ou seguir um link que leva a que o evento seja desmarcado como favorito, sendo nesse caso usada a *Disconnect Unit* denominada DesmarcarFavorito, que necessita receber a identificação do evento a ser desmarcado (idevento) e do utilizador que o está a desmarcar (oid).

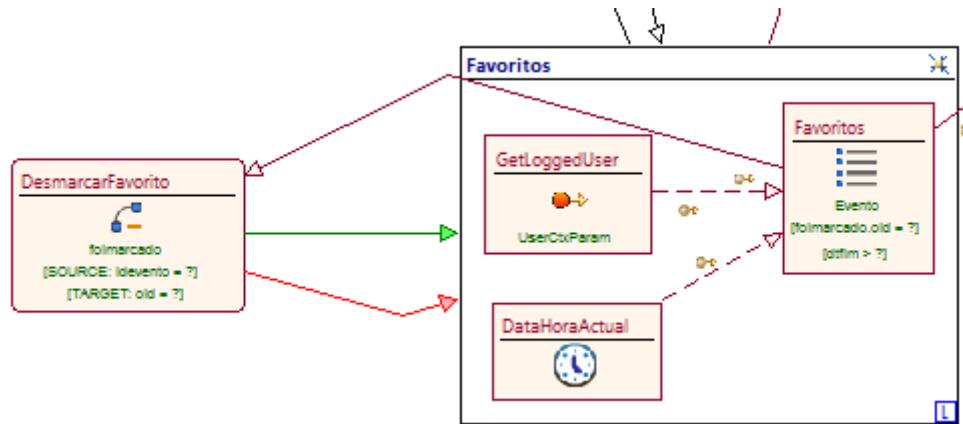


Figura 11 Página de (Eventos) Favoritos.

Na área de Edição de Registo está disponível a página que permite ao utilizador autenticado editar a sua informação de registo, que é justamente a página Página de Edição de Registo.

O *Seletor Unit* UtilizadorAutenticado é usado para preencher o formulário EditarUtilizador com a informação atual do utilizador, neste formulário o utilizador poderá com proceder a alterações ao seu registo. Para as alterações efetuadas serem refletidas nos dados da base de dados é necessário que os dados do formulário sejam enviados para a *Operation* AlterarUtilizador, sendo as alterações aplicadas com sucesso, o utilizador é redirecionado para a página principal da vista de utilizadores registados, caso contrário é enviado novamente para a página Edição de Registo.

Para que o *Seletor UtilizadorAutenticado* consiga extrair da base a informação relativa ao utilizador que está autenticado é necessário usar GetLoggedUser, que é uma *Get Unit*, para identificar o utilizador que está a utilizar a aplicação no momento.

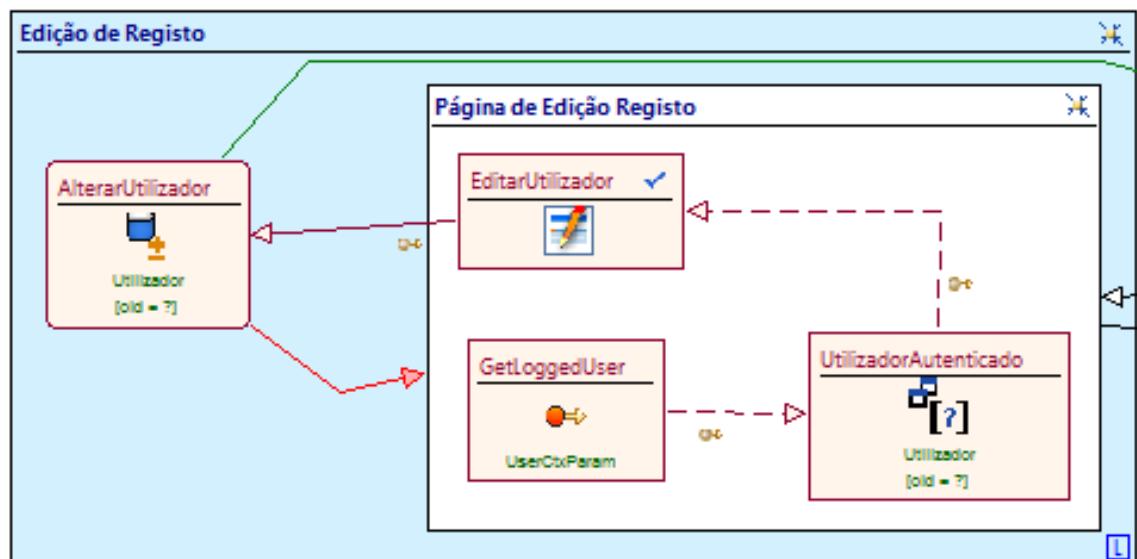


Figura 12 Área de Edição de Registo.

A área de Visualização de Eventos é a mais importante da aplicação, uma vez que é na mesma que estão concentradas a maior parte das funcionalidades que permitem a interação entre utilizadores e eventos registados no sistema.

Existem duas páginas nesta vista área, a página principal para utilizadores registados e a página do evento, na página principal para utilizadores registados é identificado o utilizador autenticado usando a *Get Unit GetLoggedUser* e são apresentados os dados iniciais, que são os tuplos latitude longitude que identificam a área geográfica que o utilizador definiu como preferencial no ato de registo, sendo essa informação usada para mostrar os eventos disponíveis na sua área preferencial.

Como já foi referido, nas páginas principais os eventos são mostrados num mapa, logo as Units DadosIniciais e MostrarEventos não vão ser usadas, pelo que são escondidas usando código do lado cliente (Javascript), no entanto a informação dada pela *Data Unit DadosIniciais* é usada para situar o mapa provido pela API da Google na zona preferencial do utilizador. A *Index Unit MostrarEventos* foi inserida exclusivamente para testes, uma vez que representa a aplicação final na plenitude, a partir dos eventos representados no mapa poderá navegar-se para a página do evento, tal como é representado na Figura 13 mas usando a *Index Unit MostrarEventos*.

Na Página do Evento é apresenta toda a informação relativa ao evento identificado através do parâmetro que chega, via link, à *Data Unit Evento* e depois é distribuído por todas as *Units* que dele necessitam, como é o caso da *Power Index Unit Comentários* que apresenta todos os comentários efetuados relativos ao evento em causa.

Há operações que só podem estar disponíveis para determinados utilizadores, por exemplo o link que permite ir para a página de alteração/remoção do evento só deve ser apresentados ao utilizador criador do mesmo, tal foi fácil de fazer uma vez que a *DataUnit Evento* tem o identificador do utilizador que criou o evento, pelo que se criou uma variável de página que compara o identificador do utilizador autenticado e a ver a página no momento com o identificador do utilizador que criou o evento. Depois usou-se uma condição de expressão que mostra o link que permite aceder à página de edição do evento caso o utilizador seja o criador e esconde o link caso não o seja.

Contudo, existiu uma situação em que não estava disponível em nenhuma *Unit* os dados necessário para resolver o problema, essa situação diz respeito ao problema de identificar se utilizador que está a visualizar a página do evento já o marcou como favorito ou

não. Devido ao explicado foi que introduzida a *Selector Unit* EstadoFavorito, que a partir do identificador do evento e do identificador do utilizador a usar a aplicação, que advém da *Get Unit* GetLoggedUser. A partir do output do EstadoFavorito já foi possível criar uma variável e uma condição de expressão que mostrasse o link desmarcar favorito no caso de o evento em visualização ser favorito do utilizador ou mostrar o link marcar como favorito na situação inversa.

Na página do Evento foi introduzido um formulário (Novo Comentário) que permite ao utilizador que está a visualizar o evento criar comentários, a informação que está no formulário vai, quando este é submetido para uma Operation Unit que insere o novo comentário (CriarComentário), no entanto é necessária mais informação do que a que está no formulário, pois é necessário o identificador do utilizador que está a criar o comentário, o identificador do evento ao qual o comentário vai ser associado e data-hora de criação do mesmo, os identificadores estão todos disponíveis em *Units* já referidas, no entanto a necessidade da data-hora da criação levou a introduzir-se um Time Unit TempoComentarios.

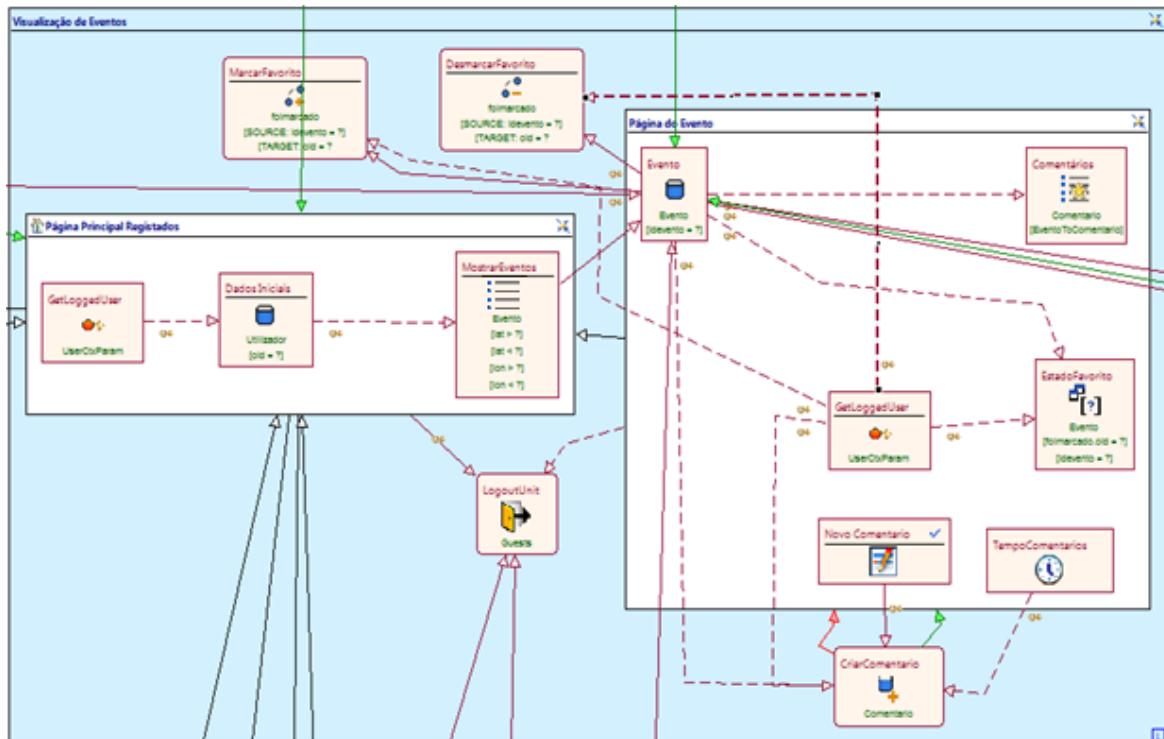


Figura 13 Área de Visualização de Eventos.

Na área de gestão de eventos estão disponíveis as páginas relacionadas com a criação, alteração e remoção de eventos. As Operation Units usadas são as RemoçãoEvento, ModificarEvento e CriarEvento. Tanto na página de Edição de Evento como na de Criar Evento foi necessário utilizar um *Seletor Unit* para extrair os tipos de eventos disponíveis da base.

Poderiam ter sido passadas as informações relativas ao evento a ser alterado para o formulário EditarEvento, no entanto, quando tal é feito por link a codificação dos caracteres acentuados fica alterada, pelo que foi utilizada a *Seletor Unit* DadosEventos.

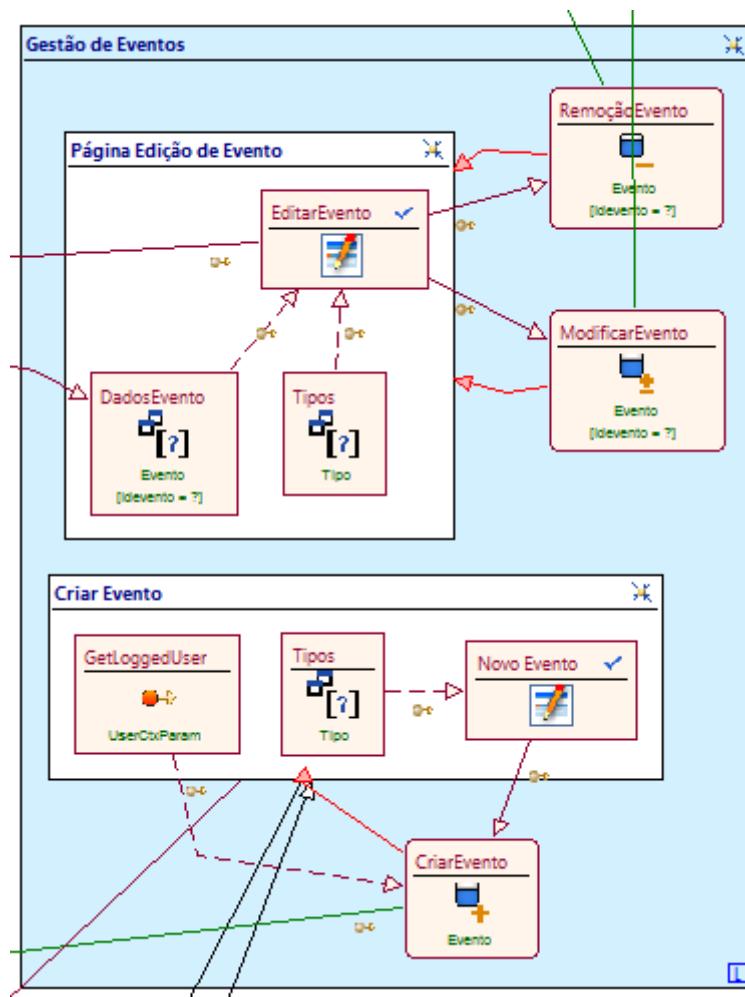


Figura 14 Área de Gestão de Eventos.

## Vista de Administrador

Os administradores da aplicação Web têm a seu cargo o envio de newsletters para os utilizadores da aplicação informando acerca de novos eventos.

Na página principal da vista para administradores podem ser vistos os eventos para os quais foi solicitado o aparecimento na newsletter seguinte, tal é realizado usando a *Index Unit* Eventos da Newsletter, o formulário de enviar newsletter só deve estar visível no caso de existirem eventos para serem publicitados, pelo que foi necessário criar uma variável e uma condição de expressão. O valor da variável de página criada advém da *Seletor Unit* NumeroEventosNewsletter, que como o próprio nome indica conta o número de eventos que estão disponíveis para fazer uma newsletter, caso não esteja nenhum a condição de expressão será “false” e por isso o formulário que permite enviar a newsletter não será mostrado.

Depois do envio da newsletter é necessário desmarcar todos os eventos que estavam marcados para serem enviados na próxima newsletter, uma vez que esses já foram publicitados, para tal usa-se uma *Query Unit* que executa na base de dados o código SQL que desmarcará os eventos. O Email é enviado usando a *Mail Unit* Enviar Email.

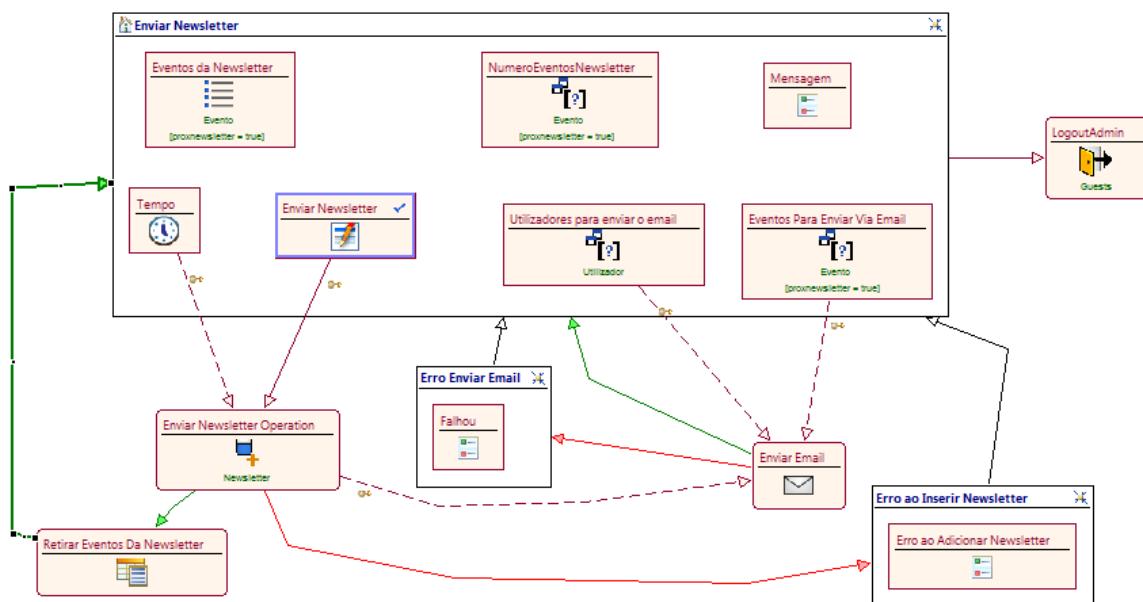


Figura 15 Páginas da Vista de Administrador.



## *Serviço de Eventos e Google Maps API*

Como foi explicado durante as fases anteriores do projeto os eventos devem ser visualizados e criados contextualizados com a localização geográfica dos mesmos, pelo que é imperativo usar uma API de Mapas disponível gratuitamente, de entre as existentes foi escolhida a Google Maps API pela facilidade de uso e informação disponibilizada.

Todo o código relativo à introdução de mapas nas páginas principais, criação e edição de registo de utilizador e criação e edição de eventos só foi introduzido depois de aplicação estar toda construída, pois tal é feito usando código Javascript e seria mais complexo fazê-lo no Webratio.

Seria impossível carregar todos os eventos disponíveis para um mapa, pelo menos numa aplicação real em que o número de eventos fosse muito elevado, pelo que surgiu a necessidade de utilizar AJAX (Asynchronous JavaScript and XML) para pedir ao servidor os eventos existentes nas zonas em visualização no mapa. Assim, se um utilizador só está a visualizar a cidade de Braga só é feito download de informação relativa a eventos na área em causa. Como é óbvio a informação resposta do servidor tem que ser entendida pelo cliente, pelo que tem que ser possível tratar o resultado usando Javascript.

As necessidades identificadas nos parágrafos anteriores levaram à necessidade de criar um Serviço Web que conseguisse responder com informação relativa a eventos de uma determinada área indicada por parâmetro, sendo a resposta numa linguagem passível de processamento pelo browser cliente, no caso XML, mas poderia ser JSON ou outra que quiséssemos criar.

O Serviço Web criado foi implementado usando uma *Solicit Unit* que é de forma simplista uma *Unit* que recebe um pedido, ao qual foi dado o nome de Area, os pedidos tem como parâmetros dois tuplos GPS que especificam uma área geográfica. Os parâmetros são enviados para uma XML Out Unit que usa um Seletor para extraír os eventos que ainda não terminaram e se situam na área parâmetro e converte em XML sendo depois enviado como resposta para o browser que fez o pedido que o interpretará.

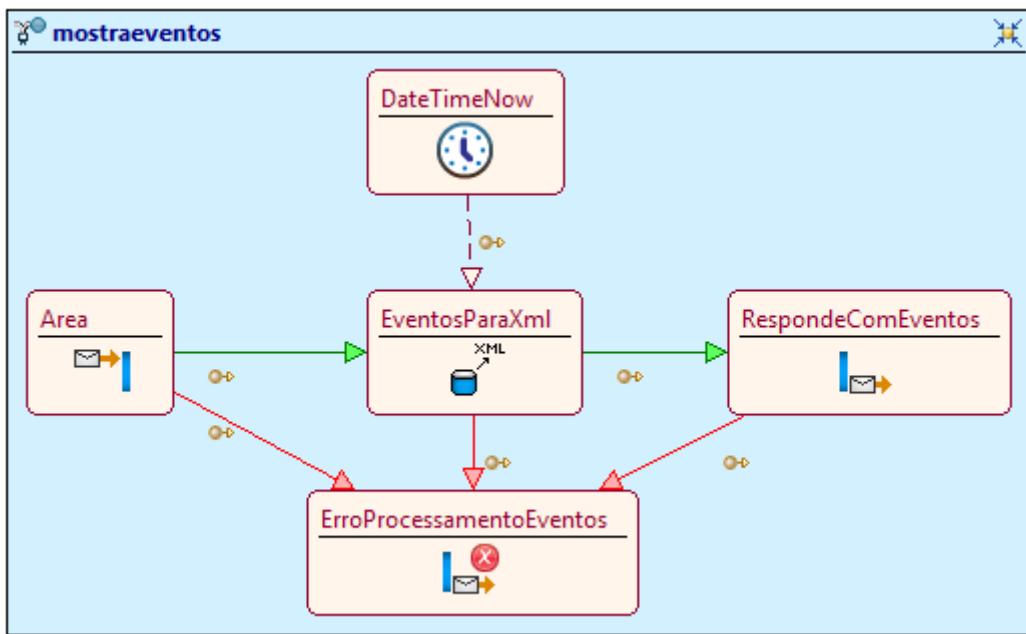


Figura 16 Serviço Web (REST) que permite retornar os eventos para uma área indicada.

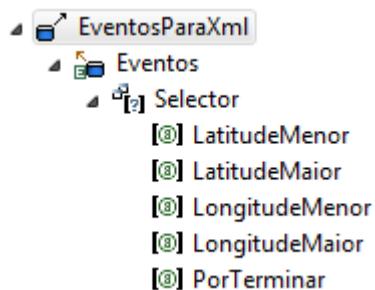


Figura 17 XML Out Unit que extrai os eventos de uma determinada área usando uma Selector Unit e converte para XML.

Podemos observar na imagem abaixo um pedido REST feito ao servidor, neste caso usando o verbo HTTP GET e a resposta XML devolvida para a área indicada.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<?xml version="1.0" encoding="UTF-8"?>
<Root>
  <Entity id="ent5" xmlOutEntityName="Eventos">
    <Instance>
      <Attribute id="att19">9</Attribute>
      <Attribute id="att25">41.61202636718747</Attribute>
      <Attribute id="att24">42.3868351423178</Attribute>
      <Attribute id="att22">Festa do Além Mar</Attribute>
      <Attribute id="role2$att6">4</Attribute>
    </Instance>
  </Entity>
</Root>

```

Figura 18 Exemplo de pedido e output XML do serviço Web.

Como o Javascript é melhor a tratar JSON (JavaScript Object Notation), pois é a notação que utiliza para representar os próprios objetos, foi decidido usar a API XMLObjectifier que permitiu converter o XML em JSON, sendo posteriormente tratado e introduzido no mapa.

No anexo 7 pode-se consultar o código *JavaScript* usado para dar suporte ao explicado nesta secção, estando o mesmo totalmente comentado.

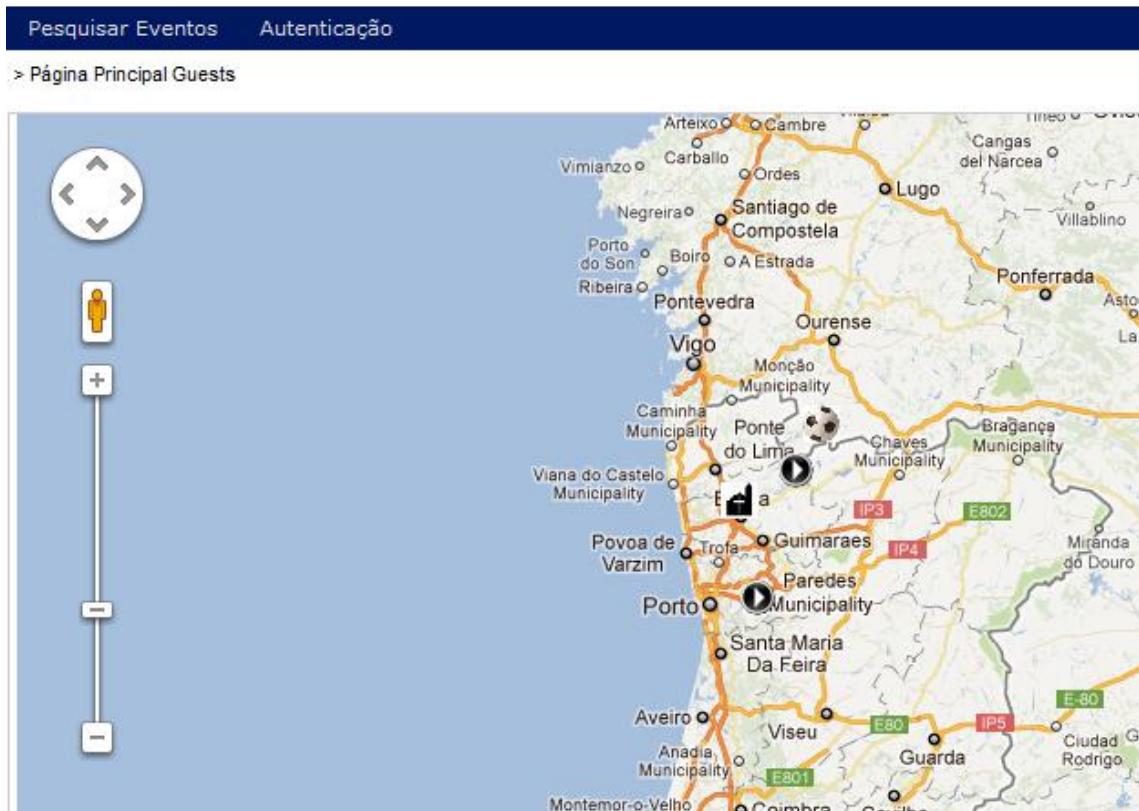


Figura 19 Resultados que advém do serviço Web.

## Componente Não Funcional

Os comentários dos utilizadores ficavam por defeito na horizontal, como a imagem ficava grande e o título do comentário e corpo de seguida, tudo na mesma linha, tal não era muito estético, pelo que foi decidido proceder à criação de um Projeto de Estilo para criar uma nova apresentação que pudesse ser aplicada na *Power Index Unit* responsável por apresentar os comentários dos utilizadores.

O resultado gráfico da alteração pode ser visualizado na figura introduzida abaixo, o código do template usado pode ser consultado no anexo 8.

The screenshot displays a user interface for managing comments. At the top, there is a header labeled "Comentários". Below it, there are two comment entries. The first entry features a small thumbnail image of a man, followed by the text "Muito interessante.". The second entry also has a thumbnail image and the text "Gostava imenso de passar por lá!!!". Below this section is another header labeled "Novo Comentario". Underneath it, there are two input fields: one for "Título" (Title) and one larger one for "Corpo" (Body). At the bottom of the form is a button labeled "Comentar" (Comment).

Figura 20 Alteração do Estilo para a introdução de imagens nos comentários.

## Trabalho Futuro

Para todos os formulários criados na implementação da aplicação foram definidas regras de validação, tendo sido todas testadas. Nos formulários que requeriam introdução de coordenadas GPS foram introduzidos mapas, sendo possível selecionar a área geográfica ou ponto geográfico a partir do mapa, ou introduzir diretamente a coordenada nas caixas de texto, tal pode ser verificado na imagem abaixo inserida.

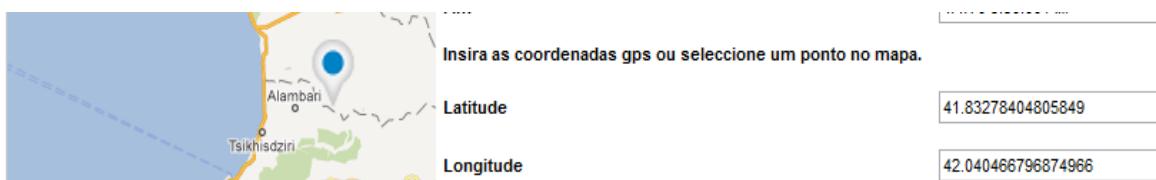


Figura 21 Exemplo de seleção de coordenada geográfica para a realização de um evento.

A possibilidade de criar uma Map Unit foi discutida, no entanto, seriam necessários conhecimentos que demorariam a adquirir, nomeadamente de *Java Server Pages*, *Groovy Script*, *Expression Language* e *JavaServer Pages Standard Tag Library*, algo que não poderia ser realizado em tempo útil, no entanto, seria muito mais interessante, pois *Unit* criada poderia ser utilizada em muitos projetos e facilitaria o *debug* e desenvolvimento da aplicação, pelo que tal tarefa é tida como primordial para evolução futura do trabalho desenvolvido.

Foram implementadas todas as funcionalidades previstas para o projeto, no entanto o modelo de negócio para a aplicação Web teria que ser completado, uma vez que não foram incluídos os pagamentos para quem quisesse que o evento fosse publicitado via newsletter.

A aplicação precisa de ser melhorada no que diz respeito ao *design* da mesma, pois devido a limites temporais, foram efetuadas poucas alterações à camada de visualização, tal é cada vez mais importante e merece atenção em futuras alterações/evoluções.

O administrador da aplicação Web deveria ter mais funcionalidades que lhe permitissem gerir a informação disponibilizada, podendo apagar informação ofensiva ou pejorativa, tal seria algo a evoluir na aplicação, mas implicaria também uma nova iteração sobre o processo de desenho, uma vez que tal não fez parte dos requisitos, pois a aplicação Web estava a ficar com demasiadas funcionalidades.



## **Notas Finais - Implementação *Webratio***

Durante a fase de implementação ficou claro que a abordagem seguida no estudo, análise e desenho da aplicação a implementar foi bem conseguido, uma vez que facilitou imenso a implementação.

A implementação da aplicação Web usando o *Webratio* limitou-se à representação em Web ML do que já tinha sido estudado e pensado na fase de *Coarse Design* da aplicação, o que foi de maior complexidade de implementação foram os mapas, que levaram à necessidade de código cliente e à criação de um serviço web, que não têm representação em *Coarse Design* daí que tal ainda não estivesse definido e a componente não funcional da implementação, que levou à criação de um projeto de estilo que melhorou a apresentação dos comentários dos eventos.

A aplicação Web é apenas um protótipo, todas as funcionalidades estão implementadas, no entanto é necessário aperfeiçoar muitas funcionalidades e ter um cuidado especial com a camada de apresentação da aplicação Web.

Desenvolver este projeto foi uma experiência muito enriquecedora, tendo ficado demonstrada a facilidade com que se implementam aplicações Web orientadas aos dados usando o *Webratio*, mas também ficou patente que quando é necessário utilizar código cliente e mecanismos como o AJAX, a experiência torna-se mais complexa, pois deve ser colocado no final do desenvolvimento em *Webratio*, sendo que uma nova geração de código apaga todo o código Javascript já incluído.

## Implementação PHP

Depois de concluída a implementação da aplicação Eventhis em *Webratio*, que era um requisito do módulo, transitamos para a nossa implementação da aplicação. Conforme foi referido e justificado na secção Tecnologias, a linguagem escolhida para a implementação da componente servidor da aplicação foi o PHP em conjunto com o frameworks *Yii*.

A implementação PHP foi tida como a principal, uma vez que era o objectivo inicial do projeto. Para além da utilização do PHP e do frameworks *Yii*, também já tinha sido definido que seria usado Javascript para implementação de componentes de cliente.

Nesta secção pretende-se elucidar/demonstrar todo o processo de produção da versão final da aplicação, ficando o mesmo totalmente documentado, para que fosse possível recomeça-lo, alterá-lo a qualquer momento sem percalços.

Foi a primeira experiência de desenvolvimento de aplicações Web com este tipo de arquitetura, que já não é assim tão simples, uma vez que já inclui serviços Web. Ao longo do desenvolvimento do projeto, nomeadamente da implementação da aplicação usando a ferramenta *Webratio*, foram ficando claras algumas características a melhorar, o que acaba por ser normal, uma vez que a referida aplicação esteve para a aplicação final como um protótipo.

Como é óbvio, tentamos introduzir na aplicação algumas das características menos positivas que detectamos na aplicação desenvolvida com o *Webratio*, algumas características relevantes são:

- Os títulos nos comentários dos eventos não são necessários, pois não é sugestivo, a maioria das aplicações Web de hoje usam apenas o texto do comentário e o username do utilizador que o fez, pelo que foi decidido retirar o título ao comentário;
- Uma vez que foi solicitada a fotografia de perfil ao utilizador no ato de registo, é possível utilizá-la melhor, tendo sido decidido apresentar a fotografia em qualquer página para qualquer utilizador autenticado;
- Hoje em dia a maioria das aplicações mais evoluídas permitem pesquisas em tempo real, isto é, já não é preciso um refreshamento da página para ver os resultados. Tal é feito usando chamadas assíncronas ao servidor (AJAX), a aplicação desenvolvida em

*Webratio* tal era um pouco complicado de implementar, uma vez que era dado pouco suporte para este tipo de procedimentos;

- Marcar um evento como favorito será certamente uma funcionalidade muito utilizada na aplicação, na aplicação anterior, marcar um evento como favorito implicava um refreshamento da página, tal também será, ao nível da aplicação PHP, substituído por chamadas assíncronas ao servidor.

As características apresentadas são claramente benéficas para os utilizadores da aplicação, pelo que, uma vez identificadas, não poderiam deixar de ser introduzidas na aplicação final.

## Framework *Yii*

O frameworks *Yii* foi o escolhido como infraestrutura de suporte ao desenvolvimento da aplicação Web em PHP, a versão usada foi a 1.1. Este *framework* é distribuído sobre a licença BSD, o que significa que as aplicações desenvolvidas com o mesmo podem ser de código aberto ou fechado (proprietárias).

O *Yii* é relativamente recente, tendo sido lança a primeira versão no final de 2009, pelo que ainda não existem muitas extensões, demos e templates. Contudo, como esta comunidade é muito ativa e a documentação disponibilizada no Web site (ver Referências Web) é muito completa, tal não é um grande entrave. Existiram apenas algumas questões devido à inexistência de um *Time Picker* (componente de interface necessário para possibilitar ao utilizador a introdução de data-hora de inicio e de fim dos eventos) e outras extensões que serão referidas ao longo desta secção.

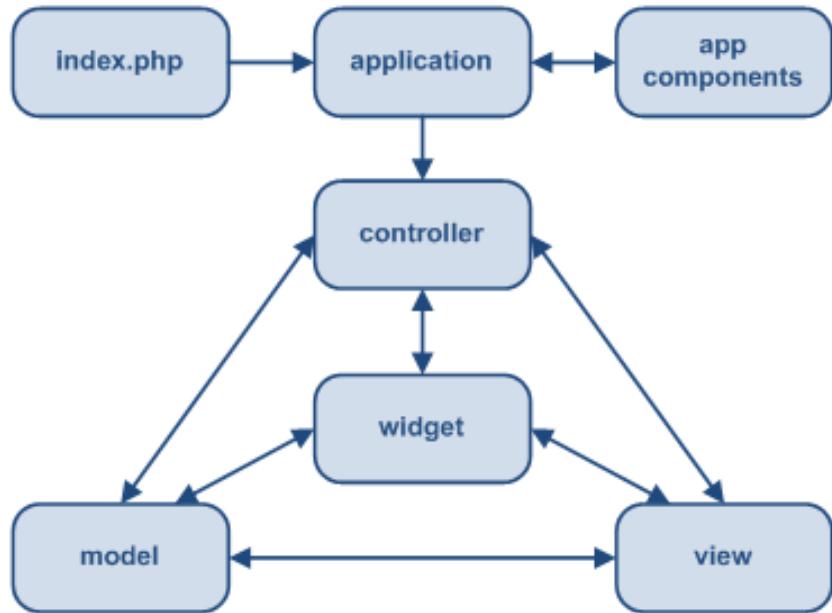


Figura 22 Arquitetura do Framework Yii.

A possibilidade de gerar código e de desenvolver a aplicação usando a mini-arquitetura MVC (Model-View-Controller), que permite a escrita de código mais independente, que pode ser reutilizado com mais facilidade, uma vez que a camada de interface gráfica e de lógica são ligadas por uma camada que controla, foi algo muito atrativo que levou a que a escolha final fosse o *Yii*.

O código que foi gerado diz apenas respeito às operações CRUD (*Create, Read, Update* e *Delete*) sobre as entidades centrais. Como é óbvio este código é útil, no entanto, foi necessário editar o código para utilizá-lo na aplicação. Veja-se, uma página de um evento tem informações sobre o evento, comentários e utilizador, se autenticado, mas o *Yii* gera um *controller*, um *model* e uma *view* por operação CRUD de cada entidade, estes *controllers* não podem ser usados por uma página como a de um evento, pois este *controller* tem que interagir com vários *models*, nomeadamente o das entidades evento, comentário e utilizador.

Algumas das características mais interessantes do *Yii* são enunciadas de seguida:

- Model-View-Controller (MVC);
- Database Access Objects (DAO), Query Builder, Active Record, DB Migration;
- Validação de Formulários;
- Suporte para autenticação;

- Segurança;
- Documentação;
- Widgets AJAX;
- Temas;
- Geração automática de código a partir da base de dados.

Por tudo o enunciado o *Yii* foi uma excelente escolha, pois permitiu o desenvolvimento de uma aplicação bastante completa em tempo útil, tendo como grande contra a dependência do framework de desenvolvimento, o que significa que se o projeto *Yii* fosse abandonado a aplicação teria que ser reescrita ou estaria a usar um framework potencialmente obsoleto.

## IDE – Netbeans

No desenvolvimento de projetos com alguma envergadura é muito útil usar um ambiente integrado de desenvolvimento, pois estas ferramentas disponibilizam funcionalidades que tornam a escrita de código bastante mais rápida, facilitando, ainda, o debug das aplicações.

Daí que no início do desenvolvimento da implementação tenhamos procurado um IDE com algumas características que proporcionassem auxílio ao desenvolvimento Web em PHP e que permitissem a configuração do framework escolhido (*Yii*), para que assim fosse possível ter acesso à API do *Yii* durante o desenvolvimento da aplicação, fazendo uso de mecanismos como o de autocomplete de código.

A escolha acabou por recair sobre o *Netbeans*, pois tem uma componente Web muito forte, existe uma versão somente para o desenvolvimento em PHP, permite a configuração de servidores HTTP, a configuração é simples e permite adição de bibliotecas, nomeadamente do *Yii*.

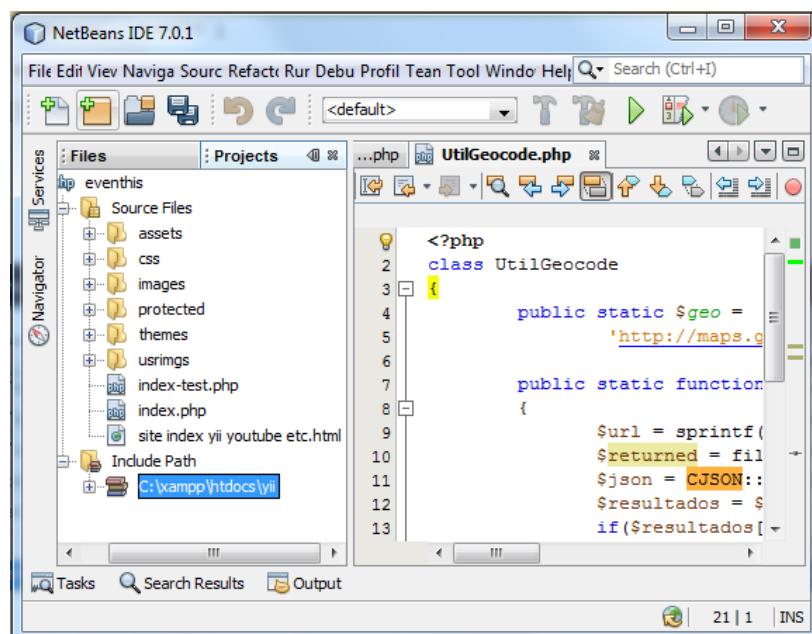


Figura 23 IDE - NetBeans com a biblioteca *Yii* incluída em Include Path.

## Componente Funcional

### Vista de Visitantes - Especificidades

A vista de visitantes é, obviamente, para utilizadores não autenticados na aplicação, a partir desta vista o visitante pode ter acesso a várias páginas, a principal, onde pode consultar os eventos no mapa, a de pesquisa de eventos e à página de registo.

O formulário que permite a autenticação está disponível em qualquer página desta vista, tendo sido criada um componente que pode ser reutilizado em vários projetos. Assim, alterando o ficheiro (views/layouts/main) pudemos fazer render da utilidade (componente de autenticação) em todas as páginas desta vista.

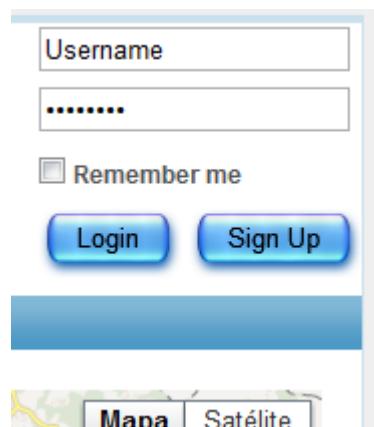


Figura 24 Componente de autenticação.

```
<?php
    $modelFL = new LoginForm();
    $this->renderPartial(
        '/utils/LoginUtil',
        array('action'=>'/site/login',
              'model'=>$modelFL
        )
    );
?>
```

Figura 25 Código necessário para a utilização do componente de autenticação.

Na página principal para visitantes, os utilizadores podem ver os eventos no mapa, no entanto, a posição inicial do mapa não coincide com a sua área de interesse, pois não é possível prevê-lo, tal poderia ser melhorado usando o IP do utilizador para determinar a área mais próxima ao mesmo, mas poderia, ainda assim, ser erróneo.

A partir da página principal, usando os ícones dos eventos, os visitantes podem navegar para as páginas dos eventos em causa, no entanto na página dos eventos estão novamente mais limitados que os utilizadores devidamente autenticados, não podendo escrever comentários relativos aos eventos, no entanto podem aceder a todas as informação relativas ao evento em causa.

Ao nível das páginas de pesquisa não existe nenhuma limitação, sendo a mesma igual para utilizadores autenticados e visitantes.

## *Vista de Utilizadores Autenticados - Especificidades*

Os utilizadores autenticados têm acesso a bastantes mais funcionalidades que os utilizadores não autenticados, o que é normal, pois relativamente ao utilizador autenticado já é possível guardar estado.

Na página principal o mapa é iniciado na zona que o utilizador definiu como preferencial no ato de registo na aplicação Web, sendo o restante comportamento igual ao da página principal de visitantes.

Na página do evento os utilizadores autenticados podem escrever comentários, marcar/desmarcar evento como favorito e, se for o utilizador criador do evento em consulta, pode alterar o evento ou apaga-lo (descativa-lo).

A partir de qualquer página da vista de utilizadores autenticados é possível aos utilizadores visualizarem uma mensagem de boas vindas, a fotografia que submeteram no ato de registo e o link que lhes permite realizarem logout. Tal é realizado usando o componente LoginUtil já apresentado na secção anterior.



Figura 26 Menu da aplicação na vista de utilizadores autenticados.

A partir do menu (Figura 26) um utilizador autenticado pode aceder a funcionalidades como a de criar evento, consultar eventos favoritos, alterar dados de registo e ver os eventos criados. Como é óbvio estas funcionalidades implicam que se guarde estado do utilizador, pelo que nunca seria possível proporcioná-las a um visitante.

## *Vista de Administrador - Especificidades*

A vista de administrador tem funcionalidades muito específicas que não podem ser usadas por mais nenhum tipo de utilizadores da aplicação, nesta vista o administrador pode

gerir os utilizadores, comentários, eventos e newsletters, sendo que na gestão dos utilizadores, eventos e comentários pode apagar, pesquisar e visualizar todos os atributos, mesmo chaves primárias.



**Figura 27** Menu da aplicação para a vista de Administradores.

Na gestão das newsletters está o modelo de negócio do projeto, em que o administrador pode visualizar os eventos para os quais foi solicitado o aparecimento na newsletter do sistema, que num caso real implicaria pagamento, atribuir um título à newsletter e enviar a newsletter.

## *Componente googlemap*

Os mapas disponibilizados pela API Javascript da Google são fundamentais para a aplicação Web dando-lhe mais vida. Tais mapas não usados em várias páginas, na página inicial para representar os eventos existentes no mapa, na página de criação de um evento para assinalar a localização do mesmo e na página inicial do utilizador para que o novo utilizador possa indicar a sua zona geográfica preferencial.

Sendo necessário a representação de um mapa em tantas páginas, não seria muito inteligente andar a replicar o código necessário para isso, até porque é possível que um componente desta natureza venha a ser necessário em futuros projetos, pelo que foi desenvolvido um módulos que inicializa um mapa, sendo possível configurar vários aspectos, desde o nome que a variável do mapa vai ter, ao identificador da “div” que o possui, a largura e altura, os eventos que se pretendem registar e que função Javascript os vai tratar e mais opções.

```
<script type="text/javascript">
// variável javascript que possibilitará o acesso ao mapa
var <?php echo $varnamemap;?>;
function initialize() {
    // inicialização do centro do mapa
    var latlng = new google.maps.LatLng(<?php echo $lat;?>,<?php echo $lon; ?>);
    // construção das opções iniciais do mapa
    var myOptions = {
        // zoom
        zoom: <?php echo $zoom; ?>,
        // centro
        center: latlng,
        // tipo de mapa
        mapTypeId: google.maps.MapTypeId.ROADMAP
    };
    // inicialização do mapa usando a API da google
    <?php echo $varnamemap;?> = new google.maps.Map(
        document.getElementById("<?php echo $id; ?>"),
        myOptions
    );
    // registo dos eventos que se pretendem registrar e do nome da
    // função javascript que os vai tratar
    <?php foreach ($events as $event) {
        echo 'google.maps.event.addListener('. $varnamemap .'," . $event[0]
            . '",'. $event[1] .');';
    }
    // Opcional: quem incializar esta API pode definir um função javascript
    // que só deva ser chamada no final da inicialização do mapa
    if(isset($functioncall))
    {
        echo $functioncall.'();';
    }
}
?>
}
</script>
```

Figura 28 Código do componente googlemap.

Usando este componente (Figura 28) é possível usar o mesmo código na representação de mapas em todas as páginas em que tal é necessário, sendo suficiente

chamar a função `renderPartial` especificando as opções, como tamanho, eventos, etc. É possível usar o componente sem especificar qualquer argumento, pois tal é testado e caso não sejam disponibilizados os argumentos necessários são usados argumentos por defeito.

## *UtilGeocode e imgmap*

A determinada altura no desenvolvimento do projeto ficou claro que usar um mapa em todas as páginas tornava a aplicação um pouco lenta e pesada, pelo que foi decidido usar em algumas páginas uma imagem do mapa em vez do mapa em si, complementando a informação da imagem com a informação relativa à morada.

Para tal foi necessário usar duas API's da Google, a *geocode* para solicitar informação relativa às moradas dos eventos a partir das coordenadas *gps* e a *static map* para solicitar uma imagem do mapa numa determinada posição GPS, com uma determinada marca.

```
<?php
if(!isset ($width)) $width = 350;
if(!isset ($height)) $height = 200;
if(!isset ($zoom)) $zoom = 10;
$src = 'http://maps.googleapis.com/maps/api/staticmap?center='.$latitude.','.
       $longitude.'&zoom='.$zoom.'&size='.$width.'x'.$height;
if( isset($linkMarca) )
{
    $src .= '&markers=icon:'.$linkMarca.'%7C'.$latitude.','.$longitude;
}
$src .= '&sensor=false';
echo CHtml::image($src);
?>
```

Figura 29 Código necessário para o pedir um mapa estático (imgmap).

Na configuração do link da imagem do mapa estático podem devem ser fornecidas a latitude e a longitude. Outros valores podem ser fornecidos, nomeadamente a largura e a altura e o zoom, no entanto para estas configurações existe um valor por defeito, caso os mesmos não sejam fornecidos. Tais valores estão definidos nas primeiras três linhas de código.

```

<?php
class UtilGeocode
{
    public static $geo =
        'http://maps.googleapis.com/maps/api/geocode/json?latlng=%s,%s&sensor=false';

    public static function getMorada($lat,$lon)
    {
        $url = sprintf(self::$geo,$lat,$lon);
        $returned = file_get_contents($url);
        $json = CJSON::decode($returned);
        $resultados = $json['results'];
        if($resultados[0]['formatted_address'])
        {
            return $resultados[0]['formatted_address'];
        } else {
            return 'Não definida.';
        }
    }
}
?>

```

Figura 30 Código necessário para realizar um pedido de morada.

Para pedir uma morada usando a UtilGeocode (Figura 31) é necessário fornecer a longitude e latitude do ponto, depois com essa informação é configurado o link do pedido faz-se o pedido à Google, usando a função *file\_get\_contents*. Como o resultado do serviço Web vem em formato JSON é necessário realizar a descodificação (CJSON::decode), posteriormente é retornada caso exista a morada da coordenada gps fornecida inicialmente.

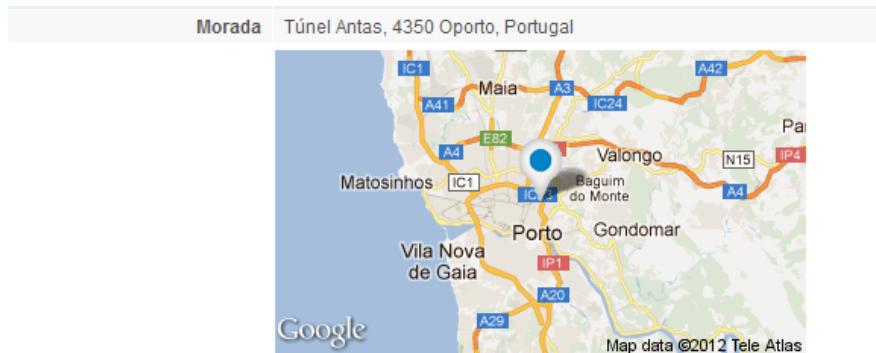


Figura 31 Resultado da aplicação do UtilGeocode e mapas estáticos.

## **Página do Evento**

A página do Evento apresenta todas as informações sobre um determinado evento, desde o seu título aos comentários de que foi alvo. Se o utilizador estiver autenticado terá oportunidade de comentar o evento usando um formulário disponibilizado no fundo da página e, para além disso, caso o utilizador esteja autenticado e seja o autor do evento em visualização terá a possibilidade de o alterar ou apagar (desativar).

Para o desenvolvimento desta página foi necessário criar várias views (`/views/pagevento/index`, `/views/pagevento/create_comment` e `_form_comment`) e um controller `PageventoController`.

As views são apenas auxiliares que representam a página, sendo o `PageventoController` que trata todos os pedidos, existindo várias ações disponíveis:

- `actionIndex`: para representar a página do evento;
- `actionMudaFavorito`: método que recebe pedidos AJAX que permitem trocar o estado do favorito;
- `actionComment`: método que permite inserir um novo comentário, associando ao utilizador autenticado e ao evento em visualização.

## Páginas Principais

A página principal é quase completamente controlada por um controller e uma view. A view é “views/site/index.php” e o controller é “controllers/EventoController.php”, no entanto, esta página não é um exemplo típico MVC, pois a página inicial comunica via um serviço web, que solicita os eventos para a nova fronteira do mapa, é claro que tal só necessário quando o utilizador altera as suas fronteiras, pelo que foi necessário registar o evento “changed\_bounds” usando a API do Google Maps.

```
$idMap = "map_canvas";
$map = "map";
$funcao = 'acertarFronteiras';
$this->renderPartial('//utils/googlemap',
    array(
        'width'=>'100%',
        'height'=>'360px',
        'id'=>$idMap,
        'varnamemap'=>$map,
        'events'=>array(
            array('bounds_changed', 'changeFronteiras'),
        ),
        'functioncall'=>$funcao
    )
);
~
```

Figura 32 Código que permite desenhar o mapa na página principal, sendo especificadas todas as opções, desde o tamanho mapa aos eventos que se pretendem registrar.

O acima exposto só é possível devido ao facto de ter sido criado o componente googlemap (secção

Componente googlemap) que permite representar um mapa que provém da API disponibilizada pela Google em todas as páginas, configurando-o sempre que necessário.

```
public function actionEventos($lat1,$lon1,$lat2,$lon2)
{
    $model = new Evento;
    // construção do critério de pesquisa para eventos da área
    $criterio = new CDbCriteria;
    $criterio->condition = "lat > $lat1 and lat < $lat2 "
        . "and lon > $lon1 and lon < $lon2 and dtfim > now()";
    $eventos = $model->findAll($criterio);
    echo CJSON::encode($eventos);
}
```

Figura 33 Método do controller EventoController que realiza a pesquisa dos eventos não terminados da área provida e devolve o resultado em JSON.

Quando um utilizador arrasta o mapa e as fronteiras mudam é chamada a função *changeFronteiras*, como pode ser verificado pelo parâmetro *events* (Figura 32), que verifica qual é a nova fronteira e envia um pedido REST para o servidor (usando AJAX), pedindo os eventos da nova área. O servidor responde em JSON (Figura 33), tendo que se tratar o output do serviço com vista a detectar os eventos que ainda não estão no mapa, para no caso de não estarem serem inseridos.

```
function changeFronteiras(){
    var ne = <?php echo $map; ?>.getBounds().getNorthEast();
    var sw = <?php echo $map; ?>.getBounds().getSouthWest();
    var lat1 = sw.lat();
    var lat2 = ne.lat();
    var lon1 = sw.lng();
    var lon2 = ne.lng();
    <?php
        echo 'var url = "'.$this->createUrl('//evento/eventos'). '&' .
            'lat1='+lat1+'&lat2='+lat2' .
            '&lon1='+lon1+'&lon2='+lon2+';';
    ?>
    var data;
    $.ajax({
        url : url,
        dataType : 'json',
        cache : false,
        success: function(data){
            inserirMarcas(data);
        }
    });
}
```

Figura 34 Função Javascript que é chamada sempre que as fronteiras do mapa mudem.

Como se pode verificar pela Figura 34 a função *changeFronteiras* utiliza a variável que da acesso ao mapa instanciado, solicita as fronteiras e constrói o link que permite realizar o

pedido ao serviço Web, sendo que parte dele é construído ainda pela componente servidor. Por fim é realizado um pedido AJAX para o url construído e em caso de sucesso é chamada a função *inserirMarcar*, que mais não faz do que inserir as marcas dos eventos no mapa e registar eventos para as mesmas, pois no caso de um utilizador clicar nas mesmas deve aparecer uma caixa com um link para a página do evento em causa.



Figura 35 Resultado da combinação do mapa com serviço Web, Javascript e AJAX.

```
    ...
    marcas[id].setDraggable(false);
    marcas[id].setClickable(true);
    marcas[id].info = new google.maps.InfoWindow();
    marcas[id].idev = id;
    google.maps.event.addListener(marcas[id],"click",function(mouse) {
        marcas[id].info.setContent(this.conteudo);
        marcas[id].info.open(map,this);
    });
}
```

Figura 36 Configuração final das marcas, eventos e janela de informação.

## Página de Pesquisa (AJAX)

Grande parte das aplicações Web disponibilizam mecanismos de pesquisa evoluídos, que vão apresentando resultados à medida que os utilizadores vão inserindo novos critérios de pesquisa. Exemplos do referido são a caixa de pesquisa do Google Search ou a do facebook.

Para implementar este tipo de pesquisa é necessário captar alguns eventos na página, como tecla premida ou mudança de seleção numa lista e realizar uma chamada assíncrona ao servidor, solicitando o resultado para os novos critérios. Quando o servidor responde apresentam-se os resultados.

```
echo CHtml::activeTextField($model,'texto',
                            array('id'=>'texto'))
                    );
echo CHtml::activeDropDownList(
    $model,'tipo',
    Tipo::getTiposPesquisa(),
    array('id' => 'tipos')
);
```

Figura 37 Definição dos componentes do formulário de pesquisa como active (*TextField* e *DropDownList*).

Como é possível a qualquer utilizador pesquisar eventos pela descrição, título ou tipo, foi necessário um campo de texto, para critérios de descrição ou título e uma Drop Down List para selecionar o tipo restrição, se tal for requerido, pois é possível que o utilizador pesquise em todos os tipos. Para tal estes elementos de interface gráfica devem estar ser active e ter um identificador único na página, para que se possa registar uma função que trate os eventos que interessam para proceder à atualização dos resultados da pesquisa.

```
$('#tipos').change(actualizaPesquisa);
$('#texto').keypress(actualizaPesquisa);

function actualizaPesquisa()
{
    var url = '<?php echo $this->createUrl('pesquisar');?>';
    url += '&texto='+$('#texto').val();
    url += '&tipo='+$('#tipos').val();
    $.ajax({
        url : url,
        type : 'GET',
        success : function(data){
            $('#<?php echo $identificador; ?>').html(data);
        }
    });
}
```

Figura 38 Código Javascript que realiza pesquisa AJAX.

Como se pode verificar pelas primeiras duas linhas do código mostrado na Figura 38 são registados eventos tecla premida para o campo de texto e mudança para a Drop Down List, sendo a função chamada quando ocorrem os eventos referidos a actualizaPesquisa, que constrói um URL e realiza um pedido AJAX para o controller PesquisaController. Como se pode verificar quando o servidor retorna o html de retorno é colocado “div” de resultados, tal não é evidente no código porque o identificador da referida “div” pode ser personalizado.

The screenshot shows a web form for searching events. At the top, there is a text input field labeled "Texto \*" containing the value "Rio" and a dropdown menu labeled "Tipo" with the option "todos" selected. A tooltip message "Displaying 1-1 of 1 result(s)." is visible. Below the form, a table lists one result: "Rock in Rio-Lisboa 2012" at "Morada Caminho das feiras". To the right of the table is a dropdown menu with various event types: todos, regional, desportivo (which is highlighted in blue), religioso, festivo, astronómico, político, artístico, and feiras. At the bottom of the table, there are columns for "Inicio" (Lisbon, Portugal) and "Fim" (2012-06-03 00:00:00 - 2012-05-25 00:00:00).

Figura 39 Formulário de pesquisa AJAX.

## TimePicker - Documentação

O framework *Yii* não tem implementado um time picker para usar em formulários, pelo que foi necessário recorrer a uma extensão. Depois de algumas pesquisas descobriu-se um widget na página do framework muito interessante, que foi usada para facilitar aos utilizadores a tarefa de indicar a data-hora de início e fim do evento, na página de criação de um evento.

A extensão usada denomina-se timepicker e a versão usada foi a 3, tendo como único requisito para o seu uso a utilização do frameworks *Yii*. Para a instalação do widget bastou extraír a zip descarregado da página do *Yii* para a pasta “/protected/widgets”. O link para a página desta extensão pode ser encontrado na secção de referências Web.

```
<?php echo $form->labelEx($model,'dtinicio'); ?>
<?php $this->widget('application.widgets.timepicker.timepicker',
    array(
        'model'=>$model,
        'name'=>'dtinicio',
    )));
?>
<?php echo $form->error($model,'dtinicio'); ?>
```

Figura 40 Utilização do widget timepicker para suportar a inserção da data de início de um evento.

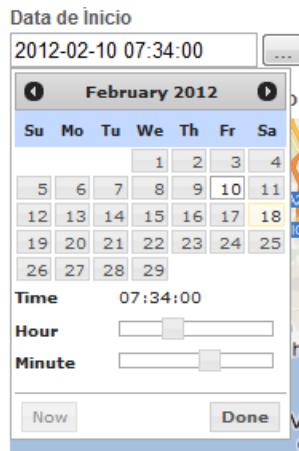


Figura 41 Resultado final da utilização do widget timepicker.

## *Introdução de Funcionalidades – Yii framework*

O framework Yii é muito bom para o desenvolvimento de aplicações Web orientadas aos dados, uma vez que, apresenta muitos auxiliares que permitem a geração de código a partir do schema da base de dados em uso, provavelmente fazendo engenharia reversa a partir dos metadados.

Como a geração de dados do Yii permite gerar uma página de gestão para as entidades centrais, como Eventos, Utilizadores e Comentários foi decidido inserir essas funcionalidades na aplicação final, pois as mesmas foram mais ou menos gratuitas no que diz respeito ao tempo de implementação necessário. Um exemplo da página de gestão gerada é mostrada na imagem abaixo inserida, como se pode verificar o gestor poderá pesquisar por qualquer atributo, apagar (desativar) qualquer instância e visualizar.

Displaying 1-2 of 2 result(s).					
Idcomentario	Texto	Dtcom	Utilizador Oid	Evento Idevento	
			1		
28	Primeiro comentário online!!!!	2012-02-18 05:31:09	1	1	
29	Tenciono ir!!!!!!	2012-02-18 05:42:52	1	3	

**Figura 42 Página de gestão de comentários.**

Estas funcionalidades não tinham sido inseridas na lista de requisitos iniciais por receio de falta de tempo para a implementação do projeto, no entanto são muito importantes, pois numa situação em que utilizadores escrevem comentários ofensivos, nesta situação o administrador da aplicação deve ter possibilidade de apagar esses comentários.

## Componente Não Funcional

O framework *Yii* induz a que se codifique a aplicação Web separando o código da camada de apresentação da camada de lógica, sendo a comunicação entre as referidas camadas mediadas por uma camada de controlo, tal é induzido pelo padrão de desenho arquitectural (MVC).

Depois de se compreender esta estrutura a alteração da camada de apresentação passa pela edição dos ficheiros relacionados com a camada de apresentação, que no framework utilizado estão na pasta *views*.

Mais especificamente, há duas coisas que se podem alterar, o próprio tema da aplicação, que está em *views/layouts*, e a apresentação de cada modelo/página, tal passa pela edição da view correspondente a cada modelo ou página.

### *Alterações transversais*

Todos os formulários relativos aos modelos das entidades principais foram alterados, pois os mesmos foram gerados fazendo engenharia reversa do schema da base de dados e tinham por isso alguns elementos totalmente desnecessários, por exemplo, eram pedidas/mostradas chaves primárias das entidades e outros elementos desnecessários.

Para tal tiveram que ser alterados alguns ficheiros relativos à visualização e edição das seguintes entidades:

- Comentário;
- Utilizador;
- Evento;
- Favorito.

A edição das views relativas às entidades referidas passou pela edição dos ficheiros contidos nas pastas */views/nome\_da\_entidade/*, mais especificamente *view.php* e *\_form.php*.

## Comentários

Como é óbvio o gerador de código do framework *Yii* não tinha como saber que a fotografia do utilizador autor dos comentários seria introduzida ao lado do comentário, tal era no entanto uma especificação do projeto, pelo que teve de ser incluída.

Como foi explicado na subsecção anterior, foi necessário alterar o ficheiro `/views/comentário/view.php`, tendo sido definida a nova forma de apresentação de um comentário (layout).



```
<div class="view">
    <table>
        <tr>
            <td width="50">
                <?php
                    $user = Utilizador::getUserById($data->utilizador_oid);
                    echo CHtml::imageButton('.'. $user->fotografia, array(
                        'width'=>50,
                        'height'=>50
                    ));
                ?>
            </td>
            <td style="float:left;">
                <?php echo $user->username; ?> diz,
                <br/>
                <b><?php echo CHtml::encode($data->texto); ?></b>
            </td>
        </tr>
    </table>
</div>
```

Figura 43 Código que permite a apresentação de um comentário.

O código exibido na figura acima inserida, só é útil na listagem de comentários, ou seja, para ser útil deve ser chamada recursivamente para um conjunto de comentários. Tal foi realizado usando um widget disponibilizado pelo frameworks *Yii* denominado *CListView*, cuja utilização é demonstrada na imagem abaixo inserida.



```
<?php
$this->widget('zii.widgets.CListView',
    array(
        'dataProvider'=>$comentarios,
        'itemView'=>'/comentario/view'
    )
);
?>
```

Figura 44 Código que permite listar os comentários.

Com isto foi obtida uma apresentação dos comentários muito mais interessante do que a inicialmente gerada pelo *Yii*, não tendo sido necessárias grandes alterações, uma vez que o código foi desenvolvido usando a modularidade induzida pelo *Yii*.



Figura 45 Exemplo de um comentário.

## Publicação da Página Web

Depois de finalizada a aplicação Web foi selecionado um servidor gratuito para a alojar, obviamente que esse servidor tem suporte para aplicações que necessitem de usar PHP e mysql, para tal foi criada uma conta no Site11 que oferece armazenamento gratuito.

Depois de se criar uma conta foi criada uma base de dados mysql, para que fosse possível configurar a aplicação para a base de dados remota antes de fazer upload do código para o servidor.

```
'db'=>array(  
    'connectionString' => 'mysql:host=localhost;dbname=eventoss',  
    'emulatePrepare' => true,  
    'username' => 'root',  
    'password' => '',  
    'charset' => 'utf8',  
) ,
```

Figura 46 Configuração da base de dados remota.

Para configurar a aplicação para a base de dados remota com a qual teria que funcionar depois de ser realizado o upload, foi necessário editar um ficheiro de configuração do frameworks Yii em /protected/config/main.php e preencher a entrada 'db' no array de configuração (Figura 46) com as informações fornecidas no Site11.

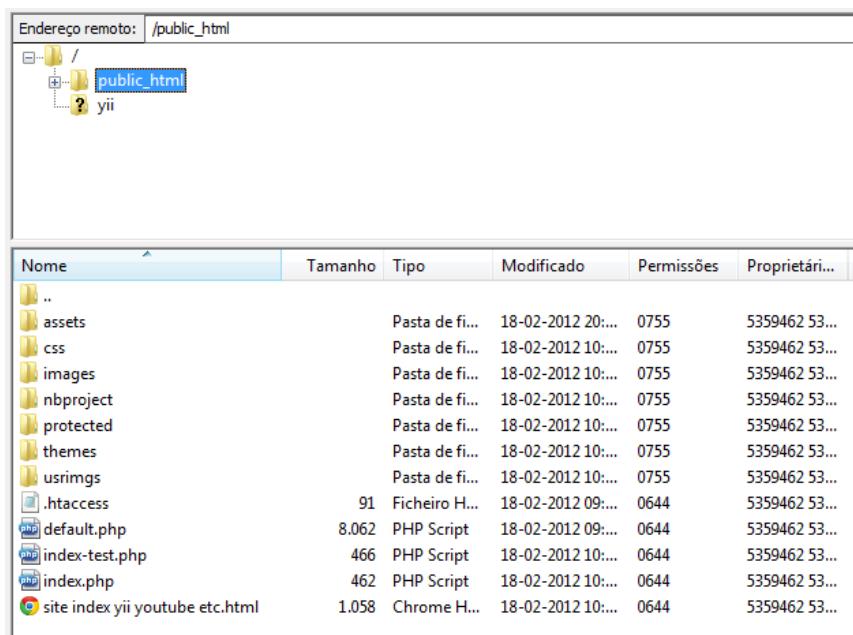


Figura 47 Estrutura remota após upload para o endereço remoto.

Posto isto, foi utilizado um programa denominado FileZilla (cliente FTP) para fazer upload do código desenvolvido para a pasta “public\_html” do web server e da biblioteca do framework *Yii* para a raiz, pois como é óbvio as bibliotecas não são para ser acedidas. No final deste processo a aplicação desenvolvida estava pela primeira vez disponível para o mundo em <http://eventhis.site11.com/>.

## Notas Finais – Implementação PHP

Desenvolver uma aplicação Web usando uma linguagem como PHP e um framework como o *Yii* é algo muito interessante, pois o desenvolvimento é realizado com muita rapidez, fazendo com que a componente de implementação de um projeto deste tipo seja apenas mais uma fase do projeto e não a totalidade do mesmo.

O suporte para o desenvolvimento de aplicações usando a linguagem PHP é enorme, pois existe uma vasta comunidade ligada a esta linguagem, o que não é de estranhar, uma vez que o PHP é uma linguagem de script open source muito usada no domínio das aplicações Web, tendo sido usada para a construção de excelentes aplicações Web mundialmente utilizadas, como é o caso do facebook.

Os frameworks são fundamentais no desenvolvimento de aplicações Web. Não utilizar um frameworks seria como construir um automóvel e reinventar as rodas, travões e todos os outros componentes, tal é evidentemente algo estúpido, pois os referidos componentes já estão muito evoluídos e compreendidos, não existindo valor acrescentado nenhum para o projeto reinventa-los.

Ficou claro durante o desenvolvimento desta aplicação Web que é necessário uma elevada compreensão dos conceitos que estão por de trás de um framework para que seja possível uma utilização efetiva e total das suas capacidades. Por exemplo, o *Yii* usando uma mini-arquitetura denominada MVC para organização e estruturação do código permite uma enorme modularidade e independência do código, sendo possível a reutilização de variadíssimos componentes em outros projetos, no entanto, se tal não fosse convenientemente entendido, essa capacidade não seria totalmente espremida.

Existiu a preocupação de comentar o código que pudesse suscitar dúvidas, não tendo sido por vezes apresentado no código exibido no presente relatório, porque o mesmo acaba por ser completamente explicado no corpo do mesmo.

Foi desenvolvido para a aplicação final um manual de utilizador, como não poderia deixar de ser num produto de software, tal é apresentado no anexo 10.

## **Conclusão e Trabalho Futuro**

Desenvolver um projeto desta natureza é algo muito enriquecedor e interessante, pois foram desenvolvidas capacidades de análise, descrição, abordagens metodológicas, documentação, construção e desenho de um produto de software, que no caso é uma aplicação Web.

Existe muito a melhorar no produto criado, até no processo/abordagem metodológica de criação do mesmo, tal é natural, uma vez que a aplicação Web Eventhis foi desenvolvida ao longo de um semestre.

Algumas evoluções óbvias para a aplicação desenvolvida seriam implementação total do modelo de negócio, que incluiria formas de pagamento para que os eventos fossem enviados nas newsletters do sistema, melhorar o serviço Web que disponibiliza os eventos, pois poderia ser mais eficiente, por exemplo, seria interessante enviar para o servidor os identificadores de todos os eventos que já estão representados no mapa, evitando um reenvio que não é usado.

Para além do referido, poderiam ser incluídas mais funcionalidades, uma vez que a abordagem seguida teve como output uma série de maquetas e documentos de suporte ao produto de software desenvolvido tornando muito mais fácil evoluir o produto. Acresce à documentação o facto de o framework usado ter induzido uma modularidade do código muito importante, sendo fácil detectar as repercussões das alterações e identificar as partes de código que é necessário alterar para a introdução ou modificação de funcionalidades.

Um exemplo de funcionalidade á a possibilidade de partilha via redes sociais dos eventos, pois as redes sociais são cada vez mais importantes e determinantes no sucesso de negócios/ideias, tal seria muito simples, se o objetivo fosse apenas a partilha, caso fosse requerido um registo da partilha, tal já implicaria modificações mais profundas, notadamente alteração do modelo de dados com vista ao registo das partilhas, o que já seria uma nova iteração da abordagem mais completa.

Muitos problemas tiveram que ser ultrapassados ao longo do desenvolvimento do produto, pois foi a primeira vez que foi usado o framework *Yii* e tal acarretou estudo ao mesmo tempo que se desenvolvia o produto, apesar de *Yii* apresentar uma curva de aprendizagem rápida, a implementação teria, certamente, menos percalços numa segunda iteração.

## Bibliografia

Martin Gaedke Michael Grossniklaus and Oscar Díaz (Eds.). Web Engineering 9th International Conference, ICWE 2009 San Sebastián, Spain, June 24-26, 2009 Proceedings

## Referências Web

- [www.yiiframework.com:](http://www.yiiframework.com)

Endereço utilizado para obter todas as informações relativamente ao framework Yii, extensões e tutoriais.

Último acesso: 17-02-2012.

- [www.blueprintcss.org:](http://www.blueprintcss.org)

Endereço utilizado para obter informação relativamente ao framework de CSS (Cascade Style Sheets) utilizado pelo *Yii* para proceder a alterações da componente gráfica.

Último acesso: 16-02-2012.

- [www.predic8.com/rest-webservices.htm:](http://www.predic8.com/rest-webservices.htm)

Endereço usado para obter informações sobre os Web Services REST que foram usados para disponibilizar os eventos.

Último acesso: 16-02-2012.

- [www.site11.com/:](http://www.site11.com/)

Web host onde foi criada uma conta para se proceder ao upload do web site com vista a disponibiliza-lo online.

Último acesso: 22-02-2012.

## **Lista de Siglas e Acrónimos**

MVC	Model-View-Controller: padrão de desenho arquitetural que permite uma separação do código entre as camadas de lógica e visualização, sendo a comunicação entre as referidas camadas mediadas por uma camada de controlo.
PHP	Linguagem script usada especialmente para o desenvolvimento da componente servidor de aplicações Web.
AJAX	Asynchronous Javascript and XML (Javascript e XML Assíncronos), tecnologia que permite uma comunicação assíncrona entre servidores e aplicações cliente.
IDE	<i>Integrated Development Environment</i> (Ambiente Integrado de Desenvolvimento), aplicação de computador que reúne características e ferramentas de apoio ao desenvolvimento de software, com o objetivo de agilizar este processo.

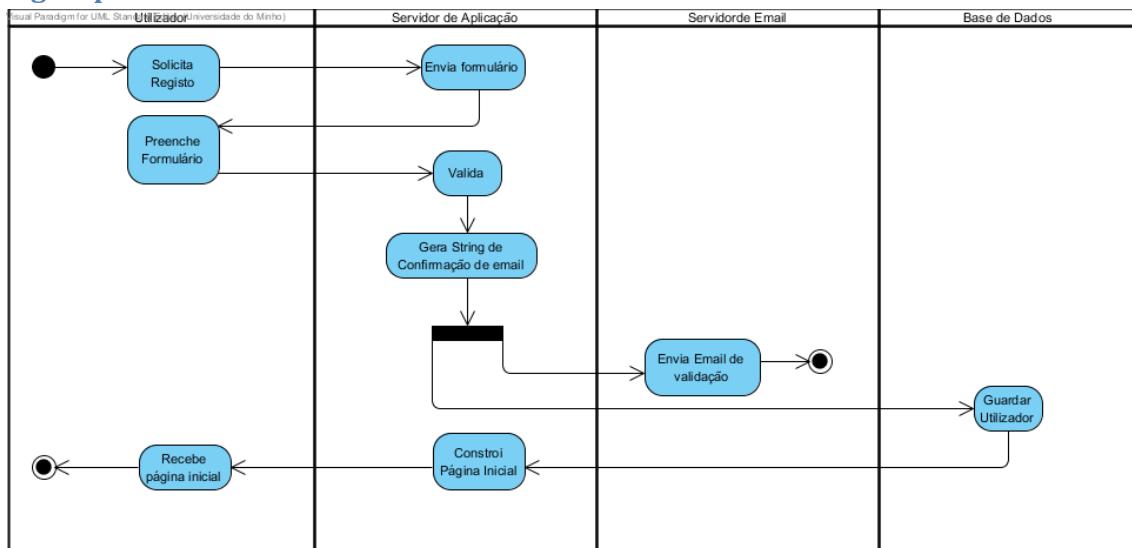
## **Anexos**

## 1. Relação entre os Grupos de Utilizadores

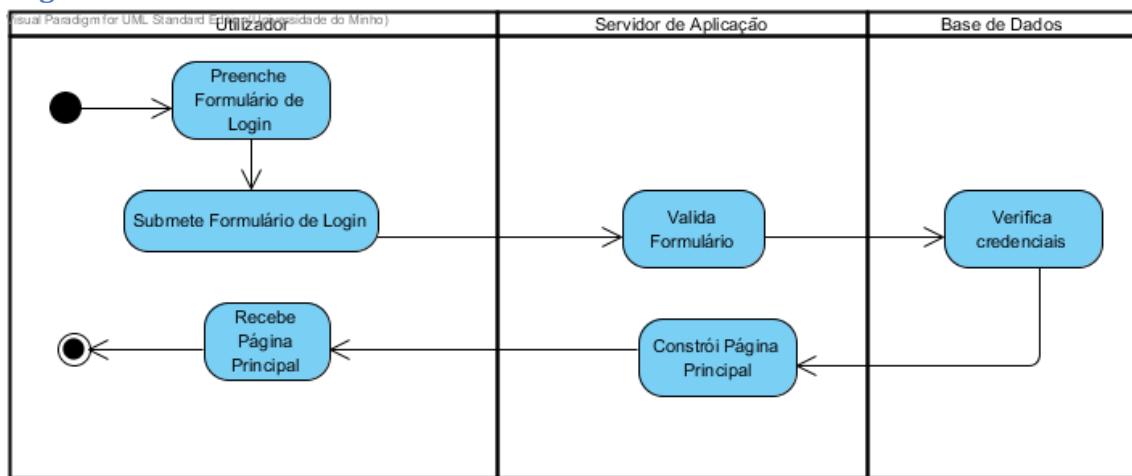


## 2. Diagramas de Atividade dos Casos de Uso

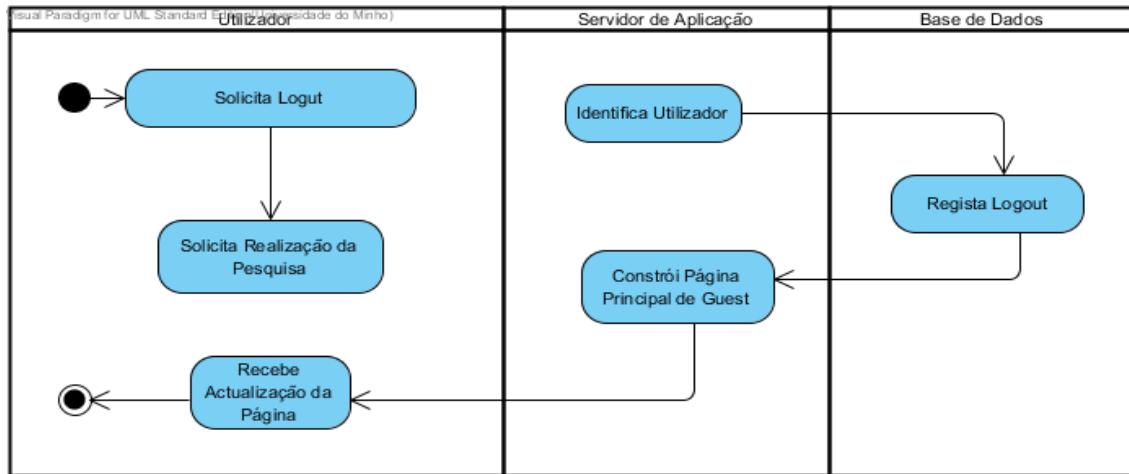
### Sign-Up



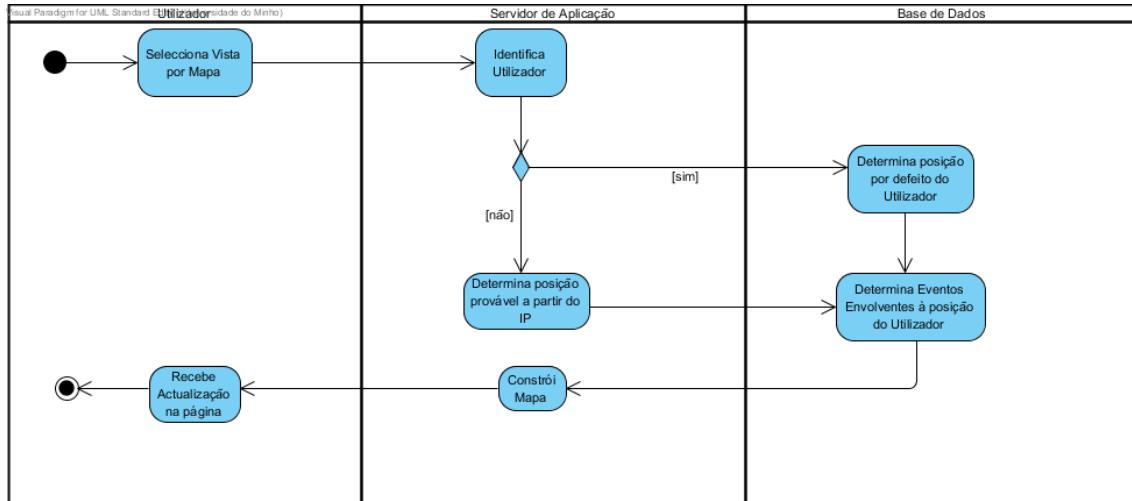
### Login



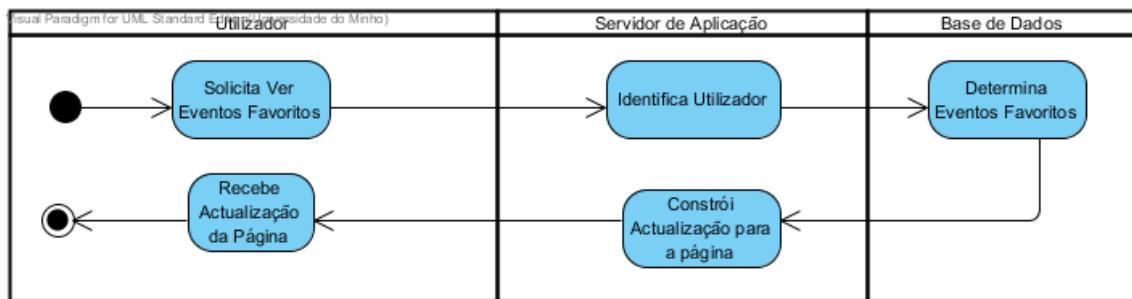
## Logout



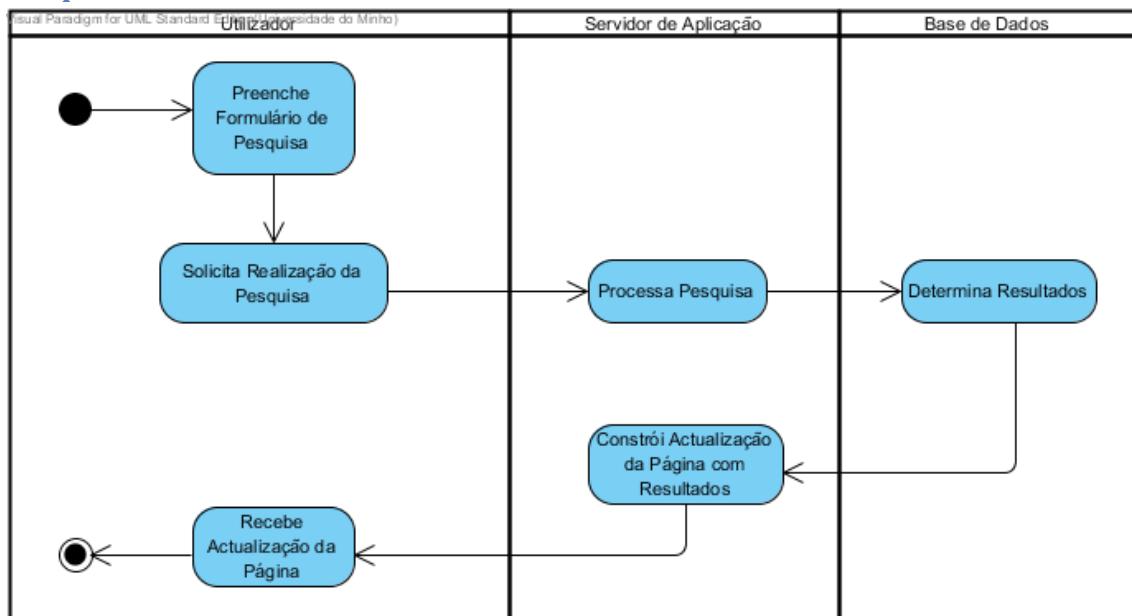
## Ver Eventos Por Mapa



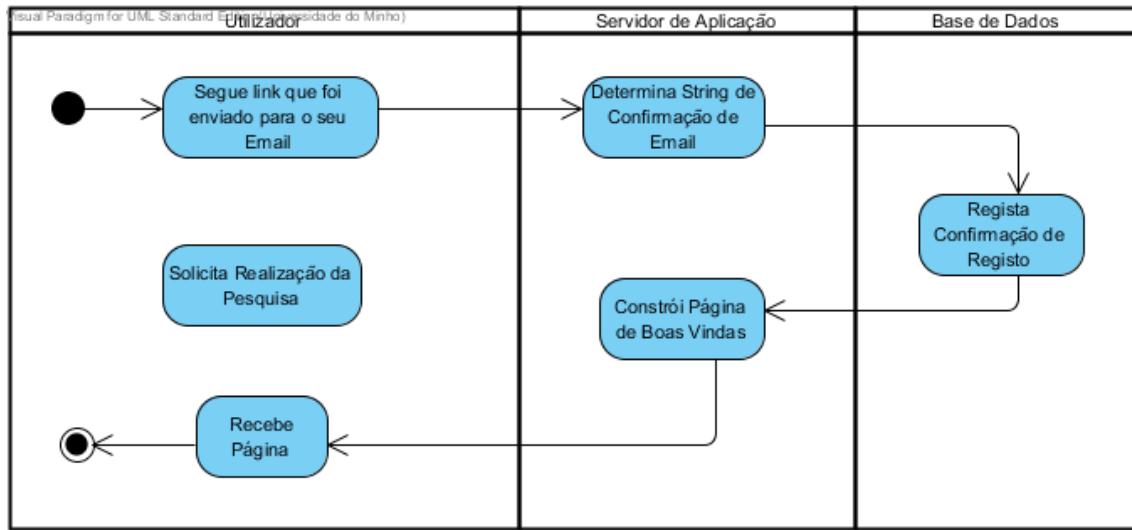
## Ver Eventos Favoritos



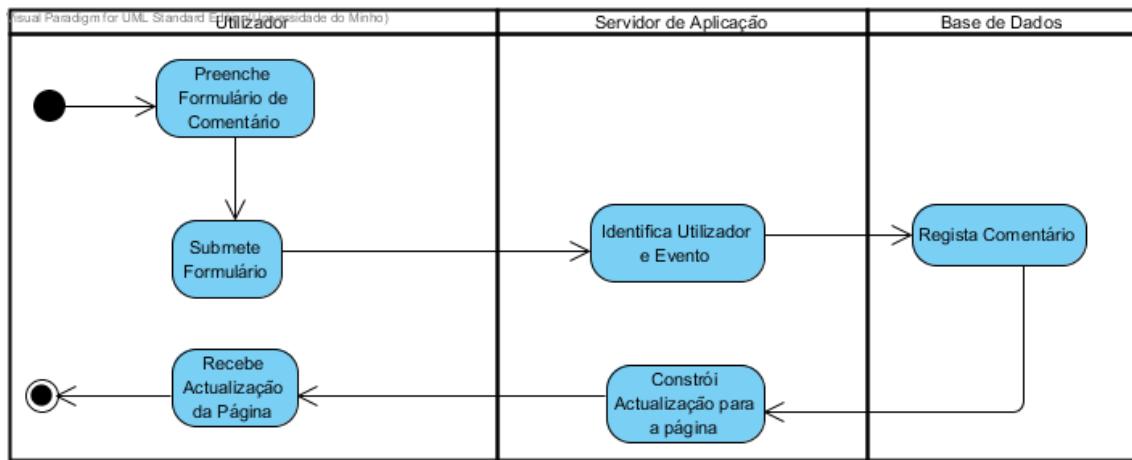
## Pesquisar Multicritérios



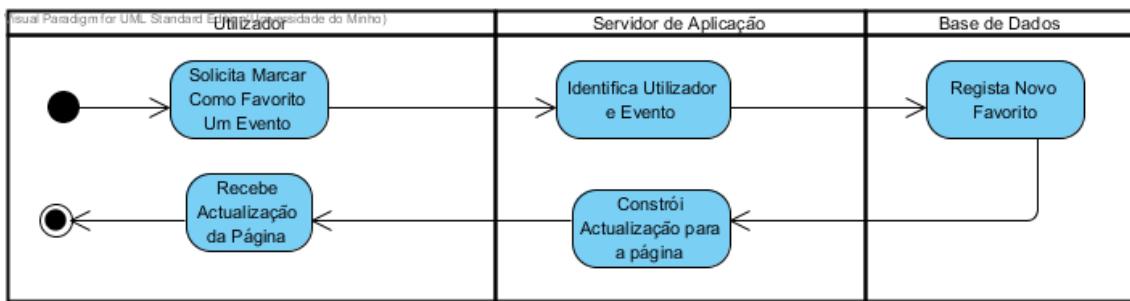
## Confirmar Endereço de Email



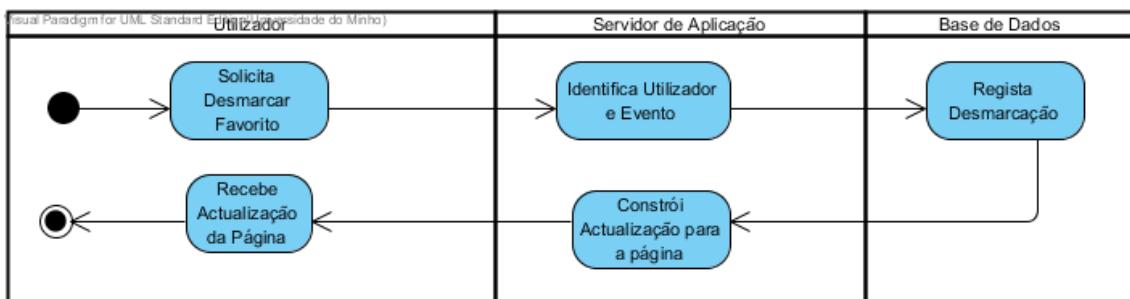
## Comentar Evento



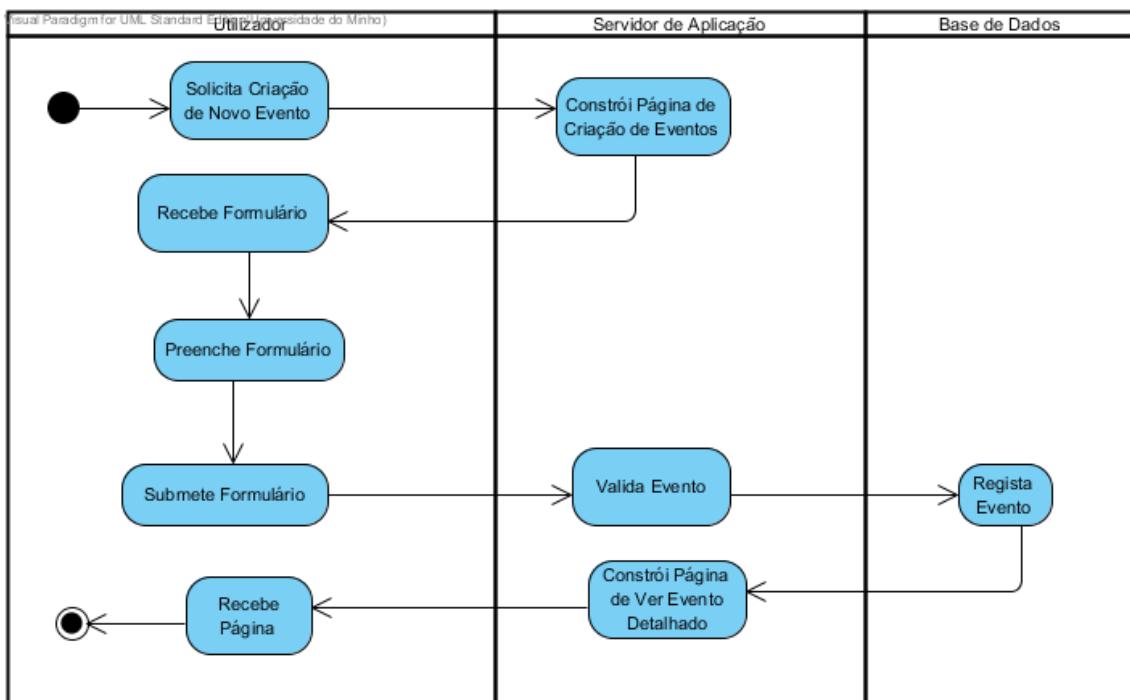
## Marcar Evento Como Favorito



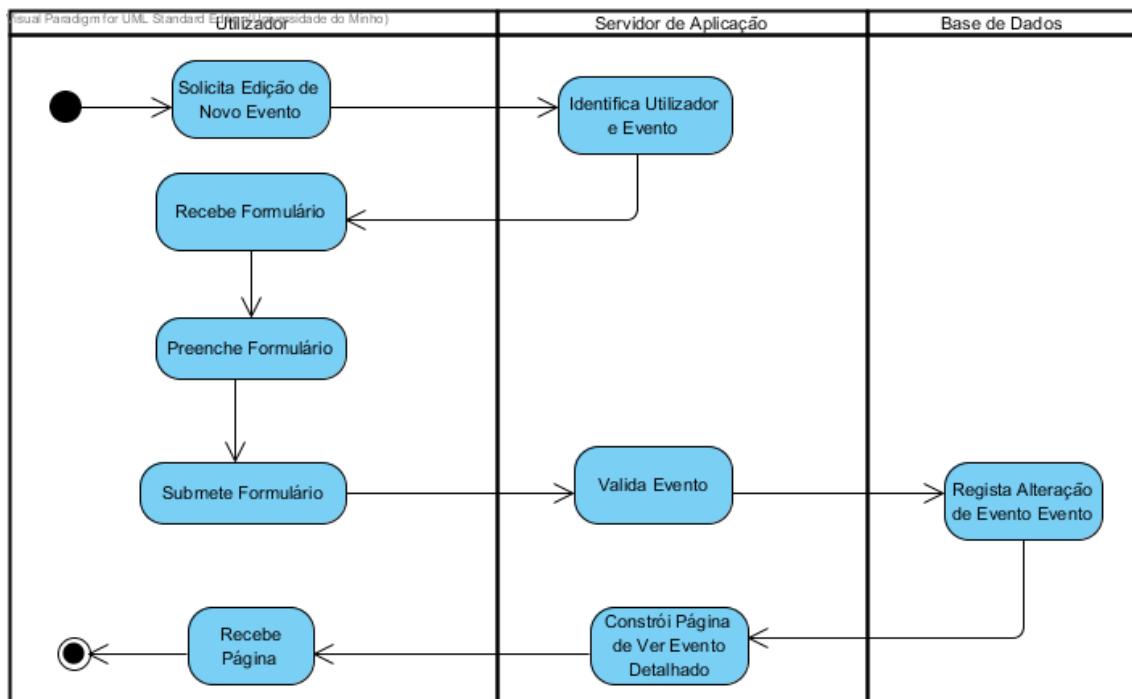
## Desmarcar Favorito



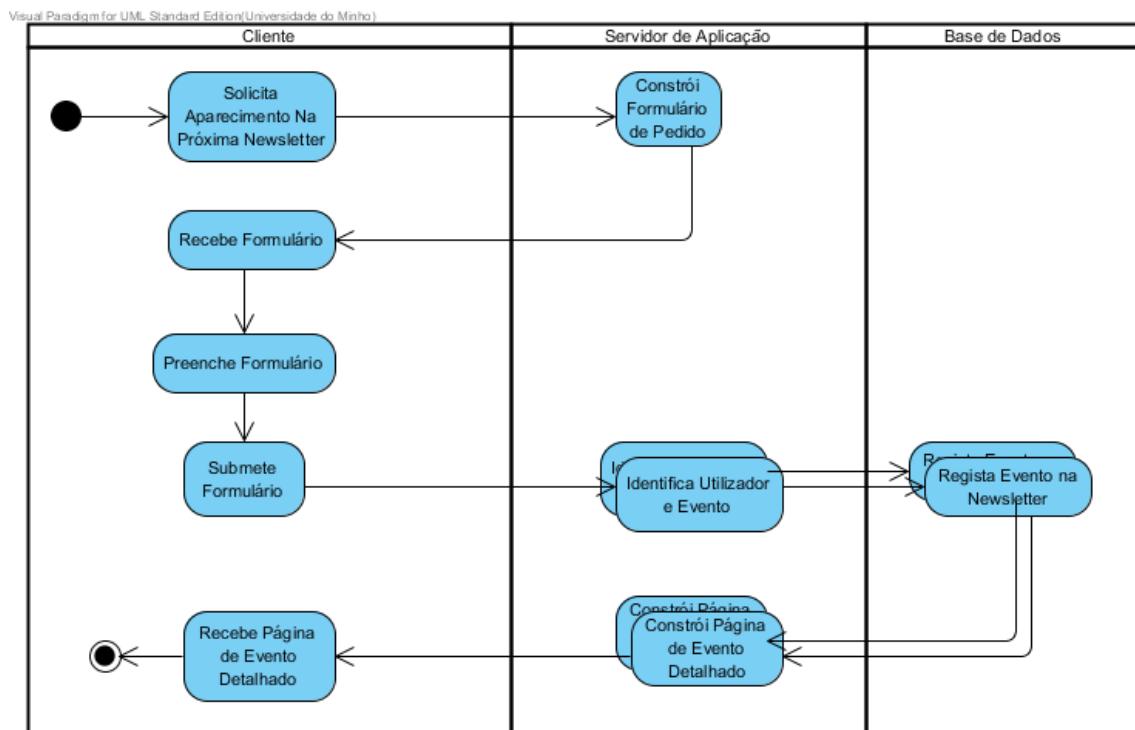
## Criar Evento



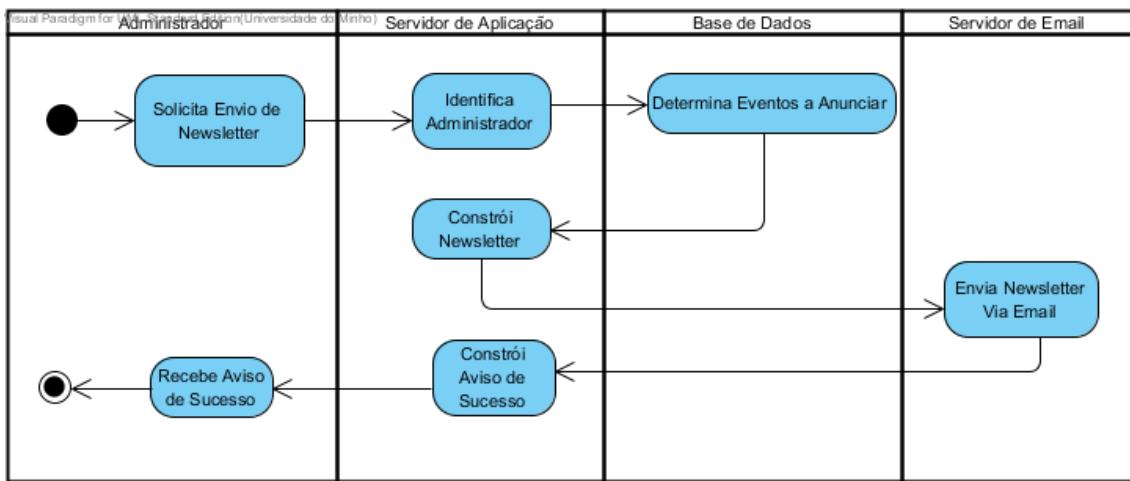
## Editar Evento



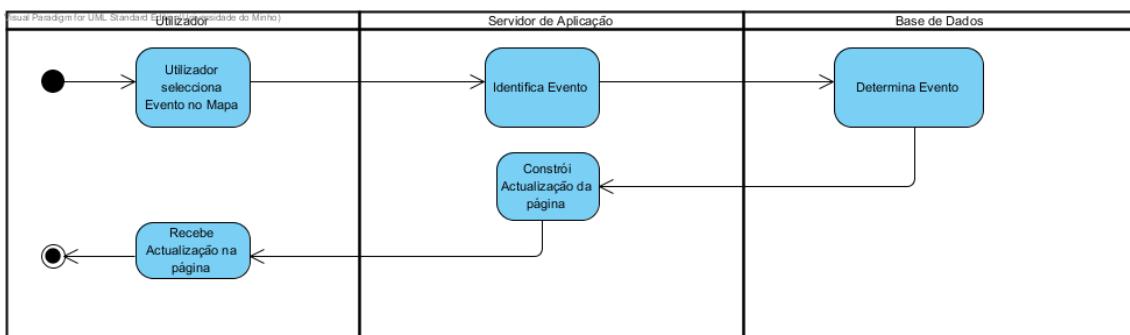
## Solicitar Aparecimento Na Próxima Newsletter



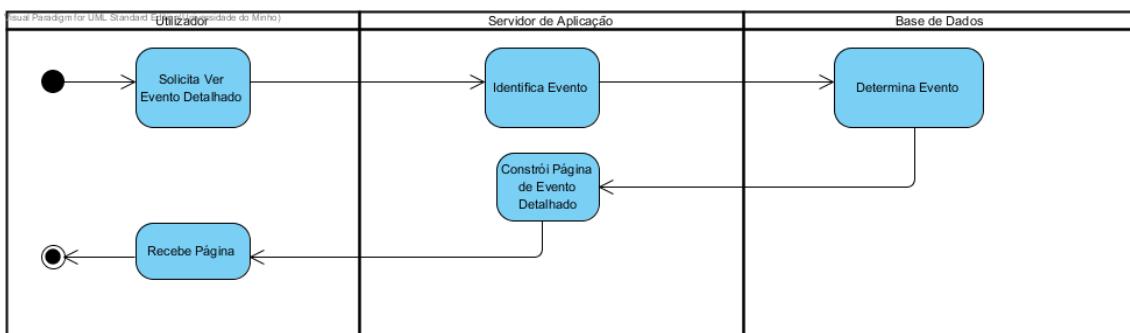
## Enviar Newsletter



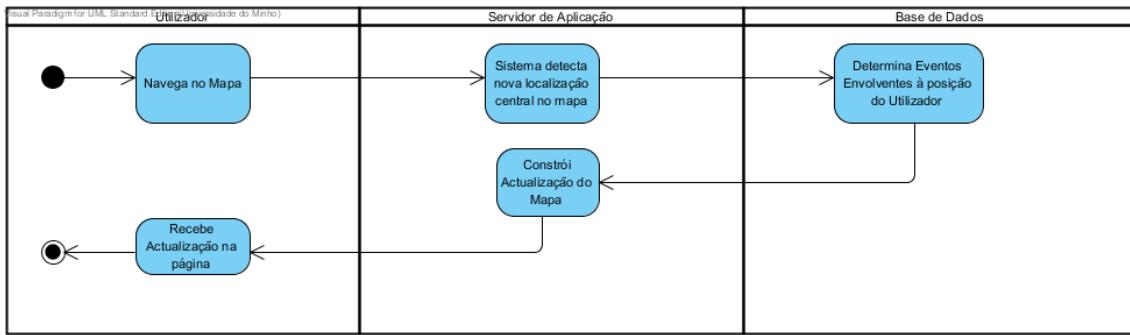
## Ver Evento Rápido



## Ver Evento Detalhado

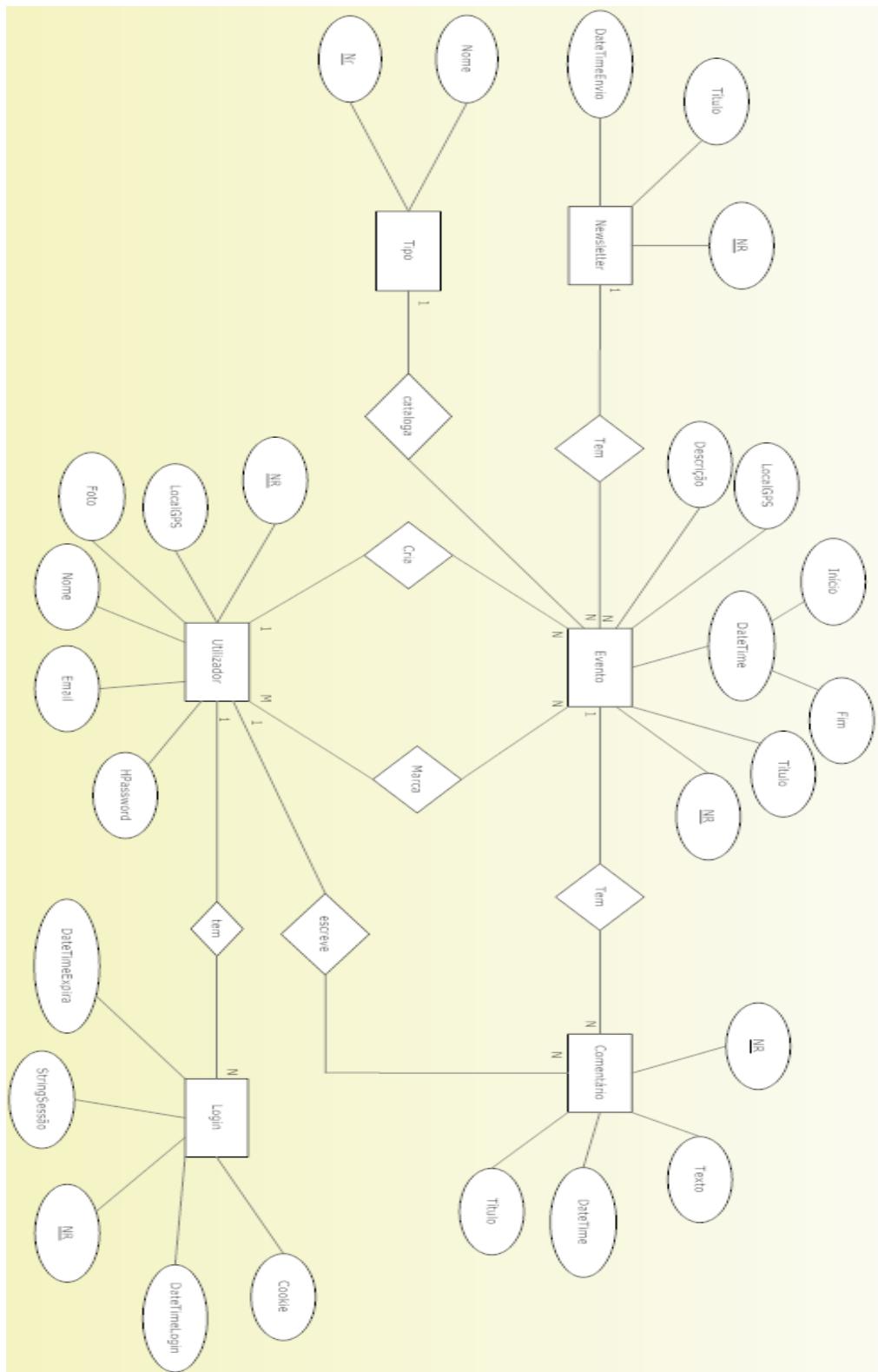


## Navegar No Mapa

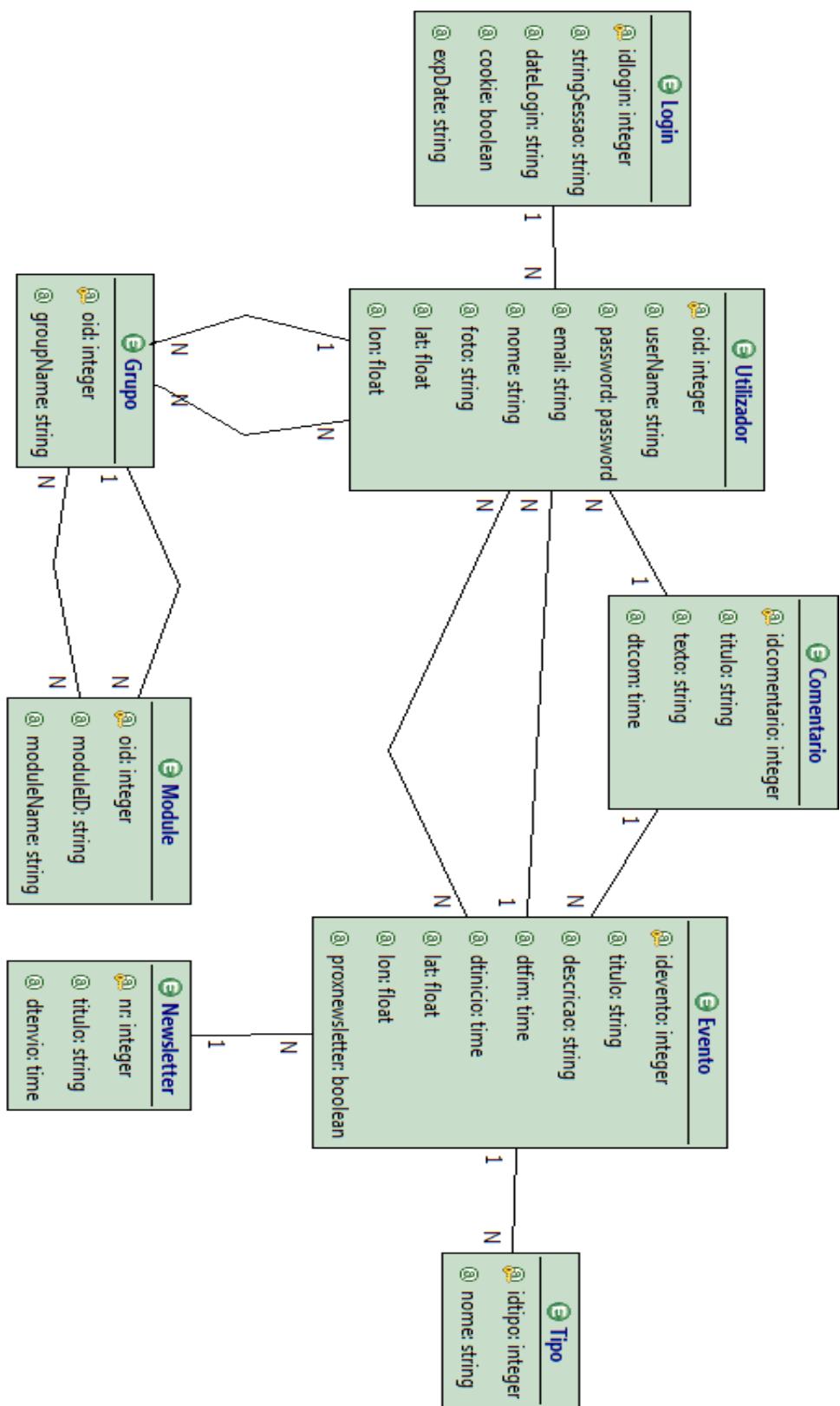




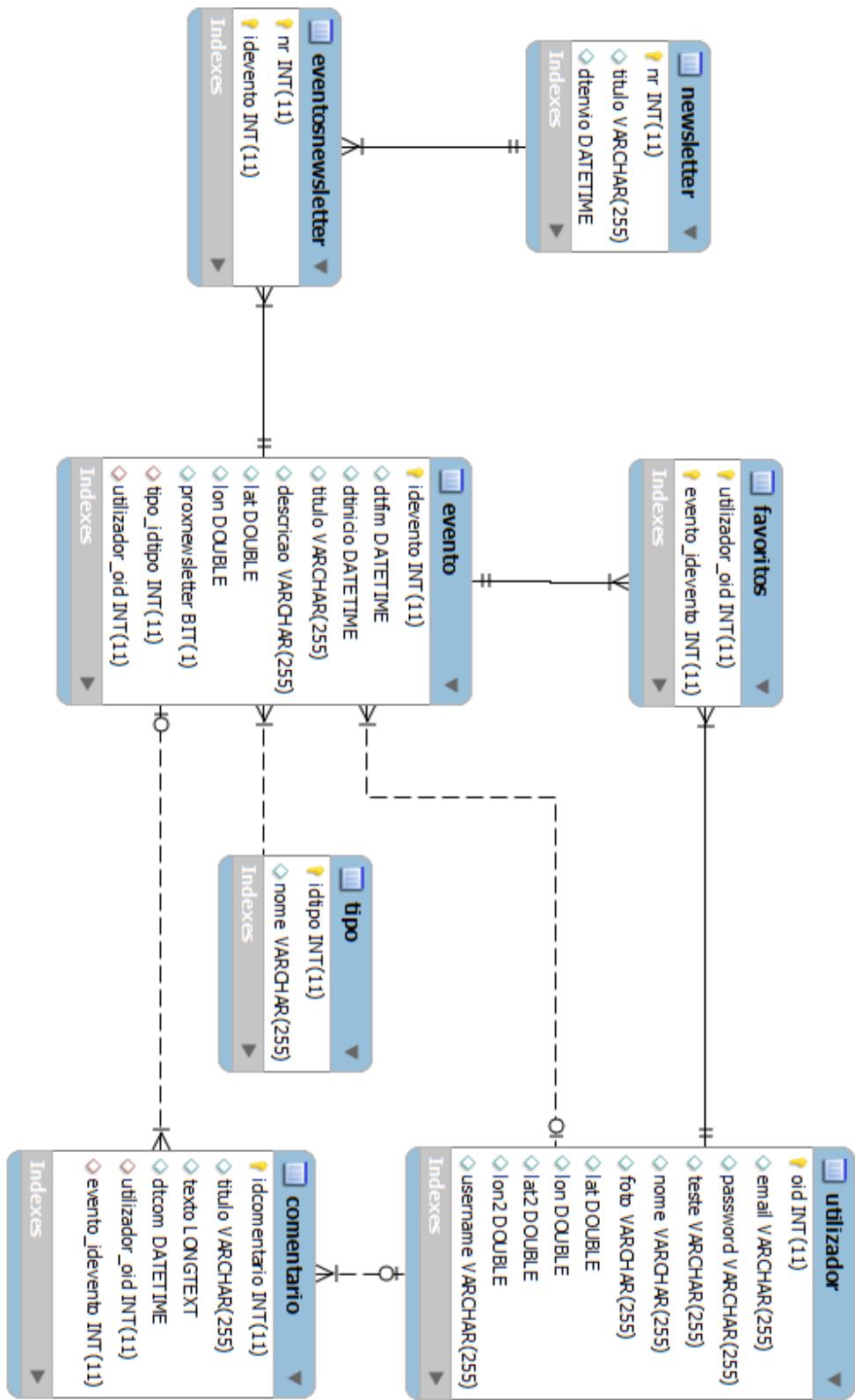
### 3. Modelo Entidades e Relacionamentos - CHEN



#### 4. Modelo de Dados - Webratio



## 5. Diagrama de Entidades Relacionamentos - Mysql



## 6. Código SQL – Base de Dados Eventhis

```
create database eventos;
use olaola;
-- Grupo [Group]
create table `grupo` (
  `oid` integer not null,
  `groupname` varchar(255),
  primary key (`oid`)
);

-- Module [Module]
create table `module` (
  `oid` integer not null,
  `moduleid` varchar(255),
  `modulename` varchar(255),
  primary key (`oid`)
);

-- Utilizador [User]
create table `utilizador` (
  `oid` integer not null,
  `email` varchar(255),
  `password` varchar(255),
  `teste` varchar(255),
  `nome` varchar(255),
  `foto` varchar(255),
  `lat` double precision,
  `lon` double precision,
  `lat2` double precision,
  `lon2` double precision,
  `username` varchar(255),
  primary key (`oid`)
);

-- Newsletter [ent1]
create table `newsletter` (
  `nr` integer not null,
  `titulo` varchar(255),
  `dtenvio` datetime,
  primary key (`nr`)
);

-- Tipo [ent2]
create table `tipo` (
  `idtipo` integer not null,
```

```

`nome` varchar(255),
primary key (`idtipo`)
);

-- Comentario [ent4]
create table `comentario` (
`idcomentario` integer not null,
`titulo` varchar(255),
`texto` longtext,
`dtcom` datetime,
primary key (`idcomentario`)
);

-- Evento [ent5]
create table `evento` (
`idevento` integer not null,
`dtfim` datetime,
`dtinicio` datetime,
`titulo` varchar(255),
`descricao` varchar(255),
`lat` double precision,
`lon` double precision,
`proxnewsletter` bit,
primary key (`idevento`)
);

-- Group_DefaultModule [Group2DefaultModule_DefaultModule2Group]
alter table `grupo` add column `module_oid` integer;
alter table `grupo` add index fk_grupo_module (`module_oid`), add constraint
fk_grupo_module foreign key (`module_oid`) references `module` (`oid`);
create index `idx_grupo_module` on `grupo`(`module_oid`);

-- Group_Module [Group2Module_Module2Group]
create table `group_module` (
`grupo_oid` integer not null,
`module_oid` integer not null,
primary key (`grupo_oid`, `module_oid`)
);
alter table `group_module` add index fk_group_module_grupo (`grupo_oid`), add constraint
fk_group_module_grupo foreign key (`grupo_oid`) references `grupo` (`oid`);
alter table `group_module` add index fk_group_module_module (`module_oid`), add
constraint fk_group_module_module foreign key (`module_oid`) references `module` (`oid`);
create index `idx_group_module_grupo` on `group_module`(`grupo_oid`);
create index `idx_group_module_module` on `group_module`(`module_oid`);

-- User_DefaultGroup [User2DefaultGroup_DefaultGroup2User]
alter table `utilizador` add column `grupo_oid` integer;

```

```

alter table `utilizador` add index fk_utilizador_grupo (`grupo_oid`), add constraint
fk_utilizador_grupo foreign key (`grupo_oid`) references `grupo` (`oid`);
create index `idx_utilizador_grupo` on `utilizador`(`grupo_oid`);

-- User_Group [User2Group_Group2User]
create table `user_group` (
  `utilizador_oid` integer not null,
  `grupo_oid` integer not null,
  primary key (`utilizador_oid`, `grupo_oid`)
);
alter table `user_group` add index fk_user_group_utilizador (`utilizador_oid`), add constraint
fk_user_group_utilizador foreign key (`utilizador_oid`) references `utilizador` (`oid`);
alter table `user_group` add index fk_user_group_grupo (`grupo_oid`), add constraint
fk_user_group_grupo foreign key (`grupo_oid`) references `grupo` (`oid`);
create index `idx_user_group_utilizador` on `user_group`(`utilizador_oid`);
create index `idx_user_group_grupo` on `user_group`(`grupo_oid`);

-- Tipo_Evento [rel1]
alter table `evento` add column `tipo_idtipo` integer;
alter table `evento` add index fk_evento_tipo (`tipo_idtipo`), add constraint fk_evento_tipo
foreign key (`tipo_idtipo`) references `tipo` (`idtipo`);
create index `idx_evento_tipo` on `evento`(`tipo_idtipo`);

-- Utilizador_Comentario [rel2]
alter table `comentario` add column `utilizador_oid` integer;
alter table `comentario` add index fk_comentario_utilizador (`utilizador_oid`), add constraint
fk_comentario_utilizador foreign key (`utilizador_oid`) references `utilizador` (`oid`);
create index `idx_comentario_utilizador` on `comentario`(`utilizador_oid`);

-- Evento_Comentario [rel3]
alter table `comentario` add column `evento_idevento` integer;
alter table `comentario` add index fk_comentario_evento (`evento_idevento`), add constraint
fk_comentario_evento foreign key (`evento_idevento`) references `evento` (`idevento`);
create index `idx_comentario_evento` on `comentario`(`evento_idevento`);

-- Utilizador_Evento [rel4]
alter table `evento` add column `utilizador_oid` integer;
alter table `evento` add index fk_evento_utilizador (`utilizador_oid`), add constraint
fk_evento_utilizador foreign key (`utilizador_oid`) references `utilizador` (`oid`);
create index `idx_evento_utilizador` on `evento`(`utilizador_oid`);

-- Favoritos [rel6]
create table `favoritos` (
  `utilizador_oid` integer not null,
  `evento_idevento` integer not null,
  primary key (`utilizador_oid`, `evento_idevento`)
)

```

```

);

alter table `favoritos` add index fk_favoritos_utilizador (`utilizador_oid`), add constraint
fk_favoritos_utilizador foreign key (`utilizador_oid`) references `utilizador` (`oid`);
alter table `favoritos` add index fk_favoritos_evento (`evento_idevento`), add constraint
fk_favoritos_evento foreign key (`evento_idevento`) references `evento` (`idevento`);
create index `idx_favoritos_utilizador` on `favoritos`(`utilizador_oid`);
create index `idx_favoritos_evento` on `favoritos`(`evento_idevento`);

-- Newsletter_Evento [rel7]
alter table `newsletter` add column `evento_idevento` integer;
alter table `newsletter` add index fk_newsletter_evento (`evento_idevento`), add constraint
fk_newsletter_evento foreign key (`evento_idevento`) references `evento` (`idevento`);
create index `idx_newsletter_evento` on `newsletter`(`evento_idevento`);

-- Script de configuração dos módulos para autenticação

INSERT INTO `module` (`oid`,`moduleid`,`modulename`) VALUES (1,'sv4','registados');
INSERT INTO `module` (`oid`,`moduleid`,`modulename`) VALUES (2,'sv1','administradores');

INSERT INTO `grupo` (`oid`,`groupname`,`module_oid`) VALUES (1,'registados',1);
INSERT INTO `grupo` (`oid`,`groupname`,`module_oid`) VALUES (2,'administradores',2);

-- Utilizadores e eventos de teste

INSERT INTO `utilizador`
(`oid`,`email`,`password`,`nome`,`foto`,`lat`,`lon`,`lat2`,`lon2`,`username`,`grupo_oid`,`teste`)
VALUES (1,'rafael@gmail.com','nova','Rafael Eduardo de Carvalho
Pereira','jm.jpg?field=page10.fld38',42.71384201504933,-
5.354429687499987,54.29017450293639,10.026429687500013,'rafael',1,'upload/images/1/j
m.jpg');

INSERT INTO `utilizador`
(`oid`,`email`,`password`,`nome`,`foto`,`lat`,`lon`,`lat2`,`lon2`,`username`,`grupo_oid`,`teste`)
VALUES (2,'admin@eventhis.com','1234567','Natal
Oliveira','http://www.google.pt/ola.jpg',0,0,100,100,'admin',2,'upload/images/1/IMG2011090
5_001.jpg');

INSERT INTO `utilizador`
(`oid`,`email`,`password`,`nome`,`foto`,`lat`,`lon`,`lat2`,`lon2`,`username`,`grupo_oid`,`teste`)
VALUES (3,'la@gmail.com','12345','Luis
Araujo','http://multimedia.fnac.pt/multimedia/PT/images_produits/PT/grandes150/4/5/8/404
4156023854.gif',0,0,100,100,'lala',1,'upload/images/1/IMG20110905_001.jpg');

INSERT INTO `utilizador`
(`oid`,`email`,`password`,`nome`,`foto`,`lat`,`lon`,`lat2`,`lon2`,`username`,`grupo_oid`,`teste`)
VALUES (4,'antonio@gmail.com','nova','António
Carvalho','http://cdn1.iconfinder.com/data/icons/softwaredemo/PNG/256x256/DrawingPin1_
Blue.png',40.824948183814335,-8.73284912109375,41.65107744417234,-
7.77154541015625,'antonio',1,'upload/images/1/IMG20110905_001.jpg');

INSERT INTO `utilizador`

```

```

(`oid`,`email`,`password`,`nome`,`foto`,`lat`,`lon`,`lat2`,`lon2`,`username`,`grupo_oid`,`teste`)
VALUES (6,'omnimonmaster@hotmail.com','ruilei','rui pereira','http://www.g-
volutiondance.com/Images/GV08home.jpg',34.044373582597856,-
15.67071533203125,47.30225321039508,-
0.28985595703125,'states',1,'upload/images/1/IMG20110905_001.jpg');

INSERT INTO `utilizador`
(`oid`,`email`,`password`,`nome`,`foto`,`lat`,`lon`,`lat2`,`lon2`,`username`,`grupo_oid`,`teste`)
VALUES (7,'rui@gmail.com','ruirui','Rui
Pereira','http://multimedia.fnac.pt/multimedia/PT/images_produits/PT/grandes150/4/5/8/40
44156023854.gif',40.21319374398207,-8.67242431640625,41.87029512726479,-
6.74981689453125,'ruipapa',1,'upload/images/1/IMG20110905_001.jpg');

INSERT INTO `utilizador`
(`oid`,`email`,`password`,`nome`,`foto`,`lat`,`lon`,`lat2`,`lon2`,`username`,`grupo_oid`,`teste`)
VALUES
(8,'luis.5.aa@hotmail.com','luis','Luis','http://data.whicdn.com/images/16658538/405194_460
s_v1_thumb.jpg',34.044373582597856,-15.67071533203125,47.30225321039508,-
0.28985595703125,'luis',1,'upload/images/1/IMG20110905_001.jpg');

-- tipos de eventos

INSERT INTO `tipo` (`idtipo`,`nome`) VALUES (1,'regional');
INSERT INTO `tipo` (`idtipo`,`nome`) VALUES (2,'desportivo');
INSERT INTO `tipo` (`idtipo`,`nome`) VALUES (3,'religioso');
INSERT INTO `tipo` (`idtipo`,`nome`) VALUES (4,'festivo');
INSERT INTO `tipo` (`idtipo`,`nome`) VALUES (5,'astronómico');
INSERT INTO `tipo` (`idtipo`,`nome`) VALUES (6,'político');
INSERT INTO `tipo` (`idtipo`,`nome`) VALUES (7,'artístico');
INSERT INTO `tipo` (`idtipo`,`nome`) VALUES (8,'feiras');

-- eventos

INSERT INTO `evento`
(`idevento`,`dtfim`,`dtinicio`,`titulo`,`descricao`,`lat`,`lon`,`proxnewsletter`,`tipo_idtipo`,`utiliza
dor_oid`) VALUES (1,'19:44:25','19:44:27','Festa da Nossa Senhora','A realizar em Salamonde
Vieira do Minho.',41.10385002812584,-8.3251953125,'0',4,1);

INSERT INTO `evento`
(`idevento`,`dtfim`,`dtinicio`,`titulo`,`descricao`,`lat`,`lon`,`proxnewsletter`,`tipo_idtipo`,`utiliza
dor_oid`) VALUES (9,'09:30:00','09:30:00','Festa do Além Mar','Festa com muita dança e
companheirismo a realizar na Georgia.',42.3868351423178,41.61202636718747,'0',4,1);

INSERT INTO `evento`
(`idevento`,`dtfim`,`dtinicio`,`titulo`,`descricao`,`lat`,`lon`,`proxnewsletter`,`tipo_idtipo`,`utiliza
dor_oid`) VALUES (5,'23:27:55','23:27:51','Feira em Honra de Nossa Senhora','Nesta feira serrá
possível comprar de tudo inclusive computadores e latas de conserva.',41.54439989652676,-
8.433385848999023,'0',3,1);

```





## 7. Código Javascript – Google Maps – Webratio

```
<!--API da google que dá suporte para a utilização de mapas -->
<script type="text/javascript" src="http://maps.googleapis.com/maps/api/js?sensor=true">
</script>
<!--API usada para converter o XML em JSON, facilitando
    O tratamento da resposta do servidor -->
<script type="text/javascript"
src="http://www.terracoder.com/scripts/saXMLUtils.js"></script>
<!--Script construído realiza os pedidos de eventos ao servidor e os insere no mapa -->
<script type="text/javascript">
function inserirNoMapa(json)
{
    for(var i = 0 ; i < json.Entity[0].Instance.length ; i++ ){
        var idEvento = json.Entity[0].Instance[i].Attribute[0].Text;
        // só insere o evento no mapa se o mesmo não estiver lá
        if(!marcas[idEvento]){
            var id = "i"+idEvento;
            var pos = new google.maps.LatLng(
                json.Entity[0].Instance[i].Attribute[2].Text,
                json.Entity[0].Instance[i].Attribute[1].Text
            );
            var titulo = json.Entity[0].Instance[i].Attribute[3].Text;
            var tipo = json.Entity[0].Instance[i].Attribute[4].Text;
            // vai buscar imagem para o tipo do evento
            var foto = getImagenTipo(tipo);
            // criação de uma nova marca para inserção
            // no mapa
            marcas[id] = new google.maps.Marker({
                position: pos,
                map: map,
                title: titulo,
                icon: foto,
                conteudo:titulo+'<br /><a href="page15.do?kcond3.att19='
                        +idEvento+'">aqui</a>'
            });
            marcas[id].setDraggable(false);
            marcas[id].setClickable(true);
            marcas[id].info = new google.maps.InfoWindow();
            marcas[id].idev = id;
            // registo do comportamento para o click numa marca
            // do mapa
            google.maps.event.addListener(marcas[id],"click",
                function(mouse){
                    marcas[id].info.setContent(this.conteudo);
                    marcas[id].info.open(map,this);
                });
        }
    }
}
```

```
function novosEventos()
{
    var xmlhttp;
    // identifica fronteiras visiveis no mapa no momento
    var ne = map.getBounds().getNorthEast();
    var sw = map.getBounds().getSouthWest()
    // constrói pedido REST
    var pedido = "EventosService/mostraeventos/Area.do?LatMenor="
                + sw.lat() + "&LatMaior=" + ne.lat() + "&LonMenor="
                + sw.lng() + "&LonMaior=" + ne.lng();
    if (window.XMLHttpRequest)
        // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp=new XMLHttpRequest();
    }
    else
    {
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    // indica o que fazer quando chegar a resposta do
    // servidor com o XML
    xmlhttp.onreadystatechange=function()
    {
        if (xmlhttp.readyState==4 && xmlhttp.status==200)
        {
            // converte XML resposta do servidor
            // para JSON, pois é mais fácil de tratar
            // com javascript
            var resp = xmlhttp.responseText;
            var json =
XMLObjectifier.xmlToJSON(XMLObjectifier.textToXML(resp));
            // chama função que insere os eventos no mapa
            inserirNoMapa(json);
        }
    }
    // realiza pedido HTTP
    xmlhttp.open("GET",pedido,true);
    xmlhttp.send();
}
```

## 8. Código do Template Power Index de Comentários - Webratio

```
#?delimiters [%], [%=, %]
<wr:LayoutParameter label="Show Header" name="show-header"
type="boolean" default="true">
Defines the rendition of the header on the top.
Allowed values are: true (default) or false.
</wr:LayoutParameter>
<wr:LayoutParameter label="Show Attributes Name" name="show-
attribute-name" type="boolean" default="false">
Defines the rendition of the attributes names.
Allowed values are: true or false(default).
</wr:LayoutParameter>
<wr:LayoutParameter label="Show Bullet" name="show-bullet"
type="boolean" default="true">
Defines the rendition of a bullet next to each line.
Allowed values are: true or false(default).
</wr:LayoutParameter>
<wr:LayoutParameter label="Use Alternate Rows" name="use-alternate"
type="boolean" default="false">
Defines the usage of alternates colours for each line.
Allowed values are: true or false(default).
</wr:LayoutParameter>
<wr:LayoutParameter label="Links Position" name="link-position"
type="enum" values="left|right|on_bullet|on_row" default="right">
Defines the rendition of the sub levels links.
Allowed values are:
- right (default): places the links on the right
- left: places the links on the left
- on_bullet: places the first link on the bullet
- on_row: places the first link on the attributes
</wr:LayoutParameter>
<wr:LayoutParameter label="Scrolling Position" name="scrolling
position" type="enum" values="top|bottom|both" default="top">
Defines where to position the scroll links.
Allowed values are:
- top (default): places the links at the top
- bottom: places the links at the bottom
- both: places the links at the top and bottom
</wr:LayoutParameter>
<wr:LayoutParameter label="Buttons Position" name="button-position"
type="enum" values="top|bottom|both" default="bottom">
Defines where to position the buttons to render submit links.
Allowed values are:
- top: places the buttons at the top
- bottom (default): places the buttons at the bottom
- both: places the buttons at the top and bottom
</wr:LayoutParameter>
<wr:LayoutParameter label="Show Jump Links" name="show-jump-links"
type="boolean" default="true">
Defines whether to show the jumps links of the scroller or not.
Allowed values are: true(default) or false.
</wr:LayoutParameter>
<wr:LayoutParameter label="Select All" name="select-all"
type="boolean" values="true|false" default="false">
Defines the rendition of a link that allows
to select or deselect all the checkboxes of
```

```

the unit.
Allowed values are: true or false(default).
</wr:LayoutParameter>
<wr:LayoutParameter label="Use Empty Unit Message" name="use-empty-unit-message" type="boolean" default="false">
Defines the usage of a message for empty units.
Allowed values are: true(default) or false.
</wr:LayoutParameter>
<wr:LayoutParameter label="Empty Unit Message" name="empty-unit-message" type="string" default="emptyUnitMessage">
Defines the key of the message to use if the unit is empty.
Default value: emptyUnitMessage
</wr:LayoutParameter>

[%import org.w3c.dom.Attr

import org.apache.commons.lang.StringUtils
import org.apache.commons.lang.math.NumberUtils

setHTMLOutput()

def unitId = unit["id"]
def blockFactor = unit["blockFactor"]
def isSortable = unit["sortable"]
def isScrollable = blockFactor != "" ? "true": "false"

def links =
unit.selectNodes("layout:Link").findAll{it["_exp"]!='t'}
def unitLink = links.empty ? null : links[0]
def unitLinkIsAjax = unitLink != null ? (unitLink["ajaxEnabled"] == "true" && isAjaxPage(page)) : false
def link = unitLink?.valueOf("@link")
def myIndex = unit.hashCode()

def showHeader = params["show-header"]
def showAttributeName = params["show-attribute-name"]
def useAlternate = params["use-alternate"]
def showBullet = params["show-bullet"]
def linkPosition = params["link-position"]
def buttonPosition = params["button-position"]
def showJumpLinks = params["show-jump-links"]
def scrollingPosition = params["scrolling-position"]
def selectAll = params["select-all"]
def useEmptyUnitMessage = params["use-empty-unit-message"]
def emptyUnitMessage = params["empty-unit-message"]

def firstLink =
unit.selectSingleNode("layout:Link[contains(@link,'First')]")
def previousLink =
unit.selectSingleNode("layout:Link[contains(@link,'Previous')]")
def blockLink =
unit.selectSingleNode("layout:Link[contains(@link,'Block')]")
def nextLink =
unit.selectSingleNode("layout:Link[contains(@link,'Next')]")
def lastLink =
unit.selectSingleNode("layout:Link[contains(@link,'Last')]")
def isAutoExpandAjax = isAjaxPage(page)
def checkable = "true" == unit["checkable"] && "true" != unit["distinct"]

//returns true if the given link must be rendered as a form button

```

```

def isButton(link) {
    def navLink = getById(link["link"])
    if(navLink == null) {
        return false
    }
    def linkParams =
navLink.selectNodes("LinkParameter").collect{it["source"]}
    if(linkParams.isEmpty()) {
        linkParams =
getAutomaticLinkParameters(navLink).collect{it["source"]}
    }
    for(param in linkParams) {
        if(param?.endsWith("Checked")) {
            return true;
        }
    }
    return false
}

def buttonsCount =
unit.selectNodes("layout:Link").findAll{ isButton(it) }.size();
[%]

[%
def getIconPath(linkLayout) {
    folder = linkLayout.parameters["icon-folder"]
    name = StringUtils.defaultIfEmpty(linkLayout.parameters["icon-
name"], linkLayout.contextElement?.attributeValue("name"))
    extension = linkLayout.parameters["icon-extension"]
    return getFilePath(folder, name, extension)
}
%]

[% if (useEmptyUnitMessage != "true") { %>
<c:if test="${not(empty <wr:UnitId/>) and (<wr:UnitId/>.dataSize gt 0)}">
[% } else { %]
<c:choose>
<c:when test="${not(empty <wr:UnitId/>) and (<wr:UnitId/>.dataSize gt 0)}">
[% } %]
<wr:Frame>
    <div class="plain <wr:StyleClass/>">
        <div class="plain PowerIndexUnit">
            <table cellspacing="1" cellpadding="2">

                <!--Ínicio dos Comentários -->
                <c:forEach var="current" varStatus="status"
items="${<wr:UnitId/>.data}">
                    <c:set var="index"
value="${status.index}" />

                    <wr:Iterate var="attr"
context="unit" select="layout:Attribute">
                        <tr>
                            <wr:Visible
position="'index'">
                                [% if ((linkPosition ==
"on_row") && (!links.empty)) { %]
                                    <td
class="

```

```

test="${<wr:UnitId/>.currentIndex eq index}">Current</c:if>[% if
(useAlternate == "true") { %}<c:if test="${index mod 2 eq
0}">Alternate</c:if>[% } %] [= attr["type"]%]">
                                                 <wr:Visible
context="unitLink" position="'index'">
                                                 <a
href=<wr:URL context="unitLink"/> class=<wr:StyleClass
context="unitLink"/> link<c:if test="${<wr:UnitId/>.currentIndex eq
index}">Current</c:if>[% if (useAlternate == "true") { %}<c:if
test="${index mod 2 eq 0}">Alternate</c:if>[% } %]">
onclick=<wr:AjaxURL context="unitLink" />" [%if
(unitLink?.attributeValue("newWindow") == "true") { %
target="_blank" [% } %]>
                                                 </wr:Visible>
                                                 <wr:Value/>
                                                 <wr:Visible
context="unitLink" position="'index'">
                                                 </a>
                                                 </wr:Visible>
</td>
[% } else { %}

[%
if( attr.attributes() [index].value == "att17") { %
                                                 <td
style="background:#F1F1F1" class=<wr:StyleClass/> value<c:if
test="${<wr:UnitId/>.currentIndex eq index}">Current</c:if>[% if
(useAlternate == "true") { %}<c:if test="${index mod 2 eq
0}">Alternate</c:if>[% } %] [= attr["type"]%]">

<wr:Value />
</td>

[%
} else {
if( attr.attributes() [index].value == "t") { %
                                                 <td
style="background:#F1F1F1" class=<wr:StyleClass/> value<c:if
test="${<wr:UnitId/>.currentIndex eq index}">Current</c:if>[% if
(useAlternate == "true") { %}<c:if test="${index mod 2 eq
0}">Alternate</c:if>[% } %] [= attr["type"]%]">

<wr:Value />
</td>

[%
} else { %
                                                 <td
rowspan="3" class=<wr:StyleClass/> value<c:if
test="${<wr:UnitId/>.currentIndex eq index}">Current</c:if>[% if
(useAlternate == "true") { %}<c:if test="${index mod 2 eq
0}">Alternate</c:if>[% } %] [= attr["type"]%]">

<wr:Value />
</td>

[% } %]

```

```

[ % } %]
</wr:Visible>
</tr>
</wr:Iterate>

</c:forEach>

<!-- Fim dos comentários --->
</table>
</div>
</div>
</wr:Frame>
[% if (useEmptyUnitMessage != "true") { %]
</c:if>
[% } else { %]
</c:when>
<c:otherwise>
<wr:Frame>
<div class="plain <wr:StyleClass/>">
<div class="plain PowerIndexUnit">
<table>
<tr>
<td><bean:message
key="<%printJSPTagValue(emptyUnitMessage)%>" /></td>
</tr>
</table>
</div>
</div>
</wr:Frame>
</c:otherwise>
</c:choose>
[% } %]

```

## 9. Screenshots da Aplicação Desenvolvida com o Webratio

---

Pesquisar Eventos Autenticação

> Autenticação

Login

Username rafael

Password ....

Log In Sign Up

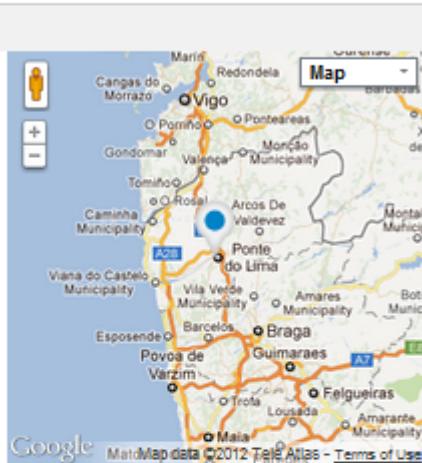
Figura 48 Página de Autenticação.

Pesquisar Eventos Favoritos  
> Visualização de Eventos > Página do Evento

#### Evento

Título	Festa do Além Mar
Descrição	Festa com muita dança e companheirismo a realizar na Georgia.
Tipo do Evento	festivo
Inicio	1/1/70 9:30:00 AM
Fim	1/1/70 9:30:00 AM
Latitude	41.763
Longitude	-8.59

[Marcar Como Favorito](#) [Editar ou Remover Evento](#)



#### Comentários



Muito Interessante.

Gostava imenso de passar por lá!!!

#### Novo Comentário

Título

Corpo

[Comentar](#)

Figura 49 Página que permite ver um evento detalhado para utilizadores autenticados.

Pesquisar Eventos Favoritos

> Gestão de Eventos > Criar Evento

**Novo Evento**

**Titulo**:

**Descrição**:

**Tipo do Evento**:

**Inicio**:

**Fim**:

**Latitude**:

**Longitude**:

**ProximaNewsletter**:  yes  no

**Criar**

Insira as coordenadas gps ou seleccione um ponto no mapa.

Map data ©2012 Tele Atlas - Terms of Use

Figura 50 Página de Criação de Evento.

Pesquisar Eventos Favoritos

> Gestão de Eventos > Página Edição de Evento

**EditarEvento**

**Titulo**:

**Descrição**:

**Tipo do Evento**:

**Inicio**:

**Fim**:

**Latitude**:

**Longitude**:

**Remover Evento** **Guardar** **Página do Evento**

Insira as coordenadas gps ou seleccione um ponto no mapa.

Map Data - Terms of Use

Figura 51 Página de Edição de Evento.

Pesquisar Eventos | Favoritos

> Favoritos

**Favoritos**

Titulo	Descrição	Inicio	Fim	Latitude	Longitude
<input checked="" type="checkbox"/> Festa do Além Mar	Festa com muita dança e companheirismo a realizar na Georgia.	1/1/70 9:30:00 AM	1/1/70 9:30:00 AM	42.387	41.612

[Desmarcar Favorito](#) [Ver Evento Detalhado](#)

Figura 52 Página Favoritos.

Pesquisar Eventos Autenticação

> Página de Registo

**Novo Utilizador**

Nome	<input type="text"/>
Username	<input type="text"/>
Email	<input type="text" value="rafael"/>
Password	<input type="password" value="..."/>
Reintroduza a Password	<input type="text"/>

**Localize zona de interesse no Mapa**

Latitude Menor	<input type="text" value="31.86906525977223"/>
Longitude Menor	<input type="text" value="-14.22052001953125"/>
Latitude Maior	<input type="text" value="45.5147369053"/>
Longitude Maior	<input type="text" value="1.16033935546875"/>

Fotografia  No file chosen

Figura 53 Página de Registo na Aplicação Web.

[Pesquisar Eventos](#) [Favoritos](#)

> Visualização de Eventos > Página Principal Rегистados.

| generated by WebRatio® |



Figura 54 Página principal para utilizadores registados.

[Pesquisar Eventos](#) [Autenticação](#)

> Página Principal Guests



Figura 55 Página principal para Visitantes.

## **10. Manual de Utilizador da Aplicação Final**

### **Nota Introdutória ao Manual**

O objetivo do presente anexo passa por uma explicação, passo-a-passo, dos procedimentos de certas funcionalidades presentes na aplicação web.

Para ser feita tal explicação e demonstração, este manual dividir-se-á em vários “capítulos”, nos quais serão feitas, para cada um deles, breves anotações e comentários dos passos que terão de ser tomados pelo utilizador para executar dada funcionalidade. Em conjunto a tais comentários serão ainda apresentadas imagens, devidamente contextualizadas, que permitirão ao leitor deste manual ter uma melhor visão daquilo que está a ser explicado.

Desta forma, e mais concretamente, este manual de utilizador abordará as funcionalidades de: “Alterar Perfil”, “Criar Evento”, “Gerir Eventos”, “Alterar Evento”, “Gerir Comentários”, “Gerir Newsletter”, “Gerir Utilizadores”, “Visualizar Página de Evento”, “Visualizar Página Inicial”, “Fazer Pesquisa”, “Ver Eventos Favoritos” e, por fim, “Ver os Meus Eventos”.

Por fim, é preciso ainda notar que, para cada funcionalidade acima apresentada, será sempre feita referência de qual ou quais o(s) tipo(s) de utilizadores que poderão usufruir da funcionalidade que está a ser descrita. Esta referência será sempre feita em lugares da descrição que surjam oportunos.

## **Alterar Perfil**

A funcionalidade de nome “Alterar Perfil” surge como uma prática executada com alguma regularidade por os utilizadores de um sistema web e, desta forma, é presente transversalmente à maioria das aplicações atuais.

Para alteração de perfil o utilizador do sistema deve ter feito anteriormente login com sucesso surgindo, assim, como utilizador autenticado. Este utilizador autenticado acederá à pagina respetiva a alteração de perfil caso pretenda alterar alguns dos seus dados pessoais. Os dados que poderão ser alterados são o email, a password, o nome, a área geográfica vista como de interesse para o utlizador em causa e a fotografia de exibição. Depois de tal atualização dos dados, as modificações deverão ser salvas para que possam passar a efetivas. A página de atualização de dados de perfil pode ser acesse clicando simplesmente no link correspondente à alteração de dados que se encontra na barra horizontal de menu (esta encontra-se sempre presente no cima de qualquer página da aplicação).

Abaixo é apresentada, a título demonstrativo, a página de alteração de perfil com alguns dados preenchidos para simular uma atualização de dados na conta do utilizador ummei.

**Eventhis**

Bem vindo ummei!

[Logout](#)

Página Inicial | Pesquisar Eventos | Ver Favoritos | Alterar Dados | Criar Evento | Os Meus Eventos

## Actualizar Perfil ummei

Fields with \* are required.

Email:

Password:

Nome:

Seleccione a sua área de interesse:

Fotografia:  Nenhum f...cionado

Copyright © 2012 by My Company.  
All Rights Reserved.  
Powered by Tiki Framework.

Abra

Organizar ▾ Nova pasta

**Biblioteca**  
Inclui: 2 localizações

- ★ Favoritos
  - Ambiente de trabalho
  - Localizações
  - Transferências
  - Dropbox
- Bibliotecas
  - Documentos
  - Imagens**
  - Música
  - Vídeos

Nome do ficheiro: EENG.jpg

Por fim é necessário rever quais os tipos de utilizadores que poderão aceder a esta página que, como já foi acima referido, são os utilizadores autenticados, isto é, os utilizadores que já tenham feito previamente login com sucesso no sistema.

## Criar Evento

A página respetiva à criação de um novo evento pode ser acedida simplesmente clicando no link que se encontra no menu horizontal da aplicação. Esta funcionalidade vem responder a um requisito do sistema que passa pela necessidade de criação de novos eventos por utilizadores que se encontrem autenticados na aplicação.

Para criar e catalogar tal evento, este deverá ser descrito por um nome e por um breve resumo descriptivo do mesmo. Além de tal, será ainda necessário definir qual a data de início e de fim para o evento em causa para, desta forma, não termos eventos no sistema cuja data de realização já passou podendo, assim, induzir em erro os clientes e utilizadores da aplicação web. Por fim, e além dos atributos já descritos, terão de ser atribuídos valores que identifiquem o evento como a localização geográfica onde irão decorrer e, ainda, o tipo de evento que se trata. Este tipo de evento terá de ser escolhido de entre uma lista de tipos já pré-definidos para etiquetarem um novo evento.

A página respetiva à criação de um novo evento encontra-se abaixo apresentada.

# Eventhis

Bem vindo ummei!



[Logout](#)

[Página Inicial](#) [Pesquisar Eventos](#) [Ver Favoritos](#) [Alterar Dados](#) [Criar Evento](#) [Os Meus Eventos](#)

## Criar Evento

Fields with \* are required.

Título do Evento

Descrição do Evento

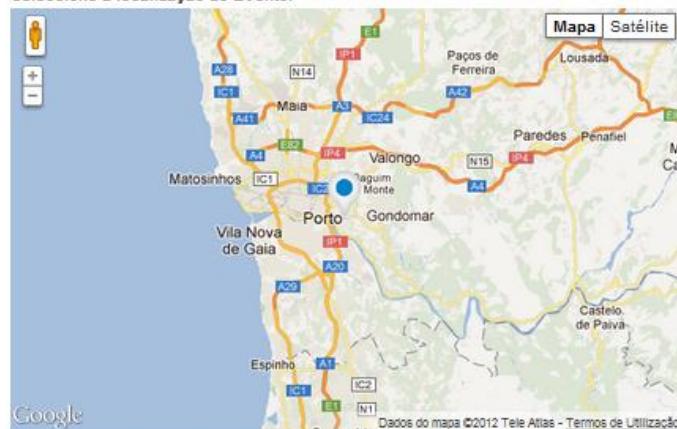
Data de Fim

 ...

Data de Ínicio

 ...

Selecione a localização do Evento:



Google

Tipo

 ▾

[Create](#)

Copyright © 2012 by My Company.  
All Rights Reserved.  
Powered by [Yii Framework](#).

Uma vez mais é necessário identificar quais os utilizadores que poderão criar um novo evento e, desta forma, aceder a tal funcionalidade e página. Estes utilizadores passam por ser os utilizadores que surjam no sistema como autenticados, isto é, utilizadores que tenham previamente feito login com sucesso na aplicação Eventhis.

## **Gerir Eventos**

A página respetiva à gestão de eventos só se encontra disponível para os administradores do sistema e, portanto, surge apenas disponível na barra horizontal de menu em contas cujo login tenha sido feito como administrador da aplicação.

Esta página poderá ser acedida clicando, a partir da homepage, no link respetivo aos eventos e, de seguida, no link respetivo à gestão ou, em inglês, “Manage”. Tal pode ser observado na figura abaixo.

Esta gestão de eventos passa pela possibilidade de vizualizar os eventos que existem no momento na base de dados do Eventhis, pela alteração de tais eventos ao nível da sua descrição, título, entre outros, e , ainda, pela possibilidade de remover eventos que, a ver do administrador, devam ser apagados.

Abaixo poderá ser observada a página respetiva à gestão de tais eventos onde surgem, a título de exemplo, alguns exemplos de eventos já existentes no sistema.

# Eventhis



[Gerir Eventos](#) [Gerir Comentários](#) [Gerir Utilizadores](#) [Gerir Newsletters](#)

[Home](#) » [Eventos](#) » Manage

## Gestão de Eventos

Podem ser usados operadores de comparação no início do valor de uma pesquisa (<, <=, >, >=, <> or =).

Displaying 1-3 of 3 result(s).

Idevento	Data de Fim	Data de Início	Título do Evento	Descrição do Evento	Latitude ▲	
1	2012-05-25 00:00:00	2012-06-03 00:00:00	Rock in Rio-Lisboa 2012	O Rock in Rio é o maior evento de música e entretenimento de todos os tempos, contando já com 10 edições realizadas no Brasil, Portugal e Espanha que reuniram perto de 6 milhões de pessoas.	38.73516380751677	
2	2012-06-23 00:00:00	2012-02-24 00:00:00	Festa S.João	O São João do Porto é uma festa popular, católica, que tem lugar de 23 para 24 de Junho na cidade do Porto, em Portugal, em que se celebra o nascimento de São João Batista.	41.14633755876385	
3	2012-05-30 19:00:00	2012-05-30 21:00:00	Jogo FCPorto-Chelsea	Jogo amigável entre duas das melhores equipas da actualidade desportiva internacional. Chelsea treinado por Terry, substituto do sr. Milhões, e FCP treinado por Domingos nervoso, que substitui VP.	41.16179519272488	

Copyright © 2012 by My Company.  
All Rights Reserved.  
Powered by [Yii Framework](#).

Como já foi dito acima, só utilizadores vistos pela aplicação como administradores terão as permissões necessárias para poderem fazer gestão dos eventos que já existam alojados no servidor de base de dados da aplicação Eventhis

## Alterar Evento

Ao contrário da página respetiva à gestão de eventos, na qual que se poderiam fazer alterações ou mesmo apagar qualquer evento que se encontrasse na base de dados da aplicação, nesta página apenas poderão ser alterados eventos que tenham sido criados pelo utilizador em causa. Desta forma é fácil compreender que, para poder aceder a tal página, o utilizador deve encontrar-se autenticado pelo sistema e, além de tal, para poder alterar algum evento, o utilizador autenticado deverá previamente ter acedido à página de criação de evento e, portanto, ter criado um novo evento na aplicação.

Os dados que poderão ser alterados referentes a um dado evento são o título do evento, a descrição do evento, a data de início do evento e data de fim do mesmo evento, a localização geográfica onde o evento se realizará e, ainda, o tipo do evento em causa. De salientar que a alteração da localização geográfica do evento deverá ser feito selecionando-se o local respetivo no mapa que se encontra na página.

Para aceder a tal página de alteração de evento o utilizador deverá previamente selecionar a opção respetiva à alteração de dados e, posteriormente, selecionar a alteração de eventos e escolher qual o evento que realmente pretende alterar. Abaixo será apenas apresentada a página respetiva a alteração de um evento já selecionado previamente e surgem ainda, a título demonstrativo, alguns campos já preenchidos que pretendem simular a alteração real de um evento.

# Eventhis



Página Inicial Pesquisar Eventos Ver Favoritos Alterar Dados Criar Evento Os Meus Eventos

## Alterar Evento Rock in Rio-Lisboa 2012

Fields with \* are required.

Título do Evento

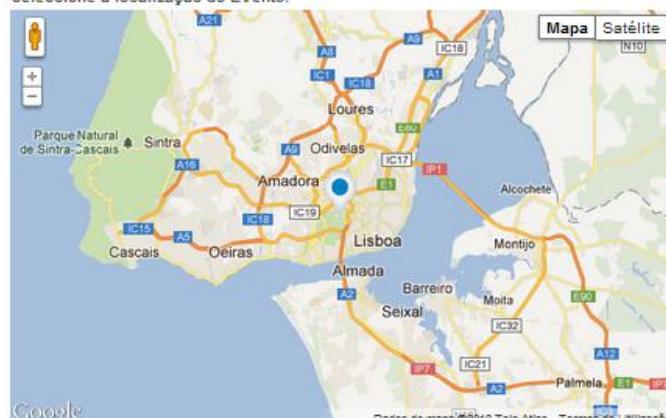
Descrição do Evento

O Rock in Rio é o maior evento de música e entretenimento de todos os tempos, contando já com 10 edições realizadas no Brasil, Portugal e Espanha que reuniram perto de 6 milhões de pessoas.

Data de Fim

Data de Ínicio

Selecione a localização do Evento:



Google

Tipo

Copyright © 2012 by My Company.  
All Rights Reserved.  
Powered by [Yii Framework](#).

Por fim, e como tradicionalmente tem sido feito, surge agora a necessidade de identificar quais os tipos de utilizadores que podem aceder a tal página e, portanto, podem usufruir de tal funcionalidade. Mais uma vez, para poder ter acesso a tal página o utilizador deverá estar autenticado porém, desta vez, além de ter feito previamente login com sucesso, o utilizador deverá ainda já ter criado algum evento para, agora, poder alterá-lo, poder atualizar os dados respetivos ao mesmo.

## **Gerir Comentários**

De forma análoga à gestão de eventos, a gestão de comentários possibilita ao administrador gerir todos os comentários que tenham sido feitos pelos utilizadores do sistema a qualquer evento presente na aplicação.

Uma vez mais, o administrador do sistema poderá, além de visualizar os comentários, modificá-los e, ainda, quando surja a necessidade de tal, apagar comentários. Para tal gestão, o comentário surge sempre com um atributo associado de nome “Idevento” que pretende associar ao comentário qual o seu evento, isto é, identifica inequivocamente que dado comentário foi feito a um evento com um dado número de identificação.

Para poder aceder a tal página o administrador apenas terá de clicar no link respetivo que se encontra no menu horizontal da aplicação. Uma vez mais é necessário salientar que tal link só aparecerá caso o sistema identifique o utilizador como sendo um administrador do Eventhis.

Abaixo surge a imagem respetiva à página em questão onde já se poderão visualizar alguns comentários que foram feitos a título demonstrativo.

Bem vindo admin!



[Logout](#)

**Eventhis**

[Gerir Eventos](#) [Gerir Comentários](#) [Gerir Utilizadores](#) [Gerir Newsletters](#)

## Gerir Comentários

Podem ser usados operadores de comparação no início do valor de uma pesquisa (<, <=, >, >=, <> or =).

Displaying 1-5 of 5 result(s).					
Idcomentario	Texto	Dtcom	Utilizador Oid	Evento Idevento	
25	Ola	2012-02-17 17:33:35	1	1	
26	Pois.... agora parece estar a funcionar convenientemente!!!!	2012-02-17 17:36:34	1	1	
27	Tenciono ir... Já tenho martelo do ano passado.	2012-02-17 17:54:12	1	2	
28	Se o Porto estiver na frente vou, caso contrário não!!!	2012-02-17 22:34:32	1	2	
29	Fake! Fake! Fake! Fake! Fakel! Fakel! Fakel!	2012-02-17 22:39:35	1	3	

---

Copyright © 2012 by My Company.  
All Rights Reserved.  
Powered by [Yii Framework](#).

Para finalizar, e para reforçar o que já foi dito em cima, tal página poderá ser acedida por administradores do sistema, isto é, por utilizadores quem tenha feito com sucesso login com credenciais de administrador da aplicação.

## Gerir Newsletter

Antes de avançarmos na explicação de utilização da página respetiva à gestão de newsletter convém explicar o nosso conceito de newsletter e como este é expostos. Desta forma, serão publicitados, numa newsletter a enviar por email, todos os eventos que tenham pago uma taxa extra para poderem usufruir de tal serviço. Estes eventos sugem apresentados na página intitulada “Gerir Newsletter”. Além de tal, e a catalogar tal newsletter, terá de ser escolhido um nome identificativo para a mesma, tal nome deverá ser sugestivo do que trata tal publicidade. O serviço de newsletter será enviado a todos os utilizadores que tenham escolhido a opção de pretensão de receber tais serviços publicitários.

Além de tudo o que acima foi referido, é importante salientar que a partir desta página tal newsletter poderá ser criada e, além de tal, poderá ainda ser enviada pelo administrador do sistema. Abaixo surge apresentada tal página com eventos, a título de exemplo, que serão publicitados.

The screenshot shows the 'Eventhis' application's newsletter management page. At the top right, it says 'Bem vindo admin!' with a user icon and a 'Logout' link. The main title is 'Eventhis'. Below it, there's a navigation bar with links: 'Gerir Eventos', 'Gerir Comentários', 'Gerir Utilizadores', and 'Gerir Newsletters' (which is highlighted). The main content area is titled 'Newsletters' and has a sub-section 'Eventos Para Newsletter'. It displays a table with three results:

Título do Evento	Data de Início	Data de Fim
Rock in Rio-Lisboa 2012	2012-06-03 00:00:00	2012-05-25 00:00:00
Festa S.João	2012-02-24 00:00:00	2012-06-23 00:00:00
Jogo FCPorto-Chelsea	2012-05-30 21:00:00	2012-05-30 19:00:00

Below the table, there's a form with a 'Título' input field and an 'Enviar Newsletter' button. At the bottom, there's a copyright notice: 'Copyright © 2012 by My Company. All Rights Reserved. Powered by [Yii Framework](#)'.

Tal página, salientando o que já foi referido, só poderá ser acedida por administradores do sistema e, portanto, as newsletters só poderão ser criadas e enviadas por utilizadores que tenham feito login com sucesso com credenciais de administradores da aplicação Eventhis.

## Gerir Utilizadores

De forma análoga às funções de gestão já apresentadas, apenas o utilizador visto pelo sistema como administrador poderão gerir os utilizadores existentes na aplicação. Esta gestão passa pela visualização dos utilizadores existentes, onde poderemos ver o seu “username”, o

seu número de identificação, o seu email e, ainda, o seu nome. Além de tal, o administrador poderá ainda fazer alterações aos campos respetivos a cada utilizador e, poderá ainda, quando surgir necessário, apagar um ou mais utilizadores da base de dados respetiva à aplicação Eventhis.

Para aceder a tal página, o utilizador apenas precisará de “clicar” o link que é respetivo à opção “Gerir Utilizadores”, link esse que se situa no menu horizontal presente no cima de cada uma das páginas da aplicação. É necessário salientar uma vez mais que tal link só será apresentado no caso do utilizador surgir como um administrador do sistema, caso contrário as funcionalidades que terá visíveis para escolher serão completamente diferentes e dependerão do tipo de utilizador que é.

Abaixo surge, como tem sido feito tradicionalmente, uma imagem respetiva a tal página de gestão de utilizadores. Esta página conta já com alguns utilizadores que surgem a título demonstrativo para este manual.

The screenshot shows the 'Eventhis' application interface. At the top, there is a header with the text 'Bem vindo admin!' and a user profile icon. Below the header, the main title 'Eventhis' is displayed. A navigation bar contains links for 'Gerir Eventos', 'Gerir Comentários', 'Gerir Utilizadores' (which is highlighted in blue), and 'Gerir Newsletters'. The main content area is titled 'Gerir Utilizadores' and includes a note about using comparison operators in searches. A table displays four user records:

Username	Oid	Email	Nome	
ummeli	1	rafaelumeli@gmail.com	Rafael Eduardo de Carvalho Pereira	
lala	5	luis@gmail.com	Luis Araújo	
olga	7	ola@gmail.com	Olga Lanhoso	
admin	8	admin@gmail.com	Rafael Eduardo de Carvalho Pereira	

At the bottom of the page, there is a copyright notice and a link to the Yii Framework.

Esta página, como facilmente se compreenderá, só poderá ser acedida por um utilizador que tenha feito um login prévio com as credenciais de administrador de sistema e, tal login, tenha sido realizado com sucesso.

## [Visualizar Página de Evento](#)

A página respetiva à visualização de evento pode ser acedida de diversos pontos, isto é, de diversas páginas da aplicação. Focaremos portanto a atenção na mesma e não em como a poderemos aceder devido ao largo espectro que esta componente teria.

Contrariamente ao que tem sido abordado, em que apenas um tipo de utilizadores poderia usufruir de uma dada funcionalidade, por sua vez, a visualização da página respetiva a um evento comprehende tanto utilizadores autenticados como simples visitantes da aplicação Eventhis. Contudo é preciso salientar que existe uma diferença na especificidade de funcionalidades disponíveis para um e para o outro. Mais concretamente, o utilizador autenticado poderá fazer comentários na página de visualização do dado evento contudo, o visitante não terá disponível tal funcionalidade. Para ter acesso à mesma terá primeiro de fazer login com sucesso usando credenciais de utilizador existente no sistema.

Os dados que poderão ser observados nesta página são, como é óbvio, respetivos a um evento concreto e pretendem catalogar e descrever tal evento. Estes dados são o título do evento, a descrição do evento, a latitude e longitude do local onde o evento ocorre ou ocorrerá, a data de início e de final do evento, o tipo do evento, a morada do evento e, ainda, uma representação visual em forma de mapa dinâmico do local onde o evento ocorre/ocorrerá. Além de todos estes dados será ainda possível visualizar-se todos os comentários que foram feitos pelos utilizadores devidamente autenticados.

Como já foi dito, além de toda a informação que pode ser encontrada neste tipo de página e que acima foi enumerada, os utilizadores que estejam autenticados poderão ainda fazer novos comentários relativos ao evento em causa.

Abaixo segue uma imagem demonstrativa daquilo que pode ser uma página de visualização de evento. De notar que esta visualização é respetiva a um utilizador que tenha feito um previo login com sucesso no sistema e, portanto, a caixa de possibilidade de fazer comentário é apresentada.



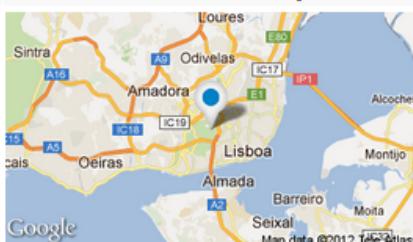
# Eventhis

Página Inicial Pesquisar Eventos Ver Favoritos Alterar Dados Criar Evento Os Meus Eventos

Operations  
Alterar Evento  
Apagar Evento

## Evento Rock in Rio-Lisboa 2012

Título do Evento	Rock in Rio-Lisboa 2012
Descrição do Evento	O Rock in Rio é o maior evento de música e entretenimento de todos os tempos, contando já com 10 edições realizadas no Brasil, Portugal e Espanha que reuniram perto de 6 milhões de pessoas.
Latitude	38.73516380751677
Longitude	-9.179045232177705
Data de Fim	2012-05-25 00:00:00
Data de Início	2012-05-03 00:00:00
Tipo	festivo
Morada	Caminho das Pedreiras, 1500 Lisbon, Portugal



### Comentários

Displaying 1-2 of 2 result(s).



ummei diz.  
Pois.... agora parece estar a funcionar convenientemente!!!!!!



ummei diz.  
Ola

Fields with \* are required.

Texto

Apenas como forma de manter a estrutura constante relativamente a todas as funcionalidades já apresentadas, surge esta conclusão à funcionalidade que, visa principalmente realçar os privilégios necessários para se poder usufruir da mesma. Desta forma, todas as funcionalidades presentes nesta página, como a visualização de dados relativos ao evento, a visualização de comentários e possibilidade de fazer comentários ao evento surgem como possíveis a todos os utilizadores vistos como autenticados pelo sistema.

Por sua vez, um utilizador visitante, isto é, um utilizador que não tenha feito login com sucesso na aplicação poderá visualizar todos os dados relativos ao evento, poderá ainda ver os comentários que já foram feitos, contudo não poderá fazer novos comentários caso pretenda. Aliás, a funcionalidade inerente à escrita de novos comentários nem será apresentada a um utilizador visitante.

Para concluir, é preciso apenas explicar o porquê da existência de uma estrela junto ao título do evento. Tal acontece pois este evento trata-se de um que foi escolhido pelo utilizador autenticado como um dos seus favoritos e, portanto, poderá ser acedido de forma rápida na sua página de favoritos. Uma abordagem mais explicativa e completa relativamente aos eventos favoritos será feito aquando a abordagem e explicação da página de “Ver Eventos Favoritos”.

## [Visualizar Página Inicial](#)

A página inicial será a primeira página a ser vista por um visitante que tenha entrado na aplicação ou, ainda, por um utilizador que já tenha feito login com sucesso no sistema, encontrando-se agora com privilégios inerentes a um utilizador autenticado. Por isso mesmo as páginas iniciais são diferentes para cada um dos tipos de utilizadores.

De salientar apenas que, para a página inicial vista por um utilizador autenticado, a mesma apresentará um mapa cujo centramento é feito pela localização escolhida como favorita aquando o registo ou atualização de dados do perfil de utilizador.

Esta página dispensa grandes apresentações e explicações e, portanto, serão apenas mostradas abaixo duas figuras respetivas à pagina inicial vista por um visitante e vista por um utilizador autenticado, dispensando, desta forma, anotações relativas às mesmas.

Página Inicial vista por um Utilizador Autenticado

# Eventhis

Bem vindo ummei!  
  
[Logout](#)

Página Inicial

Pesquisar Eventos

Ver Favoritos

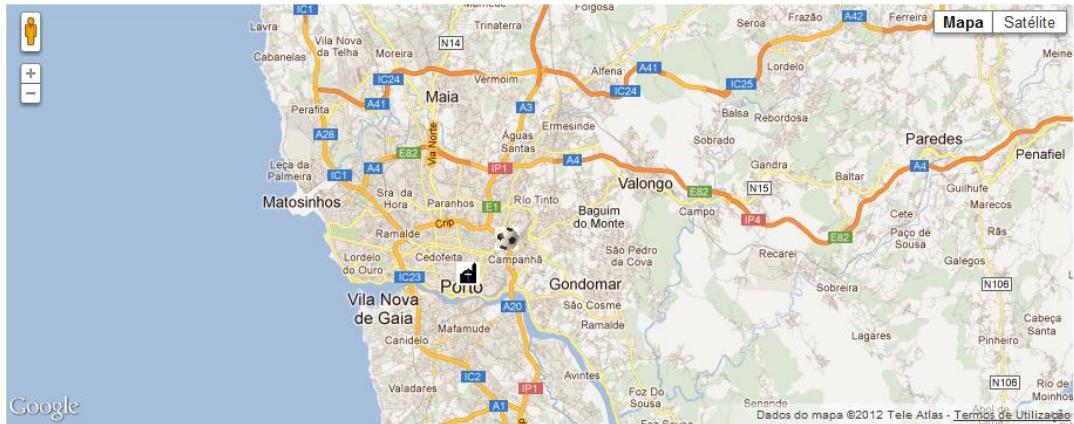
Alterar Dados

Criar Evento

Os Meus Eventos

Mapa

Satélite



## Página Inicial vista de um Utilizador Visitante

# Eventhis

Username

\*\*\*\*\*

Remember me

[Login](#)

[Sign Up](#)

Página Inicial

Pesquisar Eventos



Mapa

Satélite



Apenas para concluir, nas imagens acima apresentadas é notável um decréscimo enorme nas funcionalidades disponíveis de um utilizador autenticado para um visitante e, tal, deve-se à especificidade de privilégios que cada um tem, como tem sido referido ao longo de todo este manual de utilizador.

## Fazer Pesquisa

A funcionalidade de pesquisa presente no Eventhis passa por permitir que sejam feitas buscas por um dado evento com base em algum texto que o utilizador ache que poderá pertencer a algum atributo do evento. Além de tal, a pesquisa poderá ainda ser filtrada caso o utilizador queira escolher qual o tipo do evento, sendo este “filtro” opcional.

Para aceder a tal página de pesquisa o utilizador apenas terá de seguir o link que se encontra no menu horizontal da aplicação e que tem de nome “Pesquisar Eventos”.

Para demonstrar tal página de pesquisa de eventos, será apresentada abaixo uma imagem onde já constam alguns campos de pesquisa previamente preenchidos a título demonstrativo. Além de tal, e como se poderá constar, é encontrado um evento respetivo a tal pesquisa e o mesmo é apresentado sob a forma de tabela na página de pesquisa. O evento poderá ainda posteriormente acedido caso o utilizador clique no título associado ao evento encontrado pelos critérios de pesquisa.

The screenshot shows the Eventhis application interface. At the top, there is a navigation bar with links: Página Inicial, Pesquisar Eventos (which is highlighted in blue), Ver Favoritos, Alterar Dados, Criar Evento, and Os Meus Eventos. On the right side of the header, there is a welcome message "Bem vindo ummei!" next to a user profile icon and a "Logout" link. The main content area has a search form with a "Texto" input field containing "Rio" and a "Tipo" dropdown menu showing options like "todos", "regional", "desportivo", "religioso", "festivo", "astronómico", "político", "artístico", and "feiras". Below the search form, a message says "Displaying 1-1 of 1 result(s)". A table lists one event: "Rock in Rio-Lisboa 2012" at "Caminho das Moradas" in "Lisbon, Portugal" from "2012-06-03 00:00:00" to "2012-05-25 00:00:00". At the bottom of the page, there is a copyright notice: "Copyright © 2012 by My Company. All Rights Reserved. Powered by [Yii Framework](#)".

Por fim, é necessário salientar que apesar da imagem acima ser demonstrativa para o acesso feito por um utilizador autenticado, tal acesso a esta funcionalidade pode ser feito quer por utilizadores autenticados, quer por utilizadores que ainda não tenham feito login com sucesso no sistema e, portanto, surjam como visitantes. Este é um dos raros casos em que a funcionalidade poderá ser acedida totalmente por mais do que um tipo de utilizadores, utilizadores com privilégios de visitante e utilizadores com privilégio de autenticados, isto é, utilizadores que já fizeram login com sucesso na aplicação web.

## **Ver Eventos Favoritos**

A página respetiva à visualização de eventos favoritos nada mais é do que uma página que agrega, de forma estruturada, todos os eventos que ao longo da navegação pela aplicação foram escolhidos pelo utilizador como sendo favoritos por ele.

Tal prática é muito normal e frequentemente feita em browsers sob o nome de “bookmarks” ou “marcadores”. A sua funcionalidade é exatamente a mesma e passa por uma agravação de um conjunto de eventos marcados como favoritos por um dado utilizador autenticado. Ao longo da navegação pela aplicação, sempre que o utilizador ache que encontrou um evento interessante ou susceptível de ser importante para o mesmo, ele marca o evento como favorito e, posteriormente, poderá visualizar tal evento simplesmente clicando no título do mesmo presente na página intitulada “Ver Eventos Favoritos”.

Desta forma, comprehende-se facilmente que além de simplesmente apresentar os eventos marcados pelo utilizador como favoritos, a funcionalidade de visualização de eventos favoritos permitirá ainda o acesso aos mesmos aquando o “click” sobre o título dos mesmos.

Será apresentada abaixo a página respetiva a tal funcionalidade com eventos previamente escolhidos como favoritos pelo utilizador “ummei”. Esta escolha de eventos favoritos foi feita a título demonstrativo.

Bem vindo ummei!



[Logout](#)

**Eventhis**

Página Inicial Pesquisar Eventos Ver Favoritos Alterar Dados Criar Evento Os Meus Eventos

Eventos Favoritos

Displaying 1-3 of 3 result(s).

[Rock in Rio-Lisboa 2012](#) ★

[Jogo FCPorto-Chelsea](#) ★

[Festa S.João](#) ★

---

Copyright © 2012 by My Company.  
All Rights Reserved.  
Powered by [Yii Framework](#).

Como facilmente se compreenderá, apenas utilizadores autenticados poderão aceder à pagina “Ver Favoritos” e, apenas poderão ser visualizados eventos em tal página caso previamente se tenha feito uma escolha como favorito de algum evento. Caso contrário, e de forma obvia, não serão encontrados nenhuns eventos respetivos a esta página.

## Ver os Meus Eventos

Um utilizador autenticado poderá querer ver todos os seus eventos estruturados de uma forma organizada. É neste contexto que surge a necessidade de existência de uma página que permita a satisfação de tal funcionalidade. Esta página é a página “Ver os Meus Eventos”, que pode ser acedida simplesmente clicando no link respetivo que se encontra no menu horizontal do topo da aplicação. Como tal, sempre que o utilizador queira ver os seus eventos apenas terá de aceder à página de visualizações dos seus eventos e, poderá fazê-lo de forma rápida simplesmente clicando no apontador presente no menu horizontal que se encontra presente em toda a aplicação web.

Esta funcionalidade, para cada evento, apresenta dados respetivos ao mesmo que são: o título do evento, a descrição do evento, a data de início e de fim do evento e a localização

geográfica do mesmo sob a forma de mapa. Além de tal, existe ainda um link que nos redirecionar para a página referente ao evento em questão.

De forma a estruturar a informação a apresentar, cada evento é apresentado dentro de uma “caixa” por forma a melhor organizar e tornar visualmente mais agradável a apresentação de todos os eventos.

De seguida será apresentada a página em discussão na qual constam alguns eventos criados a título de exemplo pelo utilizador autenticado de nome “ummei”.

Bem vindo ummei!



[Logout](#)

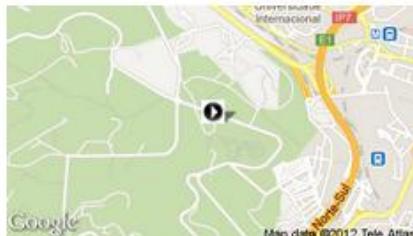
## Eventhis

Página Inicial | Pesquisar Eventos | Ver Favoritos | Alterar Dados | Criar Evento | Os Meus Eventos

### Eventos

Displaying 1-3 of 3 result(s).

Título do Evento: Rock in Rio-Lisboa 2012  
Descrição do Evento: O Rock in Rio é o maior evento de música e entretenimento de todos os tempos, contando já com 10 edições realizadas no Brasil, Portugal e Espanha que reuniram perto de 6 milhões de pessoas.  
Data de Início: 2012-06-03 00:00:00  
Data de Fim: 2012-05-25 00:00:00  
Localização:



[Página do Evento](#)

Título do Evento: Festa S.João  
Descrição do Evento: O São João do Porto é uma festa popular, católica, que tem lugar de 23 para 24 de Junho na cidade do Porto, em Portugal, em que se celebra o nascimento de São João Batista.  
Data de Início: 2012-02-24 00:00:00  
Data de Fim: 2012-06-23 00:00:00  
Localização:



[Página do Evento](#)

Título do Evento: Jogo FCPorto-Chelsea  
Descrição do Evento: Jogo amigável entre duas das melhores equipas da actualidade desportiva internacional. Chelsea treinado por Terry, substituto do sr. Milhões, e FCP treinado por Domingos nervoso, que substitui VP.

Por forma a concluir tal explicação desta funcionalidade, surge ainda a necessidade de indicar quais os tipos de utilizadores que poderão ter acesso à mesma. De facto, esta funcionalidade poderá ser acedida e utilizada por utilizadores que estejam autenticados na aplicação web.

Além do referido, um utilizador autenticado só verá eventos, nesta página de visualização dos seus eventos, caso previamente já tenha criado algum. À medida que os for criando poderá sempre, de forma atualizada, visualizar e aceder aos mesmos de forma rápida e prática, para tal apenas precisa de aceder a esta página que é intitulada de “Os Meus Eventos”.