



Universidade do Minho

Departamento de Informática

Mestrado Integrado em Engenharia Informática

Momenter

Desenvolvimento do Projeto

Projeto de Engenharia Informática

Equipa de Desenvolvimento:

Carlos Pereira, nº a61887

Diana Cristina Abreu Lopes, nº pg38925

Diogo Lopes da Silva, nº a76407

Miguel Dias Miranda, nº a74726

Rafael Costa, nº a61799

Rogério Moreira, nº a74634

Samuel Ferreira, nº a76507

Braga, 1 de Fevereiro de 2019

Conteúdo

1	Introdução	2
1.1	Objetivos	2
2	Levantamento de Requisitos	3
2.1	Estrutura dos Requisitos	3
2.2	Requisitos Funcionais e de Dados [9]	4
2.2.1	Gestão de Utilizadores	4
2.2.2	Gestão de Eventos	5
2.3	Requisitos Não Funcionais	8
2.3.1	Aparência [10]	8
2.3.2	Usabilidade [11]	9
2.3.3	Desempenho [12]	9
2.3.4	Operacionalidade [13]	10
2.3.5	Manutenção e Suporte [14]	10
2.3.6	Segurança [15]	10
3	Documentação UML	11
3.1	Modelo de Domínio	11
3.2	Diagrama Use Cases	12
3.3	Diagrama de Classes	14
3.4	Arquitetura do Sistema	15
4	Design da Interface	17
4.1	Mockups	17
4.2	Wireframes	19
5	Modelação da Camada de Dados	24
5.1	Modelo Lógico	24
5.1.1	Eventos	25
5.1.2	Utilizadores	25
5.1.3	Organizações	25
5.1.4	Categorias e <i>Tags</i>	25
5.2	Povoamento automatizado - Fontes de dados externas	28
6	Desenvolvimento e Implementação	29
6.1	Arquitetura Aplicação	29
6.1.1	Organograma	29
6.2	Módulos Desenvolvidos	31
6.2.1	Front end	31
6.2.2	Back-end	32
6.2.3	Sistema de recomendação	33
7	Trabalho Futuro	35
8	Conclusões	36

Capítulo 1

Introdução

Devido ao limitado espetro temporal disponível para desenvolver a versão inicial da plataforma *Momenter*, foi necessário traçar um conjunto de etapas de modelação e desenvolvimento bem definidas e ponderadas.

Nesse sentido, o presente documento serve como documentação de todas as principais características e decisões tomadas ao longo da fase de criação e implementação do projeto.

A contextualização do ambiente onde se insere a plataforma *Momenter* é descrito com detalhe no documento do Plano de Negócio, entregue junto com este documento. Sumariamente, *Momenter* centraliza-se na divulgação e promoção de eventos. Para os promotores, providenciam-se mecanismos de publicidade dinâmicos, capazes de distribuir a publicidade do evento por potências participantes que pertencem à comunidade *Momenter* e demonstraram ter interesse em eventos semelhantes. Na perspetiva de um utilizar, a recomendação de eventos permite que cada perfil seja único, mostrando apenas as recomendações tailhadas para aquele utilizador, com base nas suas preferências explicitamente fornecidas ou inferidas pela plataforma, à medida que este interage com a plataforma. Globalmente, fornecemos ainda os normais mecanismos de pesquisa, filtragem e visualização de informações de eventos.

Uma vez que a sua principal funcionalidade é a apresentação de eventos, questões de design e uma interação intuitiva são fundamentais. Num outro ponto de vista, a criação de um sistema de recomendação único para este contexto exige a modelação e análise de um conjunto de cenários, que permitam testar a qualidade das recomendações.

A construção de cada uma destas componentes é assim detalhada ao longo de todo o relatório.

1.1 Objetivos

Com a organização e conteúdos deste documentos, pretende-se:

- Sumarizar **alguns** dos principais requisitos (estrutura e justificação da mesma) levantados no projeto. O conjunto completo de requisito encontra-se no Documento de Requisitos enviado em Anexo;
- Descrever o conjunto de decisões de modelação de dados e arquitetura da aplicação, realizando a análise de alguns modelos UML principais;
- Apresentar o esboço inicial dos mockups da aplicação, que posteriormente levaram à interface gráfica atualmente em uso;
- Abordar o modelo de base de dados criado para suportar o contexto da aplicação;
- Documentar as principais decisões a nível de tecnologias e ferramentas utilizadas nas diferentes camadas da arquitetura da aplicação.

Capítulo 2

Levantamento de Requisitos

2.1 Estrutura dos Requisitos

No âmbito de desenvolvimento de um projeto de software, o levantamento de requisitos é geralmente dividido em dois documentos distintos:

- O Documento de Requisitos do Cliente, levantados de forma textual e clara face às intenções do stakeholder;
- O Documento de Requisitos do Sistema, usado pela equipa de desenvolvimento para compreender o enquadramento do requisito nos módulos do sistema e como implementa-lo.

Devido ao espectro temporal deste projeto, foi tomada a decisão de não produzir explicitamente dois documentos distintos. Num sentido de otimização de tempo, os requisitos levantados apresentam simultaneamente um conjunto de campos retirados dos Cartões de *Volere*, voltados para os Clientes e, o conjunto de etapas que descrevem o comportamento do requisito dentro do sistema. Desta forma, os requisitos podem ser entendidos quer por partes interessadas externas ao projeto, quer pela equipa de desenvolvimento, nas futuras fases de desenvolvimento do projeto.

Da estrutura dos cartões de *Volere*, pela forma simples e objetiva como descrevem um requisito, foram considerados como aspectos essenciais para os requisitos os campos:

- *Id* (número) do requisito;
- Tipo do requisito, considerando a numeração de 9 a 17 do modelo *Volere*;
- Descrição, clara e curta do requisito;
- Justificação, fornecendo a razão de existência do requisito;
- *Persona*, que originou o requisito;
- Prioridade, que define quais os requisitos a implementar nesta fase de desenvolvimento.

<u>ID Requisito:</u> N	<u>Tipo:</u> K
<u>Definição do requisito do utilizador:</u>	
<ul style="list-style-type: none">• Descrição: Eu <papel utilizador>, devo ser capaz/realizo <tarefa>.• Justificação: Benefícios / Breve descrição da necessidade da funcionalidade.	
<u>Persona:</u> Utilizador	

Estes campos são assim levantados para que o requisito seja descrito, compreendido e validado pelos eventuais *stakeholders* do projeto.

Num sentido mais técnico, cada template do cartão de *Volere* é acompanhado por uma *cartão* que descreve os requisitos do sistema necessários para implementar esse requisito. Para a compreensão destes procedimentos do sistema, será aconselhável rever o esquema da arquitetura da aplicação, o modelo de domínio e o diagrama de classes do sistema, como forma de compreender a interação entre os módulos referidos. Estes diagramas estão descritos no Documento de Desenvolvimento do projeto, anexado em paralelo com este Documento.

Da estrutura dos cartões com os requisitos de sistema, foram considerados como aspectos essenciais para os requisitos os campos:

- Restrições: Listagem de restrições ou limitações do requisitos, caso aplicável.
- Definição dos requisitos de sistema: Procedimentos necessários de implementar nos módulos do sistema de forma a desempenhar a tarefa do requisito.
- Prioridade: Para a priorização dos requisitos, é usada a técnica designada de *MoSCoW*. Esta técnica divide os objetos em quatro prioridades, sendo cada uma definida por:
 - **Must**: Requisitos que devem ser obrigatoriamente incluídos.
 - **Should**: Requisitos que devem ser considerados para o projeto.
 - **Could**: Requisitos que não são necessários mas são desejáveis para um bom produto.
 - **Won't**: Requisitos que poderão ser considerados posteriormente.

<u>ID Requisito:</u> 11	<u>Tipo:</u> 9
Restrições: Listagem de restrições, se aplicável. <ul style="list-style-type: none">• Restrição / limite do requisito (ex: só para users com login, só para promotores, etc);• ...;• Restrição N.	
Definição dos requisitos de sistema:	
<ol style="list-style-type: none">1. procedimento 1;2. ...;3. procedimento N.	
Prioridade: N	

Nas seguintes secções serão listados **apenas alguns dos principais requisitos levantados** no decorrer deste projeto. Recomenda-se a revisão do Documento de Requisitos para uma visão mais detalhada sobre todos os requisitos levantados e propícios de ser implementados no sistema.

2.2 Requisitos Funcionais e de Dados [9]

2.2.1 Gestão de Utilizadores

Na presente secção são levantados e descritos os requisitos funcionais relacionados com a gestão de contas de utilizadores como registo, login, cancelamento ou edição de parâmetros da conta.

A gestão de utilizadores apresenta-se idêntica para qualquer utilizador do sistema. Internamente, e conforme as funcionalidades escolhidas, o utilizador define o seu papel na plataforma, podendo aceder a funcionalidades de promotor ou de participante de eventos.

<u>ID Requisito:</u> 4	<u>Tipo:</u> 9
Definição do requisito do utilizador:	
<ul style="list-style-type: none">• Descrição: O utilizador pode preencher um questionário com as suas preferências.• Justificação: Este processo pode ser realizado aquando o registo ou em qualquer altura posterior. O questionário permitirá, por exemplo, marcar como preferenciais categorias pré apresentadas ou definir um raio em quilómetros, sobre o qual filtrar inicialmente eventos próximos de si.	
Este questionário serve como uma forma inicial para traçar um perfil de preferências do utilizador.	
Persona: Gabriela Baron	

ID Requisito: 4**Tipo:** 9**Definição dos requisitos de sistema:**

1. O sistema deve apresentar, através da *Fascade Layer*, uma opção (caixa de texto ou ícone sugestivo) referente à funcionalidade "Responder a Questionário";
2. O sistema deve solicitar várias informações referentes às preferências do utilizador. O questionário deve ser guiado e intuitivo. Nomeadamente, deve apresentar:
 - (a) Um conjunto de ícones (ou imagens) pré-definidos e sugestivas que represente as principais categorias do sistema (música, gastronomia, cinema, teatro, dança, desporto, outros, debates, convívio, outros);
 - (b) o raio de alcance sobre o qual filtrar eventos;
 - (c) A cidades (ou cidades) preferenciais. Por exemplo, selecionadas através de uma mapa.
3. Após o utilizador ter respondido ao questionário, o sistema deve armazenar as preferências indicadas utilizando o método *setUserPreferences*.
4. A interface utilizada neste requisito será utilizada tanto na inserção opcional de preferências, aquando o registo de um utilizador, ou quando o utilizador decidir atualizar/alterar os parâmetros das suas preferências.

Prioridade: *Should*

2.2.2 Gestão de Eventos

Na presente secção são levantados e descritos os requisitos funcionais relacionados com a gestão de Eventos na plataforma, por parte dos seus utilizadores.

Formalmente, um evento consiste num conjunto de atividades, realizadas numa determinada data e local. A sua intenção passa por agrupar pessoas com interesses em comum e relacionadas com o tema do evento, que observam ou interagem com as atividades a decorrer.

Nesse sentido a plataforma fornece um conjunto de serviços relacionados com eventos. Com particular importância, a plataforma permite a inserção, publicitação e visualização da descrição de eventos.

ID Requisito: 8**Tipo:** 9**Definição do requisito do utilizador:**

- **Descrição:** O utilizador, na qualidade de promotor, tem a possibilidade de criar eventos.
- **Justificação:** Um promotor consegue criar uma atividade com uma descrição do evento a promover.

Desta forma, o sistema realiza o papel de distribuição e publicitação do evento. Em paralelo, outros utilizadores, na perspetiva de participantes, visualizam as informações relativas a eventos que estejam a decorrer à sua volta.

Persona: Adriana Antunes

ID Requisito: 8**Tipo:** 9**Definição dos requisitos de sistema:**

1. Caso o utilizador se encontre a usar a plataforma com a sua conta ligada, o sistema deve apresentar, através da *Fascade Layer*, opção (caixa de texto ou ícone sugestivo) referente à funcionalidade "Inserir Evento";
2. Após a seleção da opção "Inserir Evento", o sistema deve apresentar uma interface para a inserção dos campos necessários para descrever um evento (Ver Requisito 9);
3. O utilizador (promotor) deve preencher os campos pedidos para posteriormente serem validados;
4. O utilizador pode escolher se pretende que o evento a criar seja um evento premium. Nesse sentido, deve existir uma opção (caixa de texto ou ícone sugestivo) referente à funcionalidade "Evento Premium";
Ou seja, o promotor pode pagar por um pacote premium que personaliza, de forma a tirar partido do sistema de recomendação direta e publicitação da plataforma Momenter.
A marcação desta opção e escolha entre os pacotes/serviços de publicitação disponibilizados são descritos em detalhe no Requisito 29 da Secção ??.
5. O utilizador pode escolher se pretende que o evento tenha a opção de Car Renting (ver Requisito 25). Nesse sentido, deve existir uma opção (caixa de texto ou ícone sugestivo) referente à ativação da funcionalidade "Car Renting" (ver Requisito 26);
6. O utilizador pode escolher se pretende que o evento tenha a opção de Hospedagem. Nesse sentido, deve existir uma opção (caixa de texto ou ícone sugestivo) referente à ativação da funcionalidade "Hospedagem";
7. Caso os campos inseridos sejam válidos:
 - Caso esta opção premium seja marcada, os parâmetros definidos (alcance e pacote de publicidade) são associados ao evento. Caso contrário, o evento é registado como um evento gratuito.
 - O sistema realiza o registo do evento na plataforma, recorrendo ao método *createEvent*. O evento ficará associado ao utilizador que o adicionou, na lista *organized_events*;
 - Uma mensagem de sucesso é apresentada ao utilizador (promotor).
 - O evento, independente do seu tipo, será listado no painel de eventos (ver Requisito 28);
 - Caso o evento seja premium, deve ser disparado o mecanismo de notificações de utilizadores, com base no pacote adquirido para o evento (ver Requisito 29 - eventos Premium e Requisito 33 - envio notificações).
8. Caso os campos inseridos sejam inválidos, a interface deve apresentar a anterior página de inserção dos dados do evento, indicando quais os campos mal indicados/para correção.

Prioridade: Must**ID Requisito:** 10**Tipo:** 9**Definição do requisito do utilizador:**

- **Descrição:** O utilizador é capaz de registar uma organização na plataforma.
- **Justificação:** Num sentido de representar uma entidade oficial enquanto promotor de eventos, um utilizador (promotor) pode registrar organizações. A organização representa assim a entidade que promove/financia externamente o evento.

Persona: Adriana Antunes

ID Requisito: 10**Tipos:** 9**Restrições:**

- A opção "Registar organização" deve existir apenas na interface de um utilizador registado, após login.
- O registo de uma organização deve ser validado pela equipa Momenter, no sentido de validar a credibilidade e veracidade da mesma.

Definição dos requisitos de sistema:

1. Na interface de criação de eventos (ver Requisito 8 - Registar Evento) ou, na interface das definições do utilizador, deve existir uma opção (caixa de texto ou ícone sugestivo) referente à função "Registar Organização";
2. Após a seleção dessa opção, o sistema deve transitar para a página de registo de uma organização. Devem ser solicitados os seguintes campos:
 - Nome;
 - Localização;
 - NIF;
 - Contactos (telefone, email, pombo correio ou outros)

No final da página, deve existir a opção (caixa de texto ou ícone sugestivo) relativo à opção "registar Organização", para salvar o processo.

3. Após a seleção de guardar o registo de uma organização, os campos inseridos devem ser recolhidos. O sistema deve realizar o registo da organização através do método *createOrganization*.
4. A equipa Momenter validará, internamente, a veracidade da Organização através de um contacto direto com a mesma.
Após a validação, o utilizador receberá uma notificação a confirmar a aceitação da organização.

Prioridade: Should**ID Requisito:** 19**Tipos:** 9**Definição do requisito do utilizador:**

- **Descrição:** O utilizador é capaz de visualizar um conjunto de eventos, agrupados com base nas suas preferências.
- **Justificação:** No sentido de tirar partido da principal proposta de valor da plataforma Momenter, o utilizar consegue visualizar os eventos que se adequam com as suas preferências, isto é, com base num sistema de recomendação.

Persona: Rui Fernandes

ID Requisito: 19**Tipos:** 9**Definição dos requisitos de sistema:**

1. A recomendação de eventos só se encontra disponível na interface de um utilizador registado e com a sua conta ligada (após login);
2. O sistema deve apresentar ao utilizador, através da *Fascade Layer*, opção (caixa de texto ou ícone sugestivo) referente à funcionalidade "Recomendação de Eventos";
3. Caso a referida opção seja selecionada, o sistema deve reunir o conjunto de eventos inseridos na plataforma que se adequam às preferências inferidas perante o perfil de utilização do utilizador. Para isso, deverá ser utilizado o método *getSuggestions*.
4. Os eventuais eventos encontrados devem puder ser visualizados tanto na forma de uma listagem como, projetados num mapa, conforme a sua localização (Requisito 20 - Alterar forma Visualização Eventos).
5. Caso o utilizador não tenha sessão iniciada na plataforma, o sistema:
 - Deve apresentar a listagem geral dos eventos disponíveis (Página Inicial);
 - Deve apresentar uma mensagem sugerindo o Login ou inscrição na plataforma, como forma de tirar partido de todas as funcionalidades.

Prioridade: Must

2.3 Requisitos Não Funcionais

Relativamente à forma como são apresentados os requisitos não funcionais, será feita a sua representação segundo uma adaptação dos cartões de *Volaire*, tal como apresentado na secção 2.2.

No caso concreto dos requisitos não funcionais, foi acrescentado o campo "Critério de encaixe" sempre que possível, de forma a indicar um processo que permita comprovar e testar a realização do requisito.

2.3.1 Aparência [10]

Requisitos voltados para o *look and feel* da plataforma, levantando limitações focadas na aparência e aspetto visual final do produto.

ID Requisito: 34**Tipos:** 10**Descrição:** O sistema deve apresentar uma interface com cores suaves.**Justificação:** Para que não seja visualmente cansativo e pesado, em especial em locais com pouca iluminação, o sistema deve apresentar cores suaves.**Critério de encaixe:** Reunindo um grupo de 20 utilizadores, os mesmos terão que realizar um conjunto pré definido de tarefas, intercalando entre as interfaces da aplicação por um período de 15 minutos. Se no final um conjunto superior a 5 indivíduos ficar com indisposições, dores de cabeça ou dores oculares, então o sistema apresenta cores demasiado fortes ou contrastantes.**Solução do Sistema****Prioridade:** Must

<u>ID Requisito:</u> 35	<u>Tipo:</u> 10
Descrição: O sistema, na barra superior das páginas, deve apresentar o logótipo da aplicação <i>Momenter</i> .	
Justificação: A aplicação deve ser publicitada de forma explícita no produto final, por questões de publicidade e direitos.	Solução do Sistema
	Prioridade: <i>Should</i>

2.3.2 Usabilidade [11]

Requisitos relacionados com a facilidade de utilização do produto, assim como considerações voltadas com a sua usabilidade pelos diferentes tipos de utilizadores finais.

<u>ID Requisito:</u> 41	<u>Tipo:</u> 11
Descrição: A criação de eventos deve ser simples de realizar para utilizadores que nunca tenham organizado eventos.	
Justificação: Todos os utilizadores devem conseguir criar e organizar um evento de forma simples e rápida, através do preenchimento de campos pré definidos e objetivos.	Solução do Sistema
	Prioridade: <i>Must</i>

2.3.3 Desempenho [12]

Requisitos com foco em questões como segurança, rapidez, extensão e precisão que uma funcionalidade deve apresentar para se mostrar viável e útil no contexto.

<u>ID Requisito:</u> 44	<u>Tipo:</u> 12
Descrição: O sistema fornece notificação aos promotores de eventos, à medida que os utilizadores interagem com um evento seu.	
Justificação: Integrado nas funcionalidades de gestão de um evento, o sistema disponibiliza a opção de receber notificações, de forma a conhecer estatísticas sobre os seus eventos.	Critério de Encaixe:
<ul style="list-style-type: none">• As notificações enviadas ao promotor, surgem quando um utilizador marca um evento como Favorito, reserva um bilhete ou compra um bilhete;• Para evitar um grande volume de notificações relativas ao mesmo evento, o sistema agrupa estatísticas do evento.	
	Por exemplo, se na última hora 10 utilizadores interagirem com o evento, não são geradas 10 notificações. É sempre apresentada 1 única notificação por evento, que agrupa as informações (estatísticas).
Solução do Sistema	Prioridade: <i>Could</i>

2.3.4 Operacionalidade [13]

Requisitos voltados para o ambiente de operação onde se insere o produto, assim como considerações sobre esse ambiente.

<u>ID Requisito:</u> 46	<u>Tipo:</u> 13
Descrição: A aplicação web deve funcionar nos principais <i>browsers</i> do mercado.	
Justificação: Todas as funcionalidades da aplicação devem estar disponíveis, independentemente da plataforma de acesso utilizada.	
Critério de encaixe: Todas as principais funcionalidades da aplicação (organização, visualização e pesquisa de eventos) devem ser executadas e testadas nas últimas versões dos principais <i>browsers</i> utilizados (<i>Firefox</i> , <i>Chrome</i> , <i>Edge</i> e <i>Safari</i>). Deve verificar-se que todos os processos foram terminados com sucesso e sem falhas.	
<u>Solução do Sistema</u>	<u>Prioridade:</u> Must

2.3.5 Manutenção e Suporte [14]

Requisitos relacionados com as alterações esperadas de se realizar ao sistema e as restrições associadas a esse processo.

<u>ID Requisito:</u> 48	<u>Tipo:</u> 14
Descrição: O código do <i>software</i> desenvolvido deve conter comentários. Em acréscimo, as classes desenvolvidas devem também estar organizadas nos respetivos módulos.	
Justificação: Qualquer pessoa com conhecimentos de programação e dos módulos do sistema, deve conseguir, através dos comentários, perceber o funcionamento do sistema.	
<u>Solução do Sistema</u>	<u>Prioridade:</u> Must

2.3.6 Segurança [15]

Requisitos relacionados com condições de segurança e confidencialidade do sistema.

<u>ID Requisito:</u> 52	<u>Tipo:</u> 15
Descrição: O sistema deve validar a conta de um utilizador através do email a si associado.	
Justificação: De forma a garantir que o utilizador insere um email seu, o sistema deve enviar um email de confirmação de conta para o email indicado pelo utilizador. Este email é meramente informativo, porém se o mesmo não for aberto no espaço de 30 dias a conta do utilizador deverá ficar suspensa na plataforma.	
<u>Solução do Sistema</u>	<u>Prioridade:</u> Could

Capítulo 3

Documentação UML

No presente capítulo será apresentado o conjunto de decisões de modelação UML que definem o limite do sistema envolvido pela plataforma *Momenter*.

A revisão deste capítulo é essencial para o futuro desenvolvimento do projeto e para a compreensão da relação entre as entidades e dados envolvidos neste contexto, assim como a futura modelação de Wireframes e mockups para conter os dados do sistema *Momenter*.

3.1 Modelo de Domínio

Como forma de delimitar todas as entidades e atividades envolvidas no contexto da plataforma, o modelo de domínio define a parametrização inicial do contexto da aplicação. Serve assim como forma de compreender aquele que virão a ser as principais entidades, quais os seus principais atributos e de que forma as mesmas se relacionam entre si.

O presente modelo é sujeito a eventuais alterações ao longo do desenvolvimento prático do projeto. O modelo de domínio é apresentado na Figura 3.1 e pode ser, sumariamente, descrito da seguinte forma:

- A plataforma debruça-se sobre três entidades principais: a noção de Utilizador, Organização e Evento;
- Um utilizador representa a entidade que interage com as funcionalidades da plataforma, nomeadamente a pesquisa, procura e colocação de eventos na plataforma. Sobre esta entidades são guardados os campos necessários e as suas preferências, para que lhe sejam recomendados eventos com base nas mesmas.
- Uma organização representa uma organização real ou um promotor que publicite regularmente um conjunto de evento.

A consideração desta entidade permite agrupar eventos associados a uma determinada entidade organizadora, possibilitando assim funcionalidades relacionadas com a gestão e promoção de eventos em grande número, na perspetiva de organizações que dependem dessas ferramentas.

- Por fim, um evento representa uma determinada atividade que é consumida nas diversas funcionalidades da plataforma. Em suma, os eventos são colocados por organizações ou utilizadores normais. Utilizadores normais podem ainda procurar, filtrar e guardar eventos de forma personalizável.
- Será sobre a promoção de eventos que se desenrolará o modelo de negócio da plataforma *Momenter*. Nesse sentido, o modelo de domínio abarca alguns conceitos chave relativamente à criação de eventos *premium*.

Nomeadamente, um evento *premium* é pago antes de ser colocado na plataforma. Os eventos *premium* têm acesso a funcionalidades de promoção especiais, nomeadamente através de destaque nas listagens e envio de notificações diretas a utilizadores registados que têm preferências em categorias onde o evento *premium* criado se insere.

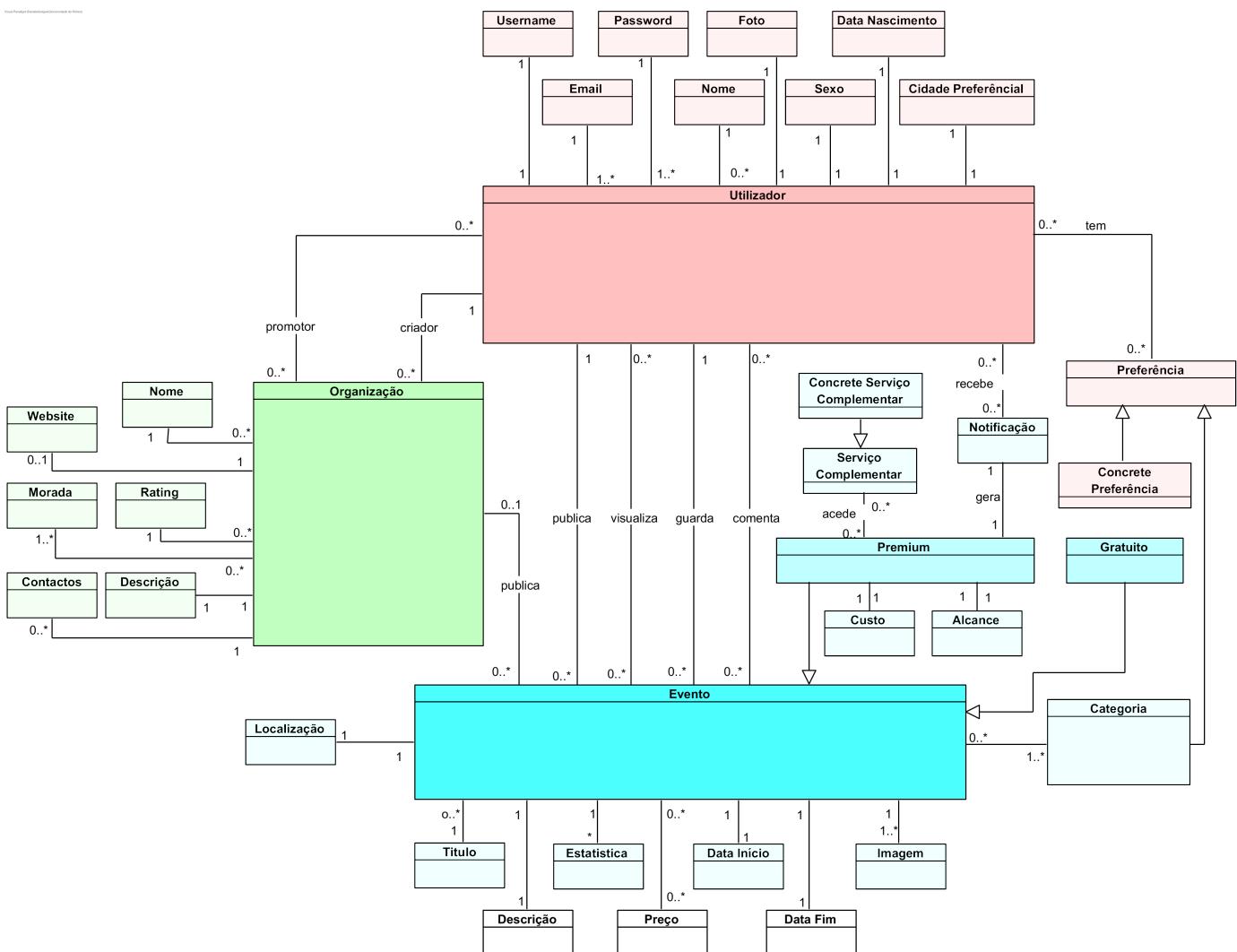


Figura 3.1: Modelo Domínio referente ao universo de entidades do contexto da aplicação.

3.2 Diagrama Use Cases

Na presente secção são apresentados um conjunto de Diagramas de Casos de Uso, que representam as principais interações dos utilizadores dentro do ambiente de funcionalidades da plataforma *Momenter*.

A Figura 3.3 apresenta uma visão geral dos cenários de acesso às principais funcionalidades. Destaca-se a divisão entre dois atores principais: O utilizador da plataforma, quer na perspetiva de participante de eventos como no papel de promotor de eventos e, o papel ativo do próprio sistema *Momenter* que providêncie e gere um conjunto de funcionalidade integrada na plataforma.

Sumariamente, os esquemas da Figura ?? descreve a versão pormenorizada das funcionalidades existentes em cada caso de uso para cada sub-grupo de tarefas. Nomeadamente: a primeira linha apresenta as interações de um utilizador no sentido de gerir a sua conta e as propriedades de pesquisa oferecidas na plataforma. Na segunda linha, são indicadas algumas das funcionalidades a nível da um promotor, quer a nível da gestão da sua organização quer a nível dos métodos disponíveis para a gestão e promoção de eventos dentro da plataforma.

De referir que por uma questão de metodologia e organização mental, as cores utilizadas nestes diagramas seguem a mesma estrutura dos anteriores esquemas, como forma de compreender a que entidades e partes interessadas estão associadas as tarefas/casos de uso levantados. (Cor vermelha relacionada com utilizadores, cor azul relacionada com eventos e cor verde relacionada com organizações)

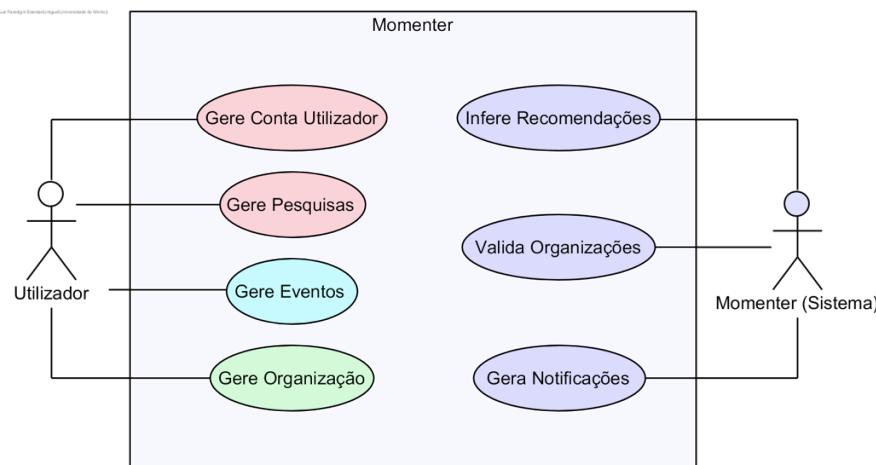


Figura 3.2: Diagrama geral de casos de Uso

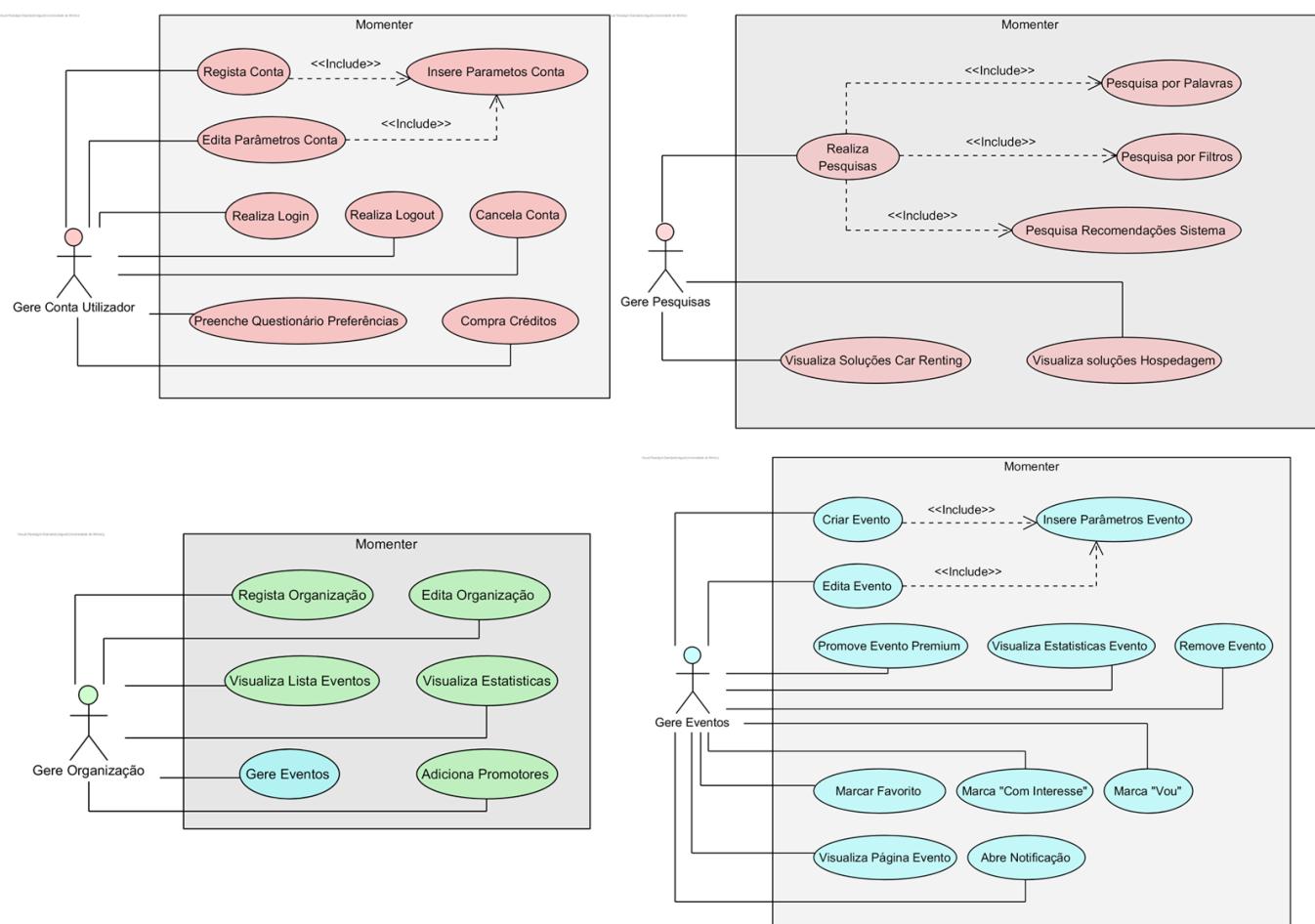


Figura 3.3: Sub-diagramas de casos de uso: Primeira linha: Casos de uso a relacionados com a conta de um utilizador e gestão de pesquisa; Segunda linha: Casos de uso a nível da gestão de uma organização e promoção de eventos na plataforma.

3.3 Diagrama de Classes

Considerando as entidades e interações descritas no decorrer das secções anteriores, na presente secção é introduzida a modelação do modelo de classes geral.

Todo o estado e informações são abstraídos sobre um conjunto de modelos, que acedem diretamente aos dados da base de dados, realizando as operações necessárias. A utilização destes modelos para alterar o estado da informação será feita através de um conjunto de métodos previamente definidos neste diagrama e levantados de acordo com as funcionalidades do sistema.

Desta forma, o modelo apresentado de classes apresentado na Figura 3.4 divide-se em dois principais componentes:

- Um conjunto de métodos, que permite o acesso aos dados da plataforma.

Estes métodos permitem, não só, aceder a funções simples de criação, recolha, atualização e remoção sobre os modelos de dados, como também a execução de tarefas mais complexas associadas à lógica e funcionalidades da plataforma.

Estes métodos estão organizados por blocos que podem ser vistos como os Módulos principais do sistema, seguindo a mesma apresentação visual e estrutura apresentada no modelo de domínio (Secção 3.1). Existe assim um nódulo responsável pela lógica do Utilizador, Organizações, Eventos e procura/recomendação de eventos.

- Um conjunto de modelos, que abstraem as entidades e entradas criadas na base de dados da plataforma.

Estes modelos podem ser vistos como classes que representam os principais objetos de dados do domínio da plataforma. É pretendido que estes modelos sejam criados de forma genérica, independente do módulo de acesso aos dados e ao motor de base de dados utilizado para suportar a plataforma.

- Os métodos descritos nos módulos fazem uso dos modelos gerados para realizar as suas funções sobre os dados da plataforma e manipular a informação entre as várias camadas do sistema. A estrutura/arquitetura do sistema será descrita de forma geral na Secção 3.4 seguinte.

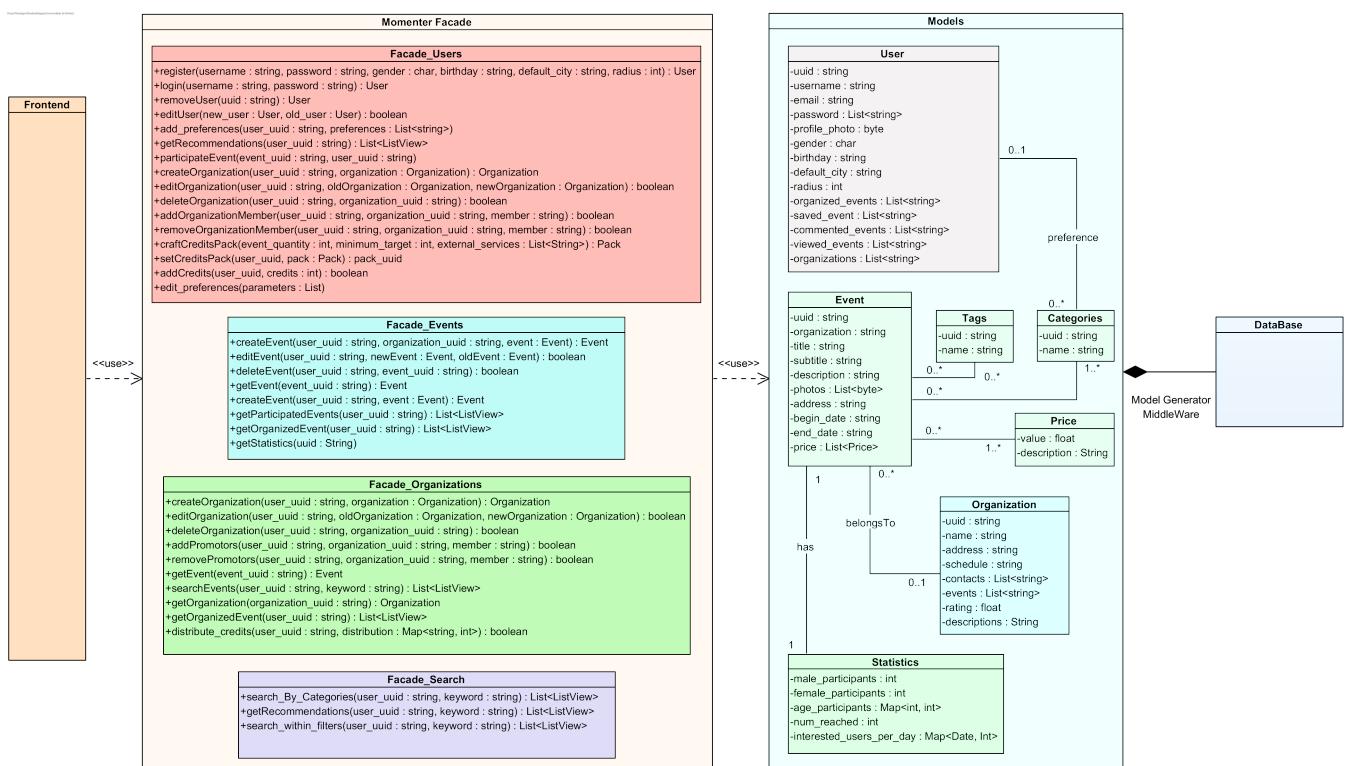


Figura 3.4: Diagrama de Classes do sistema.

3.4 Arquitetura do Sistema

Relativamente à arquitetura da plataforma *Momenter*, foram tomadas em considerações um conjunto de assunções que permitam a modularidade do sistema. Esta preocupação assenta na necessidade de reutilização dos módulos desenvolvidos a nível do *back end*, independente da forma com que se accedam às funcionalidades da aplicação.

A figura 3.5 apresenta uma descrição geral da arquitetura do sistema. Uma especificação mais detalhada desta arquitetura poderá ser encontrada no Capítulo 6 relativo à implementação prática desta estrutura.

Numa descrição global, a plataforma *Momenter* pode ser decomposta nos seguintes componentes:

- O conjunto de aplicações disponibilizadas de forma nativa para várias plataformas como forma de acesso aos recursos e funcionalidades do sistema *Momenter*. Nomeadamente, existirá uma aplicação para sistemas Android, IOS e navegadores web.

Devido ao espectro temporal deste projeto, será inicialmente apenas desenvolvida a aplicação web, que pode de qualquer forma ser accedida e utilizada por qualquer dispositivo, através de um navegador web. Esta decisão assenta também numa perspetiva de usabilidade do sistema, uma vez que determinadas tarefas de gestão e criação de eventos exigem a visualização de vários campos de informação. O acesso a estes campos de forma prática é dificultada em dispositivos de menores dimensões, podendo tornar a experiência de utilização menos produtiva.

Nesse sentido, ficará para um estágio de desenvolvimento futuro o estudo da interface para dispositivos moveis e a implementação da aplicação *Momenter* para estes ambientes.

- Um serviço de *gateway*, que abstrai a forma de acesso ao sistema *Momenter*. Nesse sentido, todos os pedidos enviados para o *back end* são uniformizados, independente da sua origem a nível de *front end*.

Este modulo é assim desenvolvimento tendo por base o futuro desenvolvimento da aplicação *Momenter* para outra plataformas. Funcionalidades de *login* e autenticações automáticas com base em *cookies* serão também implementadas nesta camada intermédia.

- A camada do *back end* engloba toda a lógica necessária para implementar e acceder às funcionalidades do sistema. Os métodos e modelos de dados descritos nas secções anteriores serão partilhados e utilizados nesta camada da aplicação.

Formalmente, existirá um módulo de *back end* principal, responsável pela implementação de todos os métodos necessários e resposta aos pedidos do serviço de *gateway*.

Existirá também, em paralelo, o módulo de recomendação. Para garantir a evolução do módulo de recomendação de forma independente dos restantes serviços do *back end*, este componente será implementado num serviço distinto. Este módulo terá acesso à base de dados do sistema e a uma base de dados própria, onde serão armazenados todos os padrões e informações relevantes extraídos pelo sistema de aprendizagem do mecanismo de recomendação.

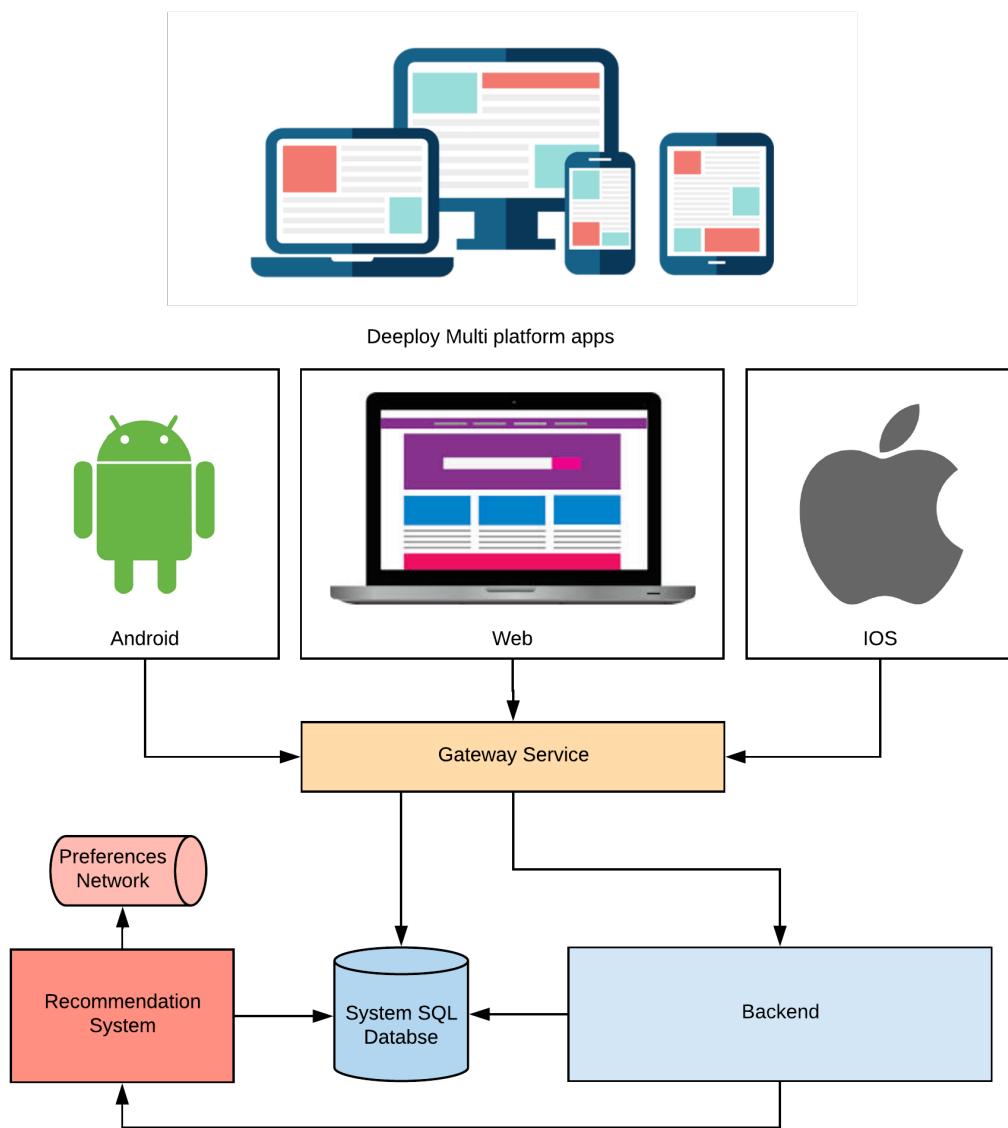


Figura 3.5: Arquitetura global da aplicação.

Capítulo 4

Design da Interface

4.1 Mockups

Num sentido de prototipagem inicial do ambiente da plataforma *Momenter*, foram inicialmente esboçados um conjunto de *mockups* das principais páginas da plataforma.

Este processo permite resolver questões relacionadas com o *flow* de atividades e possibilidades dentro do sistema e de que forma é possível navegar na plataforma. De uma forma geral:

- A plataforma abre na sua Página Início, com uma imagem de fundo características e a solicitação de registo ou *login* (Figura 4.1 - Esquema Esquerda);
- A página de *login* solicita os campos necessários para este processo, procurando ser breve e fluida para o utilizador (Figura 4.1 - Esquema Direita);
- O utilizador poderá fornecer um conjunto de preferências de forma específica. Estas preferências podem ser categorias, temáticas ou horários sobre os quais filtrar os seus eventos sugeridos. Independente destas informações, o sistema *Momenter* infere automaticamente as preferências do utilizador com a sua utilização (Figura 4.2 - Esquema Esquerda);
- O utilizador pode realizar pesquisa sobre o domínio de eventos da plataforma, utilizando filtros como palavras chave, categorias, preço, distância, etc (Figura 4.2 - Esquema Direita);
- O utilizador tem acesso a uma página com eventos recomendados. A recomendação surge baseada nas preferências explicitamente inseridas ou inferidas pela plataforma (Figura 4.3 - Esquema Esquerda);
- O utilizador pode abrir e visualizar com detalhes as informações presentes na página de um evento (Figura 4.3 - Esquema Direita);

Na secção seguinte são apresentados o conjunto de *WireFrames* que representa a modelação final destes conceitos prototipados em *mockups*.

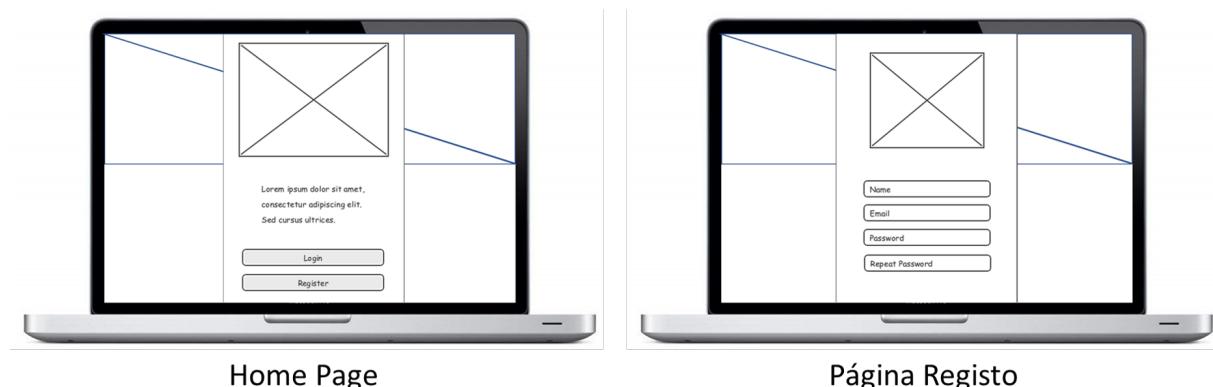


Figura 4.1: *Mockups* da interface para as páginas de Página Inicio e Registo na plataforma.



Figura 4.2: *Mockups* da interface para as páginas de inserção de preferências e pesquisa por filtros genéricos.

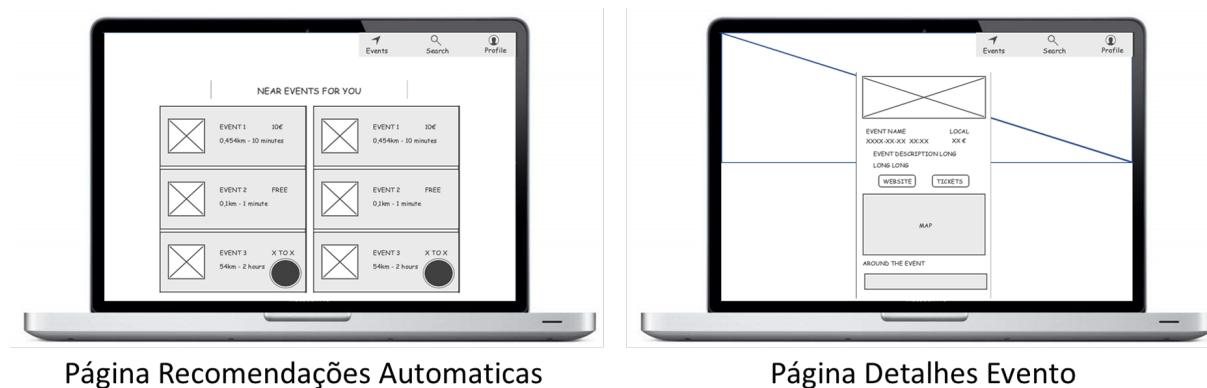


Figura 4.3: *Mockups* da interface para as páginas de visualização de eventos recomendados e da página com detalhes de evento.

4.2 Wireframes

Nesta secção são descritas as diferentes *views* existentes na plataforma **Momenter**. O conjunto destas *views* tem como finalidade a cobertura total dos requisitos elaborados no Documento de Requisitos.

Quando acedemos ao domínio da plataforma, www.momenter.xyz, é apresentada a Página Inicio observável na Figura 4.4. Esta página é apresentada desta forma a qualquer tipo de utilizador num acesso inicial à plataforma. Quando um dado utilizador pretender registar-se na plataforma, de forma a poder inserir eventos e receber notificações assim como recomendações de eventos é necessário realizar um registo.

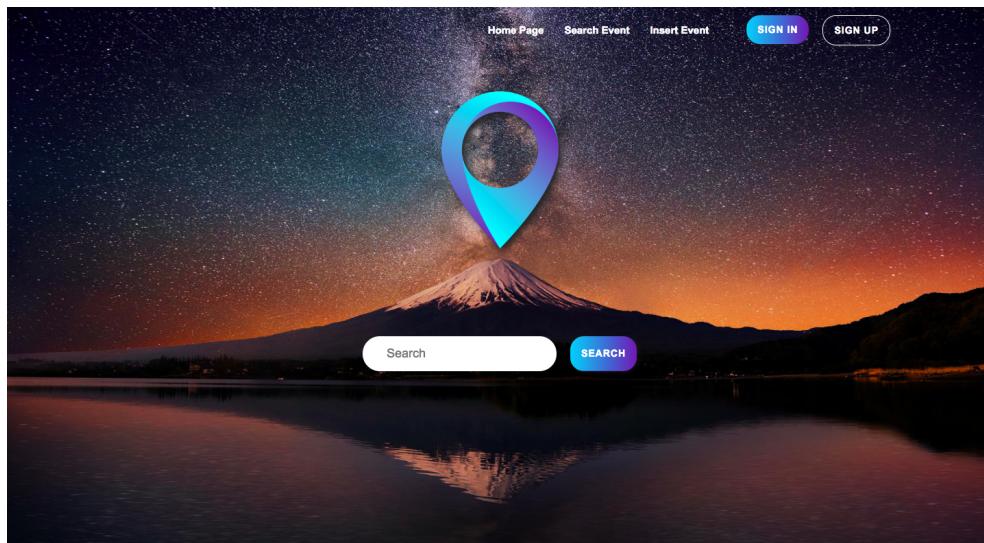


Figura 4.4: Página Inicio *Momenter*.

É possível registar-se na plataforma acedendo à opção "*Sign Up*" existente no *Header*, pedindo ao utilizador os campos necessários para realizar o registo, fig4.5, de uma forma de *popUp*.

A screenshot of a registration form (pop-up) titled "Sign Up". The form includes fields for Name, Username, Email, Password, City, Country, Gender, and Birthday. There is also a section for inserting an image, with a button to "Escolher ficheiro" (Select file). A message at the top says "Receive personalized events for you". The "SIGN UP" button is located at the bottom right of the form.

Figura 4.5: Formulário Registo *Momenter*.

Caso o utilizador que esta a aceder à plataforma já se encontre registado, pode realizar o *Login* através da opção disponível no *Header* apresentando, ao utilizador, um ambiente igual ao observável na fig4.6.

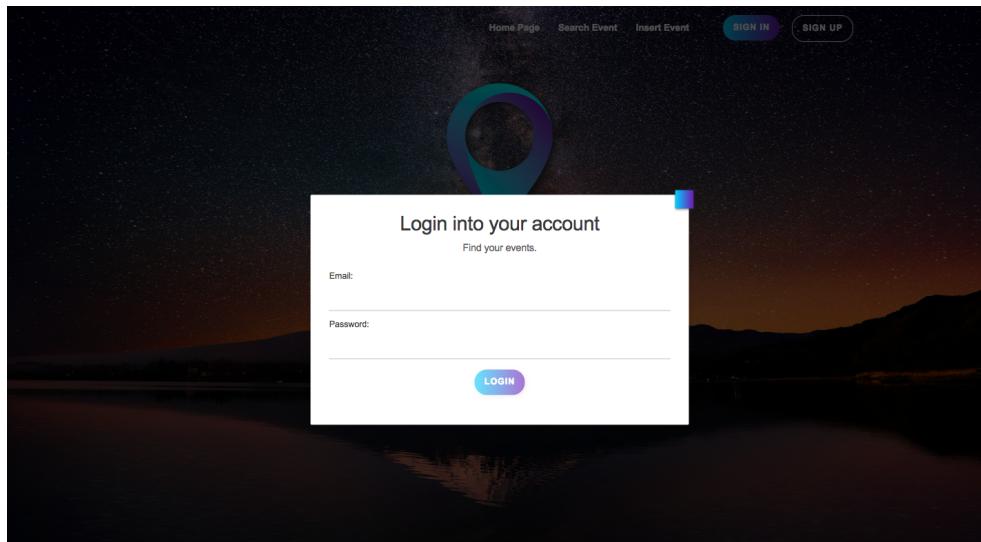


Figura 4.6: Página *LogIn* *Momenter*.

Como uma plataforma de apresentação de eventos existe uma página destinada apenas à pesquisa de eventos através de diferentes filtros.

Nesta página, figura 4.7, são listados os diferentes eventos existentes na base de dados de acordo com a pesquisa realizada pelo utilizador.

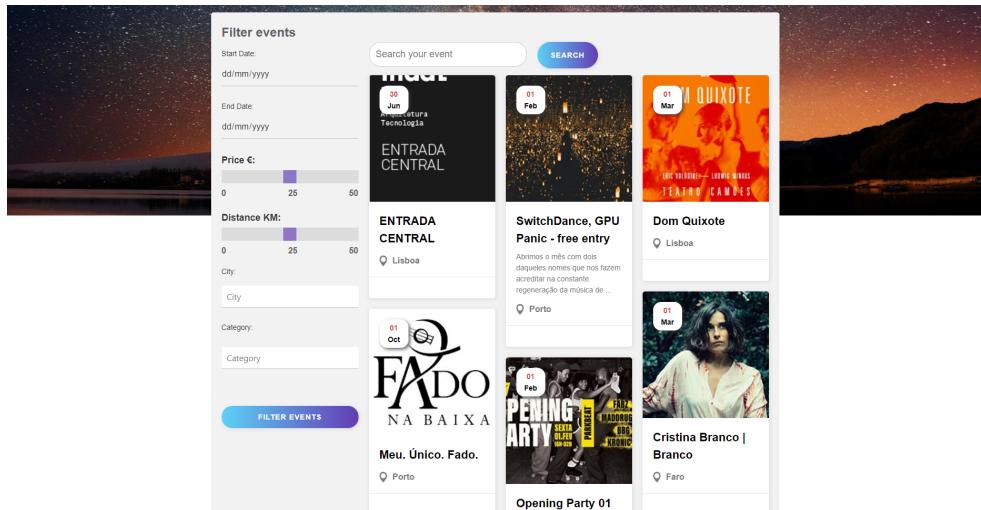


Figura 4.7: Página *Search Event* *Momenter*.

Em seguida, é possível aceder á página de um evento específico. Esta página, figura 4.8, apresenta ao destinatário final todos os campos necessários para a descrição do evento.



Figura 4.8: Página Event Momenter.

Após realizar o *login* é possível aceder à pagina do utilizador, Figura 4.9. Nesta página são apresentadas algumas informações interessantes, relativos ao utilizador em específico, como por exemplo os eventos marcados como Interesse, as Organizações que criou, a possibilidade de criar uma organização, entre outras opções e informações do próprio. Para além disto, na página em questão também é possível editar as informações inseridas anteriormente pelo utilizador.

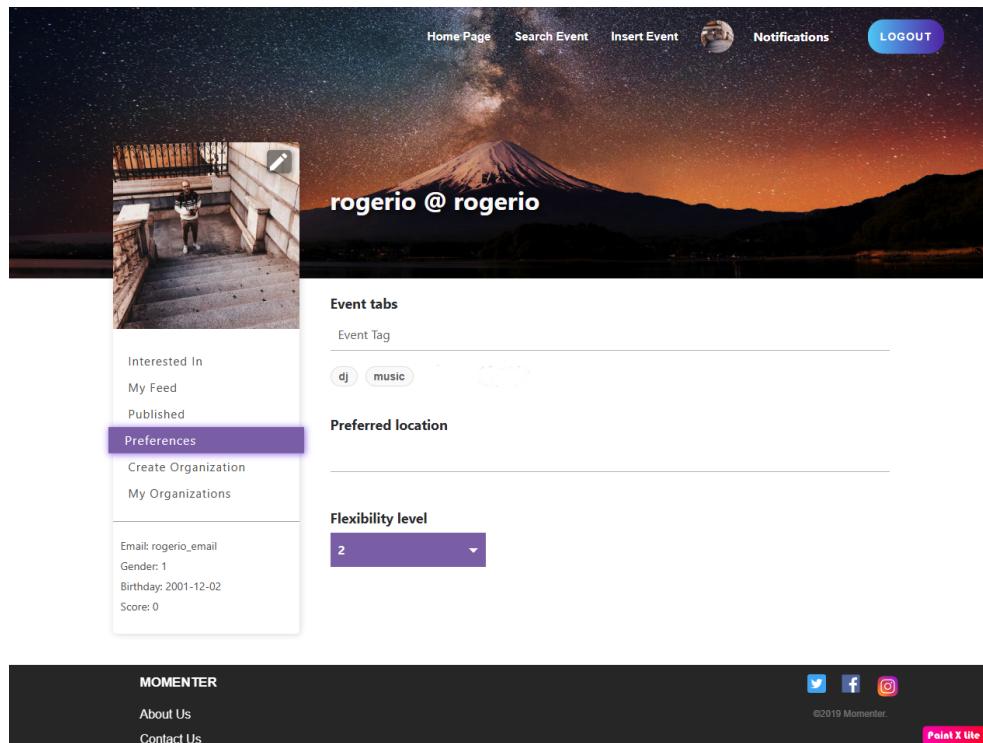


Figura 4.9: Página Perfil Utilizador Momenter.

Para além de todas as opções anteriormente disponíveis, também há a possibilidade de inserir manualmente um evento. Para inserir um evento é apresentada uma página, Figura 4.10, onde são pedidos todos os elementos necessários para o registo de um evento na plataforma *Momenter*.

The screenshot shows a 'Publish an Event' form. It has two main sections: a left sidebar with a background image of a lake at sunset and a right main area with a form. The form fields include:

- Title:** (input field)
- Sub-Title:** (input field)
- Address:** (input field) 542 W. 15th Street
- Start Date:** dd/mm/aaaa (input field)
- Start Hour:** (input field)
- End Date:** dd/mm/aaaa (input field)
- End Hour:** (input field)
- Price:** (input field) Price €
- Description Price:** (input field) Normal
- Price:** (input field) Price €
- Description Price:** (input field) Students
- Location:** (input field)
- Category:** (input field)
- Organization:** (dropdown menu) Select Organization: ▾
- Manager:** (dropdown menu) Select Manager: ▾
- Description:** (input field)
- Description:** (input field)
- Insert Image:** (button) Escolher ficheiro Nenhum ficheiro... selecionado
- More Options:** (checkboxes)
 - Car Renting
 - Hosting

A large blue 'CREATE EVENT' button is located at the bottom of the form.

Figura 4.10: Formulário criação eventos *Momenter*.

Os eventos inseridos na plataforma podem estar associados a utilizadores únicos ou a Organizações. Esta entidade é apresentada na plataforma através de uma página única, criada nos mesmos moldes base que a página do utilizador, como pode ser observado na Figura 4.11.

Nesta página é possível conhecer melhor uma organização. São listadas desde informações relativas a Eventos publicitados pela empresa que estão por acontecer, até uma secção *About Us* onde são descritos todos os utilizadores intervenientes na organização, horários, localização e descrição.

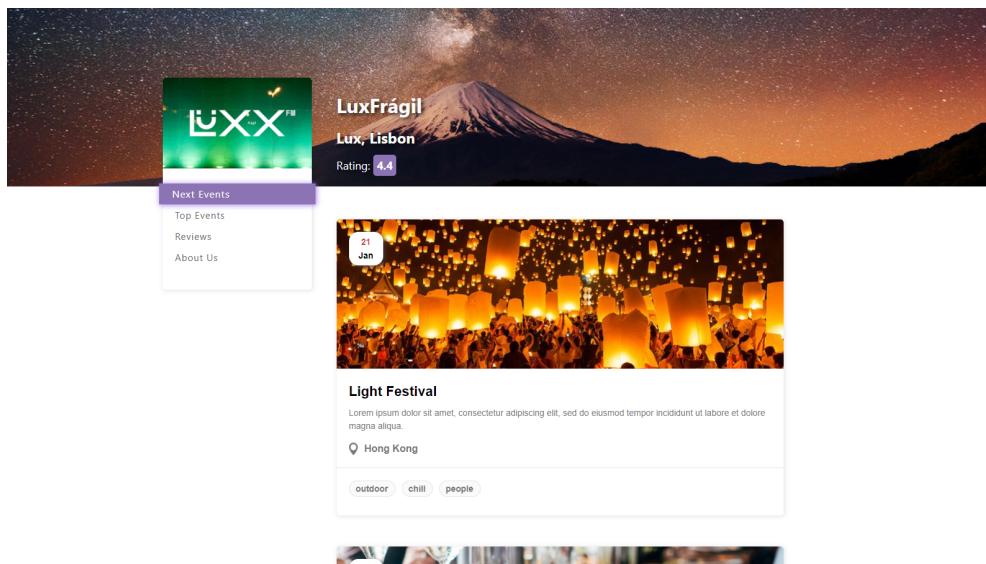


Figura 4.11: Página de Organização na plataforma *Momenter*.

Por fim, de modo a manter uma boa relação e proximidade com o nosso consumidor final, como é referido no documento de Modelo de Negócio, achou-se necessário a criação de uma página "Contactos" (Figura ??), assim como uma página de apresentação da plataforma ao consumidor final (Figura ??). Esta

página tem como principal finalidade proporcionar aos nossos utilizadores/clientes uma forma intuitiva e rápida de contactar com a equipa do projeto e apresentar o motivo da aplicação assim como a equipa de desenvolvimento.

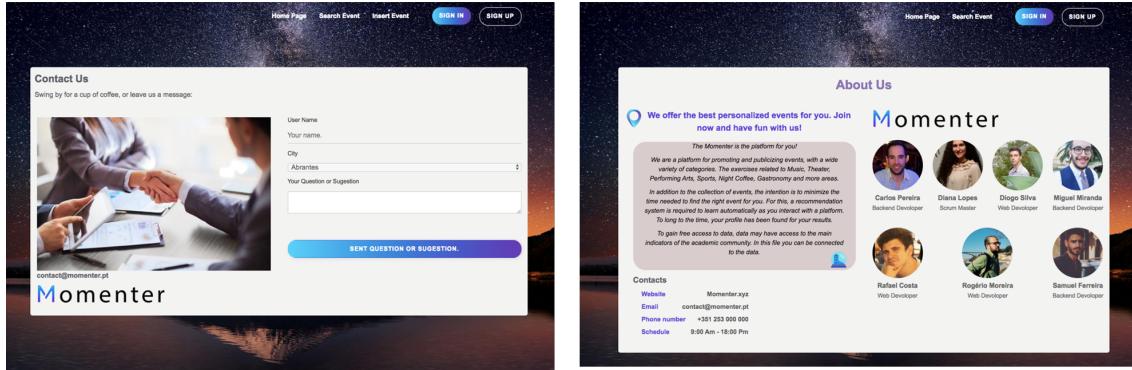


Figura 4.12: Página Contactos Momenter (Imagen esquerda) e Página com informações do Momenter (Imagen direita).

Capítulo 5

Modelação da Camada de Dados

5.1 Modelo Lógico

Tendo por base o contexto da plataforma levantado ao longo do modelo de domínio e diagramas de use cases (Secção 3, Figuras 3.1 e 3.3, respetivamente) é possível notar que entre as três principais entidades do sistema existe uma forte interligação a nível de funcionalidades, nomeadamente entre (Utilizadores - Eventos), (Eventos - Organizações) e (Organizações - Utilizadores).

Nesta perspetiva, encontrou-se todo o sentido em utilizar uma base de dados relacional, de maneira a permitir guardar as informações e relações associadas eventos, organizações, utilizadores e categorias.

Nas secções seguintes, serão descritas de forma sucinta as principais sub-regiões do modelo de dados. Para cada região é apresentada a principal entidade que a descreve e algumas das relações mapeadas pelas tabelas secundárias. A Figura 5.1 apresenta o aspetto global do modelo de dados da plataforma *Momenter*. Individualmente, as Figuras 5.2, 5.4, 5.3 e 5.5 ilustram as sub-regiões referentes a eventos, organizações, utilizadores e categorias, respetivamente. Num sentido de organização lógica, o esquema do modelo lógico mantém a mesma aparência visual (cor) e posição que os esquemas do modelo de Domínio e Classes previamente apresentado.

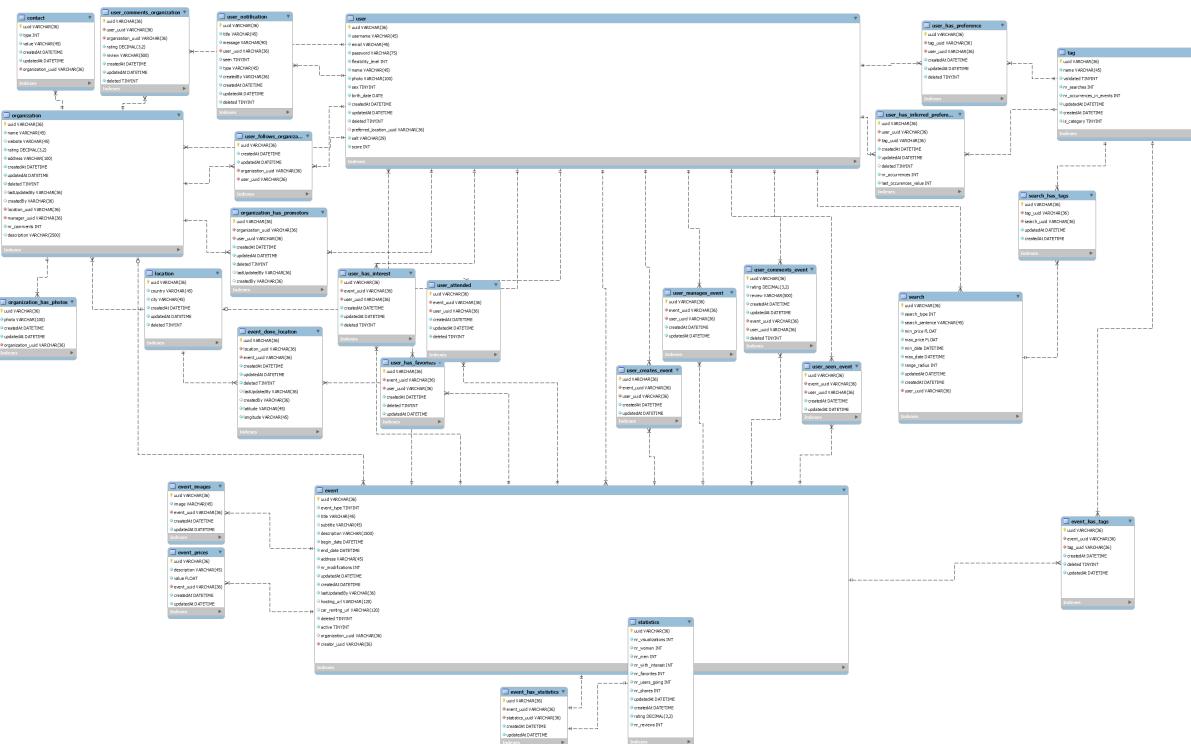


Figura 5.1: Modelo lógico da Base de dados.

5.1.1 Eventos

A entidade Evento agrupa todos os campos necessários de registar sobre um evento. Estas informações estão essencialmente associadas aos dados relevantes de listar ao utilizador que procura informações sobre eventos. Devido à lógica do sistema, alguns dados secundários são também armazenados, relativamente à organização ou criador do evento.

A nível de informações sobre eventos, é possível inferir as seguintes considerações:

- Um evento terá sempre um utilizador como criador do mesmo;
- Um evento pode pertencer a uma organização, caso seja colocado por um promotor dessa organização;
- Vários utilizadores podem gerir um evento, caso sejam promotores da organização associada ao evento;
- Um evento pode ter receber vários comentários de utilizadores;
- Um evento pode ser marcado como visto, favorito ou participado por vários utilizadores;
- Um evento tem a si associado várias categorias e *tags*;
- Um evento tem vários preços, imagens e localizações;
- Um evento tem um conjunto de estatísticas internas geridas pela plataforma.

5.1.2 Utilizadores

A entidade Utilizador define um utilizador da plataforma *Momenter*, independente do seu papel. Um utilizador por aceder tanto a funcionalidades de pesquisa e visualização de eventos como ao papel de promoção de eventos usando a plataforma *Momenter*.

A nível de informações dos utilizadores, é possível ter as seguintes relações:

- Um utilizadores podem estar associado a várias organizações, no papel de promotor das mesmas;
- Um utilizador pode comentar (avaliar) uma organização ou evento que encontre na plataforma;
- Um utilizador pode criar (promover) vários eventos próprios;
- Um utilizador pode marcar eventos como favoritos, com interesse ou participados;
- Um utilizador tem uma localização preferencial, mas que não é obrigatória;
- Um utilizador tem um conjunto de preferências associadas, que são utilizadas pelo processo de recomendação;
- Um utilizador tem várias pesquisas associadas, à medida que interage com essa funcionalidade.

5.1.3 Organizações

A entidade Organização agrupa todos os dados relacionados com uma organização ou entidade promotora de eventos. A definição desta entidade permite a oferta de funcionalidade de gestão e visualização de eventos agrupados por organização.

A nível de informações das organizações, é possível ter as seguintes relações:

- Uma organização tem um gestor principal, o seu criador;
- Uma organização pode ter vários promotores, que podem gerir os seus eventos;
- Uma organização pode ter vários eventos associados a si mesma;
- Uma organização apresenta uma localização;
- Uma organização pode receber vários comentários de utilizadores;
- Uma organização pode conter vários contactos e fotos que a descrevam.

5.1.4 Categorias e Tags

Categorias e *Tags* representam as informações que individualizam cada tipo de evento. O registo destes dados será relevante para o processo de recomendação, tendo como base categorias e *tags* associadas entre si.

A nível de informações das categorias/*tags*, é possível ter as seguintes relações:

- Uma pesquisa tem várias categorias e *tags* associadas;
- Uma categoria ou *tag* podem estar associadas a vários eventos;
- Uma categoria ou *tag* podem estar associadas a vários utilizadores, com base nas suas preferências;

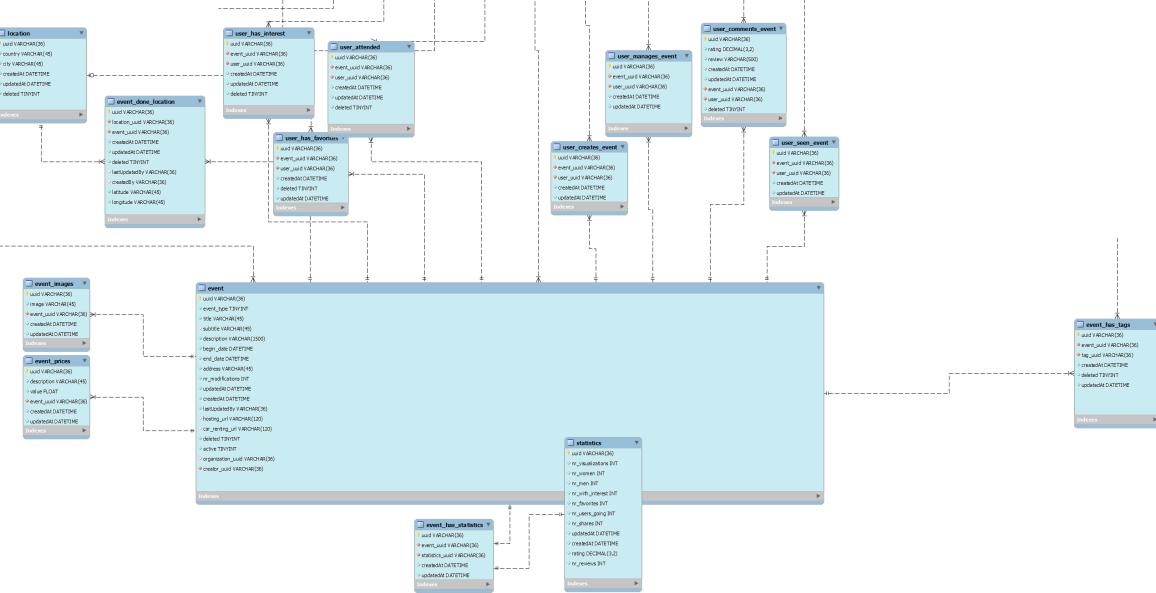


Figura 5.2: Modelo Lógico da Base de Dados - Sub-região Eventos.

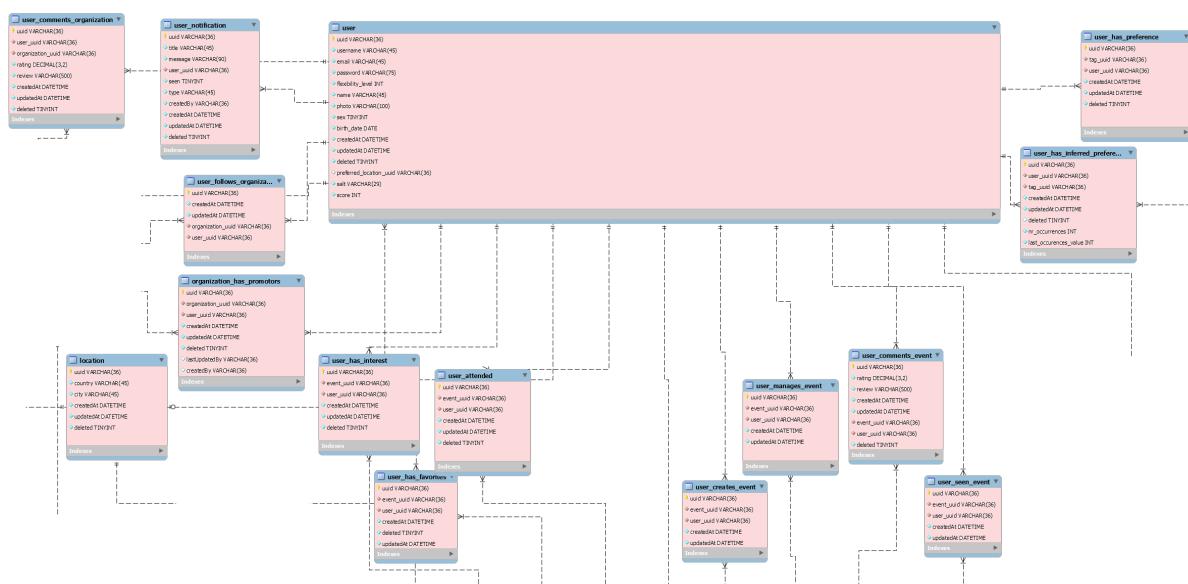


Figura 5.3: Modelo Lógico da Base de Dados da perspetiva dos utilizadores

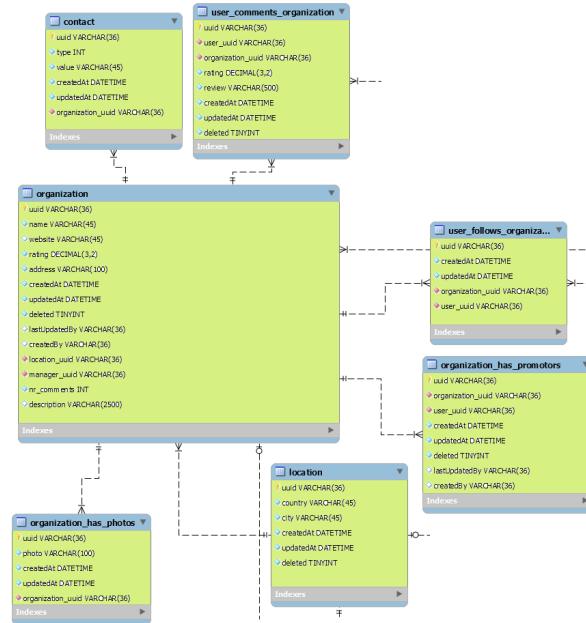


Figura 5.4: Modelo Lógico da Base de Dados - Sub-região Organizações.

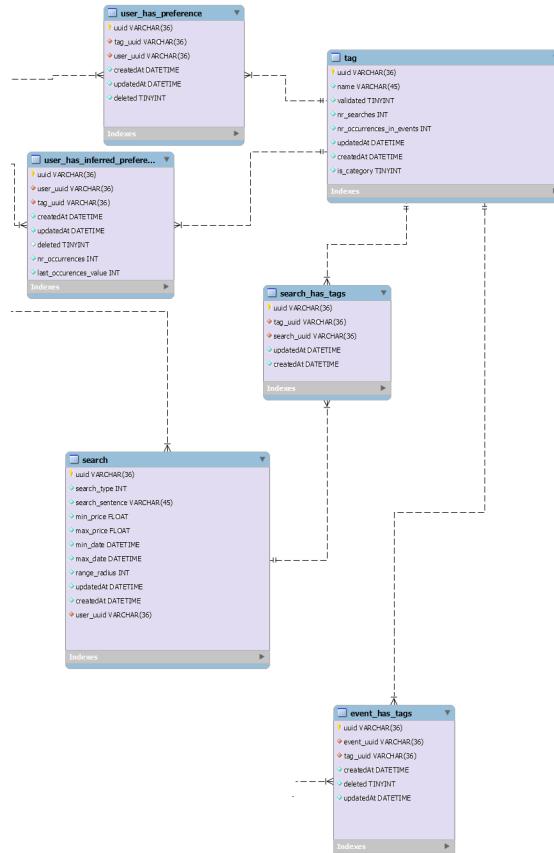


Figura 5.5: Modelo Lógico da Base de Dados - Sub-região Categorias/ Tags

5.2 Povoamento automatizado - Fontes de dados externas

Considerando o propósito da plataforma *Momenter*, é imprescindível a existência de uma oferta inicial de eventos no sentido de cativar utilizadores a interagir e utilizar a plataforma.

A existência de eventos permite cativar a comunidade inicial da plataforma o que, por sua vez, atrai promotores e organizações a promover o seus próprio eventos na plataforma, devido à comunidade regular conquistada, que irá também ver os seus eventos.

Além da perspetiva de negócio, uma forma automatizada de povoar regularmente a plataforma com novos eventos aumenta a centralidade e valor da mesma. A existência de uma vasta diversidade de eventos permite encontrar um conjunto de recomendações certo para um maior número de utilizadores com preferências distintas e peculiares. Não seria possível oferecer uma oferta personalizada de eventos se o número e diversidade de eventos na plataforma fosse, por si só, reduzido.

Os dados de eventos recolhidos são pré-processados para que sejam previamente organizados por organizações, antes de serem inseridos na plataforma. Atualmente, existem desenvolvidos dois *web crawlers*, criados para os sites Bilheteira Online e do Viral Agenda. O primeiro foi escolhido por se ter verificado que um elevado número de eventos que dependem da venda de bilhetes são colocados nesta plataforma, não sendo por exemplo encontrados noutros locais, como em redes sociais. A plataforma Viral Agenda foi escolhida como fonte de dados devido às suas parcerias com organizações municipais (Teatros, Museus, Salas Espetáculos) a nível nacional. Desta forma, é possível recolher um conjunto de eventos muito específicos associados à cultura e atividade local de cada município.

Denote-se também que algumas organizações com quem se estabeleceu contacto para parcerias indicaram o seu interesse neste tipo de serviço. Onde através de um *crawler* dos seus sites principais a plataforma *Momenter* teria acesso direto aos seus eventos. Os *web crawlers* retiraram as informações de interesse analisando o código fonte das páginas dos eventos e organizações nestes sites. Apesar do desenvolvimento inicial destes *crawlers* exigir algum conhecimento de HTML, depois de concluídos a sua utilização baseia-se na chamada de um *script* com execução automática, sem necessidade de intervenção manual. O esquema da Figura 5.6 procura descrever este procedimento automatizado.

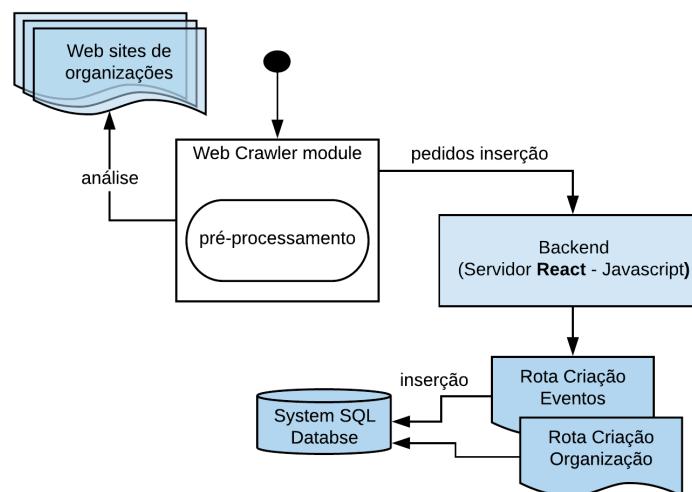


Figura 5.6: Esquema do *flow* de atividades executado pelo módulo de web *crawling* desenvolvido.

Capítulo 6

Desenvolvimento e Implementação

6.1 Arquitetura Aplicação

Considerando a fase de desenvolvimento implementada do decorrer do último trimestre, a arquitetura da Figura 6.2 descreve aquela que foi a estrutura inicial dos módulos a desenvolver para cada componente da aplicação. Sumariamente, a implementação desta arquitetura segue três frentes:

- Desenvolvimento do módulo de *front end* (Secção 6.2.1) cujo repositório pode ser encontrado em <https://github.com/rgllm/frontend>;
- Desenvolvimento do módulo de *back end* (Secção 6.2.2) cujo repositório pode ser encontrado em <https://github.com/munybt/momenter-backend>;
- Desenvolvimento do módulo do Sistema de Recomendação (Secção 6.2.3) cujo repositório pode ser encontrado em https://github.com/MikeDM16/Momenter_Recommendation_System.

Por questões de propriedade e desenvolvimento futuro, os repositórios fornecido serão novamente colocados em privado após dia 20 de Fevereiro de 2019. A plataforma Momenter, no seu estado inicial viável e funcional poderá ser encontrada no domínio www.momenter.xyz

A descrição do desenvolvimento de cada módulo será abordada com maior detalhe nas referidas secções. O desenvolvimento destas camadas da aplicação segue uma abordagem *Scrum* procurando incrementalmente estabelecer metas e pontos de situação entre a comunicação e integração dos vários módulos. Como observado ao longo da estrutura deste documento, o desenvolvimento da plataforma *Momenter* segue uma abordagem *Bottom-up* onde se começou por definir as entidades e objetos de dados. Nesse sentido, tanto a frente de desenvolvimento do *front end* como a camada de lógica do *back end* iniciam o seu desenvolvimento com uma noção inicial bem definida de quais os campos e objetos de informação necessários de manipular e integrar no contexto da aplicação.

6.1.1 Organograma

No sentido de otimizar o desenvolvimento da plataforma e tirar uma maior partido das competências individuais de cada individuo da equipa de desenvolvimento, o organograma apresentado na Figura 6.1 descreve a divisão dos elementos de trabalho pelos diferentes módulos do sistema. Denote-se que, propositadamente, alguns dos elementos da equipa realizam trabalho em várias frentes de desenvolvimento de forma a conseguir cumprir com os prazos estipulados.

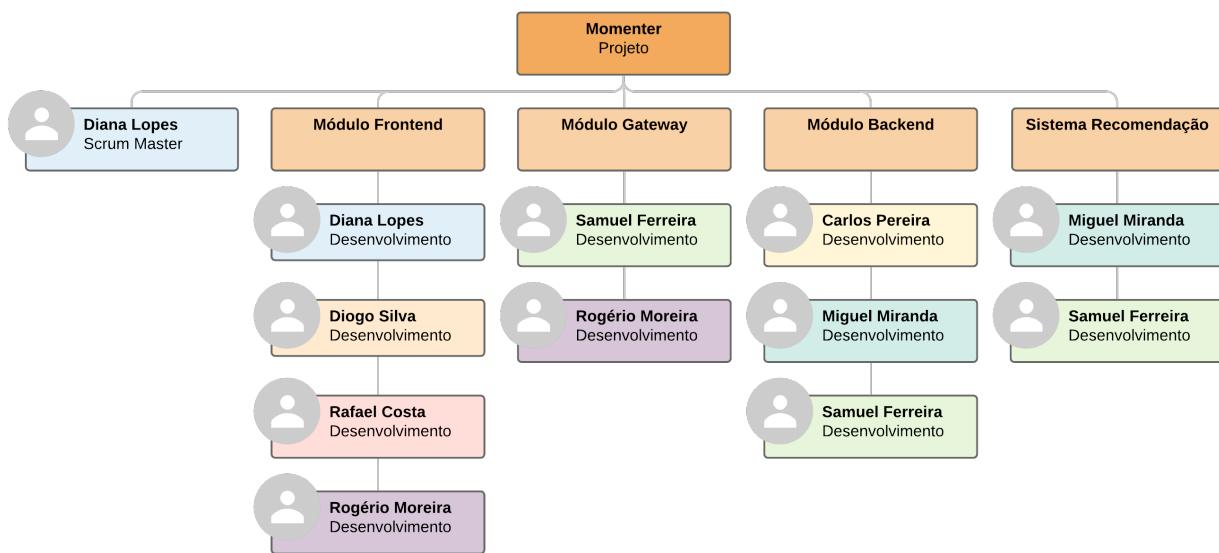


Figura 6.1: Organograma da equipa de desenvolvimento

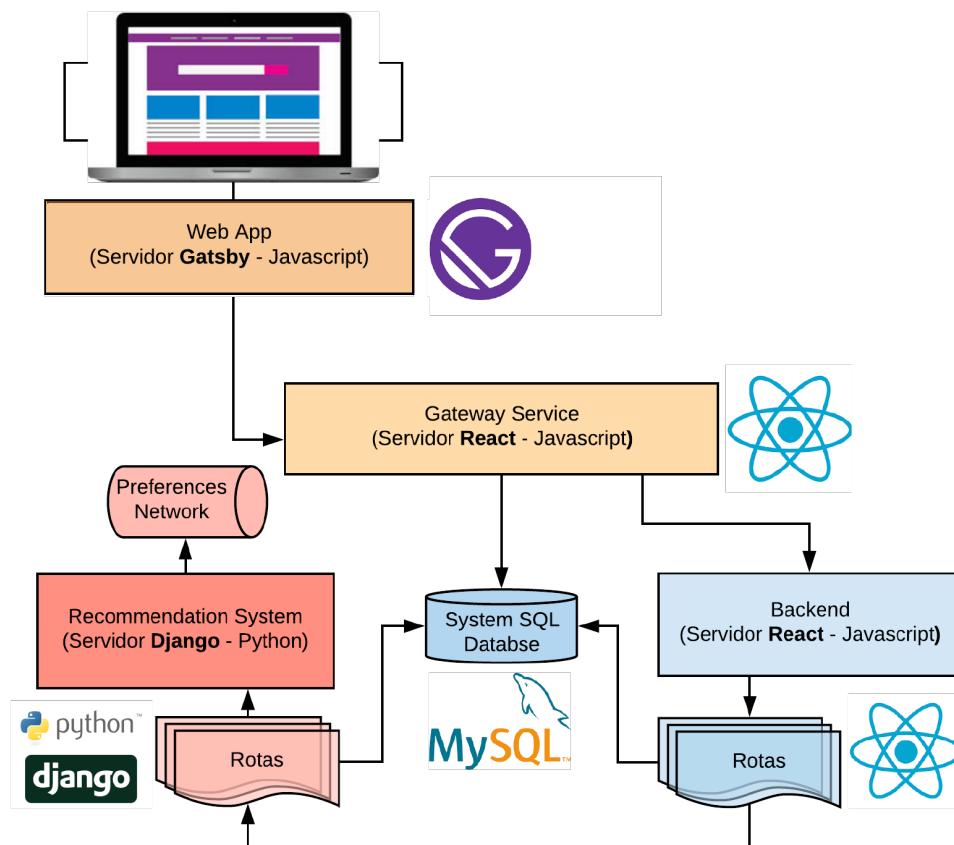


Figura 6.2: Arquitetura do sistema da aplicação *mobile*

6.2 Módulos Desenvolvidos

6.2.1 Front end

Foi necessário efetuar um estudo no sentido de perceber quais as tecnologias que melhor se enquadravam no desenvolvimento web do *front end* da aplicação. Entre os fatores tidos em conta estão o tempo de aprendizagem, já que a maior parte dos membros da equipa nunca tinham trabalhado com tecnologias web, o tempo de prototipagem, já que o projeto tinha prazos apertados, a maturação da tecnologia para não existirem limitações numa fase posterior e a sua adequação à arquitetura geral da aplicação e às tecnologias usadas em *back end*. O panorama atual das ferramentas e tecnologias para desenvolvimento é bastante extenso e diverso, deve, por isso, ser analisado com cuidado e sem preconceitos.

Há três partes distintas que usualmente fazem parte de uma aplicação web e onde têm que ser definidas tecnologias diferentes a usar.

- Camada de Lógica - é a responsável por todos os dados dinâmicos da aplicação tendo sido a camada onde as tecnologias foram analisadas em primeiro lugar já que as escolhas aqui feitas podem influenciar as escolhas nas restantes.

Partindo dos critérios e em grande parte devido ao desenvolvimento recentes das tecnologias escolheu-se usar uma *framework* de Javascript entre: React, VueJS e Angular. Depois de analisado o custo de aprendizagem de cada uma delas e graças em grande medida à divisão mais direta por componentes do React, escolheu-se esta tecnologia como ponto de partida.

Contudo, usar React sem nenhuma ferramenta de gestão incorporada seria muito dispendioso a nível de tempo já que tudo teria que ser feito sem nenhum ponto de partida. Por esta razão, incorporou-se no React o GatsbyJS que permite fazer a gestão toda a arquitetura mais lógica da aplicação e a gestão de rotas.

- Camada de Estilo - Esta é a camada responsável por tudo o estilo de uma página. Habitualmente a linguagem a utilizar é o CSS, contudo é bastante limitado em diversos pontos que seriam vitais na nossa aplicação, como por exemplo a separação de estilos para dispositivos móveis e/ou para ecrãs maiores ou a gestão das cores da aplicação. Por estas razões optamos pela utilização do SCSS.
- Camada de Conteúdo - Devido à escolha na camada de lógica, a tecnologia a utilizar nesta camada foi condicionada já que o React utiliza uma separação da aplicação em componentes, usando para isso uma sintaxe própria de HTML (a tecnologia normalmente usada). Por essa razão, a tecnologia usada foi o JSX.

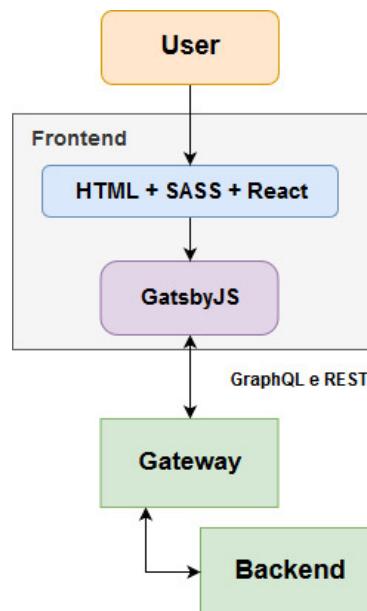


Figura 6.3: Esquema da organização do Módulo Frontend.

Relativamente ao componentes integrados:

- React - O React é uma framework de JavaScript usada para criar interfaces web. Criada pelo Facebook em 2011, o React faz o uso de componentes para criar módulos reutilizáveis entre páginas e que para além de poderem comunicar entre si têm também um estado próprio. Foi usada no contexto do frontend para criar toda a interface.
- GatsbyJS - O GatsbyJS é uma ferramenta usada no contexto do projeto como uma ferramenta de gestão para React. No frontend da aplicação foi usado para gerir todas as rotas da aplicação, as queries de GraphQL e todo o processo de compilação da aplicação.
- GraphQL - O GraphQL, também criado pelo Facebook à semelhança do React é uma linguagem de query a APIs que permite, ao contrário, do REST fazer pedidos exatamente da informação pretendida. Foi usada no contexto do frontend para queries estáticas.
- SCSS - O SCSS é um pré-compilador de CSS e permite adicionar a esta linguagem de estilos variáveis dinâmicas, endereços dinâmicos e aninhamento de classes. Todo o estilo da aplicação web é escrito em SCSS.
- Bootstrap - O Bootstrap é uma biblioteca de HTML, CSS e Javascript que permite o uso de elementos básicos para uma página web. Fornece na aplicação web todos os elementos HTML, como por exemplo botões, caixas de escrita e grelhas.
- TypeScript - O TypeScript é um superset de JavaScript criado pela Microsoft que permite a escrita de código com a utilização de tipagem estática, orientação a objetos e facilita a escrita de código com uma sintaxe de fácil compreensão

Para além disto, foram ainda usadas algumas tecnologias e ferramentas para a correta gestão e verificação da aplicação web.

- TSLint - O TSLint é uma ferramenta de verificação de código TypeScript, que permite verificar se determinado código obedece a regras pré-estabelecidas. No nosso caso estas regras incluem por exemplo, a proibição do uso de expressões lambda, indentação com espaços de comprimento 1 e a proibição de uso de expressões bitwise.
- Prettier - O Prettier é uma ferramenta de verificação de código CSS, que permite manter o código estável, obedecendo a regras pré-estabelecidas e definidas pela equipa.
- PostCSS - O PostCSS é uma ferramenta para automatizar operações de CSS. No projeto foi usado para adicionar prefixos de estilo específicos para cada browser.
- Webpack - O Webpack é uma ferramenta de compilação de HTML, CSS e Javascript. No frontend é responsável pela compilação de todo o projeto nos ficheiros que serão interpretados pelos browsers.
- Netlify - O Netlify é uma ferramenta de deploy e hospedagem de aplicações web. Na configuração para este processo temos por exemplo o deploy automático da aplicação a partir de um repositório de Github e o deploy automático de Pull Requests para verificação.

6.2.2 Back-end

Como ponto de partida para o desenvolvimento do *back end* é necessário um estudo das atuais tecnologias de desenvolvimento web, no sentido de compreender quais as vantagens de cada uma e de que forma é possível criar um conjunto de serviços versáteis e que se adaptem ao contexto de utilização do sistema *Momenter*.

Neste sentido, optou-se por implementar a camada lógica do *back end* utilizando *Node.js*. Esta foi a ferramenta escolhida por ser um interpretador de *Javascript*, agilizando, desta forma, a comunicação entre o *front end* e o *back end*. Além desta ponte comum entre as camadas da aplicação, *Node.js* foi escolhido por permitir a construção de aplicações *web* fluidas e otimizadas, preparadas para dar resposta assíncrona a um elevando número de pedidos de clientes simultâneos. Este ponto torna-se essencial devido ao contexto de utilização onde a plataforma *Momenter* se insere.

Uma vez que é necessário persistir os dados sobre os eventos, organizações e utilizadores da plataforma, foi também necessário desenhar e implementar uma base de dados. Optou-se por recorrer a uma base de dados relacional *MySQL* tendo sido implementado o modelo lógico definido no Capítulo 5.

A decisão de começar o *back end* pela implementação da Base de Dados permitiu a integração de ferramentas que auxiliam a manipulação e acesso aos dados. Nomeadamente, foram ainda integradas duas *frameworks* neste módulo de desenvolvimento: *Express* e *Sequelize*.

- *Express* é uma *framework* para *Node.js* que permite construir aplicações *web* e desenvolver *APIs*. O *Express* baseia-se na implementação de serviços *middleware* que respondem a pedidos HTTP (*GET*, *PUT*, *POST*, *DELETE*, etc.) e, a partir de um conjunto de rotas pré definidas, permite definir diferentes ações para o tratamento destes pedidos.

As rotas são organizadas conforme a lógica que manipulam. Cada rota pode ser interpretada como uma *Façade* que implementa os métodos que podem ser vistos no diagrama de classes apresentado na Secção 3.3. Por exemplo, existe uma rota para tratar todos os pedidos relacionados com dados do utilizador. O mesmo para organizações, eventos, pesquisas ou categorias.

Esta abordagem permitiu uma melhor organização da camada lógica do *back end*, onde esta modularidade facilitou o acréscimo de componentes do lado do servidor.

- Pelo facto de se a implementar o *back end* em *Node.js*, decidiu-se facilitar a comunicação entre este módulo e a camada de dados recorrendo a um *Object-relational mapping (ORM)*, nomeadamente o *Sequelize*.

O *Sequelize* é um *ORM* baseado em *Promises* que cria automaticamente os *data access object (DAO)* necessários para operar na camada de dados. Estes *DAO* são designados por *models* tendo sido já apresentados sumariamente no Diagrama de classes.

Esta ferramenta implementa também métodos próprios para efetuar operações sobre os dados, sem ser necessário definir explicitamente as *queries* pretendidas. Ou seja, o *Sequelize* constrói *statements* a partir dos métodos evocados sobre os *models* (*DAOs*), tirando do programador muita da complexidade no que diz respeito a operações como intersecções de várias tabelas.

Esta abstração permite também tornar a manipulação dos dados independente do motor de base de dados utilizado. Exige, contudo, uma definição inicial e cuidada das relações entre as entidades da base de dados, de forma a permitir tirar o maior partido das funcionalidades desta *framework*.

6.2.3 Sistema de recomendação

Como referido na descrição da arquitetura da aplicação (Secção 3.4) foi definido que o módulo que implementa o sistema de inferência de preferências e recomendação de eventos deve ser construído de forma independente dos restantes serviços de *back end*.

Este pressuposto assenta na ideia que o sistema de recomendação exigirá um tipo de desenvolvimento e análise mais experimental ao passo que, o desenvolvimento da restante lógica do *back end* é mais metodóloga e facilmente definida. Além disso, é esperado que a componente de recomendação utilize estruturas de dados (para registar as interações e preferências dos utilizadores) que não são facilmente definidas e manipuladas pelas tecnologias de desenvolvimento utilizadas na arquitetura do *back end*.

Neste sentido:

- A inferência das preferências e a geração de recomendações de eventos é alcançada utilizando uma rede que relaciona as principais categorias e *tags* de eventos entre outros eventos e utilizadores.

Sempre que um utilizador realiza uma ação na plataforma (abre uma página eventos, coloca evento como favorito, partilha evento, comenta eventos ou organização, etc.) foram definidas um conjunto de funções que atualizam o peso das ligações que ligam o nodo desse utilizador às categorias/*tags* do evento com que interagiu.

As recomendações são geradas procurando pelos principais componentes ligados da rede, quer a nível de um padrão global como partindo do nodo específico de um utilizador.

- Por implementar o sistema de recomendação com base numa rede (grafo ligado) foi tomada a decisão de implementar este sistema em *Python*, devido ao conjunto de bibliotecas fortemente otimizadas que existe nesta linguagem.
- Uma vez que o sistema de recomendação precisa também de aceder à base de dados do sistema *Momenter* para inferir novas informações relativas a eventos ou utilizadores, todo o projeto do sistema de recomendação é integrado num ambiente *Django*.

Tal como no *back end* se tiram partido das funcionalidades do *Sequelize* e *Express*, no sistema de recomendação utiliza-se a *framework Django*. *Django* permite implementar um serviço de respostas também baseado em rotas para pedidos HTTP específicos e funciona ainda como um *ORM* sobre os dados que encontra na Base de dados, abstraindo assim as entidades e operações sobre as tabelas.

- Destaca-se ainda a integração com a ferramenta de visualização de dados científicos *Gephi*. Esta ferramenta externa permite estudar e analisar visualmente grafos, fornecendo integração com vários métodos de visualização, deteção de colónias/ regiões de interesse de um grafo, assim como a sua edição e análise.

Isto permitirá avaliar de que forma as tendências inferidas se relacionam e remover eventuais ligações que se criam entre eventos e tags ou categorias. Estas relações indesejadas podem surgir fruto de eventos que foram regularmente colocados na plataforma de forma indevida, tendo sido relacionados com palavras ou categorias impróprias ou onde não se inserem.

- Como tuning e configuração inicial das ligações entre os nodos, foi realizado um questionário à comunidade académico no sentido de analisar e inferir as preferências gerais da comunidade e de que forma diferentes pessoas relacionam diferentes temáticas entre si.

Com o conjunto de questões e escalas foi possível analisar questões como, a titulo de exemplo, "Alguém que gostas Bastante de Teatro, gosta Muito de Música e Moderadamente de Conferências. Além disso, com esta distribuição de gostos, recomenda ainda as temáticas de eventos X, Y e Z".

Este conjunto de dados inicial permitiu relacionar assim os nodos da rede e providências uma fonte de recomendações base baseada naquilo que pode ser visto como as preferências da comunidade académica.

Capítulo 7

Trabalho Futuro

Como trabalho futuro refere-se a implementação dos restantes requisitos, inicialmente marcados com a prioridade *Could* e *Wont*. Depois de otimizar e implementar todas as funcionalidades principais na plataforma Momenter, existirá posteriormente margem para aprimorar características e funcionalidades específicas levantadas por um número menor de potências partes interessadas. Contudo, essas características aumentarão o valor da plataforma Momenter e farão com que a mesma atinja um maior número de utilizadores.

Levanta-se ainda a necessidade de atacar várias frente do mercado, implementando uma aplicação nativa da plataforma Momenter para sistemas Android e IOS. Esta necessidade baseia-se no crescimento inicial da plataforma, onde é necessário garantir a presença em vários ambiente de interação. Caso contrário, outras companhias concorrentes podem construir soluções semelhantes e lança-las primeiro para ambientes nativos IOS e/ou Android.

Contudo, o desenvolvimento e aprimoramento da aplicação web mantém-se o foco central do projeto Momenter. O acesso ao Momenter por um browser pode ser acedido por qualquer plataforma móvel e permite atualmente a melhor execução de algumas tarefas de gestão e manipulação de eventos.

Em alguns cenários de utilização, a área de visualização disponível afeta significativamente a experiência de utilização. Nestas situações, antes de partir para o desenvolvimento de aplicações mobile é necessário um estudo fundamentado de várias interfaces fluídas, no sentido de procurar um equilíbrio entre a experiência do utilizador (participante ou promotor) e a tarefa que o mesmo procura desempenhar.

De uma forma global, levanta-se ainda a necessidade de melhorar toda a arquitetura geral do sistema, procurando integrar novas ferramentas ou soluções que tragam vantagens para o contexto da plataforma Momenter. Em paralelo, o sistema de Recomendação irá também sofrer melhorias graduais, de forma a evitar que o mesmo siga tendências globais em vez das tendências individuais de cada utilizador.

Capítulo 8

Conclusões

Ao longo do desenvolvimento de todo o projeto houve a necessidade de estar em permanente contacto com o público alvo e possíveis clientes de modo a que o produto criado se torne atrativo e útil para os stakeholders. Isto levou a que, ao longo de todo o desenvolvimento, houvesse sempre alguns refinamentos em termos de prioridades de apresentação de dados e como eram tratados.

Destaca-se também o esforço em manter um contacto ativo com organizações locais, promotoras de eventos e outras atividades culturais diversas. Esta comunicação permitiu compreender a necessidades destas partes interessadas, que posteriormente serão uma parte integrante e essencial para o sucesso do projeto Momenter.

Relativamente ao *frontend* foram desenvolvidos vários pontos de vista, relativamente à apresentação dos objetos, ao longo do projeto. Tendo estes sido reajustados de acordo com algumas necessidades levantadas, tanto dos utilizadores como das entidades promotoras. Estes ajustes foram priorizados pela equipa de desenvolvimento, de acordo com o que achou importante ou não solucionar de imediato.

A nível da implementação dos componentes de *backend*, a sua implementação segue uma estrutura fortemente modular, capaz de dar resposta e suporte a qualquer perspetiva de escalabilidade futura da plataforma.

Destaca-se ainda o esforço na construção de um sistema de recomendação distinto de qualquer solução atualmente no mercado. Nesta vertente foi tido o cuidado de evitar a aprendizagem de tendências globais e focou-se na inferências de preferências individuais, traçando assim um perfil próprio para cada utilizador. Relativamente a este componente, a sua evolução e análise será constante e gradual, uma vez que só com o crescimento da comunidade e da sua interação com a plataforma Momenter será possível afinar os parâmetros de aprendizagem e correlação entre características ou categorias de diferentes eventos.

Em suma, avalia-se que o lançamento da plataforma Momenter no contexto académico onde a mesma surgiu servirá como um forte ponte de alavancagem para a sua comunidade inicial, permitindo posteriormente disseminar-se para outros potenciais utilizadores e cativar novos promotores de eventos a divulgarem as suas atividades na plataforma.

A plataforma Momenter, no seu estado inicial viável e funcional poderá ser encontrada no domínio www.momenter.xyz.