



Departamento de Ciências e Tecnologias de Informação

Integração e divulgação de informação desportiva em redes
sociais através de dispositivos móveis

Sérgio Filipe Miranda Vicente Carvalho

Dissertação submetida como requisito parcial para obtenção do grau de:

Mestre em Engenharia Informática

Ramo de Sistemas de Informação e Gestão do Conhecimento

Orientador:

Professor Doutor Carlos Serrão

Instituto Superior de Ciências do Trabalho e da Empresa

Instituto Universitário de Lisboa

[Julho, 2012]

Resumo

A necessidade de acesso a informação é um factor marcante da sociedade actual. Existem diversas formas pelas quais as pessoas podem garantir essa informação, sendo os dispositivos móveis com acesso à Internet uma dessas possibilidades. Assim, o segmento dos dispositivos móveis está em crescendo e as aplicações para *smartphones* têm cada vez mais utilizadores que se tornam autênticos dependentes das mesmas. Nesta perspectiva, o sector do Desporto tem verdadeiros adeptos, sedentos de informação a toda a hora.

Nesta dissertação pretende-se desenvolver uma solução baseada em dispositivos móveis inteligentes e redes sociais que permita aos seus utilizadores uma experiência mais rica e diversificada relativa a conteúdos desportivos, possibilitando que estes subscrevam e acessem a canais específicos de informação. Será ainda concebida a infraestrutura de suporte aos processos de agregação e categorização de conteúdos à medida que os mesmos forem sendo criados. Importa referir aqui o papel fundamental das redes sociais, como o *Twitter* ou o *Facebook*, como fonte de dados.

Uma das mais-valias da solução desenvolvida é permitir aos utilizadores decidir e/ou personalizar os conteúdos recebidos, tanto porque eles próprios decidem a panóplia de conteúdos do seu interesse, como porque existe a possibilidade de definir um conjunto de regras que filtram a informação apresentada.

Com o intuito de validar a solução desenvolvida foram realizados um conjunto de testes em eventos desportivos reais referentes a duas modalidades desportivas, ténis e futebol, nos quais foram obtidos resultados bastante satisfatórios que deram valor e sentido ao trabalho realizado.

Palavras-chave: dispositivos móveis, aplicações para *smartphones*, conteúdos desportivos, agregação e categorização de informação, redes sociais.

Abstract

The need to access information is a striking factor of today's society. There are several ways in which people can gather information, and mobile devices with Internet access are one of those possibilities. Thus, the segment of mobile devices is growing and applications for smartphones have an increasing group of users who become totally dependent on them. Accordingly, the Sports sector has real fans, hungry for information all the time.

This thesis intends to develop a solution based on intelligent mobile devices that allow its users a richer and diverse experience related to sports content, enabling them to access and subscribe specific channels of information. Furthermore, it will be designed the infrastructure to support the processes of aggregation and categorization of content as those are created. For this process it's important to note the central role of social networks like Twitter and Facebook, as data sources.

The implemented approach presents an added value solution to its users, giving them the possibility to decide or personalize the information they wish to receive, both providing them with the opportunity to choose the content of interest, as by allowing the definition of a set of rules which filters the information presented.

In order to validate the developed solution, a set of tests were performed on real sport events related to two main sports, tennis and football. The results obtained were very satisfying, which gave value and meaning to the work done.

Keywords: mobile devices, applications for smartphones, sports content, aggregation and categorization of information, social networks.

Agradecimentos

Ao concluir esta fase da minha vida sinto que por mais que agradeça irá sempre faltar alguém, no entanto irei tentar fazê-lo nos próximos parágrafos.

Em primeiro lugar um agradecimento especial ao professor Carlos Serrão por toda a orientação, conhecimento partilhado, paciência e acima de tudo amizade (próximo jogo de ténis está prometido). Sem a sua ajuda não seria possível ter atingido o objectivo.

Gostaria também de agradecer aos meus pais por tudo aquilo que me proporcionaram ao longo da vida, neste caso com um suporte tanto financeiro como emocional que me permitiu chegar ao final de mais uma etapa na minha formação académica.

Quero agradecer à minha namorada Mafalda, por todo o apoio e carinho ao longo deste percurso, para além do olho clínico que permitiu que este trabalho ficasse (quase) imaculado.

Por fim, agradecer a todos os meus colegas de curso, especialmente aos que me acompanharam para além da licenciatura. Foram longas as horas de estudo e únicos os momentos de amizade que partilhámos, espero por isso não perder o contacto e a amizade de cada um de vós.

Índice

| | |
|---|------|
| Lista de Figuras..... | VII |
| Lista de Tabelas | VIII |
| Dicionário de Acrónimos | IX |
| 1 Introdução | 1 |
| 1.1 Enquadramento..... | 1 |
| 1.2 Objectivos..... | 1 |
| 1.3 Requisitos | 2 |
| 1.4 Estrutura da Dissertação | 5 |
| 2 Estado da Arte..... | 7 |
| 2.1 Redes Sociais..... | 7 |
| 2.1.1 Twitter..... | 9 |
| 2.2 Agregadores e Categorizadores de Conteúdos..... | 11 |
| 2.3 Aplicações para Dispositivos Móveis..... | 17 |
| 3 Desenho | 23 |
| 3.1 Arquitectura do Sistema (Modelo)..... | 23 |
| 3.1.1 Agregação de Conteúdos | 24 |
| 3.1.2 Categorização de Conteúdos..... | 25 |
| 3.1.3 Indexação de Conteúdos | 28 |
| 3.1.4 Aplicação Móvel | 29 |
| 4 Implementação..... | 31 |
| 4.1 Arquitectura da Plataforma de Gestão de Conteúdos Desportivos | 31 |
| 4.2 Ferramentas de Implementação | 33 |
| 4.2.1 Ferramentas de back-end | 33 |
| 4.2.2 Ferramentas de front-end | 34 |
| 4.3 Estrutura de Implementação da Plataforma de Gestão de Conteúdos Desportivos | 36 |
| 4.3.1 Interação Twitter..... | 38 |
| 4.3.2 Interação Aplicação Móvel | 47 |
| 4.3.3 Agregação dos Conteúdos de Preferência..... | 48 |
| 4.3.4 Renovação Índices..... | 49 |
| 4.4 Implementação Aplicação Móvel | 50 |
| 4.4.1 Metodologia MVC..... | 50 |
| 4.5 Servidor <i>Web</i> | 68 |
| 5 Testes e Resultados | 69 |
| 5.1 Duplicação de Mensagens | 69 |
| 5.1.1 Teste entre Duas Mensagens Totalmente Iguais | 69 |

| | | |
|-------|---|----|
| 5.1.2 | Teste com Mensagem do Tipo retweet sem Texto antes do Padrão..... | 70 |
| 5.1.3 | Teste com Mensagem do Tipo retweet com Texto antes do Padrão | 71 |
| 5.2 | Plataforma de Agregação, Categorização e Indexação de Conteúdos | 72 |
| 5.2.1 | Testes Federer vs Nadal..... | 72 |
| 5.2.2 | Testes Sporting Clube de Portugal vs Sport Lisboa e Benfica..... | 76 |
| 5.2.3 | Testes Sport Lisboa e Benfica vs Gil Vicente Futebol Clube..... | 79 |
| 6 | Conclusões..... | 85 |
| 6.1 | Conclusões Gerais | 85 |
| 6.2 | Contribuições | 86 |
| 6.3 | Limitações e Trabalho Futuro..... | 87 |
| 7 | Bibliografia..... | 89 |
| 8 | Anexos..... | 93 |

Lista de Figuras

| | |
|---|----|
| Figura 1 - Crescimento <i>Twitter</i> (Nº Utilizadores) | 9 |
| Figura 2 - Crescimento <i>Twitter</i> (Nº Posts)..... | 9 |
| Figura 3 - Arquitectura do sistema | 23 |
| Figura 4 - Categorização de mensagens..... | 25 |
| Figura 5 - Diagrama Conceptual..... | 28 |
| Figura 6 - Arquitectura da plataforma de gestão de conteúdos desportivos | 31 |
| Figura 7 - Estrutura de implementação da plataforma de gestão de conteúdos desportivos..... | 37 |
| Figura 8 - Coeficiente de Similaridade de <i>Jaccard</i> | 42 |
| Figura 9 - Estrutura Sencha Touch | 50 |
| Figura 10 - Aplicação Móvel: Ecrã Instruções..... | 52 |
| Figura 11 - Aplicação Móvel: Ecrã Gerir Preferências. Acção: Escolha do tipo de informação..... | 54 |
| Figura 12- Aplicação Móvel: Ecrã Gerir Preferências | 54 |
| Figura 13 - Aplicação Móvel: Ecrã Gerir Preferências. Acção: Confirmação de preferências..... | 55 |
| Figura 14 - Aplicação Móvel: Ecrã Gerir Favoritos..... | 56 |
| Figura 15 - Aplicação Móvel: Ecrã Gerir Favoritos. Acção: Visualizar Equipas..... | 57 |
| Figura 16 - Aplicação Móvel: Ecrã Gerir Favoritos. Acção: Inserir Equipa | 57 |
| Figura 17 - Aplicação Móvel: Ecrã Gerir Favoritos. Acção: Detalhes Equipa..... | 58 |
| Figura 18 - Aplicação Móvel: Ecrã Gerir Favoritos. Acção: Eliminar Equipa..... | 58 |
| Figura 19 - Aplicação Móvel: Ecrã Gerir Favoritos.Acção: Adicionar Hashtag Equipa..... | 59 |
| Figura 20 - Aplicação Móvel: Ecrã Gerir Favoritos. Acção: Visualizar Hashtags Equipas..... | 59 |
| Figura 21 - Aplicação Móvel: Ecrã Inserir Mensagem..... | 60 |
| Figura 22 - Aplicação Móvel: Ecrã Inserir Mensagem. Acção: Escolher Hashtag | 61 |
| Figura 23 - Aplicação Móvel: Ecrã Inserir Mensagem. Acção: Escrever Mensagem.... | 62 |
| Figura 24 - Aplicação Móvel: Ecrã Inserir Mensagem. Acção: Confirmar Mensagem . | 63 |
| Figura 25 - Aplicação Móvel: Ecrã Timeline Mensagens | 63 |
| Figura 26 - Mensagens duplicadas totalmente iguais | 70 |
| Figura 27 - Mensagens duplicadas sem texto antes do padrão retweet | 70 |
| Figura 28 - Mensagens duplicadas com texto antes do padrão retweet..... | 71 |
| Figura 29 - Hashtag: #ausopen; Tipo: Geral | 73 |
| Figura 30 - Hashtag: #ausopen. Tipo: Informação | 73 |
| Figura 31 - Hashtag: #Federer. Tipo: Geral | 73 |
| Figura 32 - Hashtag: #Federer. Tipo: Informação | 74 |
| Figura 33 - Hashtag: #Nadal. Tipo: Geral..... | 74 |
| Figura 34 - Hashtag: #Nadal. Tipo: Informação..... | 74 |
| Figura 35 - Hashtag: #Sporting. Tipo: Informação..... | 77 |
| Figura 36 - Hashtag: #Sporting. Tipo: Geral..... | 77 |
| Figura 37 - Hashtag: #Benfica. Tipo: Geral..... | 77 |
| Figura 38 - Hashtag: #Benfica. Tipo: Informação..... | 78 |
| Figura 39 - Hashtag: #Benfica. Tipo: Geral..... | 80 |
| Figura 40 - Hashtag: #Benfica. Tipo: Informação..... | 80 |
| Figura 41 - Hashtag: #GilVicente. Tipo: Geral..... | 80 |
| Figura 42 - Hashtag: #GilVicente. Tipo: Informação..... | 81 |
| Figura 43 - Hashtag: #tliga. Tipo: Geral..... | 81 |
| Figura 44 - Hashtag: #tliga. Tipo: Informação..... | 81 |

Lista de Tabelas

| | |
|---|----|
| Tabela 1 - Requisitos Módulo Agregador..... | 3 |
| Tabela 2 - Requisitos Módulo Categorizador..... | 4 |
| Tabela 3 - Requisitos Módulo Móvel | 5 |
| Tabela 4 - Dados Estatísticos Twitter (2010). Fonte: Articlesbase | 10 |
| Tabela 5 - Sistemas Operativos Móveis (Milhares de unidades vendidas e quota de mercado). Fonte: Gartner (Fevereiro 2012) | 18 |
| Tabela 6 - Campos e validações <i>NewMessageModel</i> | 65 |
| Tabela 7 - Campos e validações <i>PreferencesModel</i> | 65 |
| Tabela 8 - Campos e validações <i>TeamsModel</i> | 66 |
| Tabela 9 - Campos e validações <i>PlayersModel</i> | 66 |
| Tabela 10 - Campos e validações <i>EventsModel</i> | 66 |
| Tabela 11 - Campos e validações <i>HashTeamsModel</i> | 66 |
| Tabela 12 - Campos e validações <i>HashPlayersModel</i> | 67 |
| Tabela 13 - Campos e validações <i>HashEventsModel</i> | 67 |
| Tabela 14 - Dados Gerais Federer vs Nadal..... | 72 |
| Tabela 15 - Resumo Resultados Federer vs Nadal | 75 |
| Tabela 16 - Dados Gerais Sporting vs Benfica..... | 76 |
| Tabela 17 - Resumo Resultados Sporting vs Benfica..... | 78 |
| Tabela 18 - Dados Gerais Benfica vs Gil Vicente..... | 79 |
| Tabela 19 - Resumo Resultados Benfica vs Gil Vicente | 82 |

Dicionário de Acrónimos

SO – Sistema Operativo

URL - Uniform Resource Locator

API - Application Programming Interface

SOA - Service Oriented Architecture

RSS - Really Simple Syndication

HTML - HyperText Markup Language

CSS – Cascading Style Sheets

MVC – Model-View-Controller

JSON – JavaScript Object Notation

JSONP – JSON with padding

1 Introdução

1.1 Enquadramento

Com o crescimento do mercado dos dispositivos móveis, em particular o segmento dos *smartphones*, é possível termos acesso a informação de diversas fontes, em qualquer lugar e em qualquer hora. É hoje possível, graças a um conjunto de dispositivos (baseados em *Windows Phone 7*, *Android*, *RIM Blackberry*, *iOS*, entre outros), ter acesso a informação útil, interactiva e em tempo real de uma forma completamente ubíqua (independente do local e da hora).

Por outro lado, a proliferação deste tipo de dispositivos tem levado a uma nova área de negócio que consiste no desenvolvimento e comercialização de aplicações para *smartphones* (a *AppStore* da *Apple* para os dispositivos com *iOS* (*iPhone*, *iPad*, *iPod*) é um exemplo disto).

Algumas das aplicações de sucesso e com potencial crescimento neste sector de aplicações para *smartphones*, estão directamente relacionadas com desporto. Existem aplicações para divulgar os resultados da UEFA, da FIFA, das diversas ligas nacionais de futebol, resultados de F1 e Moto GP, entre muitos outros. A maior parte destas aplicações são auto-contidas e têm apenas os resultados relacionados com o desporto e com a área geográfica do mesmo, para além disto, os conteúdos são gerados pela própria entidade que disponibiliza a aplicação. Neste sentido, a abordagem proposta nesta dissertação, onde os conteúdos serão, em grande medida, produzidos e distribuídos pelos próprios utilizadores e colocados em redes sociais, difere das referidas anteriormente.

1.2 Objectivos

Pretende-se com esta dissertação realizar o estudo e desenvolvimento de uma infra-estrutura integrada para a divulgação em tempo (quase) real de conteúdos desportivos subscritos pelo utilizador, através das redes sociais. Assim, este trabalho irá procurar tirar partido das redes sociais e da *Web-social*, como forma de obter informação sobre os diferentes eventos desportivos, assim como informação adicional (comentários, fotos e vídeos) por parte de múltiplos utilizadores que estejam a assistir ao evento. Por outro lado, este projecto terá uma forte componente de categorização dos conteúdos agregados, com o intuito de possibilitar aos utilizadores que acedem aos

Integração e divulgação de informação desportiva em redes sociais através de dispositivos móveis

mesmos (a partir dos seus dispositivos móveis) obterem informação melhor personalizada.

Podem ser definidos dois grandes objectivos à partida:

1) Estudo e desenvolvimento de uma infra-estrutura de suporte para agregação e categorização dos resultados que foram sendo produzidos à medida que os mesmos ocorrem;

2) Desenvolvimento de aplicação para dispositivos móveis inteligentes (*smartphones, tablets*) que permite a criação e/ou alimentação de resultados desportivos para a plataforma, assim como outra informação (comentários, fotos e vídeos) sobre um determinado evento em questão, ou ainda subscrever e aceder a canais de informação que tenham a ver com o evento desportivo que se pretende acompanhar.

1.3 Requisitos

De acordo com os objectivos propostos em cima, podemos enumerar um conjunto de requisitos funcionais relativos às funcionalidades a desenvolver. O conjunto de funcionalidades estará dividido em três módulos:

- (A) Agregador – Neste módulo serão definidos os requisitos necessários em termos de agregação de informação;
- (C) Categorizador – Neste módulo serão identificadas funcionalidades relativas ao processo de categorização de informação;
- (M) Móvel – Neste módulo serão identificadas funcionalidades relativas à aplicação móvel.

A identificação escolhida para apresentar o conjunto de requisitos funcionais é composta por uma letra e um número. A letra indicará o módulo e poderá ser uma das seguintes:

- A – Agregador;
- C – Categorizador;
- M – Móvel.

| Requisitos Módulo Agregador | |
|------------------------------------|--|
| A001 | Deverá ser possível agregar conteúdos de equipas a partir de diversas redes sociais |
| A002 | Deverá ser possível agregar conteúdos de jogadores a partir de diversas redes sociais |
| A003 | Deverá ser possível agregar conteúdos de eventos a partir de diversas redes sociais |
| A004 | Deverá ser possível agregar conteúdos escritos em português e inglês a partir de diversas redes sociais |
| A005 | Deverá ser possível agregar conteúdos de equipas a partir da própria aplicação móvel |
| A006 | Deverá ser possível agregar conteúdos de jogadores a partir da própria aplicação móvel |
| A007 | Deverá ser possível agregar conteúdos de eventos a partir da própria aplicação móvel |
| A008 | Deverá ser possível agregar conteúdos escritos em português e inglês a partir da própria aplicação móvel |

Tabela 1 - Requisitos Módulo Agregador

| Requisitos Módulo Categorizador | |
|--|--|
| C001 | Deverá ser possível categorizar informação por equipa |
| C002 | Deverá ser possível categorizar informação por individualidade/jogador |

| | |
|-------------|---|
| C003 | Deverá ser possível categorizar informação por evento desportivo |
| C004 | Deverá ser possível categorizar informação por tipo (geral, informação) |
| C005 | Deverá ser possível categorizar informação por língua (português ou inglês) |
| C006 | Deverá ser possível descartar conteúdos com informação imprópria |
| C007 | Deverá ser possível descartar conteúdos com informação imprópria |
| C008 | Deverá ser possível descartar conteúdos com número de caracteres insuficiente |
| C009 | Deverá ser possível descartar conteúdos duplicados |

Tabela 2 - Requisitos Módulo Categorizador

| Requisitos Módulo Móvel | |
|--------------------------------|---|
| M001 | A aplicação deverá permitir a definição de equipa(s) preferida(s) |
| M002 | A aplicação deverá permitir a definição de jogador(es) preferido(s) |
| M003 | A aplicação deverá permitir a definição de evento(s) favorito(s) |
| M004 | A aplicação deverá permitir que o utilizador realize upload de conteúdos multimédia sobre equipas |
| M005 | A aplicação deverá permitir que o utilizador realize upload de conteúdos multimédia sobre jogadores |
| M006 | A aplicação deverá permitir que o utilizador realize upload de conteúdos multimédia sobre eventos |

| | |
|-------------|--|
| M007 | A aplicação deverá permitir que o utilizador defina a(s) língua(s) das mensagens que quer visualizar |
| M008 | A aplicação deverá permitir que o utilizador defina a(s) fonte(s) das mensagens que quer visualizar |
| M009 | A aplicação deverá permitir que o utilizador defina o(s) tipo(s) das mensagens que quer visualizar |
| M010 | A aplicação deverá fornecer instruções sobre o uso da mesma |

Tabela 3 - Requisitos Módulo Móvel

1.4 Estrutura da Dissertação

A estrutura apresentada ao longo deste documento reflecte os diferentes momentos de execução do trabalho desenvolvido. No capítulo 2 apresenta-se o estado da arte relativamente às principais temáticas exploradas. No capítulo 3 é apresentado o modelo construído para dar resposta ao desafio proposto, enquanto no capítulo 4 é explicada a implementação do conjunto de funcionalidades e ferramentas utilizadas que permitiram implementar o modelo definido. No capítulo 5 são exibidos os testes e os resultados obtidos, e por fim, no capítulo 6 apresentam-se as conclusões e propostas para trabalho futuro.

Integração e divulgação de informação desportiva em redes sociais através de dispositivos móveis

2 Estado da Arte

2.1 Redes Sociais

A obtenção e partilha de informação são muito importantes na sociedade actual. O uso da Internet facilita essas tarefas e são cada vez mais as pessoas que procuram usufruir deste meio como forma de responder às suas necessidades. Dados relativos a este uso em Portugal referem que 56,4% da população que utiliza Internet usa as redes sociais como actividade de comunicação (Taborda, et al., 2010). O mesmo estudo revela outros dados interessantes no âmbito desta dissertação, tais como: 84% das pessoas que aderem às redes sociais fazem-no com o intuito de partilhar comentários, vídeos e fotos, e 20% dos utilizadores procuram conteúdos desportivos nos referidos sítios de partilha (Taborda, et al., 2010). Tendo como base estes dados, faz todo o sentido pensar nas redes sociais como uma boa fonte de conteúdos multimédia para a plataforma que se pretende desenvolver.

Segundo (Boyd, et al., 2008) uma rede social (*online*) define-se como um conjunto de serviços na *web* que permitem aos seus utilizadores, entre outras coisas, construir um perfil público ou semi-público dentro de um sistema limitado, criar uma lista de pessoas com quem se relacionam e visualizar ou percorrer as listas criadas por outros utilizadores com os quais partilham uma ligação. Embora existam inúmeras redes sociais com características técnicas específicas, todas elas têm por base a existência de um perfil de utilizador no qual é exibida uma lista de amigos que são utilizadores da mesma rede social.

Os mesmos autores defendem que a primeira rede social *online* foi lançada em 1997, tendo a denominação *SixDegrees*¹. Com a integração de um conjunto de funcionalidades características das redes sociais (criação de perfis, listagem de amizades e possibilidade de percorrer a lista de outros utilizadores), *SixDegrees* foi o primeiro serviço a combinar funcionalidades que já existiam noutros serviços mas não em conjunto. Infelizmente, esta rede social cessou actividade três anos após a sua criação, segundo o seu fundador por estar à frente do seu tempo, já que muitas pessoas já utilizavam a Internet mas a sua rede de amizades *online* ainda era pouco extensa.

Uma das principais formas de diferenciar as redes sociais *online* existentes é através das suas características estruturais relativas a visibilidade e acesso. Algumas

¹ <http://www.sixdegrees.com/>

redes sociais permitem acesso aos perfis dos seus utilizadores até a pessoas que não possuem perfil criado nas mesmas, como Friendster² ou Tribe.net³, por outro lado, temos redes sociais que controlam a visualização de conteúdos tendo por base o tipo de conta do utilizador – a rede social LinkedIn⁴ é disso exemplo dividindo utilizadores com conta paga dos restantes. Existem também redes sociais que permitem aos seus utilizadores decidir se querem o seu perfil visível publicamente, ou apenas para amigos - MySpace⁵. Ainda, a abordagem utilizada no Facebook⁶, onde utilizadores que pertencem à mesma rede têm acesso aos perfis uns dos outros, no entanto se um utilizador decidir que quer limitar essa permissão é-lhe dada essa possibilidade.

Existem diversas categorias de serviços para a partilha de conteúdos, entre as quais temos *micro-blogging*, notícias, vídeo, fotos e música (Gupta, et al., 2009). Graças ao aumento da popularidade das redes sociais, muitos utilizadores exploram informação relativa a serviços distintos, criando-se assim a necessidade de existirem serviços de agregação de conteúdos (que disponibilizam informação relativa a múltiplos serviços de uma forma centralizada).

No estudo de um desses agregadores, o *FriendFeed*⁷, obtiveram-se dados que demonstram a importância dos serviços de *micro-blogging* (*Micro-blogging* é um fenómeno definido como “uma forma de publicação de blogue que permite aos utilizadores realizarem actualizações breves de texto (geralmente com menos de 200 caracteres) e publicá-las para que sejam vistas publicamente ou apenas por um grupo restrito escolhido pelo utilizador.” (Wikipédia, 2007)). Segundo esse estudo, os serviços de *micro-blogging* geram cinco vezes mais conteúdo que qualquer outro serviço, sendo o *Twitter*⁸ o principal responsável por grande parte desse conteúdo (Gupta, et al., 2009).

² <http://www.friendster.com/>

³ <http://www.tribe.net/welcome>

⁴ <http://www.linkedin.com/>

⁵ <http://www.myspace.com/>

⁶ <https://www.facebook.com/>

⁷ <http://friendfeed.com/>

⁸ <http://twitter.com/>

2.1.1 Twitter

O *Twitter* surgiu em Agosto de 2006 mas deu o seu grande “salto”, suscitando o interesse de inúmeros utilizadores, quando venceu um prémio na conferência *SXSW*⁹ *Web Awards* em Março de 2007.

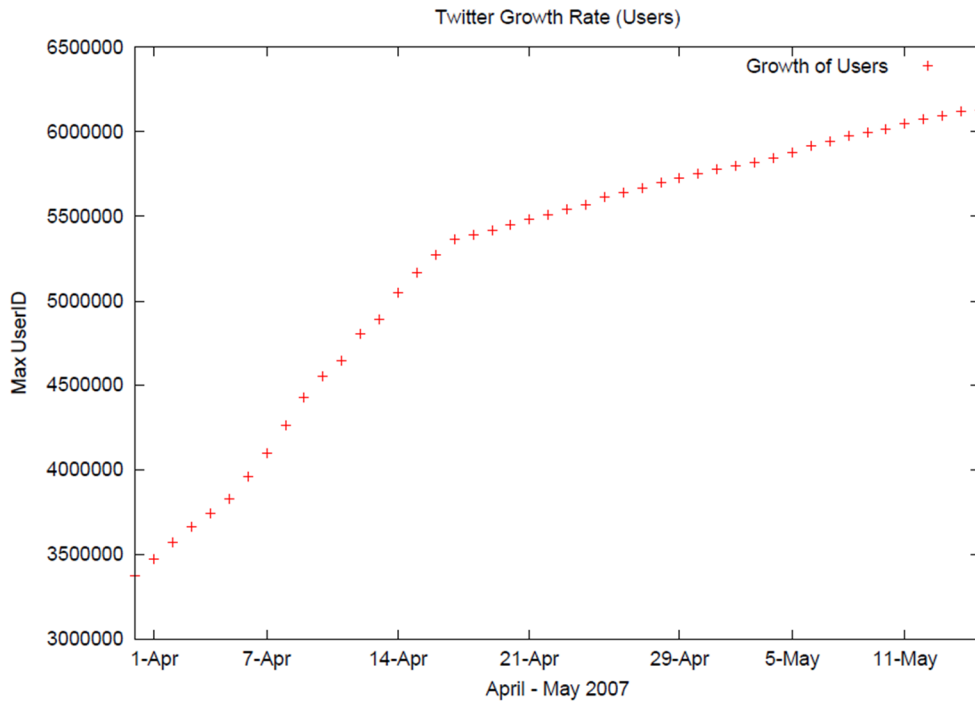


Figura 1 - Crescimento *Twitter* (Nº Utilizadores)

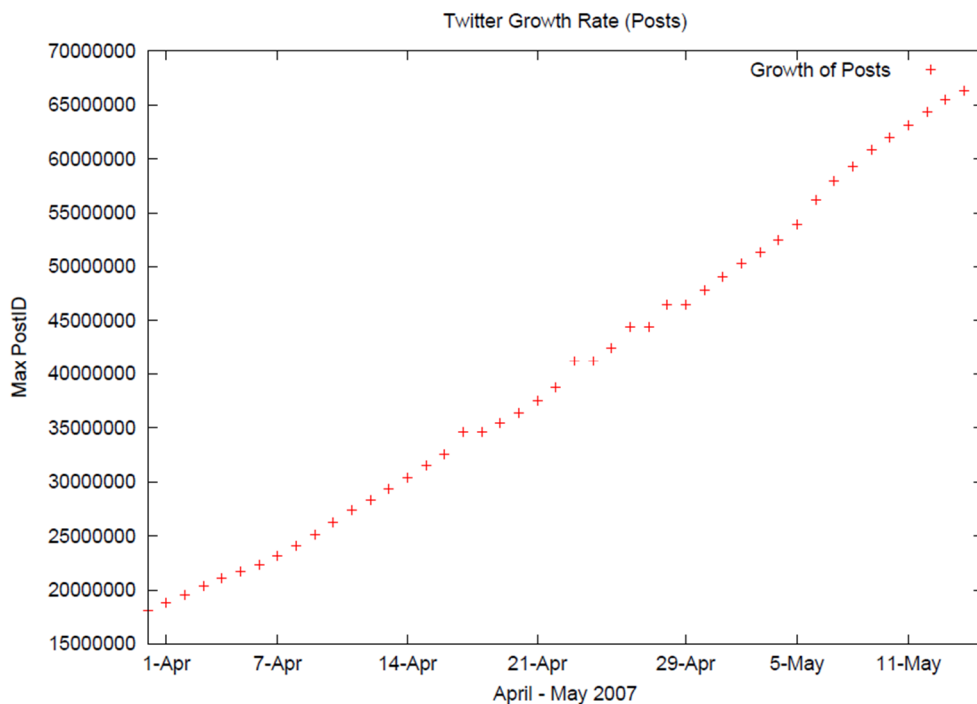


Figura 2 - Crescimento *Twitter* (Nº Posts)

⁹ <http://sxsw.com/>

Os valores apresentados na Figura 1, em cima, demonstram que houve uma grande curiosidade a partir do início do mês de Abril de 2007, tendo esse entusiasmo acalmado um pouco a partir do meio do mesmo mês. Por outro lado, e como podemos ver na Figura 2, o número de *posts* realizados seguiu um crescimento uniforme e constante, duplicando a cada mês (o que indica uma base estável de geração de conteúdos por parte dos utilizadores) (Asshay, et al., 2007).

Um conjunto de estatísticas interessantes reforça o sucesso do Twitter. Alguns dos dados mais relevantes são apresentados na Tabela 1, dos quais se destacam a existência de mais de 175 milhões de contas criadas, 95 milhões de *tweets* diários e a língua portuguesa ser uma das quatro mais utilizadas nesta rede social (ArticlesBase.com, 2010).

| Estatísticas <i>Twitter</i> |
|--|
| + 175 milhões de contas criadas |
| Criação de 300,000 contas de utilizador por dia no <i>Twitter</i> |
| Geração de 3 mil milhões de pedidos diários a partir de 180 milhões de visitantes |
| 95 milhões de <i>tweets</i> por dia, correspondente a 640 <i>tweets</i> por segundo |
| Motor de pesquisa do <i>Twitter</i> recebe 600 milhões de pesquisas diariamente |
| Quinta e Sexta são os dias de maior actividade no <i>Twitter</i> |
| A hora de maior actividade no <i>Twitter</i> é entre as 22 e as 23 |
| As línguas mais usadas são Inglês, Português, Japonês e Espanhol |
| Estados Unidos, Índia, Japão, Alemanha, Grã-Bretanha, Brasil, Canadá, Indonésia, Austrália e Espanha são os países que mais utilizam o <i>Twitter</i> |
| Mais de 70,000 aplicações foram criadas usando a API do <i>Twitter</i> |
| Mais de 67% das mensagens enviadas através do <i>Twitter</i> estão sob a forma de estado actual do utilizador, conversas privadas e ligações para notícias ou artigos de blogues pessoais |

Tabela 4 - Dados Estatísticos *Twitter* (2010). Fonte: Articlesbase

No que respeita à realidade portuguesa, e ao estudo levado a cabo pela OberCom¹⁰ (Taborda, et al., 2010), os dados revelam que o *Twitter* é a terceira rede

¹⁰ <http://www.obercom.pt/content/home>

social mais usada no nosso país, atrás de *Hi5*¹¹ e *Facebook*, ultrapassando por exemplo o *Orkut*.

Em 2008, o uso do *Twitter* em Portugal não tinha qualquer expressão, algo que já não acontece a partir de 2010, com 7,9% dos internautas inquiridos a afirmar a sua utilização, percentagem que sobe para os 24,2% se restringirmos o escalão etário aos mais jovens (dos 15 aos 24 anos) (Taborda, et al., 2010).

O “MaisFutebol”, um dos jornais desportivos com edição *online* mais conhecidos em Portugal, para além de possibilitar o acompanhamento das notícias através da sua conta no *Twitter*, disponibilizou uma página que acompanha todas as informações futebolísticas em tempo real produzidas a partir de múltiplas fontes desportivas no *Twitter*, e uma lista que acompanha os jogadores que também usufruem da rede social. Assim, é-nos dada a hipótese de seguir uma lista que apresenta as ideias e opiniões vindas directamente dos “craques” que também gostam de usar o *Twitter* (Maisfutebol, 2011) (Maisfutebol, 2011).

2.2 Agregadores e Categorizadores de Conteúdos

A necessidade de agregação e categorização de conteúdos sejam eles textuais, imagens ou vídeos, entre outros, é um problema muito comum e existente em áreas totalmente heterogéneas.

A nível das ciências da computação e do ramo da inteligência artificial um campo com grande exploração a nível científico é o processamento de linguagem natural. Neste campo existem algumas tarefas que estão ligadas aos processos de agregação e categorização de informação, tais como extracção de informação ou recuperação de informação. Uma abordagem comumente utilizada por investigadores nas áreas descritas em cima, com objectivos relacionados com prospecção de dados, é denominada *clustering*.

Uma definição abstracta de *clustering* será “o processo de organizar objectos em grupos cujos membros são similares de alguma forma”. *Clustering* é muitas vezes considerado o problema mais importante de aprendizagem não supervisionada que procura encontrar determinada estrutura numa colecção de dados sem rótulos.

¹¹ <http://hi5.com/friend/displayHomePage.do>

Um *cluster* pode ser entendido como uma colecção de objectos similares entre si e diferentes relativamente a objectos pertencentes a outros *clusters*. O conceito de similaridade pode variar: por um lado podemos ter um critério baseado na distância, ou seja, dois ou mais objectos pertencem ao mesmo *cluster* se são considerados parecidos de acordo com uma distância específica (distância geométrica) – chamado agrupamento baseado na distância (*distance-based clustering*). Por outro lado temos agrupamento conceptual, onde dois ou mais objectos pertencem ao mesmo *cluster* se este define um conceito comum a todos esses objectos. Por outras palavras, os objectos são agrupados de acordo com conceitos definidos, em contrapartida a simples medidas de similaridade.

Um dos componentes específicos associados a análise de agrupamento é o algoritmo de agrupamento ou método de agrupamento a utilizar.

São diversas as abordagens defendidas quanto aos métodos e algoritmos de agrupamento existentes. Uma possível classificação dos diferentes métodos tem a ver com a forma como o subconjunto de dados pertencentes aos diferentes *clusters* fica organizado. Uma distinção encontrada define a existência de dois tipos: agrupamento divisivo (*hard clustering*) e agrupamento difuso (*fuzzy clustering*). Enquanto no agrupamento divisivo um objecto pode pertencer apenas a um único *cluster*, significando que os dados são divididos em subconjuntos mutuamente exclusivos, no agrupamento difuso os dados podem pertencer a vários *clusters* em simultâneo com diferentes graus de associação.

Ao nível dos algoritmos uma divisão proposta divide os algoritmos de *clustering* em quatro opções: *clustering* exclusivo/divisivo, *clustering* de sobreposição, *clustering* hierárquico e *clustering* probabilístico (Jain, et al., 1999) (Matteucci, 2004) (Steinbach, et al., 2004):

- 1) *Clustering* exclusivo: representa o método divisivo descrito em cima. Um objecto que pertence a um *cluster* não poderá fazer parte de qualquer outro. Um exemplo deste tipo de algoritmos é o *K-Means Clustering* que funciona da seguinte forma: dado que cada *cluster* está associado a um ponto central (*centroid*) e cada ponto/objecto é atribuído ao *cluster* que contiver o ponto central mais próximo de si, inicialmente é necessário escolher o número de *clusters* e os pontos centrais de cada *cluster*, depois associa-se cada objecto ao grupo/*cluster* com ponto central mais próximo e quando todos os objectos já estiverem associados a um *cluster* recalcula-se a posição dos pontos

- centrais de cada *cluster*. A associação dos objectos aos *clusters* e recálculo dos pontos centrais repete-se até que não exista mudança dos pontos centrais;
- 2) *Clustering* de sobreposição: utiliza conjuntos difusos para agrupar os dados, permitindo assim que cada objecto/ponto pertença a vários *clusters* com graus de associação diferentes. Um exemplo deste tipo de algoritmos é o *Fuzzy C-Means* que é bastante usado para realizar reconhecimento de padrões;
 - 3) *Clustering* hierárquico: este algoritmo baseia-se na união entre os dois *clusters* mais próximos. A condição inicial é a de que cada objecto/ponto deve ser entendido como um *cluster* e depois de algumas iterações atingimos o conjunto final de *clusters* desejados. Existem dois tipos principais de *clustering* hierárquico: aglomerativo e divisivo. No tipo aglomerativo inicialmente cada objecto é visto como um *cluster* individual e a cada iteração os dois *clusters* mais próximos são unidos num único até à última iteração onde a totalidade dos objectos pertence ao mesmo *cluster*. No tipo divisivo, a condição inicial é a de termos um único *cluster* totalmente inclusivo, que engloba todos os objectos, e a cada iteração divide-se o(s) *cluster(s)* existente(s) até cada objecto simbolizar um *cluster*;
 - 4) *Clustering* probabilístico: este tipo de algoritmos utiliza uma abordagem totalmente probabilística, baseada num modelo que procura otimizar o agrupamento dos dados.

(Messina, et al., 2008) definiram a utilização de um algoritmo de agrupamento hierárquico baseado numa matriz de equivalências para agregar conteúdos existentes em artigos de jornais e transmissões televisivas. O ponto de partida para o processo de agregação eram *feeds RSS*. Sobre os itens *RSS* é feita uma análise linguística ao nível do título e descrição, e marcados com uma etiqueta de acordo com uma categoria gramatical (adjectivo, conjunção, verbo, nome, etc). De seguida efectuam-se pesquisas baseadas na análise realizada.

O núcleo do sistema desenvolvido é baseado num algoritmo de agrupamento híbrido, no sentido de utilizar fontes heterogéneas (*feeds RSS* e programas televisivos) de informação e relaciona essas fontes de acordo com a sua similaridade semântica.

(Yifan, et al., 2007) apresentaram o estudo de uma plataforma que realiza extracção semântica de eventos relativos à modalidade desportiva basquetebol. É usada

uma abordagem que recorre a um algoritmo de agrupamento não supervisionado com o objectivo de agrupar as descrições ou textos recolhidos em diferentes grupos que correspondem a eventos específicos. O algoritmo de agrupamento usado designa-se *k-means* (Manning, et al., 2008) e permitiu agrupar as descrições em nove grupos diferentes que correspondem a nove eventos de basquetebol distintos. De seguida e de forma a encontrar um conjunto de palavras-chave associados aos eventos determinados é utilizada a técnica *Latent Semantic Analysis (LSA)* (Deerwester, et al., 1990), bastante usada em questões de processamento de linguagem natural - assume a existência de estruturas latentes no uso das palavras que correspondem a significados semânticos dos documentos – juntamente com o modelo *tf-idf* (Ramos, 2003). Depois de definidas as palavras-chave era realizada a detecção dos eventos nas descrições recolhidas.

Outra abordagem muito explorada por investigadores nas áreas relacionadas com manipulação de informação é a utilização de modelos por tópicos. Vários autores acreditam que a melhor forma de compreender e categorizar informação partilhada, em especial nas plataformas sociais, é recorrendo a este tipo de modelos.

Modelos por tópicos são um tipo de modelo estatístico que procura descobrir tópicos abstractos com ocorrência numa colecção de documentos e de que forma estes se relacionam e evoluem ao longo do tempo. A ideia que está na base destes modelos sugere que os documentos podem ser vistos como uma mistura de tópicos, onde cada tópico é representado por uma distribuição probabilística sobre um conjunto de palavras. Os algoritmos que implementam estes modelos não requerem qualquer tipo de anotações prévias ou rótulos sobre os documentos a analisar, fazendo emergir os tópicos a partir da análise dos documentos originais (Blei, 2011) (Steyvers, et al., 2007).

(Ramage, et al., 2010) apresentaram a implementação de um modelo por tópicos de aprendizagem parcialmente supervisionada denominado *Labeled LDA (Latent Dirichlet Allocation)* (Ramage, et al., 2009) que tenta mapear os *tweets* em quatro dimensões: substância (relativa a eventos, ideias, coisas ou pessoas), estilo (ligada a tendências mais amplas de uso da linguagem), estado (associada a actualizações pessoais) e características sociais (ligada a fins de comunicação social). Este modelo, sendo uma extensão do *LDA* puro (Blei, et al., 2003), permite a introdução de novas etiquetas que se aplicam apenas a um subconjunto das mensagens, o que permite aprender conjuntos de palavras associadas a etiquetas particulares (como *hashtags*). Os autores consideram ainda que as *hashtags* estão associadas à categoria substância

(eventos e pessoas) e que embora a perspectiva comum seja diferente, a utilização da categoria substância é o dobro comparativamente à categoria estado (actualizações pessoais). Este conjunto de perspectivas é muito interessante no âmbito desta dissertação e do modelo definido para agregação e categorização dos conteúdos recolhidos.

(Hong, et al., 2010) tentaram perceber como treinar um modelo por tópicos eficazmente em ambientes onde as mensagens têm um tamanho mais reduzido, como é o caso do *Twitter*. Procuraram, entre outros objectivos, classificar mensagens e utilizadores segundo categorias, entre as quais desporto. Utilizaram dois tipos de modelos por tópicos, *LDA* e Autor-Tópico (Rosen-Zvi, et al., 2004) (este modelo estende o modelo *LDA* ao introduzir informação de autoria, ou seja, cada autor é associado de acordo com uma distribuição multinomial aos tópicos gerados). Para as tarefas de classificação de mensagens e utilizadores, o modelo padrão *LDA* apresentou melhores resultados que o modelo Autor-Tópico.

Recentemente, (Zhao, et al., 2011) procuraram comparar os conteúdos existentes no *Twitter* e os disponibilizados por um meio de comunicação tradicional, o jornal “*New York Times*”, utilizando um modelo não supervisionado por tópicos. Assim, para descobrir e classificar os conteúdos heterogéneos, definiram três conceitos: tópico, que representa um assunto discutido em um ou mais documentos; categoria de tópico, que agrupa tópicos pertencentes ao mesmo assunto (ex. Artes, Negócio, etc); tipos de tópico, que caracteriza a natureza do tópico – identificaram três tipos de tópicos: orientados a eventos, entidades e assuntos globais. Descobriram que o *Twitter* e os meios de comunicação tradicionais cobriam um número de categorias similares relativas a tópicos de informação, no entanto, as categorias mais importantes em cada um eram totalmente diferentes. Enquanto na descoberta de tópicos, no “*New York Times*” foi aplicado directamente o modelo *LDA* tradicional, para o *Twitter* foi desenvolvida uma extensão a essa modelo, que os autores denominaram *Twitter LDA*. Enquanto o tradicional *LDA* pressupõe que cada documento tem associado um conjunto de tópicos, esta extensão definia que um *tweet* diz respeito a um tópico em exclusivo.

Algumas investigações optam por desenvolver modelos considerados “à medida”, sem recorrer a algoritmos complexos e abrangentes, mas desenvolvendo características específicas que permitam gerir determinados conteúdos.

(Xu, et al., 2006) apresentaram uma abordagem inovadora na detecção de eventos desportivos recolhendo informações de jogos de futebol a partir de sítios *Web* desportivos dedicados. O processo é simples: realizava-se a extracção dos textos que descrevem um determinado evento desportivo através da correspondência com um conjunto de palavras-chave definidas à partida. Ou seja, definiram vários eventos existentes no futebol e para cada evento estudou-se quais as palavras-chave que melhoram o caracterizam.

(Sarmiento, et al., 2009) introduziu um método automático para criação e actualização de uma colecção de dados relativos a opiniões políticas. A abordagem pressupunha três fases: inicialmente recolheram opiniões colocadas por leitores de um jornal *online*; de seguida, aplicaram um conjunto de regras léxico-sintácticas (definidas manualmente) a esses comentários/opiniões de forma a identificar um conjunto de frases que revelassem opiniões, positivas ou negativas, relativamente a entidades políticas relevantes; por fim, as opiniões identificadas são propagadas para as restantes frases que fazem parte dos comentários que mencionam as entidades políticas, com o objectivo de se possuir um conjunto mais alargado de frases associadas a opiniões. O conjunto de regras léxico-sintácticas foi traduzido na criação de padrões que reflectiam a maneira como as frases eram construídas. Esta investigação está, entre outras, relacionada com o projecto *REACTION*¹², que permitiu o desenvolvimento do *Twitómetro*¹³, um instrumento que permitiu perceber os sentimentos revelados pelos utilizadores do *Twitter*, relativamente aos cinco líderes partidários portugueses mais relevantes nas eleições legislativas de 2011.

Numa altura em que esta dissertação estava próxima do seu final, também no âmbito do projecto *REACTION*, o SAPO lançou o “*Twitteuro*”¹⁴ referente ao evento desportivo de futebol “Euro 2012”. O “*Twitteuro*” - resultado da colaboração entre o *Labs SAPO*, centros de investigação e algumas universidades - funciona como “um barómetro de popularidade na *Twittosfera* das equipas e dos jogadores que participam no Euro 2012” (Meios e Publicidade, 2012), ideia que se assemelha aos objectivos base deste trabalho conforme será explicado nos seguintes capítulos - o que vem cimentar e validar a utilidade do mesmo.

¹² <http://xldb.di.fc.ul.pt/wiki/Reaction>

¹³ <http://legislativas.sapo.pt/2011/twitometro/>

¹⁴ <http://twitteuro.sapo.pt/>

No âmbito desta dissertação a opção de definir um conjunto de padrões tem bastante interesse. Enquanto os autores em (Sarmiento, et al., 2009) definiram padrões que permitiram identificar opiniões políticas, numa perspectiva de categorizar informação desportiva essa definição permitirá reconhecer um conjunto de padrões que identificam conteúdos desportivos relevantes. Assim, indo de encontro aos objectivos específicos desta dissertação, será desenvolvido um mecanismo singular para gerir todos os conteúdos, onde a metodologia utilizada neste projecto resulta dos pressupostos defendidos pelos autores em (Sarmiento, et al., 2009) – neste caso, apostando na criação de um conjunto de padrões específicos associados a desporto, e, por outro lado, a definição de um conjunto de palavras-chave defendidas pelos autores em (Xu, et al., 2006) que funcionarão como a base das pesquisas realizadas pela plataforma, o que se assemelha ao que foi feito pelos investigadores do “*Twitteuro*” (Meios e Publicidade, 2012).

2.3 Aplicações para Dispositivos Móveis

O desenvolvimento de aplicações para ambientes móveis está directamente relacionado com o mercado associado ao conjunto de dispositivos móveis que permitem o acesso ou compra das mesmas. A venda deste tipo de dispositivos, com particular relevo para um deles – os *smartphones* – mostra um crescimento exponencial, e tal facto motiva os programadores que desenvolvem aplicações para este mercado específico. Num estudo realizado pela *International Data Corporation (IDC)*¹⁵ demonstrou-se que a venda de *smartphones* atingiu valores nunca antes vistos, com a venda destes dispositivos a superiorizar-se à de computadores pessoais pela primeira vez, no último trimestre de 2010. Neste período, os fabricantes indicaram a venda de 100,9 milhões de *smartphones* comparativamente aos 92 milhões de computadores pessoais (PCPro, 2010).

Ao contrário dos telemóveis que possuem uma espécie de sistema proprietário, os *smartphones* integram um sistema operativo que permite um conjunto mais alargado de funcionalidades, como acesso a correio electrónico e redes sociais, acesso à Internet, visualização de conteúdos multimédia, entre outros. Existem empresas especializadas no desenvolvimento de sistemas operativos para dispositivos móveis.

¹⁵ <http://www.idc.com/>

| Sistema Operativo | 4º Trimestre 2011 Unidades | 4º Trimestre 2011 Quota de Mercado (%) | 4º Trimestre 2010 Unidades | 4º Trimestre 2010 Quota de Mercado (%) |
|---------------------------|----------------------------|--|----------------------------|--|
| <i>Android</i> | 75,906.1 | 50.9 | 30,801.2 | 30.5 |
| <i>iOS</i> | 35,456.0 | 23.8 | 16,011.1 | 15.8 |
| <i>Symbian</i> | 17,458.4 | 11.7 | 32,642.1 | 32.3 |
| <i>Research in Motion</i> | 13,184.5 | 8.8 | 14,762.0 | 14.6 |
| <i>Bada</i> | 3,111.3 | 2.1 | 2,026.8 | 2.0 |
| <i>Microsoft</i> | 2,759.0 | 1.9 | 3,419.3 | 3.4 |
| Outros SOs | 1,166.5 | 0.8 | 1,487.9 | 1.5 |
| Total | 149,041.8 | 100.0 | 101,150.3 | 100.0 |

Tabela 5 - Sistemas Operativos Móveis (Milhares de unidades vendidas e quota de mercado). Fonte: Gartner (Fevereiro 2012)

Os sistemas operativos/empresas mais importantes encontram-se discriminados na Tabela 5, em cima. Através dos dados presentes nesta tabela podemos verificar que existem seis sistemas operativos com maior importância no mercado dos dispositivos móveis nos últimos anos: *Android*, *iOS*, *Symbian*, *Blackberry OS (Research in Motion)*, *Bada* e *Windows Phone (Microsoft)*.

Dados referentes aos últimos três meses de 2011 mostram que o sistema operativo *Android*, pertencente à gigante *Google*, já consegue ultrapassar os 50% de quota de mercado, ultrapassando em larga escala a sua concorrência. O facto de ser uma plataforma flexível e aberta contribui para este domínio e crescimento exponencial.

Embora os números não tenham sido tão expressivos como os conseguidos pelo sistema operativo *Android*, o *iOS* também cimentou uma posição no mercado nos últimos anos sendo o segundo sistema operativo com mais utilizadores a nível mundial. Baseado no sistema operativo *Mac OS X* desenvolvido para máquinas *desktop*, o *iOS* é considerado o melhor e mais estável sistema operativo móvel ao nível da performance, no entanto tem a particularidade de ser fechado (aplicações desenvolvidas para *iOS* necessitam de aprovação) o que pode ser visto como um ponto fraco.

O crescimento de uns reflecte-se no decréscimo de importância de outros e nesse sentido um dos sistemas operativos em claro declínio é o *Symbian*. No final de 2010 era

o sistema operativo com maior importância, tendo sido verificada uma queda abrupta relativamente a igual período no ano seguinte. Este facto deve-se essencialmente ao facto dos telemóveis *Nokia*, uma das marcas com mais mercado a nível mundial, terem migrado do sistema operativo *Symbian* para o *Windows Phone 7*.

O sistema operativo *Blackberry OS*, pertencente à empresa *Research in Motion*, é outro dos sistemas operativos que sofre com a forte concorrência dos sistemas operativos *Android* e *iOS*. Inicialmente desenvolvido com fins associados ao negócio, o *Blackberry OS* tinha como principal função ser um assistente pessoal digital, sendo reconhecido pelo seu sistema de gestão de correios electrónicos.

O *Windows Phone (7)* é o sistema operativo da *Microsoft*. Apresentado em Fevereiro de 2010, ainda está a encontrar a sua posição no mercado. Um ano após a sua apresentação, em Fevereiro de 2011, foi comunicada a parceria entre *Nokia* e *Microsoft* fazendo do *Windows Phone 7* o principal sistema operativo dos dispositivos *Nokia*. Tal facto pesará na afirmação de que este sistema operativo terá provavelmente uma preponderância mais efectiva no mercado futuro. Recentemente, em finais de Junho de 2012, foi anunciado o *Windows Phone 8*, que ao partilhar a mesma base de *software* com o *Windows 8* garante a compatibilidade entre as duas plataformas (*smartphones* e *desktops*) algo muito aliciante para programadores.

Por fim, o sistema operativo *Bada* que manteve uma posição estável no mercado dos dispositivos móveis entre 2010 e 2011. O *Bada* foi desenvolvido pela *Samsung*, uma das empresas com maior preponderância na venda de *smartphones* em todo o mundo, sendo a plataforma proprietária dos dispositivos desta marca. É visto como um “oceano azul” no mercado dos dispositivos móveis, apostando em *smartphones* mais rentáveis na relação qualidade preço (GSMarena.com, 2012) (Microsoft, 2012) (Bada, 2011).

Os sistemas operativos móveis descritos em cima estão associados a ambientes de desenvolvimento definidos, o que leva a maioria dos programadores a desenvolver aplicações para plataformas específicas e por conseguinte sistemas operativos específicos. Este facto levou ao aparecimento de locais que possibilitam a disponibilização e compra dessas aplicações, as chamadas lojas virtuais. O *iPhone*, *smartphone* da *Apple*, tem mais de 500000 aplicações disponíveis através do *iTunes* e

da sua *App Store*¹⁶. Para dispositivos equipados com *Android*, entre outros, existe o *Google Play*¹⁷ com mais de 600000 aplicações disponíveis, utilizadores de *Symbian* têm acesso a aplicações móveis através da *Ovi Store*¹⁸. Equipamentos com *Blackberry OS* podem obter aplicações através da *Blackberry Webstore*¹⁹. Finalmente, os utilizadores do *Windows Phone 7* podem pesquisar e adquirir aplicações utilizando o *Windows Marketplace*²⁰.

Existe outro paradigma no que diz respeito à disponibilização de aplicações no mercado móvel ligado ao desenvolvimento de aplicações para *Web* e/ou através de tecnologias *Web*. Neste caso as aplicações, denominadas aplicações móveis web, são acedidas a partir do *browser* existente no *smartphone*. O desenvolvimento deste tipo de aplicações é baseado em três tecnologias: *HTML*, *CSS* e *Javascript*.

Este paradigma apresenta um conjunto de vantagens relativamente ao desenvolvimento de aplicações nativas, como: ser multiplataforma (é acedido a partir do *browser*), têm um tempo de desenvolvimento mais rápido (graças ao desenvolvimento ser feito através de tecnologias *Web* que muitos programadores já conhecem), são mais baratas ou fáceis de manter - visto não ser necessário realizar as actualizações para diversos tipos de dispositivos - e têm um acesso mais simples ou ubíquo já que não é necessário descarregar a aplicação, basta introduzir o *URL* de acesso à mesma.

Por outro lado, também existem um conjunto de desvantagens das aplicações móveis *web* face às aplicações nativas. Os *browsers mobile* ainda têm algumas limitações face aos *browsers desktop*. Também, existe a limitação deste tipo de aplicações não conseguirem aceder a funcionalidades dos dispositivos, como câmara ou *GPS*. Ainda, é necessário existir uma ligação à Internet para ter acesso aos dados, o que pode ser um factor negativo tanto por questões de performance (*site* não estar disponível ou existir demora na devolução dos conteúdos), como por questões de custos associados à transferência de dados.

Ainda, resultado das limitações das duas abordagens descritas em cima, nasceu um terceiro paradigma de desenvolvimento para *smartphones* denominado aplicações híbridas. Neste caso, a abordagem híbrida procura misturar o que as duas anteriores têm

¹⁶ <http://www.apple.com/iphone/apps-for-iphone/>

¹⁷ <https://play.google.com/store>

¹⁸ <http://store.ovi.com/>

¹⁹ <http://appworld.blackberry.com/webstore/>

²⁰ <http://marketplace.windowsphone.com/>

de melhor, com as aplicações a serem disponibilizadas a partir do *browser* e a disponibilizarem o acesso a funcionalidades dos dispositivos através de uma aplicação nativa “embutida” na aplicação principal (combinação de *standards Web* com código nativo) (Lionbridge, 2012).

Quando se discute o desenvolvimento de aplicações móveis, a principal preocupação é a experiência proporcionada ao utilizador. O sucesso de uma aplicação está naturalmente ligado à satisfação dos seus utilizadores. Num artigo comparativo entre desenvolvimento *web* versus nativo de aplicações (Charland, et al., 2011) os autores defendem que o sucesso de um projecto de *software* está dependente de duas categorias principais: contexto e implementação.

O contexto representa os elementos que não podem ser mudados ou controlados por quem desenvolve as aplicações, ou seja, elementos não programáveis dos dispositivos. Por um lado temos *hardware* que difere ao nível do *display* (tamanho físico do dispositivo, resolução do ecrã), *input* (*touchscreen*, teclado físico) ou capacidade (capacidade de processamento, armazenamento). Existem também as convenções associadas a cada plataforma. A convenção definida para aceder ao menu de contexto ou mesmo à simples funcionalidade de regredir de um ecrã para o anterior difere totalmente entre plataformas. Por fim, não menos importante, temos o ambiente. Tentar perceber em que condições os utilizadores irão usufruir das aplicações - à luz do dia ou durante a noite, de pé ou sentados, parados ou em movimento – é um desafio visto serem inúmeras as situações que podem ocorrer.

Os elementos controláveis nas aplicações simbolizam a implementação. Podemos definir diversos elementos, tais como performance, *design* e integração com funcionalidades dos dispositivos. O elemento com maior importância, se pensarmos em ambientes móveis, é a performance que pode ser quantificada através de duas métricas: latência e tempo de execução. Problemas de performance ao nível da latência resultam da necessidade de descarregar aplicações, ou dos consumos associados à transferência de dados através da rede. Por outro lado, a latência está também ligada à inicialização do código que depois de estar em memória tem de ser compilado, o que pode gerar alguns problemas de performance. Relativamente ao tempo de execução, este problema está directamente ligado ao desenvolvimento *web*, onde é realizada a interpretação de código (*Javascript*) e quanto mais código for necessário interpretar maior é o tempo de execução. Este argumento é contraposto por quem apoia o desenvolvimento *web*

Integração e divulgação de informação desportiva em redes sociais através de dispositivos móveis

argumentando que o código escrito em *Javascript* corre em múltiplos dispositivos sem necessidade de ser adaptado (umas pequenas mudanças eventualmente), enquanto no desenvolvimento de aplicações nativas é necessário ter em consideração que o código desenvolvido terá de ser modificado consoante as plataformas e os dispositivos, sendo igualmente o processo de manutenção muito mais trabalhoso.

3 Desenho

Neste capítulo apresenta-se o modelo de arquitectura do sistema desenvolvido para gerir informação desportiva e apresentar essa informação através de uma aplicação móvel. Serão descritas as ideias relativas aos três principais processos existentes na parte de *back-office*: agregação, categorização e implementação, sendo também explicadas as funcionalidades existentes na aplicação móvel.

3.1 Arquitectura do Sistema (Modelo)

A arquitectura do sistema, exibida na Figura 3, pode ser dividida em três blocos, dois deles externos – as redes sociais e os dispositivos móveis – e outro interno, o bloco central, que simboliza o conjunto de metodologias, processos e funcionalidades desenvolvidas. Assim, a arquitectura do bloco central pretende reflectir a existência de uma plataforma integrada de agregação, categorização e indexação dos conteúdos recolhidos a partir dos agentes externos, e ainda o desenvolvimento de uma aplicação móvel que permite a exibição dos dados recolhidos na plataforma, possibilitando igualmente alimentar a própria plataforma com conteúdos.

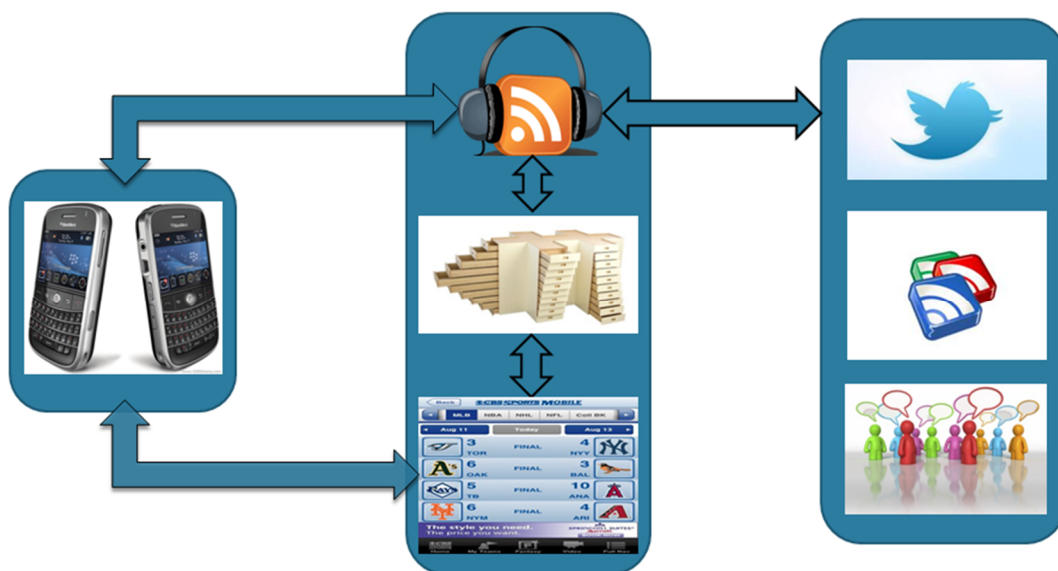


Figura 3 - Arquitectura do sistema

3.1.1 Agregação de Conteúdos

A escolha relativa às fontes de informação foi realizada tendo por base vários factores. Entendeu-se a escolha do *Twitter* como fonte principal dos conteúdos recolhidos essencialmente devido a duas perspectivas: das duas redes sociais mais importantes, *Twitter* e *Facebook*, o *Twitter* tendo um carácter mais minimalista e informativo, especialmente quando possuem marcadores ou *hashtags* (explicados de seguida), fundamentado por (Ramage, et al., 2010) quando distinguiram os *tweets* em categoria substância ou estado. Por outro lado, os utilizadores do *Facebook* na sua maioria definem os seus perfis como privados, o que poderia tornar mais difícil a recolha de informação relevante e em quantidades significativas.

Para além do *Twitter* a própria aplicação móvel funcionará como uma fonte de informação desportiva.

O modelo definido para o processo de agregação de conteúdos é genérico, embora neste trabalho tenha sido adaptado para o caso específico do *Twitter*. A plataforma realiza a agregação de conteúdos de carácter desportivo, nesse sentido, foi pensado um mecanismo de recolha associado ao desporto, baseado numa categoria principal – modalidade – e em três subcategorias – equipa, jogador e evento. A associação entre categoria principal e as subcategorias realiza-se através da definição de equipas, jogadores ou eventos desportivos em função da modalidade a que pertencem. Para cada subcategoria serão definidos marcadores ou *hashtags* usados na pesquisa de informação. Os marcadores são uma funcionalidade característica do *Twitter* e têm um papel fulcral na categorização de mensagens, ao termos uma palavra precedida pelo símbolo “#” significa que essa palavra é uma palavra-chave ou tópico da mensagem. Se clicarmos na palavra marcada podemos ver todos os *tweets* que fazem parte da mesma categoria, ou seja, que foram marcados por essa “*hashtag*” (Twitter, 2011). A ideia base desta dissertação pressupõe um paradigma diferenciador de agregação de informações desportivas, no sentido de serem baseadas em conteúdos gerados por utilizadores, utilizando as redes sociais e a própria aplicação móvel como fontes, ao contrário das aplicações móveis desportivas habitualmente disponíveis no mercado português que estão associadas a entidades desportivas. Em paralelo com esta perspectiva, decidiu-se que a panóplia de conteúdos, neste caso marcadores, usados nas pesquisas partiriam dos próprios utilizadores da aplicação móvel, ou seja, as palavras-chave usadas nas pesquisas são o conjunto de marcadores de interesse para esses utilizadores. Assim, não

só a visualização de conteúdos é feita em função dos interesses dos utilizadores da plataforma móvel como também a pesquisa e posterior agregação, fazendo com que o processo de agregação seja realizado em função dos utilizadores e dos seus interesses no momento.

3.1.2 Categorização de Conteúdos

Após o processo de agregação, a etapa seguinte é a categorização dos conteúdos agregados. Na Figura 4 podemos ver um diagrama de actividades que resume a metodologia utilizada.

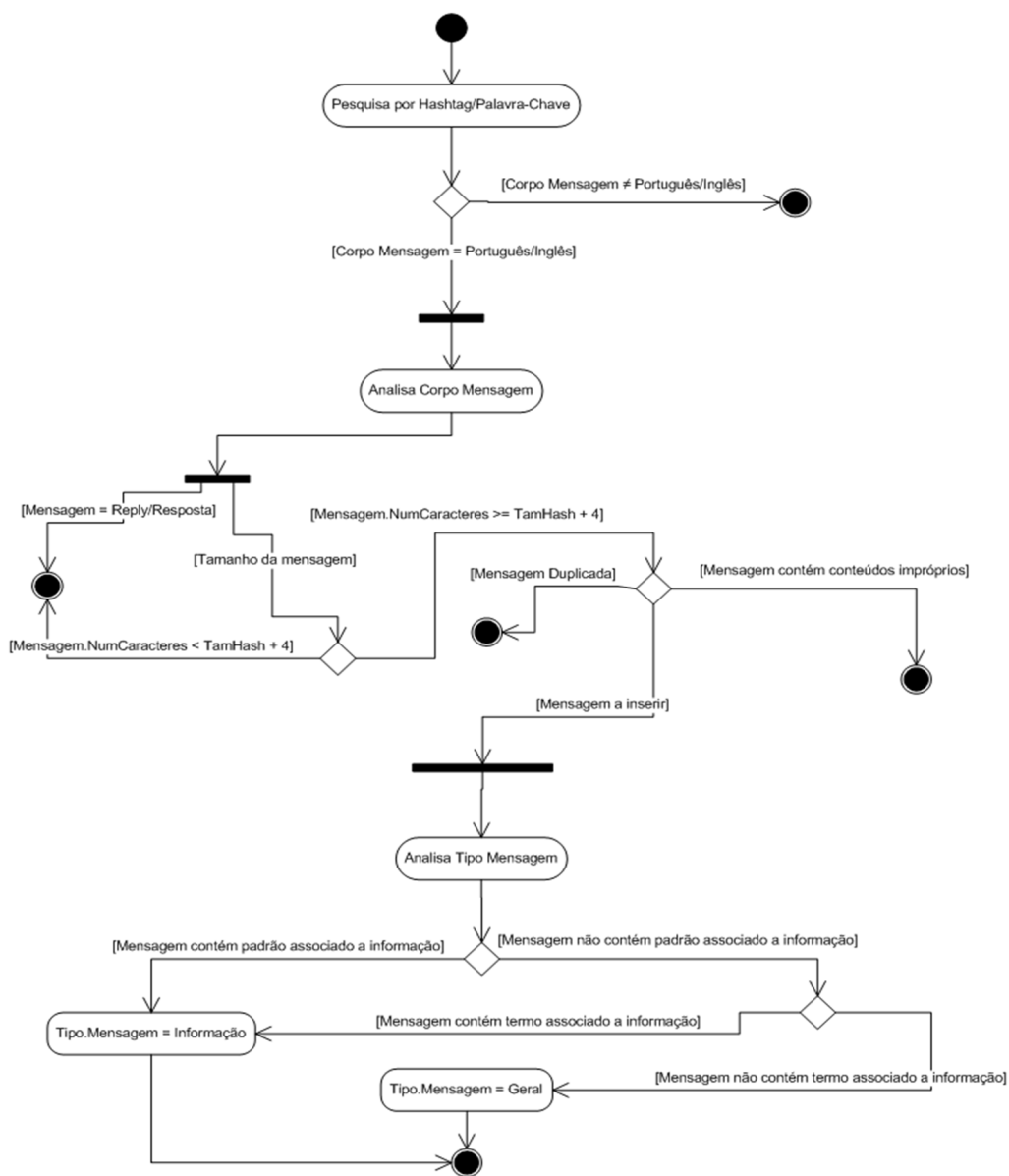


Figura 4 - Categorização de mensagens

O processo de categorização segue um conjunto de etapas para validar as mensagens a guardar no sistema, com o objectivo de agregar apenas aquelas que farão sentido apresentar aos utilizadores da aplicação móvel. Foi tomada a decisão de agregar conteúdos em duas línguas (português e inglês) essencialmente por este ser um sistema pensado para utilizadores portugueses e pelo inglês ser uma das principais línguas utilizadas no *Twitter*, e como se sabe a língua universal. Também, muitos utilizadores portugueses do *Twitter* usam a língua inglesa para criarem conteúdos.

As etapas a ultrapassar na categorização são as seguintes:

- a) Língua da mensagem: caso a mensagem não se encontre na língua portuguesa ou inglesa é imediatamente descartada;
- b) Mensagem do tipo resposta: no *Twitter* existe a funcionalidade de realizar uma resposta à mensagem de outro utilizador, ao clicar na opção “*Reply*” sobre o *tweet* específico. Neste caso é criada uma nova mensagem com a particularidade de ser colocado o nome do utilizador a quem está a ser feita a resposta precedido do símbolo “@” (Twitter, 2011). Considerou-se que as mensagens deste tipo não seriam do interesse comum, ou não conteriam informação relevante para a comunidade, portanto mensagens do tipo resposta são também descartadas;
- c) Tamanho da mensagem: com vista a eliminar mensagens habitualmente enquadradas no conceito de “*spam*”, ou seja, mensagens sem conteúdo relevante para os utilizadores, é feita uma contagem de caracteres da mensagem para evitar a recolha de conteúdos que não atinjam um número mínimo. Decidiu-se que o valor mínimo, seria o tamanho da “*hashtag*” pesquisada mais três caracteres. A decisão foi tomada neste sentido porque o número mínimo de caracteres associado a padrões específicos que revelam informação numa mensagem é de três (exemplo “*#Benfica 5*” que indica cinco minutos de jogo);
- d) Duplicações: o processo de remoção de duplicados não se resume à verificação de mensagens totalmente iguais. No *Twitter*, existe uma funcionalidade específica denominada “*Retweet*”, usada quando um utilizador deseja partilhar com os seus seguidores uma mensagem do seu interesse (Twitter, 2011). As mensagens “retuitadas” ficam com um padrão específico comparativamente à mensagem original, acrescentando no início da mesma o padrão “*RT @nome_utilizador*”. Existe ainda a particularidade de algumas aplicações permitirem ao utilizador que efectua um *retweet* adicionar texto à

mensagem, o que levou à definição de uma regra específica para este caso: mensagens “retuitadas” com texto antes do padrão específico são comparadas com a mensagem original como qualquer outra mensagem, mensagens que sejam apenas o *retweet* da original, estando a mensagem original já armazenada no sistema, são descartas. Para o processo de remoção de mensagens parcialmente idênticas definiu-se como similaridade mínima entre mensagens um coeficiente de 80%;

e) Conteúdos impróprios: no sentido de garantir que as mensagens recolhidas e armazenadas no sistema não continham linguagem imprópria, foram definidos dois dicionários de termos, em português e inglês, associados a esse tipo de linguagem e conforme a língua associada à mensagem verifica-se o conjunto de termos específicos, descartando a mensagem no caso da mesma conter um desses termos;

f) Tipo da mensagem: o processo final de categorização das mensagens é composto por dois momentos: inicialmente faz-se a verificação se a mensagem contém algum padrão associado a informação específica (tempo de jogo, resultado actual) e em caso positivo a mensagem fica categorizada com o tipo informação. Caso não seja encontrado nenhum padrão de informação definiram-se dicionários de termos desportivos associados a eventos que ocorrem nas diferentes modalidades. Conforme a língua da mensagem e a modalidade associada à “*hashtag*” existente na mensagem são verificados todos os termos do dicionário específico associado a essa modalidade para mensagens nessa língua. As duas modalidades testadas no âmbito deste trabalho foram o futebol e o ténis. O futebol foi escolhido porque é o desporto mais visto e comentado pela grande maioria dos portugueses. A escolha do ténis deveu-se ao facto de existir um conjunto alargado e importante de eventos tenísticos ao longo do ano, importante para testar a plataforma e validar a mesma, e por ser um desporto com características totalmente diferentes do futebol, importante para averiguar a heterogeneidade da plataforma.

3.1.3 Indexação de Conteúdos

O assegurar da manutenção e integridade dos dados associados ao sistema é realizado segundo duas formas distintas para as mensagens e restantes actores envolvidos.

Na Figura 5, podemos observar o diagrama conceptual com as principais classes do sistema de informação:

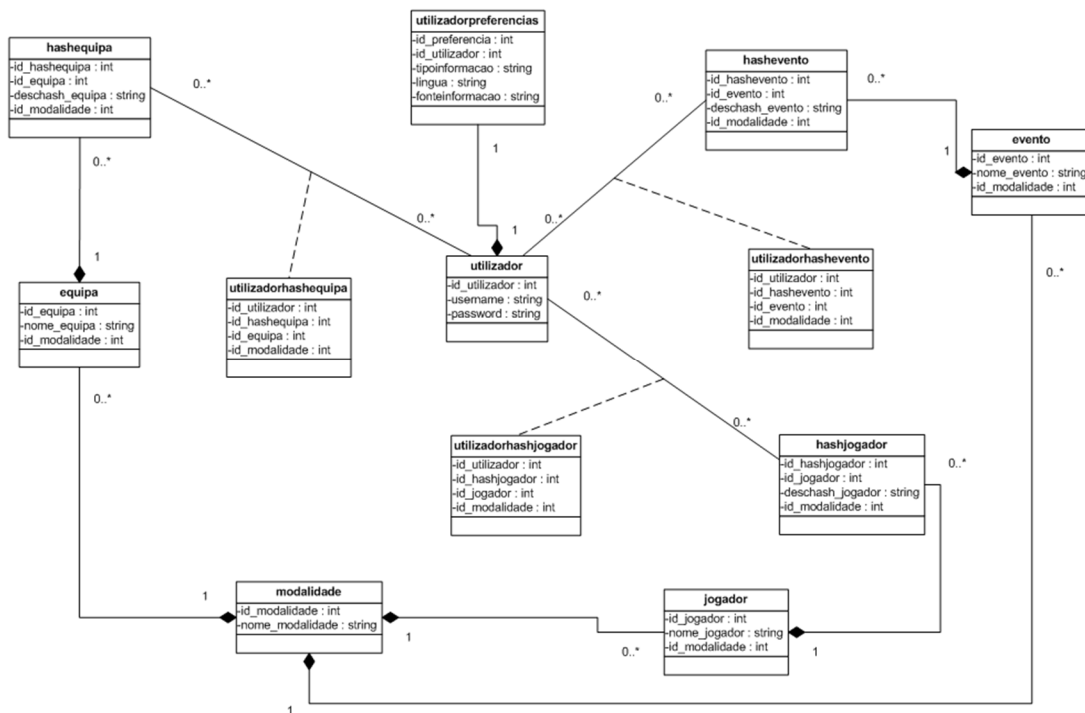


Figura 5 - Diagrama Conceptual

O diagrama apresentado revela o modelo idealizado para a manutenção de informação desportiva. Tal como foi referido no processo de agregação, equipas, jogadores e eventos são definidos em função de uma modalidade desportiva, podendo cada modalidade conter diversas equipas, jogadores ou eventos associados a si. Em função de cada equipa podemos definir vários marcadores (ex. Equipa: *Sport Lisboa e Benfica*; Marcadores/Hashtags: *#SLBenfica*, *#Benfica*), acontecendo o mesmo para jogadores ou eventos. Os marcadores associados às subcategorias equipa, jogador e evento ficam definidos nas tabelas “hashequipa”, “hashjogador” e “hashevento”. Cruzando a informação das tabelas dos marcadores com a tabela “utilizador”, conseguimos definir o conjunto de marcadores de equipas, jogadores e eventos sobre os quais os utilizadores pretendem saber informações. Essa informação encontra-se nas tabelas “utilizadorhashequipa”, “utilizadorhashjogador” e “utilizadorhashevento”. É

ainda possível definir um conjunto de preferências para cada utilizador acerca da informação recebida.

São três os conceitos a definir:

- a) Tipo de informação: tal como foi explicado no processo de categorização, a informação está dividida em dois tipos: informação e geral. O utilizador define se quer receber os dois tipos ou apenas um deles;
- b) Língua: é dada a possibilidade do utilizador receber informações apenas na língua portuguesa, apenas na língua inglesa ou ambas;
- c) Fonte de informação: o sistema agrega informações a partir do *Twitter* e da própria aplicação móvel. Neste caso, o utilizador pode escolher apenas uma das fontes ou ambas.

Relativamente às mensagens, a metodologia usada será diferente. Com recurso às funcionalidades de uma ferramenta de indexação e pesquisa textual chamada *Lucene* (esta ferramenta será descrita no capítulo seguinte), serão guardadas as principais características de cada mensagem: autor, fotografia do autor, marcador/tópico da mensagem, corpo da mensagem, língua (português ou inglês), data, tipo (informação ou geral) e fonte (*Twitter* ou aplicação móvel).

3.1.4 Aplicação Móvel

O desenvolvimento da aplicação móvel reflectirá a existência de quatro funcionalidades principais, existindo ainda um ecrã de instruções que pode ser visto como o manual de utilização da aplicação.

O conjunto de funcionalidades é o seguinte:

- a) *Timeline* de Mensagens: permite ao utilizador visualizar as mensagens relativas aos seus conteúdos de interesse. Para além de visualizar autor, fotografia do autor, mensagem, data da mensagem e fonte de todas as mensagens, o utilizador poderá actualizar os conteúdos apresentados neste ecrã. A actualização dos conteúdos estará dependente do servidor;
- b) Inserir Mensagem: permite ao utilizador enviar mensagens sobre os seus conteúdos favoritos. Estas mensagens estarão divididas pelos três tipos de actores favoritos definidos na funcionalidade “Gerir Favoritos”, sendo necessário definir “*hashtag*” e língua associada à mensagem;

- c) Gerir Favoritos: permite ao utilizador definir as suas equipas, jogadores e eventos favoritos, e ainda os marcadores ou “*hashtags*” associados a estes actores;
- d) Gerir Preferências: permite ao utilizador identificar os tipos de informação (informação e/ou geral), língua (inglês e/ou português) e fontes de informação (*Twitter* e/ou *SocialSports* – nome escolhido para a aplicação móvel desenvolvida) que deseja visualizar no ecrã que exhibe as mensagens.

4 Implementação

Neste capítulo descreve-se a implementação do modelo apresentado no capítulo anterior. Inicialmente, será dada uma explicação sucinta dos objectivos de implementação da plataforma de gestão de conteúdos desportivos definindo-se a sua arquitectura, de seguida serão descritas as ferramentas que permitiram cumprir as necessidades da plataforma, revelando-se também a tecnologia associada ao desenvolvimento da aplicação móvel. Por fim, explica-se a estrutura de implementação e as funcionalidades cumpridas através das ferramentas anteriormente descritas.

4.1 Arquitectura da Plataforma de Gestão de Conteúdos Desportivos

A arquitectura do modelo apresentada no capítulo de Desenho (Figura 3), resulta em termos de implementação da plataforma de recolha dos conteúdos desportivos no conjunto de ferramentas e funcionalidades existentes na Figura 6. A figura demonstra a existência de três processos consecutivos desde a recolha ou agregação dos conteúdos desportivos, a categorização desses conteúdos e posterior indexação e/ou armazenamento dos dados relativos aos actores do sistema.

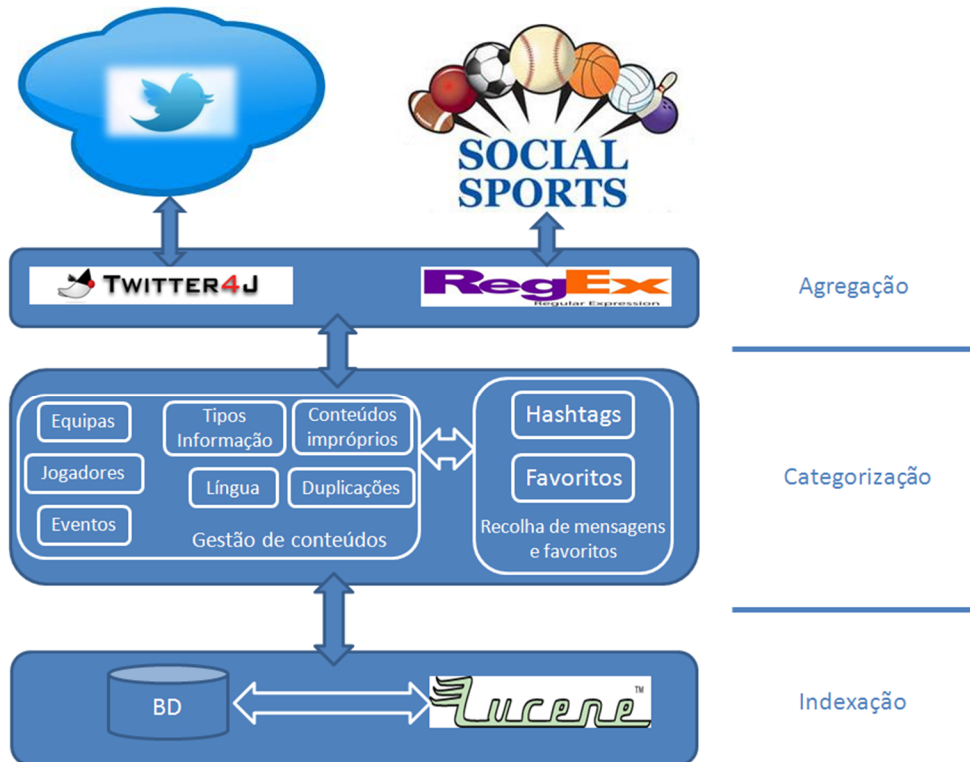


Figura 6 - Arquitectura da plataforma de gestão de conteúdos desportivos

O processo de agregação de conteúdos tem em conta as duas fontes de informação do sistema. Por um lado, a recolha de mensagens está ligada tanto à rede social *Twitter* como à aplicação móvel - “*Social Sports*”, sendo que a recolha de mensagens a partir do *Twitter* é feita recorrendo à ferramenta *Twitter4J* e a recolha de mensagens a partir da aplicação móvel é feita através da utilização de expressões regulares. Por outro, a recolha de informação não referente a mensagens – equipas, jogadores, eventos e preferências – é em exclusivo realizada a partir da aplicação móvel. A explicação das ferramentas e processos de agregação estará nos próximos subcapítulos.

O processo de categorização difere igualmente no que respeita a mensagens e restantes actores. Relativamente às mensagens, o processo de categorização é mais complexo e passa pela avaliação das suas características, verificando língua, tipo de mensagem, duplicações ou existência de conteúdos impróprios nas mesmas. Esta categorização permite seleccionar apenas mensagens de interesse a integrar na plataforma. No que diz respeito a favoritos – equipas, jogadores e eventos – e preferências recolhidos a partir da aplicação móvel é verificada a necessidade de inserir ou actualizar os dados recolhidos na plataforma.

O processo final de armazenamento ou indexação é também ele feito em função dos diferentes actores do sistema e por isso dividiu-se o processo em duas ferramentas distintas. Isto é, decidiu-se atribuir um carácter mais volátil às mensagens recolhidas na plataforma, ou seja, as mensagens mantidas pela plataforma de conteúdos desportivos são apenas as recolhidas a cada iteração de pesquisa, permitindo assim manter apenas aquelas mais recentes e de interesse para os utilizadores. Neste caso, foi usado um sistema de índices, através da ferramenta *Lucene* descrita na próxima secção. Por outro lado, e relativamente a favoritos e preferências, actores com informação menos volátil, foi desenhado um sistema de base de dados em *MySQL* para manter as suas informações.

De seguida são descritas as ferramentas de implementação da plataforma e da aplicação móvel.

4.2 Ferramentas de Implementação

As ferramentas de implementação dividem-se em dois tipos: as necessárias para desenvolver a plataforma de agregação de conteúdos desportivos, ou seja, ferramentas de *back-end*, e as ferramentas utilizadas na construção da aplicação móvel disponibilizada aos utilizadores, no *front-end*.

4.2.1 Ferramentas de *back-end*

As ferramentas de *back-end* utilizadas nesta dissertação possibilitaram resolver e/ou facilitar problemas específicos levantados pelo desenvolvimento deste trabalho, tais como integração do *Twitter* com *Java*, detecção de padrões em informação não estruturada e indexação de informação, entre outros.

4.2.1.1 *Twitter4J*

Tendo em conta que se pretende desenvolver uma plataforma baseada na linguagem de programação *Java*, tornou-se necessário identificar as ferramentas existentes, neste caso bibliotecas, que irão facilitar a recolha de dados através do *Twitter*. No site de APIs do *Twitter*²¹, consultado numa fase inicial desta dissertação, estavam definidas as bibliotecas mais importantes de ligação através do *Java*, onde se destacavam: *twitter4j*²², *java-twitter*²³ e *jttwitter*²⁴. A escolha recaiu sobre a biblioteca *Twitter4J* por ser a que maior consenso recebia por parte da comunidade, por ter um grupo alargado de contribuidores e ser a que demonstrava uma continuidade mais sustentada. Esta biblioteca foi inicialmente desenvolvida por *Yusuke Yamamoto* e permite realizar tarefas como: ligação do utilizador à sua conta pessoal, actualizações de conta através da própria API, obter todas as actualizações de outros utilizadores que uma conta segue no *Twitter* (acedendo às últimas mensagens na sua *timeline*) e realizar pesquisas de *tweets* segundo *queries*, permitindo identificar, por exemplo, autor e corpo dessas mensagens (*Twitter4J*, 2007).

²¹ <http://dev.twitter.com/pages/libraries#java>

²² <http://twitter4j.org/en/index.html>

²³ <http://code.google.com/p/java-twitter/>

²⁴ <http://www.winterwell.com/software/jttwitter.php>

4.2.1.2 Expressões Regulares

Uma das funcionalidades específicas desenvolvidas na plataforma de gestão dos conteúdos desportivos foi a categorização das mensagens. Uma das formas de categorizar as mensagens como informativas é a detecção de padrões associados a resultados ou tempos de jogo. As expressões regulares permitem identificar padrões específicos numa cadeia de caracteres através de um método flexível de reconhecimento de palavras ou grupos de palavras num texto (Techopedia, 2010). Fazem parte da plataforma *Java* desde a versão 1.4, e estão contidas no pacote *java.util.regex* (Oracle, 2011).

4.2.1.3 Apache Lucene

A necessidade de criar índices com informação relativa às mensagens recolhidas pela plataforma de conteúdos desportivos, incentivou a procura de uma ferramenta com essa capacidade. (Messina, et al., 2008) enfrentaram um problema idêntico servindo-se da biblioteca *Apache Lucene*. Esta biblioteca que segue o movimento de desenvolvimento em código aberto, totalmente escrita em *Java*, pertence à fundação *Apache Software Foundation (ASF)*. Realiza pesquisa e indexação de alta performance em recursos textuais. A nível de indexação possui um conjunto de funcionalidades interessantes, tais como a capacidade de indexar 20 MB por minuto (num *Pentium M* a 1.5 GHz) e o tamanho dos índices gerados ocupa apenas 20 a 30% comparativamente ao tamanho do texto indexado. Consegue realizar pesquisas por classificação (devolve primeiro os melhores resultados), vários tipos de consultas (por frases, proximidade, intervalo, etc) e pesquisa textual (ex. título, autor) (Apache Software Foundation, 2011).

4.2.2 Ferramentas de front-end

As ferramentas de *front-end* permitiram desenvolver a aplicação móvel. A definição da metodologia utilizada e ferramentas necessárias é explicada de seguida.

4.2.2.1 Sencha Touch

O desenvolvimento da aplicação móvel foi inicialmente pensado segundo a perspectiva de desenvolvimento de uma aplicação nativa. As hipóteses colocadas seriam desenvolver para *Android*, *iOS* ou *Blackberry*. No entanto, e com o evoluir das tecnologias associadas ao desenvolvimento de aplicações para dispositivos móveis

associadas a tecnologias *Web*, com uma curva de aprendizagem mais fácil e rápida, decidiu-se recorrer a uma *framework* denominada *Sencha Touch*. Baseada em *standards Web* como *HTML5*, *CSS3* e *Javascript* esta *framework* formou-se a partir da combinação de três outros projectos : *Ext JS*²⁵, *jQTouch*²⁶ e *Raphael*²⁷. Especialmente desenvolvida para dispositivos móveis com interfaces *touch*, permite desenvolver aplicações móveis *Web* que se assemelham a aplicações nativas. Possui um conjunto de funcionalidades muito interessantes ao nível dos componentes para criação de interfaces gráficas, efeitos de transições *CSS*, gestão de eventos por toque e gestão de dados (Sencha Inc, 2011) (Sencha Inc, 2011). Na fase inicial de implementação desta dissertação a última versão estável da biblioteca *Sencha Touch* era a 1.1.0, sendo essa a versão utilizada até à conclusão do projecto. No entanto, já existe uma nova versão da *framework*, *Sencha Touch 2*, que sofreu várias alterações destacando-se a utilização de um novo sistema de classes associado ao *Ext JS 4* (Spencer, 2011) (Sencha Docs, 2011) (Spencer, 2011).

Uma das grandes vantagens da utilização desta *framework* e das aplicações móveis *Web* é serem multiplataforma, sendo apenas necessário um *browser* para correr a aplicação. A única particularidade exigida ao *browser* é possuir um motor de renderização *WebKit* necessário para transformar o conteúdo em linguagem de marcadores lida pelo browser. O *Google Chrome* e *Safari* são exemplo de dois navegadores que utilizam este motor.

No âmbito desta dissertação e para teste da aplicação móvel foi usado o *Google Chrome* num computador *desktop* que permite interagir com os componentes através da utilização do rato, imitando a utilização dos dedos num dispositivo móvel para realizar acções de toque.

Por outro lado, também foi testada a aplicação a correr num *smartphone* de forma híbrida, isto é, embora a aplicação seja *Web*, através de um conjunto de *plug-ins* do *Eclipse* foi possível empacotar a aplicação e deixá-la apta a um mercado de aplicações ao estilo das aplicações nativas.

Neste caso, e por ter disponível um *smartphone* com sistema operativo *Android*, foi necessário proceder à instalação dos componentes necessários para correr a aplicação no *smartphone*:

²⁵ <http://www.sencha.com/products/extjs>

²⁶ <http://jqtouch.com/>

²⁷ <http://raphaeljs.com/>

- a) *Android SDK* – disponibiliza as ferramentas e bibliotecas necessárias para desenvolver e testar aplicações que correm em sistemas *Android*, permite por exemplo configurar dispositivos virtuais para correr as aplicações;
- b) *ADT Plug-in – Android Development Tools (ADT) plug-in* para o *IDE Eclipse* que disponibiliza um ambiente integrado para construção de aplicações *Android*;
- c) *MDS AppLaud PhoneGap Eclipse Plug-in – plug-in* para o *IDE Eclipse* que facilita o desenvolvimento de aplicações para *Android* ao integrar *frameworks* de interface de utilizador, como o *Sencha Touch* ou *jQuery Mobile*, com o *plug-in* da *framework Phonegap*²⁸.

4.3 Estrutura de Implementação da Plataforma de Gestão de Conteúdos Desportivos

A implementação da plataforma que gere os conteúdos desportivos segue uma estrutura específica composta por um conjunto de acções que permitem gerir tantos os conteúdos provenientes do *Twitter*, como aqueles agregados a partir da aplicação móvel. São ainda realizadas duas acções que possibilitam compor as preferências dos utilizadores e renovar os índices onde a informação é mantida.

A implementação da plataforma de gestão de conteúdos desportivos segue a estrutura exibida na Figura 7, em baixo, sob uma perspectiva geral. Existem duas macro-sequências exibidas na figura que representam duas interacções distintas, a primeira realizada entre a plataforma e o *Twitter*, a segunda entre a plataforma e a aplicação móvel. De seguida, são gerados os conteúdos de interesse do utilizador e renovados os índices. O processo geral, definido na figura por “*Main*”, é realizado segundo iterações onde se realizam as duas macro sequências e conjunto de processos associados às mesmas, seguidas da geração dos dados em função da preferência dos utilizadores e renovação dos índices.

²⁸ <http://phonegap.com/>

Main

Interacção Twitter:

- Pesquisa Equipas;
- Pesquisa Jogadores;
- Pesquisa Eventos.

Interacção Aplicação Móvel:

- Gestão Equipas;
- Gestão Jogadores;
- Gestão Eventos;
- Gestão Preferências;
- Gestão Hashtags Equipas;
- Gestão Hashtags Jogadores;
- Gestão Hashtags Eventos;
- Gestão Mensagens.

Agregação Conteúdos Preferência

Renovação Índices

Figura 7 - Estrutura de implementação da plataforma de gestão de conteúdos desportivos

Ou seja, o *Workflow* de processos é o seguinte:

- 1) Interacção com o *Twitter*:
 - a. Pesquisa Equipas: Agrega Mensagens Equipas → Categoriza Mensagens Equipas → Gere Índices Equipa;
 - b. Pesquisa Jogadores: Agrega Mensagens Jogadores → Categoriza Mensagens Jogadores → Gere Índices Jogadores;
 - c. Pesquisa Eventos: Agrega Mensagens Eventos → Categoriza Mensagens Eventos → Gere Índices Eventos.
- 2) Interacção com a aplicação móvel:
 - a. Gestão Equipas: Agrega Equipas → Verifica Equipas → Gere Equipas;
 - b. Gestão Jogadores: Agrega Jogadores → Verifica Jogadores → Gere Jogadores;
 - c. Gestão Eventos: Agrega Eventos → Verifica Eventos → Gere Eventos;

- d. Gestão Preferências: Agrega Preferências → Verifica Preferências → Gere Preferências;
 - e. Gestão Hashtags Equipas: Agrega Hashtags de Equipas → Verifica Hashtags de Equipas → Gere Hashtags de Equipas;
 - f. Gestão Hashtags Jogadores: Agrega Hashtags de Jogadores → Verifica Hashtags de Jogadores → Gere Hashtags de Jogadores;
 - g. Gestão Hashtags Eventos: Agrega Hashtags de Eventos → Verifica Hashtags de Eventos → Gere Hashtags de Eventos;
 - h. Gestão Mensagens: Agrega Mensagens → Verifica Novas Mensagens → Categoriza Mensagens → Gere Índices Mensagens;
- 3) Agregação Conteúdos Preferência: Verifica Opções de preferência → Agrega Dados de Preferência → Constrói Ficheiro com Dados Preferência;
- 4) Renovação Índices: Verifica Índices (*Twitter*) → Elimina Índices.

De seguida explica-se a implementação dos processos exibidos na Figura 7 e detalhados em cima.

4.3.1 Interacção *Twitter*

4.3.1.1 Pesquisa/Recolha de Mensagens

O processo de recolha ou pesquisa de mensagens/*tweets* provenientes da rede social *Twitter* é igual para equipas, jogadores e eventos. Uma das principais características nesta dissertação é a recolha de conteúdos ser baseada em “*hashtags*”. De acordo com este princípio e com a modelação definida para a plataforma existiam duas hipóteses: recolher informação de todas as “*hashtags*” existentes nas tabelas “*hashequipa*”, “*hashjogador*” e “*hashevento*”, ou recolher informação das tabelas “*utilizadorhashequipa*”, “*utilizadorhashjogador*” e “*utilizadorhashevento*” que dependem das tabelas anteriores e da tabela “*utilizador*” (ver Figura 5 – Diagrama Conceptual). A diferença reside no facto de se manter uma “*hashtag*” de equipa, jogador e evento nas tabelas “*hashequipa*”, “*hashjogador*” e “*hashevento*” mesmo que não exista qualquer utilizador com interesse nessa “*hashtag*”, enquanto as tabelas “*utilizadorhashequipa*”, “*utilizadorhashjogador*” e “*utilizadorhashevento*” contêm apenas “*hashtags*” de interesse dos utilizadores no momento actual. A decisão de manter esta divisão foi tomada porque qualquer utilizador pode decidir receber

informações relativas a uma “*hashtag*” e mais tarde cancelar essa subscrição. No entanto, no futuro esse ou outro utilizador pode voltar a definir interesse na mesma “*hashtag*”, tornando-se mais simples manter a “*hashtag*” nas tabelas associadas apenas a “*hashtags*” e realizar as operações de eliminação ou actualização nas tabelas associadas a utilizadores. Assim, e também pelo facto de a plataforma ter o intuito de fornecer apenas conteúdos mais recentes sem perspectivas de armazenar um histórico alargado de informação, a decisão final passou por realizar pesquisas apenas pelas “*hashtags*” existentes nas tabelas “*utilizadorhashequipa*”, “*utilizadorhashjogador*” e “*utilizadorhashevento*”, permitindo poupar em larga escala o processamento associado a esta tarefa.

Neste processo foi necessário compreender a melhor forma de interagir com a *API* do *Twitter* para recolha de informação. A *API* do *Twitter* é composta por três partes: duas *REST APIs* (*REST API* e *Search API*) e uma *Streaming API*. A primeira decisão prendeu-se com a utilização de uma *REST API* ou da *Streaming API*. Por ter maior simplicidade de aprendizagem e utilização, e por este projecto de dissertação ser uma prova de conceito e não um projecto complexo a larga escala decidiu-se utilizar uma *REST API* (140Dev, 2010). Tomada esta decisão foi necessário compreender entre as duas *REST APIs* qual utilizar; neste caso pesou o facto da *REST API* ter uma taxa limitativa de 150 pesquisas por hora, tendo por isso sido escolhida a *Search API*, que embora não permita avaliar tanta informação ao nível do utilizador, consegue garantir todas as informações necessárias relativamente às mensagens recolhidas e armazenadas na plataforma (Twitter, 2011) (Twitter, 2011).

Como referido anteriormente (ver subcapítulo “Ferramentas de Implementação”), foi utilizada a biblioteca *Twitter4J* que permite integrar aplicações desenvolvidas em Java com o *Twitter*. No processo de agregação/pesquisa de conteúdos através desta biblioteca as principais classes utilizadas foram:

- 1) *Twitter*: interface que estende o interface *Search Methods* e permite através do método ‘*search*’ realizar a pesquisa;
- 2) *Query*: classe de dados que representa uma *query* a pesquisar;
- 3) *QueryResult*: interface de dados que representa a resposta a uma pesquisa;
- 4) *Tweet*: classe de dados que representa um *tweet* na resposta a uma pesquisa.

Exemplos:

a) Utilização da classe *Query*:

```
Query query = new Query(hashtag).page(i);
query.rpp(j);
```

Neste exemplo instanciou-se a classe *Query*, definindo-se a pesquisa em função de uma “*hashtag*” (pode ser qualquer “*hashtag*” de equipa, jogador ou evento). De notar o método ‘*page*’ que indica o número de páginas a retornar na pesquisa. Em função da instância ‘*query*’ da classe *Query* o método ‘*rpp*’ permite definir o número de *tweets* a retornar por página. Para testar a plataforma decidiu-se recolher 150 novos *tweets* por iteração, repartindo-se a pesquisa por 2 páginas com 75 *tweets* cada (i=2 e j=75 no exemplo em cima).

b) Utilização dos interfaces *QueryResult* e *Twitter*:

```
TwitterFactory twitterFactory = new TwitterFactory();
Twitter twitter = twitterFactory.getInstance();
QueryResult qresult = twitter.search(query);
```

Criada uma instância do interface *Twitter*, através do método ‘*search*’ conseguimos realizar uma pesquisa em função da *query* definida no primeiro exemplo. O resultado da pesquisa é um interface de dados *QueryResult*.

c) Utilização do interface *QueryResult* e da classe *Tweet*:

```
List<Tweet> listaTweets = qresult.getTweets();
Iterator<Tweet> itTweet = listaTweets.iterator();
while (itTweet.hasNext()){
    Tweet tweet = itTweet.next();
    ...
}
```

O método ‘*getTweets*’ da instância ‘*qresult*’ criada para o Interface *QueryResult* alberga uma lista de *tweets*, podemos iterar essa lista e obter informações relativas a cada *tweet* recolhido. Para cada instância ‘*tweet*’ da classe *Tweet* temos:

- 1) *tweet.getIsoLanguageCode()*: devolve a língua associada à mensagem;
- 2) *tweet.getFromUser()*: devolve o username do utilizador que escreveu a mensagem;

- 3) *tweet.getProfileImageUrl()*: devolve a fotografia do utilizador que escreveu a mensagem;
- 4) *tweet.getText()*: devolve o corpo da mensagem;
- 5) *tweet.getCreatedAt()*: devolve a data da mensagem.

4.3.1.2 Categorização de Mensagens

O processo de categorização de mensagens é composto por seis etapas, sendo que as cinco primeiras servem para eliminar mensagens que não cumprem um conjunto de requisitos definidos e a última é a etapa onde se processa a categorização real das mensagens recolhidas.

Identificando as etapas e respectivos processos de implementação temos:

- 1) Verificação da língua associada à mensagem: este processo está directamente relacionado com o processo de agregação, já que a língua é obtida a partir da informação fornecida pelo método *'getIsoLanguageCode()'* da classe *Tweet* da biblioteca *Twitter4J*. Este método devolve o código de linguagem, código composto por duas letras, associado ao perfil/mensagem do utilizador. O conjunto de códigos disponíveis é público (Iana, 2009), sendo o código para a língua inglesa as letras *'en'* e o código para a língua portuguesa as letras *'pt'*. Qualquer mensagem recolhida com código diferente aos dois mencionados é descartada;
- 2) Verificação de mensagem tipo resposta: como foi referido anteriormente (capítulo de Desenho), este é um tipo de mensagens/*tweets* específicos da rede social *Twitter*. A característica essencial é que a mensagem no seu início é composta pelo *username* do utilizador para quem é dirigida a mensagem precedido do símbolo '@'. Assim, nesta etapa é simplesmente feita a verificação do primeiro carácter da mensagem, se corresponder ao símbolo '@' a mensagem é descartada;
- 3) Verificação do tamanho da mensagem: nesta etapa são verificados tamanho de mensagem e *"hashtag"*. Se o tamanho total da mensagem for inferior à soma do tamanho da *"hashtag"* com três caracteres a mensagem é descartada. Esta decisão foi tomada porque alguns utilizadores podem, por exemplo, decidir escrever uma mensagem simplesmente com a própria *"hashtag"*, mensagem sem qualquer interesse para a comunidade. A decisão de colocar a restrição a partir da soma do tamanho da *"hashtag"* com três

caracteres foi devido a não existir qualquer palavra-chave ou expressão regular com menos de três caracteres que caracterize a mensagem (referido na sexta e última etapa de categorização);

- 4) Verificação de mensagem duplicada: no processo de avaliação de mensagens duplicadas foi usado o coeficiente de similaridade de *Jaccard* (Bank, et al., 2008) definido como a medida estatística que mede a similaridade entre dois conjuntos (neste caso mensagens), ou seja, o coeficiente é definido pela cardinalidade da intersecção de dois conjuntos dividida pela cardinalidade da união dos mesmos:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

Figura 8 - Coeficiente de Similaridade de *Jaccard*

A classe *Jaccard* (Cohen, 2011) contida num pacote de *software* adicional denominado “*com.wcohen.secondstring*” permite realizar o processo que avalia duplicações através do método ‘*score*’ que recebe dois conjuntos de *strings* e compara-os. O valor desta comparação varia entre 0.0 e 1.0, sendo que quanto mais alto o valor maior é a similaridade das mensagens. Foi definido que o grau de similaridade a partir do qual duas mensagens eram consideradas duplicadas é 0.8 – foi escolhido este valor porque queremos considerar mensagens duplicadas não apenas as que sejam totalmente iguais mas também as que ultrapassem um grau específico de similaridade e noutros estudos anteriormente realizados com recurso ao coeficiente de *Jaccard* foi esse o grau definido (Eraç, et al., 2007). Neste processo foi verificado um caso específico para as mensagens recolhidas a partir do *Twitter*, que pode ser explicado pelo seguinte exemplo:

1ª Mensagem: “Benfica vence Gil Vicente”;

2ª Mensagem: “RT @utilizador: Benfica vence Gil Vicente”;

3ª Mensagem: “Vitória por 2-0!! RT @utilizador: Benfica vence Gil Vicente”.

A comparação entre 1ª e 2ª mensagens e entre 1ª e 3ª mensagens é diferente. Enquanto a 2ª mensagem corresponde a um *retweet* sem acrescentar qualquer informação antes do padrão *retweet*, a 3ª mensagem acrescenta texto antes do mesmo com informação que pode ser importante. A

plataforma no caso da 2ª mensagem, ao fazer a comparação retira o conjunto de caracteres associado à operação *retweet*, fazendo com que as mensagens sejam totalmente iguais para o exemplo definido em cima (coeficiente de similaridade = 1.0), já no caso da 3ª mensagem, por conter texto antes do padrão *retweet* a comparação irá englobar toda a mensagem.

Um conjunto de experiências para verificar mensagens duplicadas vai ser demonstrado no capítulo de Testes e Resultados.

- 5) Verificação de mensagem com conteúdos impróprios: esta etapa está directamente relacionada com a língua de cada mensagem. Após passarem a etapa de duplicação cada mensagem percorre um dicionário de termos na língua de origem, existindo por isso um dicionário de termos em português e outro em inglês. Os dicionários de termos foram construídos tendo por base duas acções: primeiro foi feito um levantamento das palavras mais conhecidas menos próprias de cada língua, depois foi aproveitado um momento específico associado ao futebol chamado mercado de transferências. No último dia do mercado de transferências de Inverno, que na maioria dos países europeus ocorre no dia 31 de Janeiro, existem muitas mexidas nos clubes que deixam os adeptos dos clubes respectivos totalmente eufóricos. Nesse dia, através da pesquisa pelo nome dos jogadores e equipas envolvidos nas transferências mais mediáticas conseguiu-se completar o dicionário com um conjunto mais alargado de referências/palavras com sentimentos de indignação.
- 6) Análise do tipo de mensagem: caso a mensagem ultrapasse todas as cinco fases eliminatórias anteriores são efectuados dois processos que permitem categorizar a mensagem. Antes de identificar os dois processos, é necessário esclarecer que as mensagens podem ser de dois tipos: geral ou informação. Esta categorização está dependente dos seguintes dois processos:
 - a. Identificação de padrões de informação específicos: para este processo recorreu-se a expressões regulares, já explicadas anteriormente (ver ‘Ferramentas de Implementação’), que funcionam essencialmente como um conjunto de caracteres especiais que permitem descrever um padrão de pesquisa. Podemos encontrar no pacote “*java.util.regex*” as classes *Pattern* (Oracle, 2011) e *Matcher*

(Oracle, 2011) essenciais para desenvolver esta tarefa. A expressão regular especificada como uma “*string*” tem de ser inicialmente compilada como uma instância da classe *Pattern*. Esse padrão pode depois ser usado na criação de um objecto da classe *Matcher* a partir do qual se tenta fazer as correspondências entre a expressão regular e um conjunto de caracteres arbitrário. No âmbito das modalidades desportivas testadas neste trabalho foram definidos os seguintes padrões a pesquisar:

- i. Padrão resultado: a expressão regular descrita pela *string* "[0-9].*(-|x|X|/|vs|Vs|VS).[0-9]" permite identificar diversas maneiras dos utilizadores colocarem o resultado de um jogo de futebol ou ténis (exemplos: 1-1, 1x1, 6/4, 2 vs 0);
 - ii. Padrão minuto antes: a expressão regular descrita pela *string* "[0-9].*('|min)" permite identificar várias formas dos utilizadores colocarem tempos de jogo (exemplos: 30', 30 min, 30 minutos);
 - iii. Padrão minuto depois: a expressão regular descrita pela *string* "('|min).[0-9]" permite identificar outras formas dos utilizadores colocarem tempos de jogo (exemplos: '5, min 19, *minute* 10);
- b. Pesquisa de termos específicos de informação desportiva em dicionários de termos: este processo é similar à verificação de conteúdos impróprios nas mensagens, neste caso a verificação de termos é no sentido de garantir informação relevante. Foi seguida a metodologia utilizada em (Xu, et al., 2006), utilizando o conjunto de palavras-chave definidas pelos autores com ligeiras modificações. No estudo apresentado as palavras-chave estão definidas para a modalidade Futebol e língua inglesa, neste projecto foram criadas mais três listas de palavras-chave (uma para a modalidade futebol e língua portuguesa e outras duas para a modalidade Ténis nas duas línguas testadas).

4.3.1.3 Gestão de Índices

O processo de indexação das mensagens é o terceiro e último que as mensagens recolhidas sofrem. Este processo está directamente relacionado com a ferramenta *Apache Lucene*, também descrita anteriormente (ver ‘Ferramentas de Implementação’), que disponibiliza o conjunto de funcionalidades necessárias para garantir a persistência dos dados. Na indexação de conteúdos através desta biblioteca as principais classes utilizadas foram:

- 1) *Standard Analyzer*: classe que permite contruir um fluxo de símbolos para realizar análise textual. Representa uma política de extracção de termos existentes no índice;
- 2) *Directory*: classe que representa uma lista de ficheiros onde os índices vão ser criados e mantidos. Existem diversos tipos de implementações disponíveis, neste projecto foi usada uma *FSDirectory* pela necessidade de manter os índices em disco por um tempo limitado;
- 3) *IndexWriter*: classe que permite criar e manter os índices ou arquivos de um índice;
- 4) *Document*: classe que representa a unidade de indexação e pesquisa realizada num índice. Um documento representa um conjunto de campos onde cada campo é composto por um par nome-valor.

Neste projecto definiu-se para cada “*hashtag*” uma directoria onde os índices são mantidos para mensagens recolhidas tanto a partir do *Twitter* como da aplicação móvel. Sendo os índices compostos por documentos e os documentos compostos por campos foi necessário identificar o conjunto de campos a manter para cada mensagem e a forma como estes campos seriam mantidos.

Assim, e recorrendo a um exemplo, temos:

```
Document doc = new Document();

doc.add(new Field("hashtag", "#Benfica", Field.Store.YES,
Index.ANALYZED));
doc.add(new Field("utilizador", "@SLBenfica", Field.Store.YES,
Index.NOT_ANALYZED));
doc.add(new Field("mensagem", "Vitória por 2-1 #Benfica",
Field.Store.YES, Index.NOT_ANALYZED));
(...)
```

```
doc.add(new Field("tipo_mensagem", "info", Field.Store.NO,
Index.ANALYZED));
doc.add(new Field("fonte", "Twitter", Field.Store.YES,
Index.ANALYZED));

index_writer.addDocument(doc);
```

No exemplo anterior definimos uma instância da classe *Document* e para essa instância definimos/adicionamos um conjunto de campos compostos por pares nome-valor. Na linha “`doc.add(new Field("hashtag", "#Benfica", Field.Store.YES, Index.ANALYZED));`”, definimos um campo composto pelo nome “*hashtag*” e o valor “#Benfica”. A definição de cada campo é ainda composta por dois argumentos que definem a possibilidade dos campos serem pesquisáveis ou resultados de pesquisa:

- a) *Field.Store.YES*: significa que o conteúdo do campo assim definido pode ser visto como resultado de uma pesquisa. Por exemplo, o campo “mensagem” pode ser visto quando for feita a pesquisa que devolverá os dados a visualizar na aplicação móvel;
- b) *Field.Store.NO*: significa que o conteúdo do campo assim definido não pode ser visto como resultado de uma pesquisa. No exemplo, o campo “tipo_mensagem” não pode ser visualizado numa pesquisa;
- c) *Index.ANALYZED*: significa que o campo assim definido é pesquisável, ou seja, podem ser feitas pesquisa em função deste campo. No exemplo, o campo “fonte” é pesquisável já que podemos querer fazer pesquisas em função da fonte de informação que pode ser o *Twitter* ou a aplicação Móvel;
- d) *Index.NOT_ANALYZED*: significa que o campo assim definido não é pesquisável, ou seja, não podem ser feitas pesquisa em função deste campo.

No exemplo, o campo “mensagem” não é pesquisável já que não tem sentido fazer pesquisas em função de uma mensagem específica.

Depois de adicionarmos o conjunto de campos ao documento é necessário adicionar o documento ao índice. Definida uma instância da classe *IndexWriter*, através do método ‘*addDocument*’ conseguimos realizar essa operação.

4.3.2 Interacção Aplicação Móvel

A interacção entre plataforma e aplicação móvel é feita com o intuito de recolher na plataforma informação dos diversos actores do sistema. Inicialmente estava prevista a execução das tarefas de gestão de conteúdos da aplicação móvel através do modelo *MVC*, onde tínhamos os interfaces de utilizador na aplicação móvel (*view*), um *servlet* (*controller*) que trataria o *input* proveniente da aplicação móvel e informava a plataforma (*model*) dos pedidos de mudança dos dados aplicativos. Infelizmente, a *framework Sencha Touch* não permite realizar o método *POST*, não sendo assim possível ao cliente (aplicação móvel) enviar dados para o servidor, por exemplo, para criação de uma nova equipa de interesse. Como alternativa, e tendo em conta que o projecto funciona como prova de conceito, foi encontrada uma solução que se baseia numa nova funcionalidade associada ao *HTML5 – webStorage*. Existem dois novos objectos de armazenamento de dados através do *browser* associados a *webStorage*. São eles a *sessionStorage* e a *localStorage*. A diferença entre ambos é a sua persistência a longo termo. Quaisquer dados armazenados através de *localStorage* persistem entre novas sessões, estão disponíveis após reabertura do *browser* ou mesmo após o utilizador reiniciar o seu computador. Outra característica importante é o limite estabelecido pelo *W3C* para cada objecto deste género, sendo *5 megabytes* o limite inicial estabelecido – limite claramente acima do necessário para os objectivos deste projecto. São diversos os *browsers* que já suportam *webStorage*, no *Google Chrome* este suporte começou na versão 5 (Spencer, 2010), sendo esse o *browser* escolhido para teste da aplicação móvel.

A partir da plataforma é possível recolher as informações guardadas localmente no *browser* utilizando expressões regulares. Como foi explicado antes, as expressões regulares permitem identificar padrões específicos de texto, bastando por isso identificar os padrões de dados geridos localmente relativos aos objectos modelados na aplicação móvel (será explicada a forma como o *Sencha Touch* faz a gestão dos dados no próximo subcapítulo ‘Implementação Aplicação Móvel’).

De seguida é apresentado um exemplo, relativo a equipas, de como a informação é guardada localmente:

- a) Equipa 1: “*equipa-lista-1id:1,name:Benfica,modalidade:Futebol*”;
- b) Equipa 2: “*equipa-lista-2id:2,name:Sporting,modalidade:Futebol*”.

Como podemos ver, pode ser facilmente definido um padrão que consiga extrair estas informações, já que existe um conjunto de informação específica associada à lista criada: nome da lista, id, nome da equipa e modalidade à qual a equipa está associada.

Para os restantes objectos modelados na aplicação móvel, o padrão é definido consoante os campos que definem esses objectos. Para além de equipas, são também recolhidas informações sobre jogadores, eventos, “*hashtags*” de equipas, “*hashtags*” de jogadores, “*hashtags*” de eventos, preferências e mensagens.

4.3.3 Agregação dos Conteúdos de Preferência

Após a realização das duas macro-sequências de interacção entre plataforma, *Twitter* e aplicação móvel dá-se a agregação/criação dos conteúdos preferidos do utilizador. Este processo é composto por quatro fases:

- 1) Verificação de preferências: os dados anteriormente recolhidos a partir da aplicação móvel permitem saber as preferências do utilizador tanto ao nível das “*hashtags*” como ao nível dos tipos e fontes de informação e língua associados às mensagens. Nesta fase é feita uma verificação da tabela “*utilizadorpreferencias*” onde são mantidas as preferências dos utilizadores relativamente às condicionantes das mensagens a recolher (tipo, fonte e língua). É também realizada a recolha do conjunto de “*hashtags*” pertencentes a equipas, jogadores e eventos de interesse do utilizador, mantidas respectivamente nas tabelas “*utilizadorhashequipa*”, “*utilizadorhashjogador*” e “*utilizadorhashevento*”;
- 2) Criação da lista de mensagens: de acordo com todas as preferências recolhidas na fase anterior, a segunda fase passa pela construção da lista de mensagens que o utilizador visualizará na aplicação móvel. Nesta fase, é importante referir que as mensagens provenientes do *Twitter* são tratadas de forma diferente das provenientes da aplicação móvel, ou seja, enquanto as mensagens recolhidas a partir do *Twitter* são mantidas apenas por iteração e não é necessário verificar quais as mais recentes (são mantidas nos índices apenas o número de mensagens que irão ser inseridas na lista), por outro lado, antes de ser feita a recolha das mensagens provenientes da aplicação móvel é verificado o tamanho dos índices e caso o número de mensagens seja superior a um máximo, eliminam-se as mensagens mais antigas e

devolvem-se apenas as mais recentes. Para cada “*hashtag*”, serão mantidas respectivamente as 150 e 100 mensagens mais recentes provenientes do *Twitter* e aplicação móvel;

- 3) Remoção de duplicados e ordenação: após recolha e colocação numa lista das mensagens relativas ao conjunto de “*hashtags*” de interesse do utilizador é necessário verificar duplicações na lista porque esta alberga todas as mensagens de todas as “*hashtags*”. Dado que a mesma mensagem pode conter mais do que uma “*hashtag*” e no acto inicial de recolha as duplicações são verificadas apenas por “*hashtag*”, não foi salvaguardado o caso da mesma mensagem ter sido recolhida para duas “*hashtags*” diferentes. Depois de realizada a remoção de duplicações, as mensagens são ordenadas em função da sua data com o objectivo de possuímos as mais recentes no topo da lista para serem colocadas no ficheiro *JSON*;
- 4) Criação de ficheiro *JSON*: a última fase do processo de agregação dos conteúdos de preferência consiste na criação do ficheiro que será a resposta ao pedido efectuado a partir da aplicação móvel. *JSON*²⁹ é um formato textual simples para intercâmbio de dados computacionais, totalmente independente de linguagem mas que utiliza um conjunto de convenções conhecidas por programadores das linguagens *C*, *C#*, *Java*, *Javascript*, *Perl*, *Python*, entre outras. A aplicação móvel tem a capacidade de ler uma resposta do servidor no formato *JSON*, sendo esse o motivo para a definição destes ficheiros.

4.3.4 Renovação Índices

O último processo realizado pela plataforma de gestão de conteúdos desportivos é a renovação dos índices. Este processo é distinto para os índices referentes a mensagens provenientes do *Twitter* e da aplicação móvel. Enquanto os índices das mensagens recolhidas a partir da aplicação móvel são mantidos entre iterações, eliminando mensagens em excesso no processo de recolha dos conteúdos preferidos - como foi identificado na segunda fase desse processo (Criação da lista de mensagens) - a gestão dos índices de mensagens provenientes do *Twitter* é diferente e realizado após a

²⁹ <http://www.json.org>

Integração e divulgação de informação desportiva em redes sociais através de dispositivos móveis

criação dos ficheiros *JSON*. Estes índices são mantidos apenas por iteração e após a criação dos ficheiros são eliminados, sendo criados novos índices no início de cada nova iteração. A decisão de realizar a manutenção dos índices desta forma foi tomada porque neste projecto o tráfego de informação gerada a partir do *Twitter* é muito superior à gerada a partir da aplicação móvel.

4.4 Implementação Aplicação Móvel

O desenvolvimento de projectos através da *framework Sencha Touch* segue uma estrutura específica:

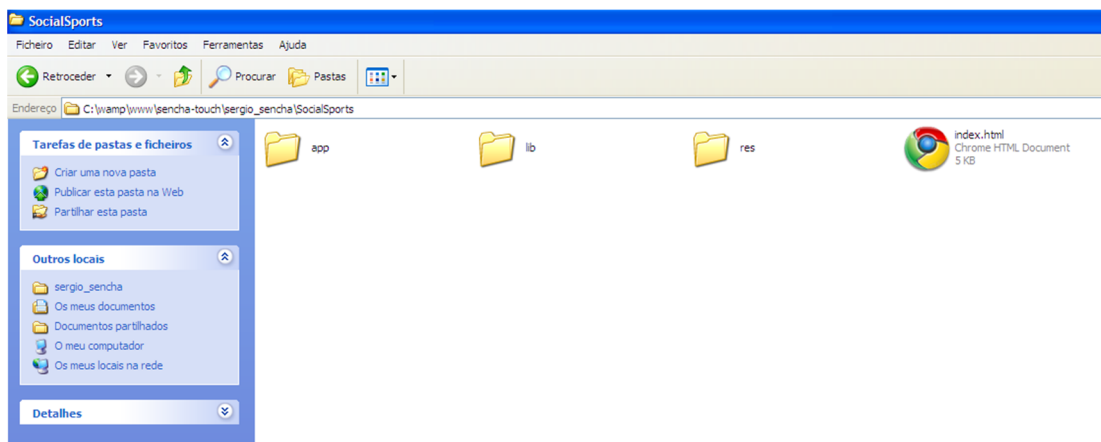


Figura 9 - Estrutura Sencha Touch

O projecto, denominado *SocialSports*, mantém a seguinte estrutura:

- 1) *app*: directoria onde é colocado todo o código da aplicação (ficheiros *.js*). Dentro desta directoria existe um conjunto de subdirectorias que permitem uma melhor organização do projecto, segundo uma metodologia de desenvolvimento *MVC*;
- 2) *lib*: directoria que contém a *framework Sencha Touch*;
- 3) *res*: directoria onde são colocadas todas as imagens e ficheiros *CSS*;
- 4) *index.html*: este ficheiro é o ponto de entrada a partir do qual a aplicação é iniciada. É o único ficheiro *html* necessário em toda a aplicação.

4.4.1 Metodologia MVC

A metodologia de desenvolvimento *MVC* é suportada e incentivada na criação de aplicações através da *framework Sencha Touch*. O primeiro passo na definição de uma aplicação é definir a própria aplicação:

```
SocialSports = new Ext.regApplication({
    name: 'SocialSports',
    (...),
    launch: function()
    {
        (...),
    },
});
```

Desta forma é criada a aplicação, através de um objecto de configuração composto pelo nome e uma função que serve essencialmente para carregar a aplicação. O nome da aplicação é essencial porque é desta forma que criamos um *namespace* único e global a partir do qual são criadas as visões (*views*), controladores (*controllers*) e modelos (*model*). O código de criação da aplicação, segundo as normas de desenvolvimento de projectos *Sencha Touch*, é colocado num script com o nome *app.js*. Este script está anexado ao documento *index.html* e é executado quando esse documento é carregado.

De seguida são apresentadas as principais visões, controladores e modelos constituintes da aplicação móvel.

4.4.1.1 Visão (View)

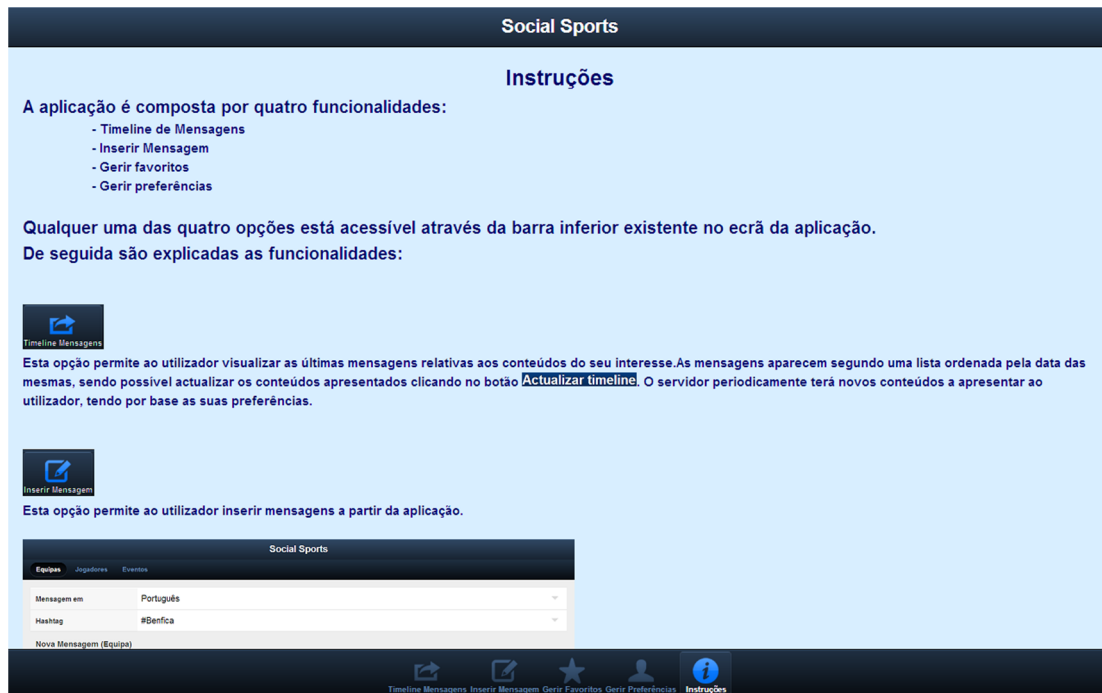


Figura 10 - Aplicação Móvel: Ecrã Instruções

As *views* são responsáveis pela apresentação dos dados ao utilizador. Neste projecto foi definida uma *view* principal a partir da qual são carregadas/acedidas as restantes. A *view* principal denominada *Viewport* foi definida como um *Panel*, um dos tipos de *container* disponibilizados pela *framework*, no qual foi definida uma propriedade (*dockedItems*) que possibilita adicionar componentes ao *Panel* numa posição fixa (topo, direita, esquerda ou fundo). Foram adicionados dois componentes ao *Panel*: no topo adicionou-se um componente do tipo *ToolBar* que serviu apenas para colocar o nome da aplicação. Adicionou-se também um componente do tipo *TabBar* no fundo do *Panel*, onde cada item deste componente permite visualizar um dos cinco ecrãs principais que a aplicação contém.

No caso da Figura 10, está seleccionado o último item do componente *TabBar* que representa o ecrã de instruções da aplicação.

De seguida são explicados os cinco ecrãs principais da aplicação e suas funções.

4.4.1.1.1 Ecrã Instruções

O ecrã “Instruções” encontra-se exibido na Figura 10 para demonstrar como funciona o componente *TabBar*. Este ecrã tem o intuito de explicar ao utilizador os restantes quatro ecrãs principais da aplicação: para que serve a definição das

preferências de utilizador, como gerir os seus conteúdos favoritos, como inserir mensagens baseadas nos seus conteúdos favoritos e exibição de mensagens de acordo com as preferências definidas.

A sua característica principal é a possibilidade de se fazer *scroll* para visualizar informações que estão descritas ao longo da página. Esse conteúdo foi totalmente escrito como *HTML* puro.

4.4.1.1.2 Ecrã Gerir Preferências

O ecrã “Gerir Preferências” possibilita ao utilizador definir o conjunto de preferências relativamente aos conteúdos que deseja visualizar no ecrã “*Timeline Mensagens*”. É composto por um componente do tipo *FormPanel* que funciona como *container* para dois outros componentes: componente *FieldSet* que permite agrupar itens ou campos do mesmo tipo, neste caso agrupa três *SelectField* (Tipo(s) de Informação, Língua(s) e Fonte(s) de Informação); componente *Button*, que neste caso serve para confirmar as preferências escolhidas.



Figura 12- Aplicação Móvel: Ecrã Gerir Preferências

Na Figura 11, podemos ver a utilização do componente *SelectField* que permite ao utilizador, no caso dos tipos de informação, escolher entre três opções o tipo de conteúdos que pretende ver. A metodologia é a mesma para língua e fontes de informação.



Figura 11 - Aplicação Móvel: Ecrã Gerir Preferências. Acção: Escolha do tipo de informação

Depois de escolhidas as preferências é necessário guardá-las:

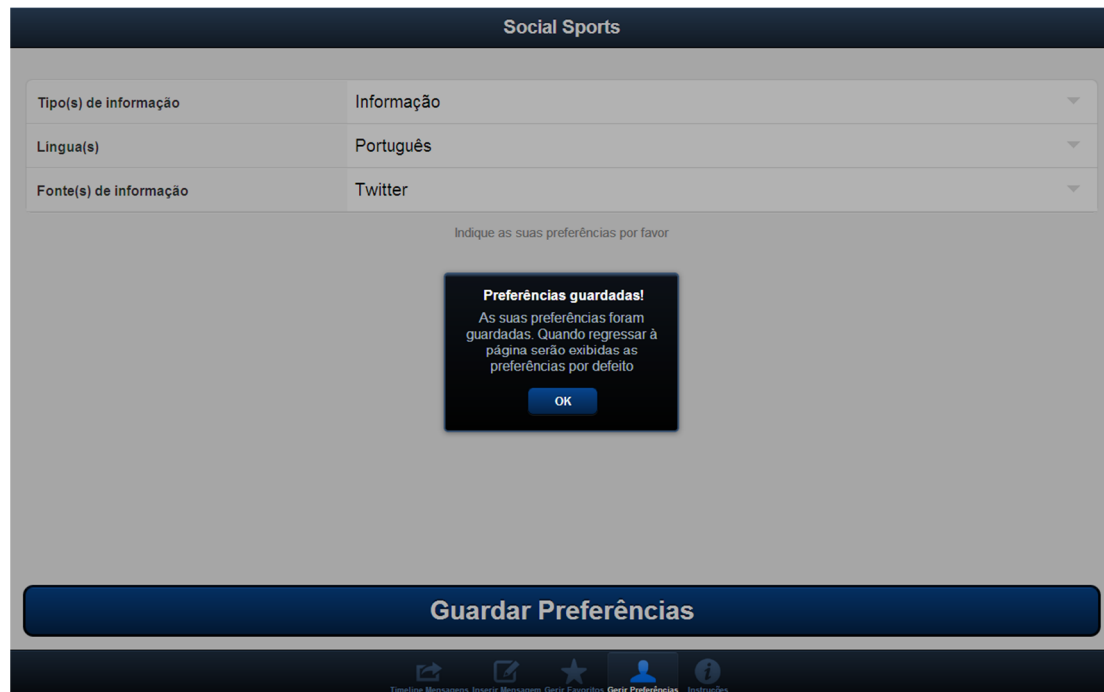


Figura 13 - Aplicação Móvel: Ecrã Gerir Preferências. Ação: Confirmação de preferências

O utilizador ao clicar no botão 'Guardar Preferências' é-lhe exibida uma mensagem de confirmação da mudança nas preferências, como pode ser visto na Figura 13.

4.4.1.1.3 Ecrã Gerir Favoritos

O ecrã "Gerir Favoritos" permite interagir com as equipas, jogadores e eventos favoritos do utilizador.

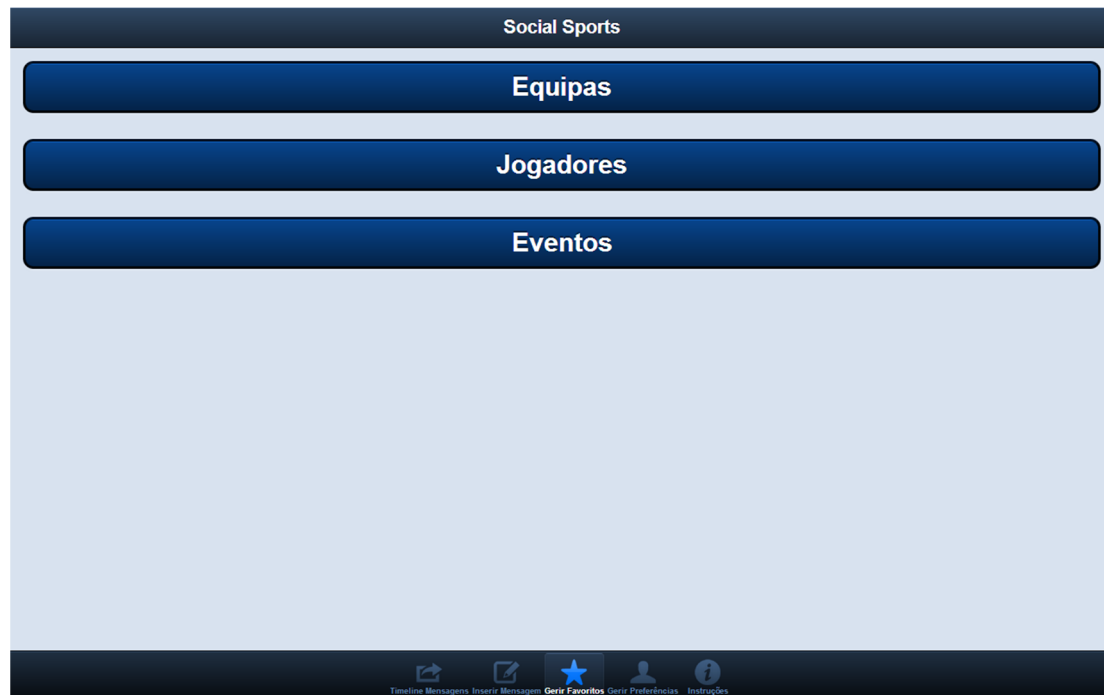


Figura 14 - Aplicação Móvel: Ecrã Gerir Favoritos

Este ecrã possibilita ao utilizador escolher os seus “actores” desportivos favoritos. É composto por três componentes do tipo *Button*, onde cada um representa um desses actores (Equipas, Jogadores e Eventos). O funcionamento é idêntico para os diversos actores. Se escolhermos as Equipas, por exemplo:

Na Figura 15, podemos verificar o ecrã apresentado ao utilizador quando este escolhe visualizar as Equipas. Este ecrã é composto por dois componentes: uma *Toolbar* que foi colocada no topo, tem como título ‘Equipas’ e funciona como *container* para três componentes do tipo *Button* (um botão para retroceder, um botão para aceder às “*hashtags*” relativas a equipas e um botão para adicionar equipas). Existe também um componente do tipo *List* que permite exibir um conjunto de dados, neste caso equipas, em forma de lista. Se clicarmos no botão sinalizado com ‘+’ podemos inserir novas equipas:

Integração e divulgação de informação desportiva em redes sociais através de dispositivos móveis

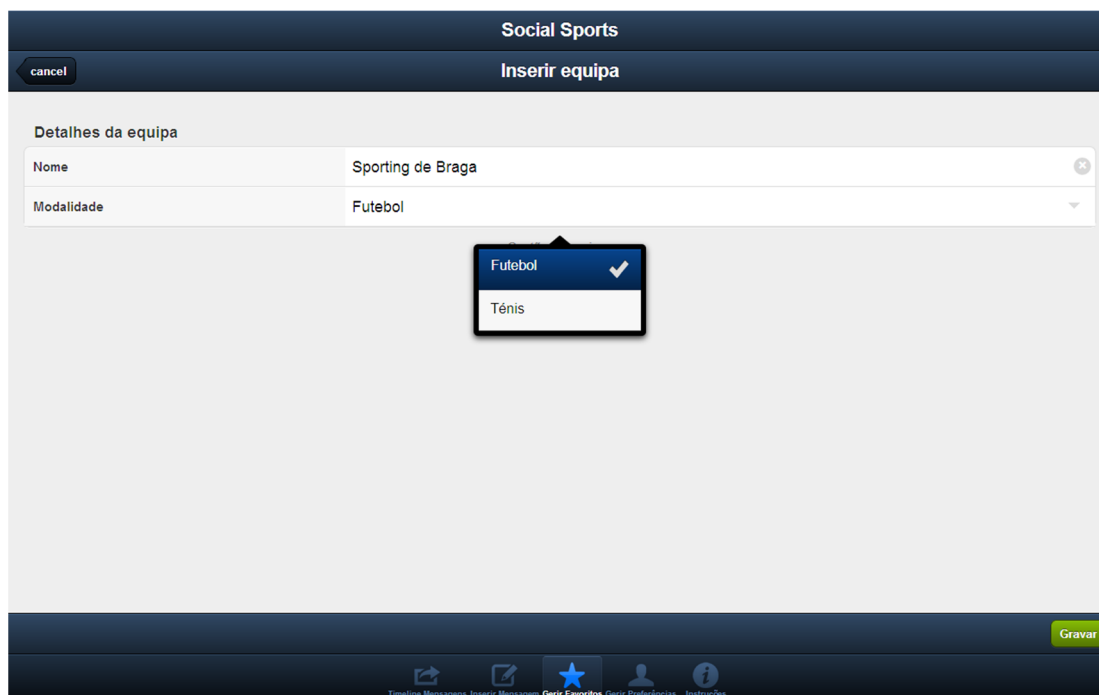


Figura 15 - Aplicação Móvel: Ecrã Gerir Favoritos. Acção: Visualizar Equipas

Quando inserimos novas equipas, temos de definir o nome da equipa e escolher qual a modalidade à qual está associada. Depois de definirmos estes dados podemos gravá-los e a equipa aparece na lista exibida na Figura 16.

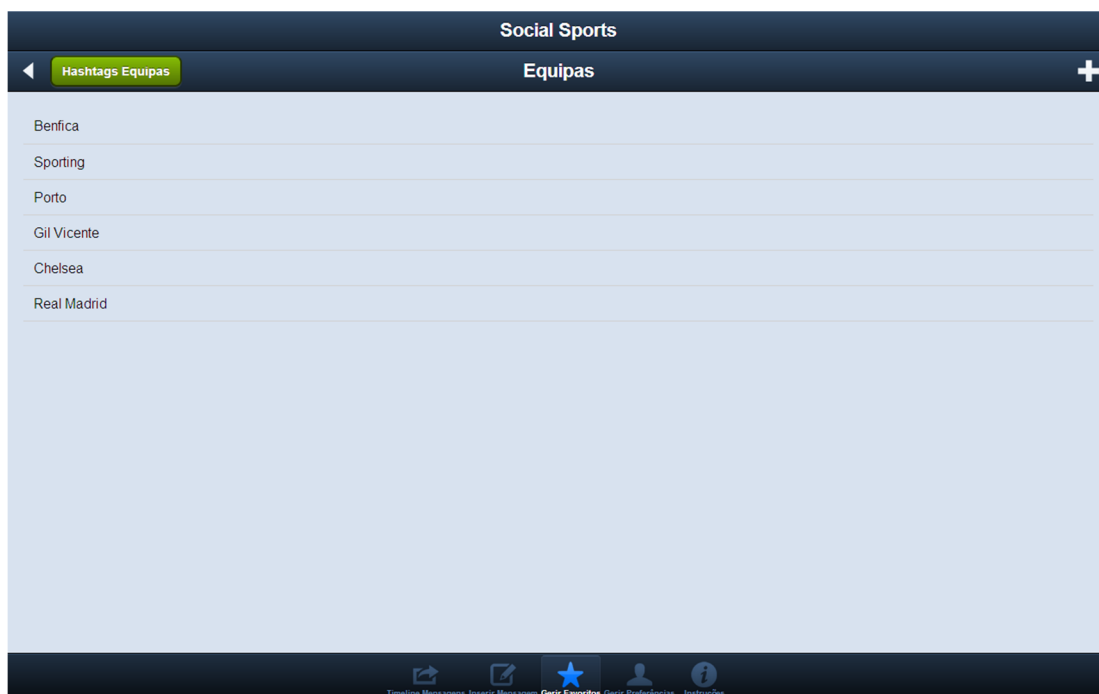


Figura 16 - Aplicação Móvel: Ecrã Gerir Favoritos. Acção: Inserir Equipa

Para além de possibilitar inserir novas equipas, ver quais as “hashtags” de equipas que o utilizador deseja seguir ou voltar ao ecrã anterior, qualquer item da lista

Integração e divulgação de informação desportiva em redes sociais através de dispositivos móveis

pode ser escolhido/acedido, possibilitando remover esse item (equipa) ou actualizar o seu conteúdo.

Como podemos ver na Figura 17, são apresentados os detalhes da equipa escolhida e podemos actualizar os seus dados (mudar o nome da equipa ou mudar a modalidade associada à equipa) ou apagar a equipa. Se decidirmos apagar a equipa, é sempre exibido uma mensagem para confirmar a operação, como podemos ver na Figura 18.

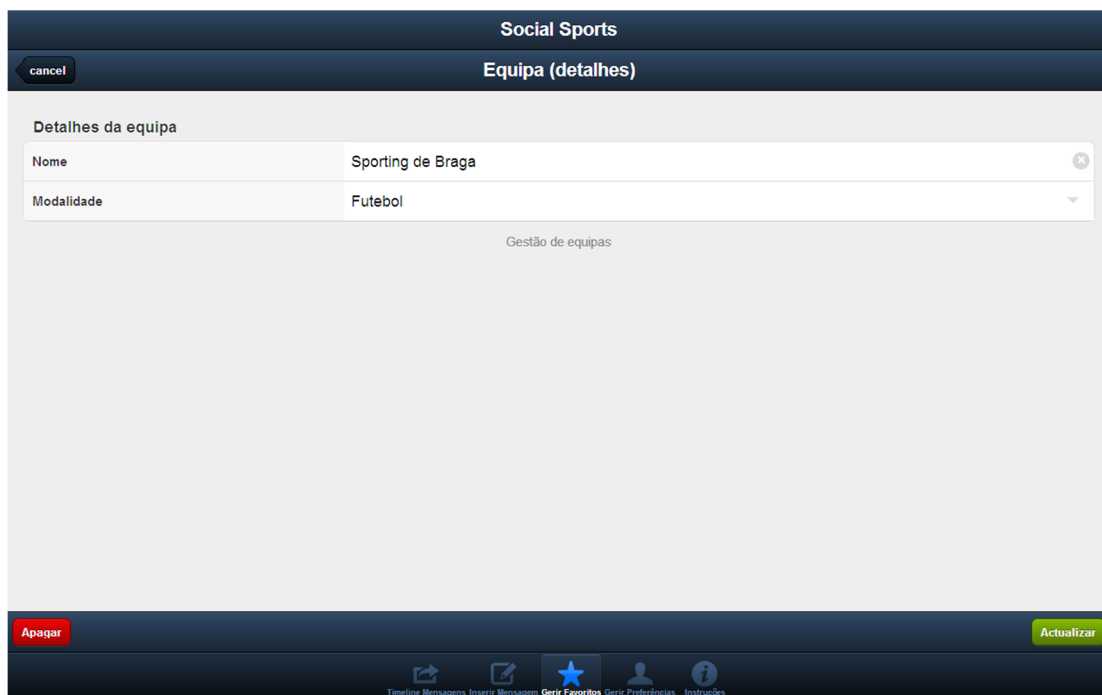


Figura 17 - Aplicação Móvel: Ecrã Gerir Favoritos. Acção: Detalhes Equipa

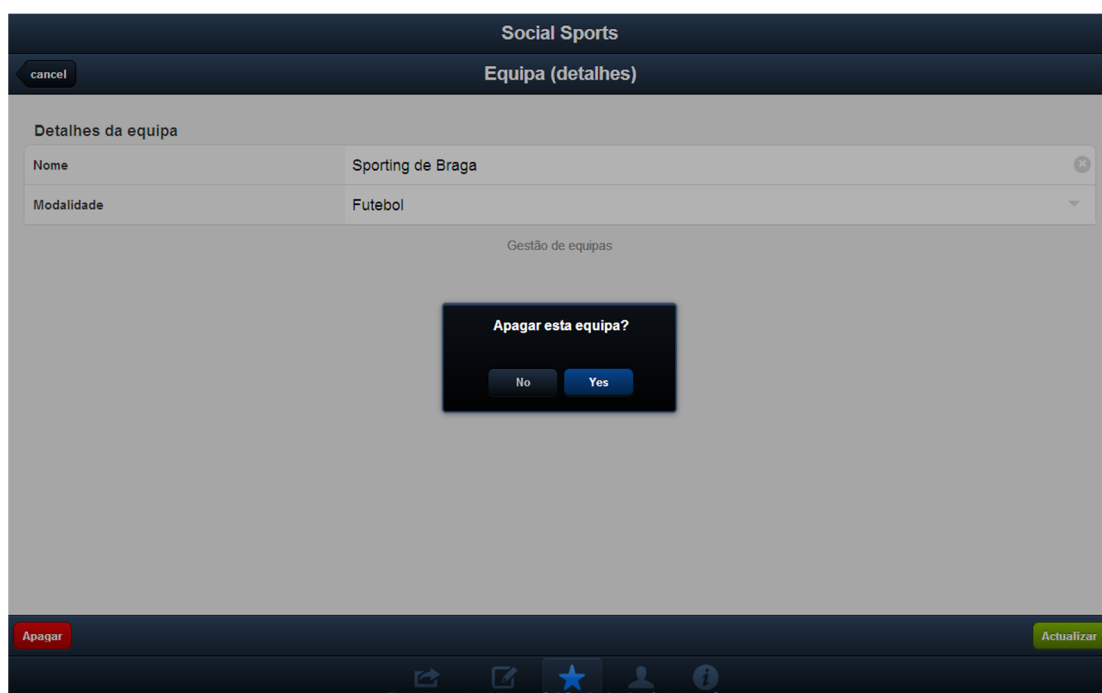


Figura 18 - Aplicação Móvel: Ecrã Gerir Favoritos. Acção: Eliminar Equipa

Integração e divulgação de informação desportiva em redes sociais através de dispositivos móveis

No ecrã inicial das equipas, Figura 15, temos a possibilidade de aceder às “*hashtags*” de equipas com interesse para o utilizador, se clicarmos no botão “*Hashtags* Equipas”. Na Figura 19, podemos verificar que este ecrã é em tudo semelhante ao das equipas, permitindo interagir com uma lista de “*hashtags*” de equipas. Podemos adicionar novas “*hashtags*”, actualizar ou apagar “*hashtags*” já criadas.

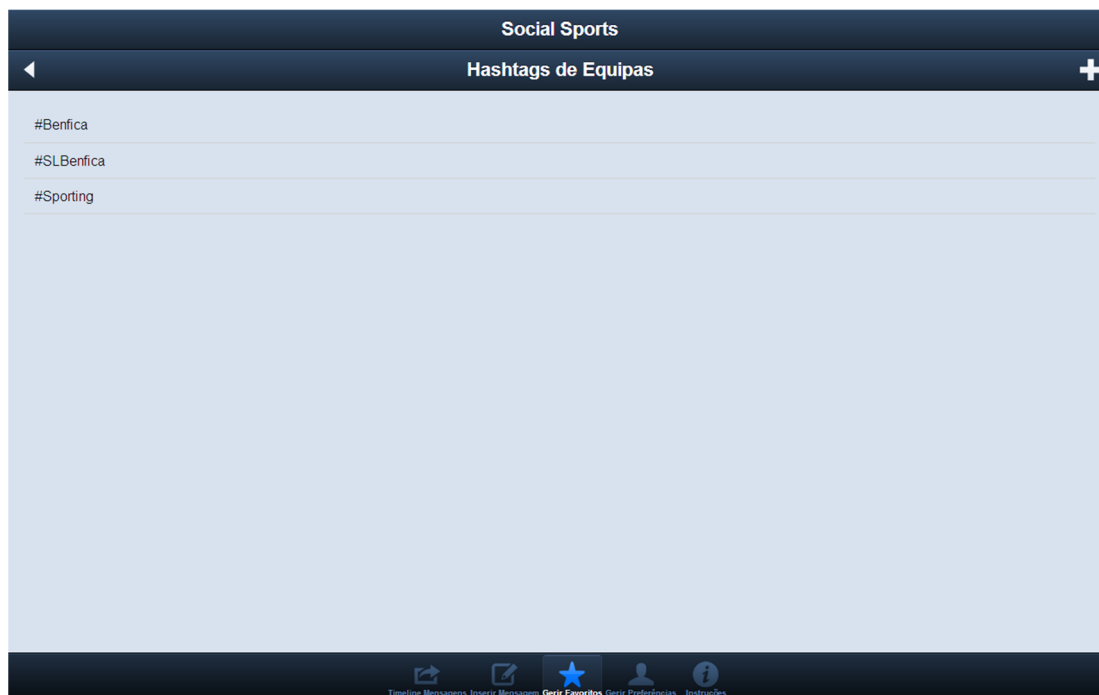


Figura 20 - Aplicação Móvel: Ecrã Gerir Favoritos. Acção: Visualizar Hashtags Equipas

A principal diferença para o ecrã de interacção de Equipas, é termos de definir para cada “*hashtag*”, além da sua identificação/nome e modalidade associada, também a equipa à qual pertence:

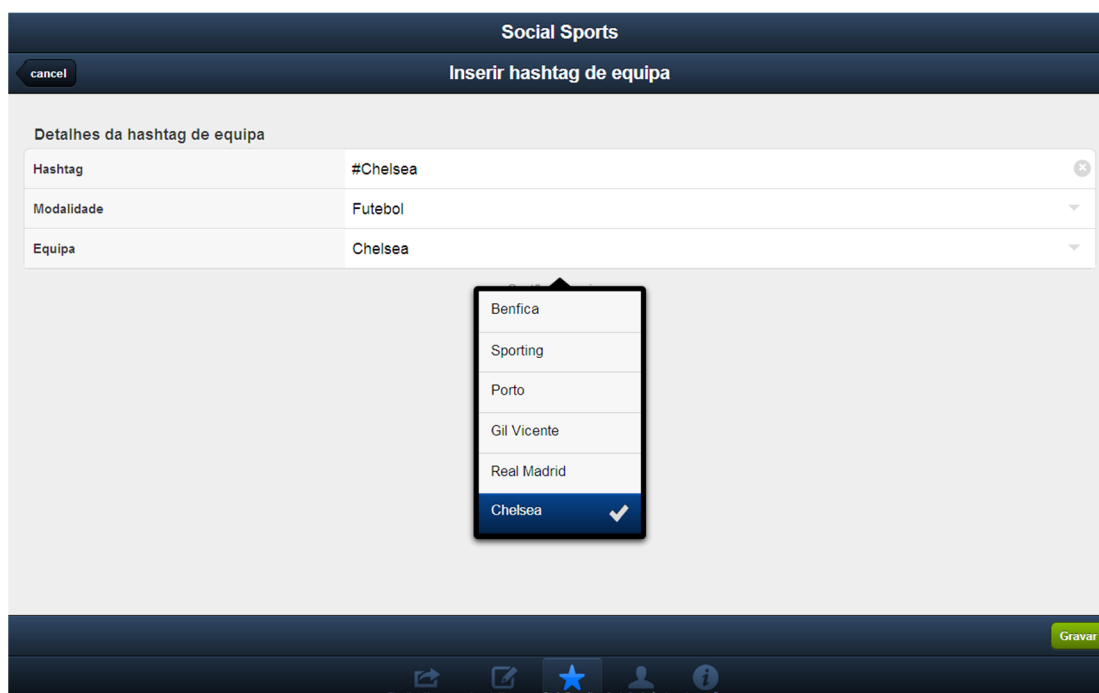


Figura 19 - Aplicação Móvel: Ecrã Gerir Favoritos. Acção: Adicionar Hashtag Equipa

Na Figura 19, podemos verificar que as equipas disponíveis, são as definidas anteriormente no ecrã de gestão de equipas.

4.4.1.1.4 Ecrã Inserir Mensagem

O ecrã “Inserir Mensagem”, exibido na Figura 21, permite ao utilizador inserir mensagens através da aplicação móvel.



Figura 21 - Aplicação Móvel: Ecrã Inserir Mensagem

Foi definido através de um *container TabPanel* que permite agrupar diversos componentes segundo *tabs*. Neste caso, cada *tab* é um *Form* (formulário) que representa o conjunto de favoritos, respectivamente Equipas, Jogadores e Eventos. Dentro de cada formulário existem três *FieldSet*: o primeiro representa a escolha da língua em que a mensagem vai ser escrita (Português ou Inglês), o segundo representa a “*hashtag*” sobre a qual a mensagem é escrita (conforme o formulário escolhido é colocada a lista de “*hashtags*” correspondentes a equipas, jogadores ou eventos, definidas no ecrã “Gerir Favoritos”), o terceiro representa a própria mensagem e é composto pelo título e um campo *TextArea* onde a mensagem é escrita. Para além dos três *FieldSet* cada formulário contém um componente *Button* para enviar a mensagem.

Na Figura 22, podemos verificar que na escolha da “*hashtag*” são apresentadas as “*hashtags*” que anteriormente definidas pelo utilizador. Foi esta a metodologia utilizada porque permite por um lado que o utilizador não seja obrigado a inserir a própria “*hashtag*” na mensagem, embora o possa fazer, e por outro porque facilita a

Integração e divulgação de informação desportiva em redes sociais através de dispositivos móveis

forma como a plataforma irá gerir os dados. Mais, a plataforma verifica a existência da “hashtag” na mensagem e caso a “hashtag” não integre o conteúdo da mensagem, antes de inserir nos índices, faz-se a transformação inserindo a “hashtag” no final da mensagem a indexar.

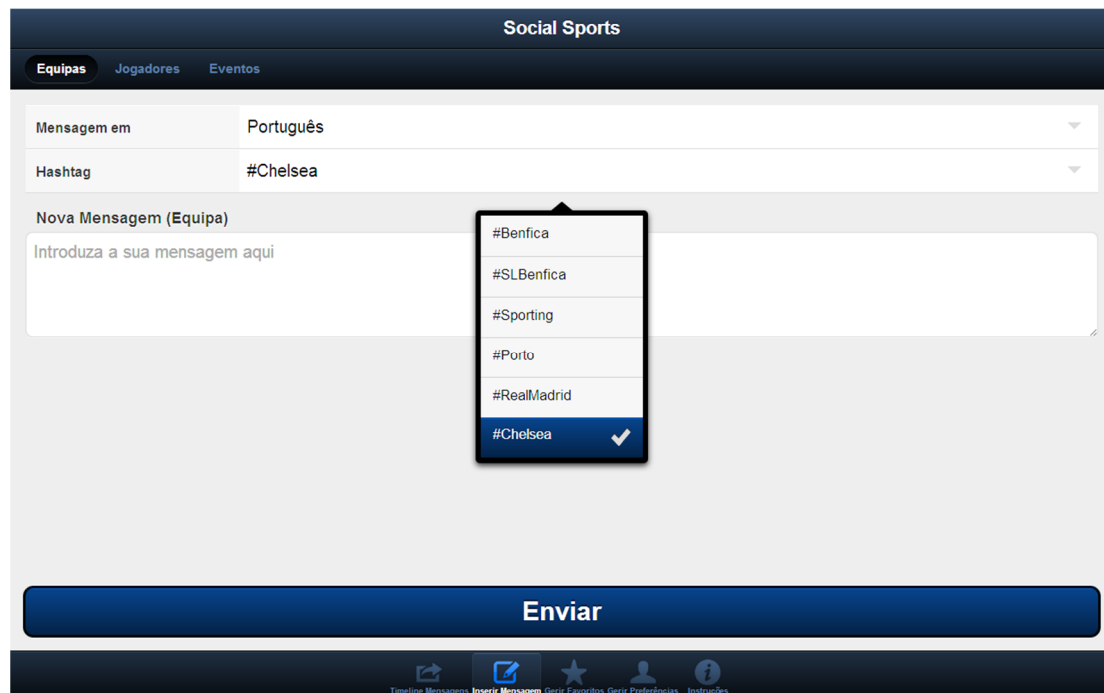


Figura 22 - Aplicação Móvel: Ecrã Inserir Mensagem. Acção: Escolher Hashtag

Na Figura 23, podemos ver o exemplo de uma mensagem relativa à “hashtag” #Chelsea. A mensagem foi escrita em português embora a equipa seja inglesa.

Integração e divulgação de informação desportiva em redes sociais através de dispositivos móveis

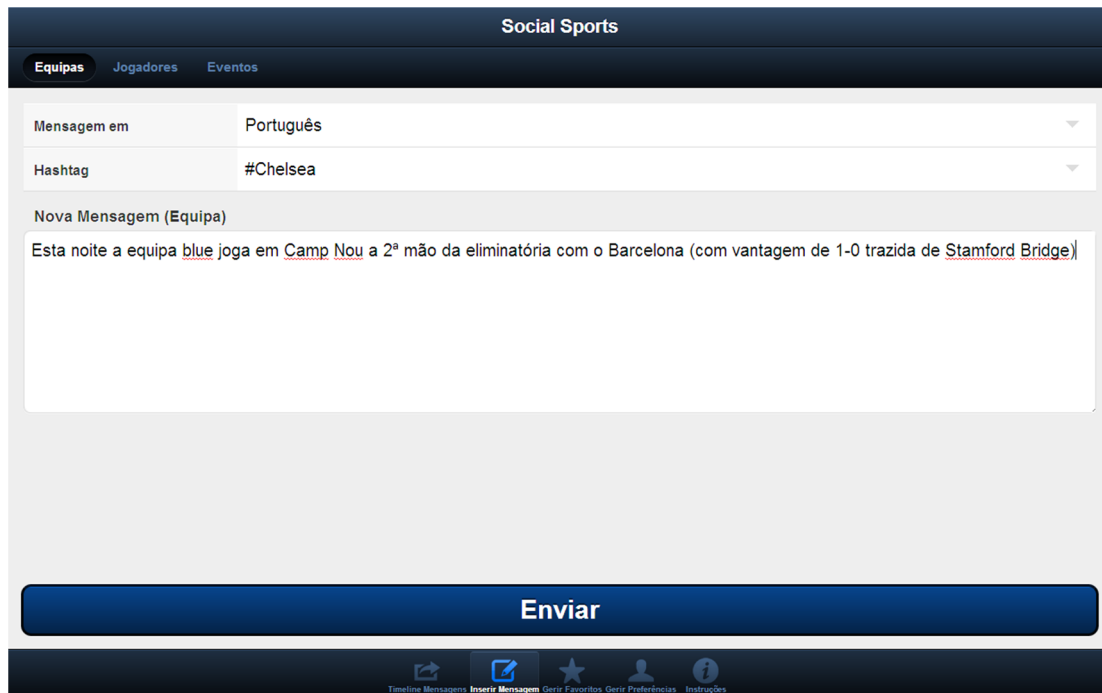


Figura 23 - Aplicação Móvel: Ecrã Inserir Mensagem. Acção: Escrever Mensagem

Após ser definida a língua, “*hashtag*” associada e ser escrito o corpo da mensagem, basta clicar no botão “Enviar” para submeter a mensagem.

Integração e divulgação de informação desportiva em redes sociais através de dispositivos móveis

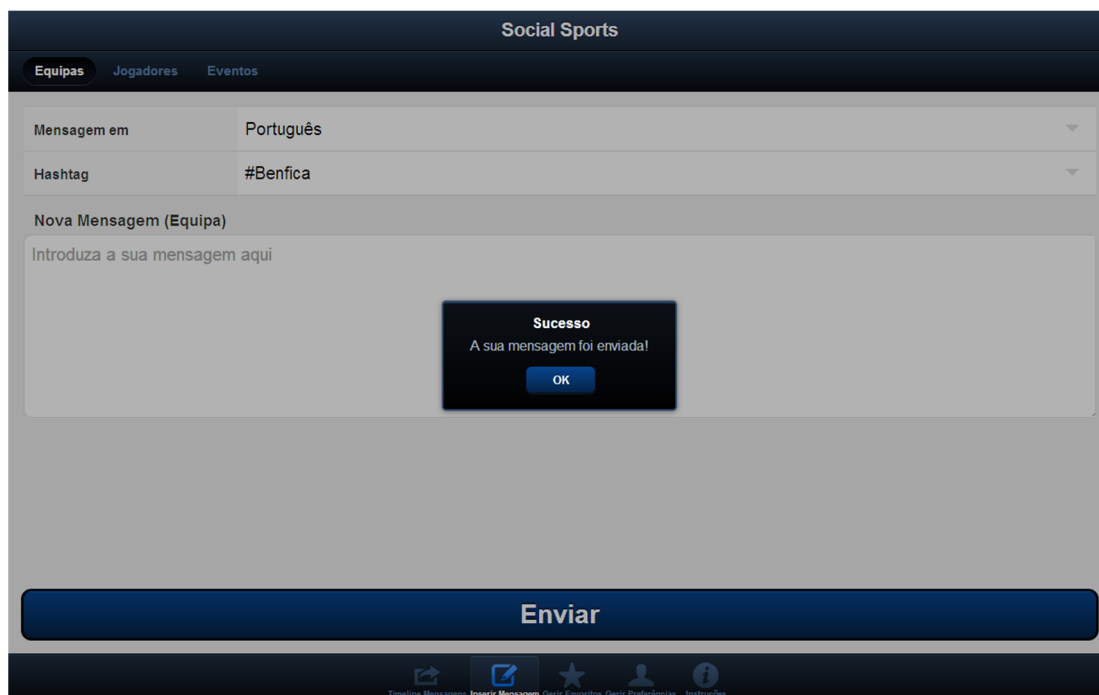


Figura 24 - Aplicação Móvel: Ecrã Inserir Mensagem. Acção: Confirmar Mensagem

Conforme realizado o envio da mensagem recebemos uma mensagem a confirmar a operação.

4.4.1.1.5 Ecrã Timeline Mensagens

O ecrã “*Timeline Mensagens*” permite exibir as mensagens de interesse do utilizador.



Figura 25 - Aplicação Móvel: Ecrã Timeline Mensagens

Este ecrã foi definido através de um *container* do tipo *Panel* composto por um componente do tipo *Button* que permite actualizar os dados exibidos (realiza um pedido ao servidor por novos dados) e por um *Panel* que contém uma *DataView* que permite exibir os dados de acordo com um *template*. Neste *template* definiu-se a forma como a informação ia estar disposta, como podemos ver na Figura 25, são exibidos autor, corpo, data e fonte de cada mensagem.

4.4.1.2 Controlador (*Controller*)

Os *controllers* são responsáveis por dar respostas a eventos que ocorrem dentro da aplicação. Um exemplo da utilização de um controlador na aplicação é quando o utilizador carrega no botão ‘Enviar’ no ecrã ‘Inserir Mensagem’ – o controlador toma a acção de guardar os dados relativos à mensagem, exibe uma mensagem a confirmar a operação e limpa o formulário.

Cada *controller* pode conter um conjunto de funções/acções despoletadas a partir da interacção dos utilizadores com as *views*. A ligação é feita a partir das *views* através da definição de um caminho/*route*. Um caminho é um identificador para um par controlador/acção, ou seja e voltando ao caso da inserção de uma mensagem, se tivermos um controlador de nome ‘*NewMessagesController*’ e uma acção nesse controlar com o nome ‘*save*’, o caminho é ‘*NewMessagesController/save*’ e através dele fazemos a ligação entre a *view* e o *controller*.

Nesta aplicação foram utilizados nove controladores:

- 1) *HomeController*: controla a escolha e renderização dos cinco ecrãs principais da aplicação;
- 2) *NewMessagesController*: recolhe os dados das mensagens inseridas pelos utilizadores;
- 3) *PreferencesController*: recolhe os dados de preferência dos utilizadores;
- 4) *TeamsController*: controla as acções relativas a equipas (apresenta lista de equipas, permite inserir, actualizar e remover equipas);
- 5) *PlayersController*: controla as acções relativas a jogadores (apresenta lista de jogadores, permite inserir, actualizar e remover jogadores);
- 6) *EventsController*: controla as acções relativas a eventos (apresenta lista de eventos, permite inserir, actualizar e remover eventos);

- 7) *HashTeamsController*: controla as acções relativas a “*hashtags*” de equipas (apresenta lista de “*hashtags*” de equipas, permite inserir, actualizar e remover “*hashtags*” de equipas);
- 8) *HashPlayersController*: controla as acções relativas a “*hashtags*” de jogadores (apresenta lista de “*hashtags*” de jogadores, permite inserir, actualizar e remover “*hashtags*” de jogadores);
- 9) *HashEventsController*: controla as acções relativas a “*hashtags*” de eventos (apresenta lista de “*hashtag*” de eventos, permite inserir, actualizar e remover “*hashtags*” de eventos).

4.4.1.3 Modelo (Model)

Os modelos representam um objecto gerido pela aplicação e são definidos através de um conjunto de campos, métodos e propriedades relevantes ao próprio modelo. Neste projecto temos diversos tipos de objectos como mensagens, equipas ou preferências. Para os diferentes modelos usados na aplicação temos um conjunto de campos e validações:

- 1) *NewMessagesModel* (modelo que define as mensagens):

| Campos | | | |
|----------------|---------------|-----------|--|
| nome | tipo | validação | mensagem validação |
| lingua | <i>string</i> | presença | / |
| <i>hashtag</i> | <i>string</i> | presença | / |
| mensagem | <i>string</i> | presença | 'O campo da mensagem não pode estar vazio' |

Tabela 6 - Campos e validações *NewMessageModel*

- 2) *PreferencesModel* (modelo que define as preferências):

| Campos | | | |
|------------------|---------------|-----------|--------------------|
| nome | tipo | validação | mensagem validação |
| tipoinformacao | <i>string</i> | presença | / |
| lingua | <i>string</i> | presença | / |
| fontesinformacao | <i>string</i> | presença | / |

Tabela 7 - Campos e validações *PreferencesModel*

3) *TeamsModel* (modelo que define as equipas):

| Campos | | | |
|-------------------|---------------|-----------|---|
| nome | tipo | validação | mensagem validação |
| <i>name</i> | <i>string</i> | presença | 'Deve introduzir uma equipa antes de tentar gravá-la' |
| modalidade | <i>string</i> | presença | / |

Tabela 8 - Campos e validações *TeamsModel*

4) *PlayersModel* (modelo que define os jogadores):

| Campos | | | |
|-------------------|---------------|-----------|---|
| nome | tipo | validação | mensagem validação |
| <i>name</i> | <i>string</i> | presença | 'Deve introduzir um jogador antes de tentar gravá-lo' |
| modalidade | <i>string</i> | presença | / |

Tabela 9 - Campos e validações *PlayersModel*

5) *EventsModel* (modelo que define os eventos):

| Campos | | | |
|-------------------|---------------|-----------|--|
| nome | tipo | validação | mensagem validação |
| <i>name</i> | <i>string</i> | presença | 'Deve introduzir um evento antes de tentar gravá-lo' |
| modalidade | <i>string</i> | presença | / |

Tabela 10 - Campos e validações *EventsModel*

6) *HashTeamsModel* (modelo que define as “hashtags” de equipas):

| Campos | | | |
|-------------------|---------------|---------------|--|
| nome | tipo | validação | mensagem validação |
| <i>hashtag</i> | <i>string</i> | <i>format</i> | 'Deve introduzir uma hashtag válida. Ex. #hashtag' |
| equipa | <i>string</i> | presença | / |
| modalidade | <i>string</i> | presença | / |

Tabela 11 - Campos e validações *HashTeamsModel*

7) *HashPlayersModel* (modelo que define as “*hashtags*” de jogadores):

| Campos | | | |
|-------------------|---------------|---------------|--|
| nome | tipo | validação | mensagem validação |
| <i>hashtag</i> | <i>string</i> | <i>format</i> | 'Deve introduzir uma hashtag válida. Ex. #hashtag' |
| jogador | <i>string</i> | presença | / |
| modalidade | <i>string</i> | presença | / |

Tabela 12 - Campos e validações *HashPlayersModel*

8) *HashEventsModel* (modelo que define as “*hashtags*” de eventos):

| Campos | | | |
|-------------------|---------------|---------------|--|
| nome | tipo | validação | mensagem validação |
| <i>hashtag</i> | <i>string</i> | <i>format</i> | 'Deve introduzir uma hashtag válida. Ex. #hashtag' |
| evento | <i>string</i> | presença | / |
| modalidade | <i>string</i> | presença | / |

Tabela 13 - Campos e validações *HashEventsModel*

Os campos com validação do tipo *format* verificam se o primeiro carácter é o símbolo ‘#’ já que essa validação é feita para “*hashtags*”. Esta verificação foi conseguida através do *matcher* `/^#[a-zA-Z0-9]+$/`.

Explicação:

- Carácter ‘^’ significa início da *string*;
- Caracteres ‘#+’ significa que o primeiro carácter é um # que será concatenado com o resto da *string*;
- Caracteres ‘[a-zA-Z0-9]+’ significa que a *string* pode conter qualquer letra de ‘a’ a ‘z’ minúscula ou maiúscula, e/ou qualquer conjunto numérico. O carácter ‘+’ obriga à ocorrência do padrão uma ou mais vezes para não permitir *strings* vazias;
- Carácter ‘\$’ significa fim de *string*.

4.4.1.3.1 *Model, Store e Proxy*

A persistência nos projectos desenvolvidos através da *framework Sencha Touch* é garantida recorrendo ao pacote de dados que permite assegurar essa persistência apoiando-se em classes do tipo *Model, Store e Proxy*. Estes três tipos de classes estão directamente relacionadas: como referido um modelo representa um objecto gerido na aplicação, uma *store* funciona como uma *cache client-side* que permite encapsular um conjunto de instâncias dos objectos definidos no modelo e carrega esses dados através de uma classe *proxy* (Spencer, 2011). No desenvolvimento da aplicação, um exemplo de uso deste tipo de classes em comum, ocorreu no ecrã ‘*Timeline Mensagens*’ onde se definiu uma *store* que agrega instâncias de um modelo ‘*Mensagens*’ (modelo definido internamente com um conjunto de campos relativos às mensagens a apresentar) e utiliza um *proxy* do tipo *ScriptTagProxy* que permite enviar pedidos a um servidor localizado num domínio diferente através de *JSON-P* (o servidor está num domínio diferente da aplicação móvel). O servidor irá enviar uma resposta ao pedido proveniente da aplicação móvel no formato *JSON*, por isso é necessário definir no *proxy* a forma de descodificar os dados. O *proxy* tem um objecto de configuração *reader* que permite indicar o tipo de dados a receber, neste caso *JSON*.

4.5 Servidor Web

A necessidade de garantir comunicação entre aplicação móvel e plataforma de gestão de conteúdos desportivos levou à utilização de um servidor *Web*. Foi escolhido o *Apache Tomcat* (versão 7.0).

Este servidor tem uma implementação simples, onde realiza uma operação singular que consiste em receber o pedido *HTTP GET* do *browser/aplicação móvel*, identificar o último (mais recente) ficheiro *JSON* gerado de acordo com as preferências do utilizador e enviar um objecto do tipo *JSON* para a aplicação móvel que o descodifica e apresenta os dados ao utilizador.

5 Testes e Resultados

Neste capítulo apresentam-se dois tipos de testes desenhados para validar a plataforma de conteúdos desportivos e respectivos resultados. O primeiro tipo de testes permitiu validar o processo de avaliação de mensagens repetidas, com os casos específicos *retweet* associados à rede social *Twitter*. O segundo tipo de testes, mais complexos, permitiu avaliar a plataforma de agregação, categorização e indexação dos conteúdos desportivos no seu todo.

Para além dos testes apresentados de seguida, é importante referir que ao longo da implementação, tanto nos processos de *back-end* como no desenvolvimento da aplicação móvel, foram sendo realizados testes para validar o conjunto de operações desenvolvidas.

5.1 Duplicação de Mensagens

A verificação de duplicações é uma das operações fundamentais da plataforma para garantir que não são recolhidas mensagens repetidas. Dada a utilização da rede social *Twitter*, foi necessário prevenir o caso específico que acontece quando um utilizador faz *retweet* de uma mensagem. Assim, o algoritmo definido faz a verificação da existência de texto antes do padrão *retweet*, caso não existe qualquer texto antes do padrão é retirada essa parte da mensagem, caso contrário é comparada toda a mensagem.

5.1.1 Teste entre Duas Mensagens Totalmente Iguais

O primeiro teste realizado verifica a comparação entre duas mensagens totalmente iguais, sem qualquer delas conter o padrão *retweet*.

Na Figura 26, podemos observar a comparação entre duas mensagens totalmente iguais. Como já foi referido anteriormente, para a verificação de duplicações foi utilizado o coeficiente de similaridade de *Jaccard*, no caso do exemplo em cima calculado através do método “*removeDups*”. Dadas duas *strings* faz-se a computação do valor de similaridade e é devolvido o resultado no tipo de dados primitivo *double*. Caso seja superior a 0.8 (valor mínimo definido para duas mensagens serem consideradas duplicadas) a mensagem mais antiga é descartada. Neste exemplo, como podemos verificar o valor da comparação é 1.0, valor que identifica mensagens totalmente iguais, logo duplicadas.

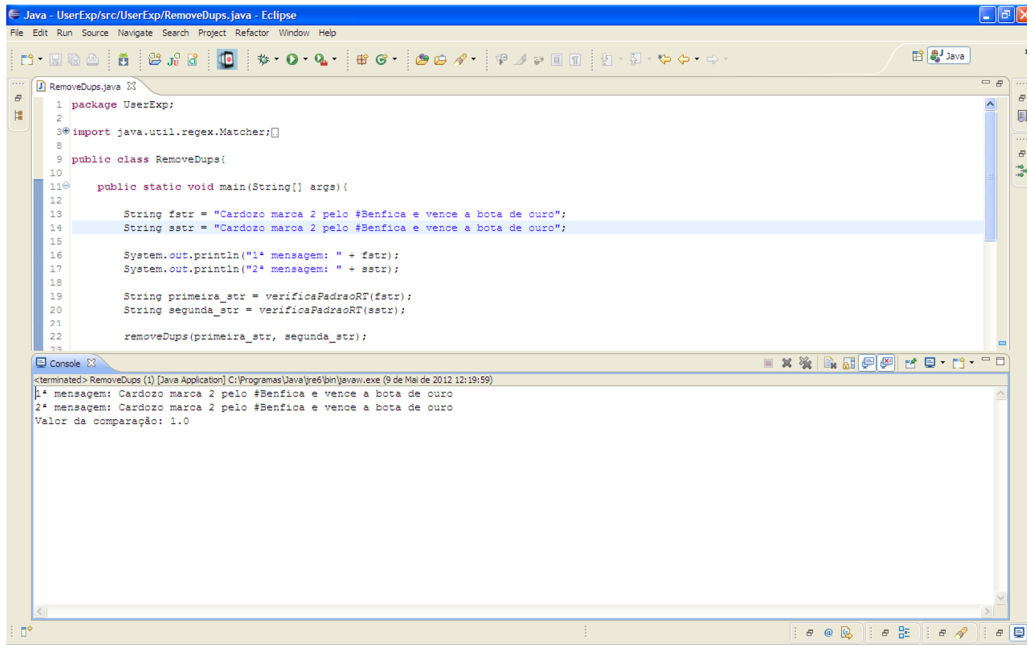


Figura 26 - Mensagens duplicadas totalmente iguais

5.1.2 Teste com Mensagem do Tipo retweet sem Texto antes do Padrão

O segundo teste realizado verifica a comparação entre duas mensagens totalmente iguais, no entanto uma das mensagens contém também o padrão *retweet* no seu início.

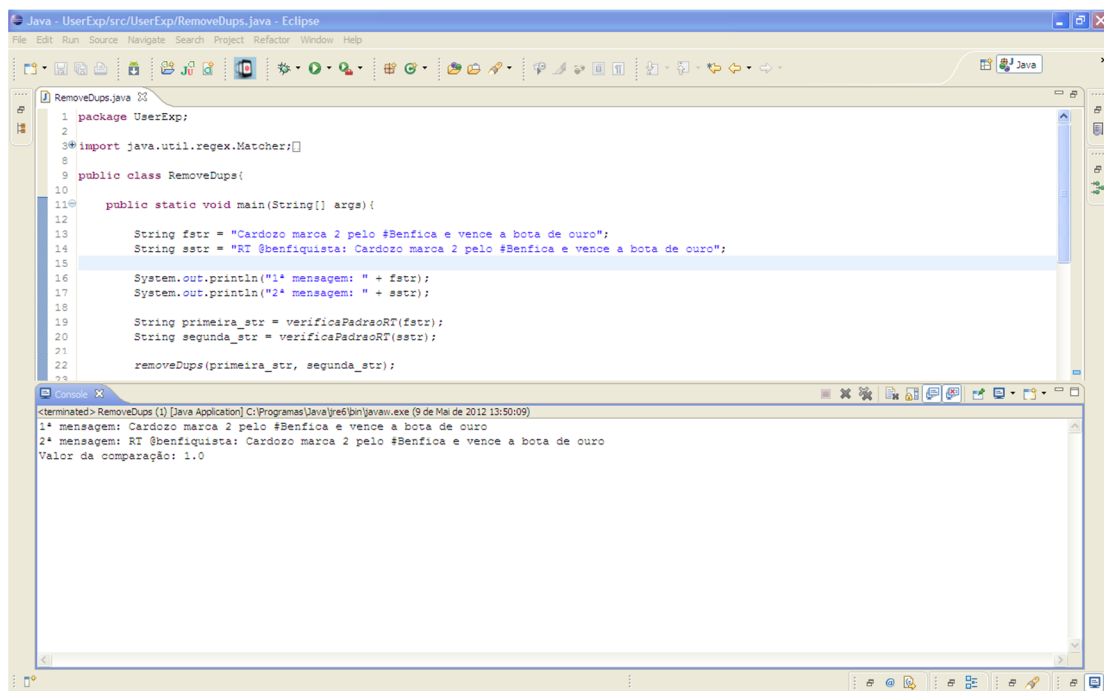


Figura 27 - Mensagens duplicadas sem texto antes do padrão retweet

Na Figura 27, podemos verificar que o valor da comparação entre as duas mensagens volta a ser 1.0. No entanto, como podemos observar a segunda mensagem tem o padrão *retweet* que sem a definição da regra de retirar esse padrão tornaria as duas mensagens substancialmente diferentes. O método “*verificaPadraoRT*” permite retirar o padrão específico, ficando a variável “*segunda_str*” apenas com o conteúdo importante da mensagem, conteúdo esse totalmente igual ao da primeira mensagem. Por isso, as mensagens são consideradas iguais, com coeficiente de similaridade de *Jaccard* igual a 1.0.

5.1.3 Teste com Mensagem do Tipo *retweet* com Texto antes do Padrão

O terceiro teste realizado verifica a comparação entre duas mensagens onde uma das mensagens contém toda a informação comparativamente à outra, o padrão *retweet* e um conjunto de informação antes desse padrão que poderá ser relevante.

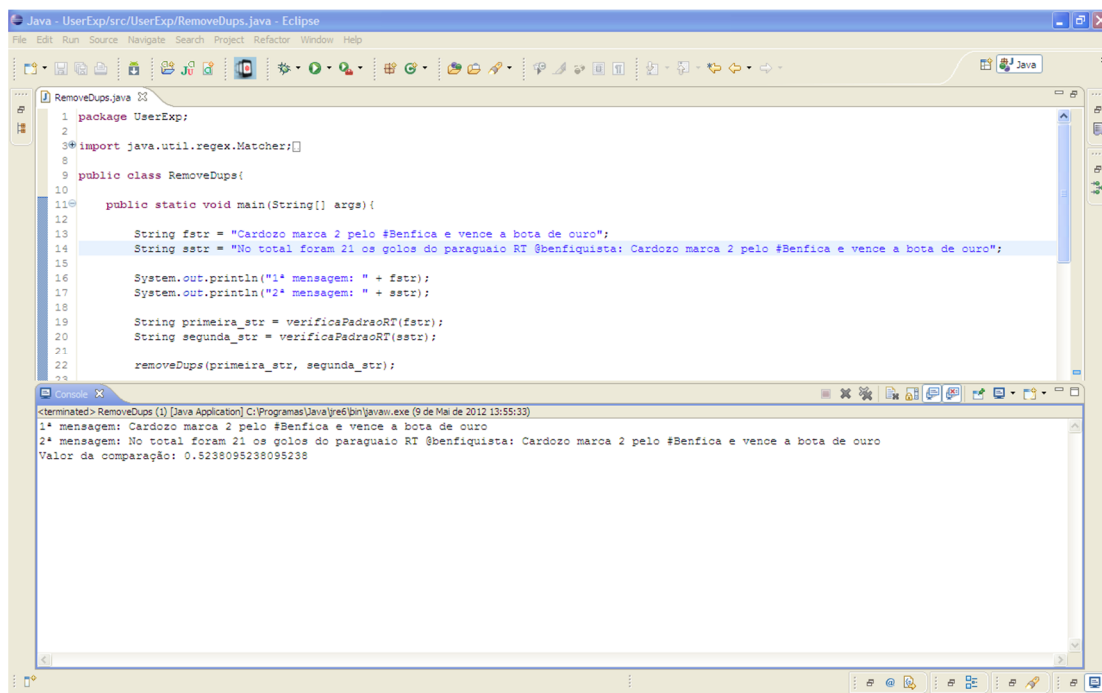


Figura 28 - Mensagens duplicadas com texto antes do padrão *retweet*

Na Figura 28, verificamos o terceiro caso possível. Embora a segunda mensagem seja um *retweet* da primeira, o utilizador acrescenta conteúdo que pode ter valor antes do padrão específico. Assim, a comparação é realizada sem mudar o conteúdo da segunda mensagem, o que resulta num valor de comparação inferior ao mínimo para o qual as mensagens são consideradas duplicadas. Neste caso, o resultado

do coeficiente de similaridade é aproximadamente 0.52, fazendo com que as duas mensagens sejam recolhidas na plataforma.

5.2 Plataforma de Agregação, Categorização e Indexação de Conteúdos

Para aferir ou validar o funcionamento da plataforma de agregação, categorização e indexação de conteúdos desportivos foram realizados um conjunto de testes que permitiram avaliar o conjunto de mensagens recolhidas. Foram aproveitados três eventos desportivos, um ligado ao ténis e dois ligados ao futebol, para colocar a plataforma em funcionamento. A metodologia utilizada foi a mesma para os três eventos: a plataforma recolhe 150 novas mensagens por “*hashtag*” a cada cinco minutos e apenas a duração de funcionamento da plataforma difere em cada teste.

De seguida são apresentados os testes e resultados para os três momentos de avaliação da plataforma, com a especificação dos parâmetros definidos.

5.2.1 Testes Federer vs Nadal

5.2.1.1 Dados Gerais do Teste

Na Tabela 14, estão descritos os dados que serviram de teste à plataforma, relativos ao evento desportivo Open da Austrália, referente à modalidade desportiva ténis.

| | |
|--------------------------|--|
| Data: | 26-01-2012 |
| Evento: | Meia-final do <i>Open</i> da Austrália |
| Modalidade: | Ténis |
| Hora Inicial: | 9:09 |
| Hora Final: | 9:39 |
| Duração: | 30 Minutos |
| Hashtags Equipa: | / |
| Hashtags Jogador: | #Federer, #Nadal |
| Hashtags Evento: | #ausopen |

Tabela 14 - Dados Gerais Federer vs Nadal

5.2.1.2 Resultados

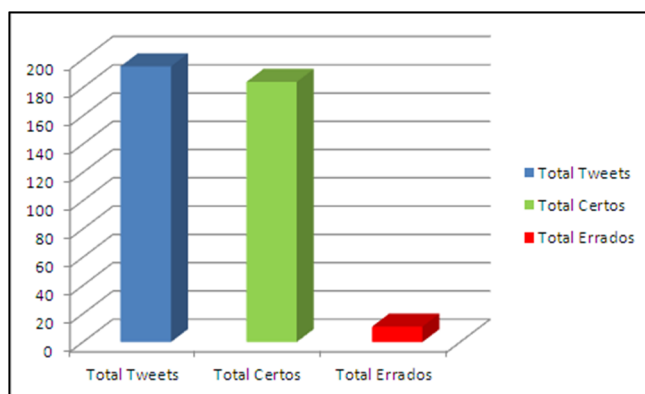


Figura 29 - Hashtag: #ausopen; Tipo: Geral

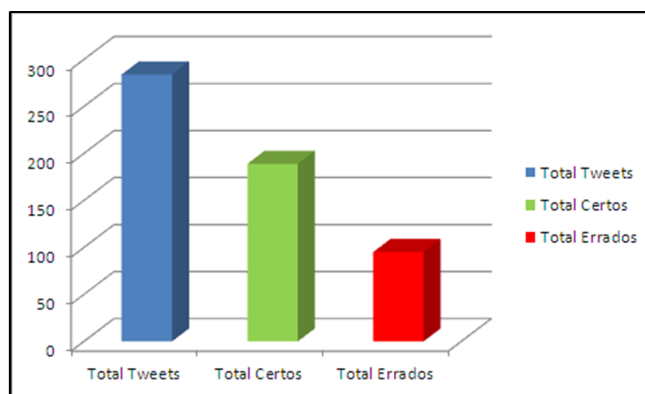


Figura 30 - Hashtag: #ausopen. Tipo: Informação

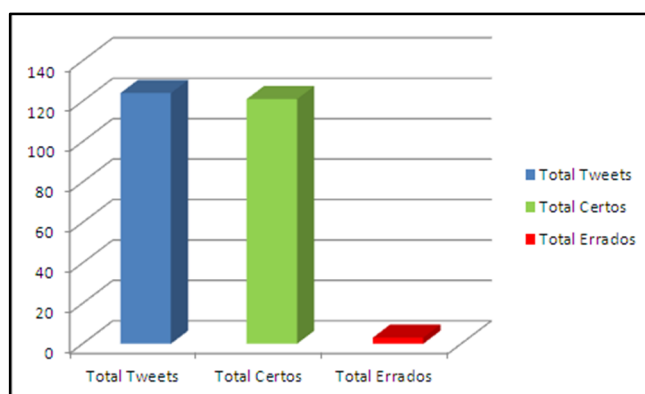


Figura 31 - Hashtag: #Federer. Tipo: Geral

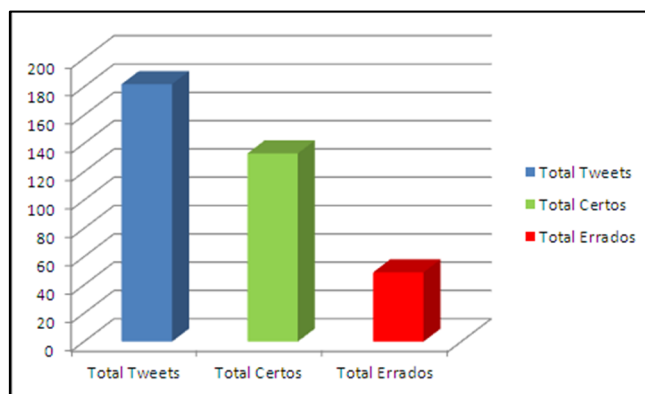


Figura 32 - Hashtag: #Federer. Tipo: Informação

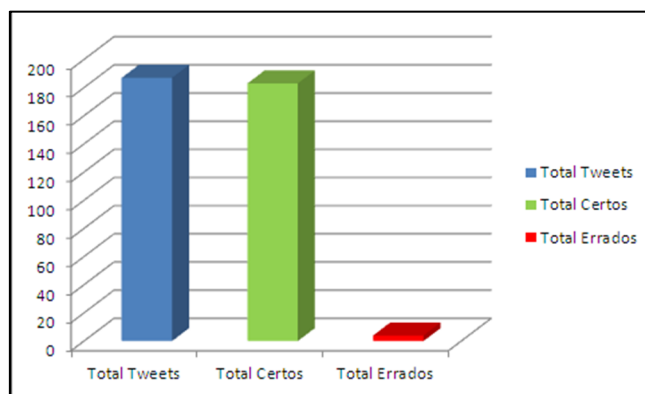


Figura 33 - Hashtag: #Nadal. Tipo: Geral

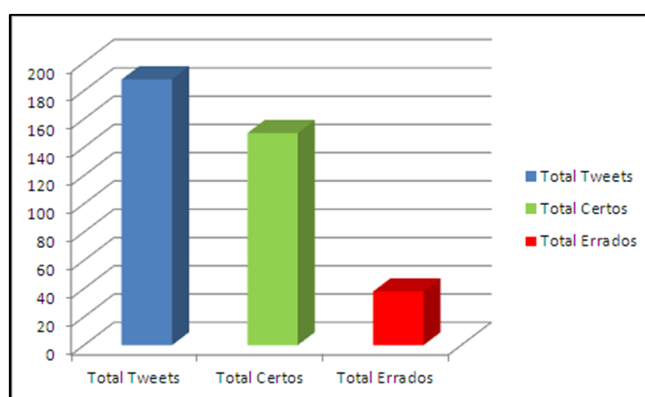


Figura 34 - Hashtag: #Nadal. Tipo: Informação

| <i>Hashtag</i> | <i>Tipo</i> | <i>Total Tweets</i> | <i>Total Certos</i> | <i>Total Errados</i> | <i>% Certos</i> |
|-----------------|-------------|---------------------|---------------------|----------------------|-----------------|
| #ausopen | Informação | 284 | 189 | 95 | 0,67 |
| #ausopen | Geral | 195 | 184 | 11 | 0,94 |
| #Federer | Informação | 182 | 133 | 49 | 0,73 |
| #Federer | Geral | 124 | 121 | 3 | 0,98 |
| #Nadal | Informação | 188 | 150 | 38 | 0,80 |
| #Nadal | Geral | 187 | 183 | 4 | 0,98 |

Tabela 15 - Resumo Resultados Federer vs Nadal

5.2.1.3 Análise dos Resultados

Um dos benefícios do sistema desenvolvido é dar a possibilidade ao utilizador da aplicação móvel escolher o tipo de mensagens que deseja ver na sua *timeline*. Assim, estes testes permitiram validar o processo de avaliação das mensagens recolhidas. Como aspectos mais relevantes a notar temos:

- 1) A categorização das mensagens do tipo “Informação” obteve bons resultados embora menos positivos comparativamente às mensagens do tipo “Geral”, tanto para a “*hashtag*” de evento como para as “*hashtags*” de jogadores. Estes resultados menos positivos deveram-se essencialmente a dois factores: à utilização de palavras-chave por parte dos utilizadores para incentivar os jogadores (ex. “*C'mon Roger let's break an take the first set! #Federer*”), ou à utilização de palavras-chave por parte dos utilizadores para expressar os seus sentimentos do momento (ex. “*I gotta feeling. Its gonna be a spectacular epic match once again. #Federer v #Rafa #aussieopen*”). Destes resultados, foi tomada a decisão de retirar algumas palavras-chave da lista que identificava as mensagens como do tipo Informação (a palavra ‘*love*’, muito utilizada no ténis para indicar um resultado, era uma das palavras-chave definida que foi retirada por ser muito utilizada para expressar admiração pelos jogadores);
- 2) A quantidade de mensagens recolhidas para os dois tipos possíveis definidos, revelou um grande equilíbrio no número de mensagens associadas aos mesmos. Isto possibilita a utilização da aplicação móvel tanto por parte de utilizadores que estejam mais interessados em saber apenas informação

específica do evento desportivo, como utilizadores que, por exemplo, tenham interesse em descobrir outras opiniões e sentimentos de pessoas que estejam a assistir ao evento e poderão nutrir admiração por um mesmo atleta.

O conjunto de mensagens recolhidas prova que nas redes sociais existem diversos utilizadores que gostam de partilhar informação desportiva relevante acerca de um evento que estejam a assistir, dando valor a este novo paradigma baseado em informações provenientes dos próprios utilizadores. Assim, como alguns exemplos das mensagens recolhidas temos:

- 1) *“Sharp backhand volley by #Federer. 3 set points, 6-3 1st set tiebreak. #Nadal serving. #ausopen semifinal”*;
- 2) *“Career won-lost record in tiebreaks (all singles matches): #Federer 307-157 (66%), #Nadal 140-87 (62%) ausopen”*;
- 3) *“Erros não-forçados: #Federer 14 x @RafaelNadal 10. #Federer acertou um e errou outro desafio. % PG 1º saque, 72x77. Porém, 1 mini-break”*.

5.2.2 Testes Sporting Clube de Portugal vs Sport Lisboa e Benfica

5.2.2.1 Dados Gerais do Teste

Na Tabela 16, estão descritos os dados que serviram de teste à plataforma, relativos ao evento desportivo Liga ZON Sagres 2011/2012, referente à modalidade desportiva futebol.

| | |
|--------------------------|---|
| Data: | 09-04-2012 |
| Evento: | 26ª Jornada da Liga ZON Sagres 2011/2012 |
| Modalidade: | Futebol |
| Hora Inicial: | 20:00 |
| Hora Final: | 22:00 |
| Duração: | 2 Horas |
| Hashtags Equipa: | #Sporting, #Benfica |
| Hashtags Jogador: | / |
| Hashtags Evento: | / |

Tabela 16 - Dados Gerais Sporting vs Benfica

5.2.2.2 Resultados

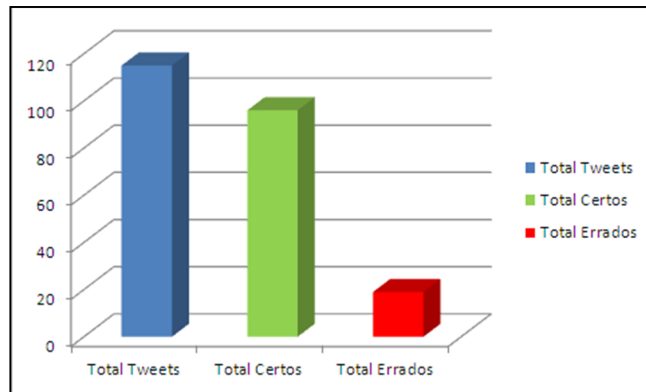


Figura 35 - Hashtag: #Sporting. Tipo: Informação

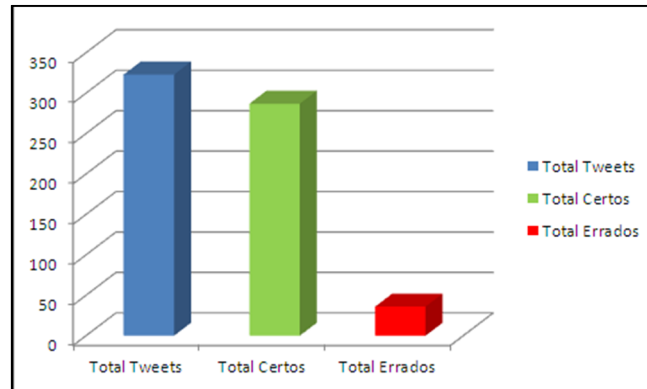


Figura 36 - Hashtag: #Sporting. Tipo: Geral

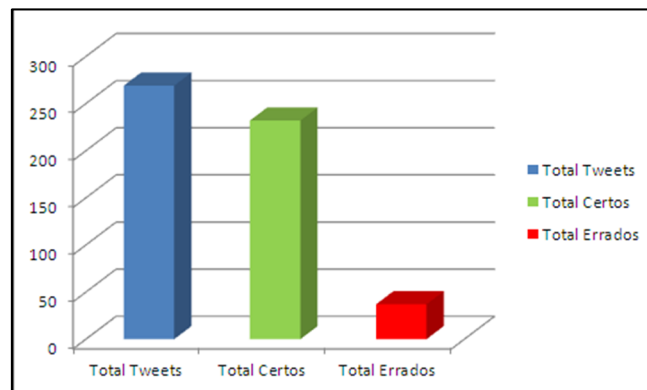


Figura 37 - Hashtag: #Benfica. Tipo: Geral

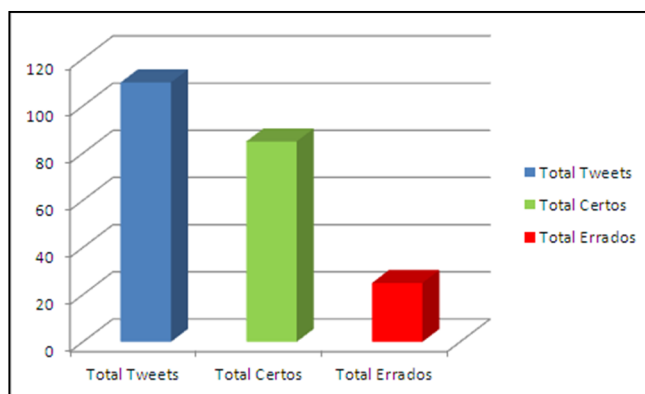


Figura 38 - Hashtag: #Benfica. Tipo: Informação

| Hashtag | Tipo | Total Tweets | Total Certos | Total Errados | % Certos |
|-----------|------------|--------------|--------------|---------------|----------|
| #Sporting | Informação | 115 | 96 | 19 | 0,83 |
| #Sporting | Geral | 323 | 287 | 36 | 0,89 |
| #Benfica | Informação | 110 | 85 | 25 | 0,77 |
| #Benfica | Geral | 269 | 232 | 37 | 0,86 |

Tabela 17 - Resumo Resultados Sporting vs Benfica

5.2.2.3 Análise dos Resultados

O segundo teste da plataforma difere em larga escala do realizado anteriormente, tanto por ser feito tendo em conta uma modalidade desportiva diferente, como por ser um evento nacional ao contrário do teste anterior, onde se avaliou um dos mais emblemáticos torneios de ténis a nível mundial. Neste caso, foi testado o jogo de futebol entre duas equipas que rivalizam em Portugal há mais de cem anos e simbolizam o chamado “Derby Eterno” ou “Derby da Capital”, característica que poderá ter influenciado os resultados obtidos. Assim, e como resultados principais a destacar temos:

- 1) A quantidade de mensagens categorizadas com o tipo Geral foi bastante superior às mensagens do tipo Informação. Este facto, terá ocorrido devido ao facto do futebol ser um desporto em que a expressão individual tem mais relevo no sentido de se contestar decisões do árbitro, pôr em causa determinado jogador, etc. Também terá contribuído para estes valores, o facto

de ser um *derby* entre duas equipas da mesma cidade que eleva as quezílias entre adeptos rivais. Apesar destes valores, o número de mensagens do tipo “Informação” não deixa de ser relevante e possibilitar recolher inúmeras informações oportunas relativas ao evento em questão;

- 2) A precisão na categorização dos dois tipos de mensagens foi menos díspar relativamente aos resultados apresentados no teste anterior. Embora as mensagens do tipo “Geral” continuem a obter melhores resultados, os valores apresentados para o tipo Informação aproximam-se de uma média de precisão de 80%.

Foi possível apurar diversas mensagens que demonstram a relevância da plataforma desenvolvida. Alguns exemplos dessas mensagens são:

- 1) “*Golo do #Sporting. Marca Van Wolfswinkel, de penákti. 1-0. #SCP #Benfca #SLB*”;
- 2) “*HT: #Sporting 1-0 #Benfca. Goal: SCP - 18' Ricky van Wolfswinkel (PK). Four players on a yellow. Two a piece*”;
- 3) “*wow Artur denied Van Wolfswinkel at a 1 on 1 run great save #sporting #Benfca*”.

5.2.3 Testes Sport Lisboa e Benfica vs Gil Vicente Futebol Clube

5.2.3.1 Dados Gerais do Teste

Na Tabela 18, estão descritos os dados que serviram de teste à plataforma, relativos ao evento desportivo Taça da Liga 2011/2012, referente à modalidade desportiva futebol.

| | |
|--------------------------|---------------------------------|
| Data: | 14-04-2012 |
| Evento: | Final da Taça da Liga 2011/2012 |
| Modalidade: | Futebol |
| Hora Inicial: | 20:00 |
| Hora Final: | 23:00 |
| Duração: | 3 horas |
| Hashtags Equipa: | #Benfca, #GilVicente |
| Hashtags Jogador: | / |
| Hashtags Evento: | #tliga |

Tabela 18 - Dados Gerais Benfica vs Gil Vicente

5.2.3.2 Resultados

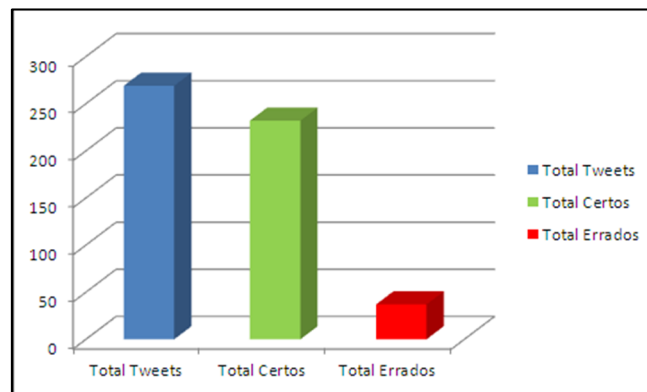


Figura 39 - Hashtag: #Benfica. Tipo: Geral

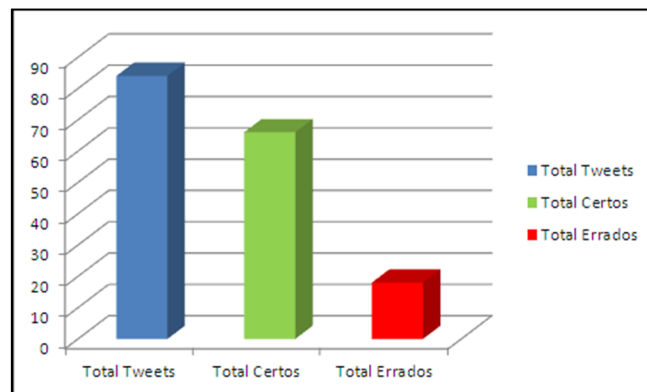


Figura 40 - Hashtag: #Benfica. Tipo: Informação

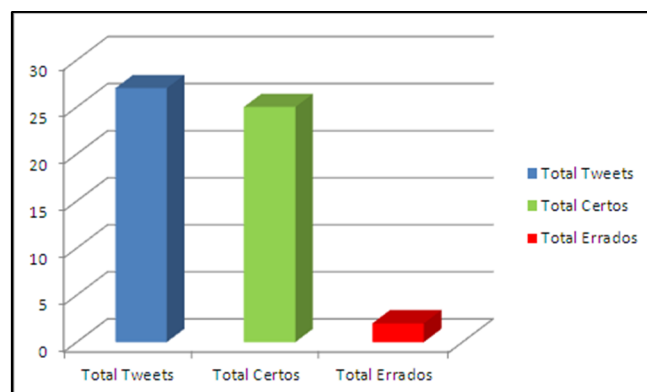


Figura 41 - Hashtag: #GilVicente. Tipo: Geral

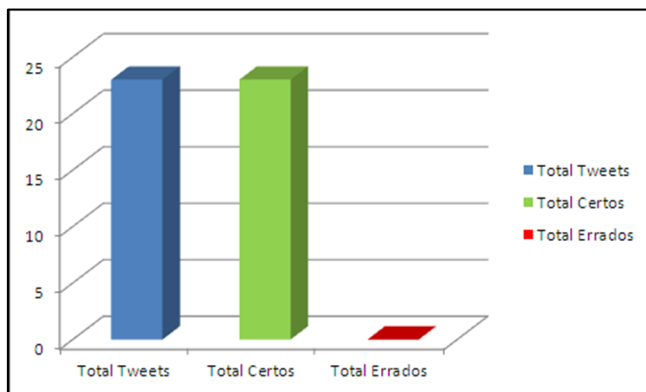


Figura 42 - Hashtag: #GilVicente. Tipo: Informação

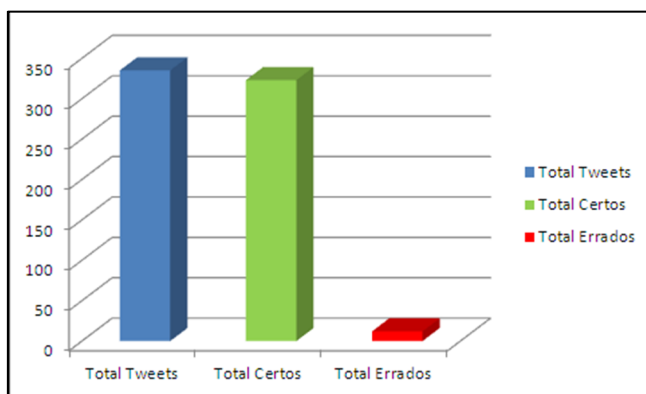


Figura 43 - Hashtag: #tliga. Tipo: Geral

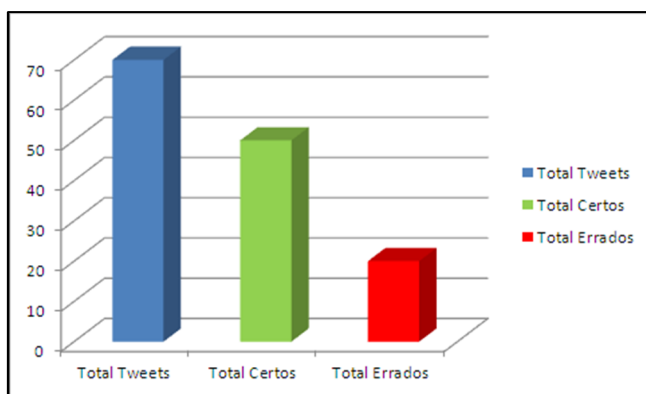


Figura 44 - Hashtag: #tliga. Tipo: Informação

| <i>Hashtag</i> | Tipo | Total Tweets | Total Certos | Total Errados | % Certos |
|--------------------|-------------|-------------------------|-------------------------|--------------------------|---------------------|
| #tliga | Informação | 70 | 50 | 20 | 0,71 |
| #tliga | Geral | 335 | 323 | 12 | 0,96 |
| #Benfica | Informação | 84 | 66 | 18 | 0,79 |
| #Benfica | Geral | 298 | 263 | 35 | 0,88 |
| #GilVicente | Informação | 23 | 23 | 0 | 1,00 |
| #GilVicente | Geral | 27 | 25 | 2 | 0,93 |

Tabela 19 - Resumo Resultados Benfica vs Gil Vicente

5.2.3.3 Análise dos Resultados

No conjunto final de testes, a plataforma foi testada novamente para um evento relativo à modalidade desportiva Futebol. Neste caso, a grande diferença relativa ao teste anterior, reside no facto de ter sido testada uma “*hashtag*” de evento - #tliga, representativa da final da Taça da Liga, época 2011/2012 – para além das “*hashtags*” principais das duas equipas envolvidas. As características principais dos resultados obtidos são as seguintes:

- 1) A diferente importância das duas equipas no panorama nacional ficou evidenciada pelo número de mensagens referentes às duas “*hashtags*” de equipas envolvidas. A “*hashtag*” relativa à equipa do Gil Vicente, recolhe um número de mensagens muito inferior ao ocorrido para a “*hashtag*” relativa à equipa do Benfica;
- 2) Reforçando o resultado obtido no teste anterior, com excepção da “*hashtag*” relativa à equipa do Gil Vicente, a quantidade de mensagens do tipo “Geral” foi bastante superior às mensagens específicas de informação, o que apoia a tese de que o futebol, comparativamente ao ténis, revela-se uma modalidade onde os utilizadores expressam mais os seus sentimentos relativos ao evento a que estão a assistir;
- 3) Relativamente à precisão na categorização das mensagens, mais uma vez os resultados revelaram-se melhores na avaliação de mensagens do tipo “Geral”, com excepção da “*hashtag*” relativa à equipa do Gil Vicente que

com um número reduzido de mensagens conseguiu atingir a máxima precisão.

Tal como no teste anterior, algumas mensagens merecem destaque pela importância e suporte que prestam à utilização desta plataforma. Alguns exemplos são:

- 1) *“1-0 #Benfica...Rodrigo with the opener .. credit to Cesar for a fine tackle to win the ball and then deliver an inch perfect cross!”*;
- 2) *“Fim do jogo: #Benfica 2-1 #GilVicente. É a quarta Taça da Liga consecutiva para o #SLB, que não perde há 20 jogos na competição. #tliga”*.

A análise dos resultados permitiu, no caso da modalidade desportiva futebol, encontrar novas palavras-chave que permitirão melhorar a precisão de categorização das mensagens. Algumas palavras novas identificadas a partir das mensagens recolhidas são: cruzamento, livre, convocado(s), titular(es), intervalo, etc.

Integração e divulgação de informação desportiva em redes sociais através de dispositivos móveis

6 Conclusões

6.1 Conclusões Gerais

A ideia inicial que despoletou o desenvolvimento deste trabalho baseou-se na criação de um meio alternativo para disponibilização de conteúdos desportivos. Os utilizadores com interesse nesse tipo de informações estão muitas vezes limitados às escolhas das entidades que gerem as aplicações. Assim, e apostando num novo paradigma, onde a informação parte dos próprios utilizadores, foi possível desenvolver um projecto cimentado em conceitos modernos e em grande expansão como são as redes sociais e o desenvolvimento de aplicações móveis.

Através da revisão de literatura foi possível demonstrar a importância das redes sociais no quotidiano das pessoas e definir que o *Twitter* seria uma fonte rica e ampla em termos de conteúdos desportivos, para além de existir uma aposta em agregar conteúdos a partir da própria aplicação móvel. A literatura existente permitiu ainda tomar decisões importantes relativas a técnicas e ferramentas de categorização e indexação de informação, duas questões essenciais no trabalho desenvolvido. Dado o propósito muito específico deste trabalho, das técnicas mais comuns apresentadas procurou-se extrair a forma como cada uma poderia ajudar no problema a resolver, tendo sido pensada uma solução à medida sem recorrer a algoritmos muito complexos. Por outro lado, demonstrou-se que actualmente existe um enorme mercado ligado ao desenvolvimento de aplicações para plataformas móveis, sendo tomada a opção de explorar uma solução baseada em desenvolvimento *web*, através da *framework Sencha Touch*, com o intuito de não restringir a utilização da aplicação a um único sistema operativo ou tipo de dispositivo.

Através da elaboração de um conjunto de requisitos foi possível determinar as necessidades do projecto e dividir o conjunto de funcionalidades a desenvolver. Destas necessidades surgiu a definição do modelo que sustentou a plataforma de gestão de conteúdos desportivos e a aplicação móvel.

A implementação deste projecto dividiu-se em duas partes essenciais, modeladas anteriormente. Por um lado, implementaram-se três funções essenciais: agregação de conteúdos, recorrendo à ferramenta *Twitter4J*, categorização de informação, com a definição de técnicas específicas e fazendo uso de expressões regulares na detecção de padrões específicos na informação, e ainda indexação de conteúdos, com recurso à

ferramenta *Lucene*, que permitiu manter os dados de interesse dos utilizadores. Por outro lado, apostou-se numa abordagem híbrida de desenvolvimento de aplicações móveis, recorrendo à ferramenta *Sencha Touch*, tendo como resultado uma aplicação que permite aos seus utilizadores parametrizarem e visualizarem toda a informação de acordo com os seus gostos, permitindo ainda alimentar directamente a plataforma de conteúdos desportivos.

Os resultados obtidos, relativos a eventos desportivos reais, demonstram a validade do trabalho desenvolvido e/ou da solução proposta. Foi possível validar todo o processo de gestão dos conteúdos existentes na plataforma, sendo ainda testada e validada a aplicação móvel que permite ao utilizador gerir os seus conteúdos desportivos de interesse.

6.2 Contribuições

O desenvolvimento deste trabalho contribui para a exploração de um paradigma distinto no que toca à disponibilização de conteúdos, neste caso desportivos, a utilizadores que apostam na mobilidade e/ou ubiquidade da informação. Assim, e em sintonia com os objectivos propostos para esta dissertação podemos enumerar as seguintes contribuições:

- 1) Estudo e explicação de um conjunto de ferramentas *open-source* que facilitam processos de agregação, categorização e indexação de dados (*Twitter4J* e *Lucene*);
- 2) Especificação, modelação e implementação de uma infra-estrutura de suporte a gestão de conteúdos desportivos que integra informações recolhidas a partir de fontes distintas;
- 3) Especificação e implementação de uma aplicação móvel que permite gestão e acesso a conteúdos desportivos de acordo com as preferências dos utilizadores.

Para além das anteriores contribuições, esta dissertação deixou o seu contributo para a comunidade científica, ainda numa fase inicial do seu desenvolvimento, através da submissão e aceitação do artigo denominado “Agregação e categorização de informação desportiva baseada em conteúdos gerados por utilizadores”, apresentado à comunidade na 11^a Conferência da Associação Portuguesa de Sistemas de Informação

(CAPSI 2011) que decorreu entre os dias 19 e 21 de Outubro de 2011 (Carvalho, et al., 2011).

6.3 Limitações e Trabalho Futuro

No âmbito desta dissertação propõe-se como trabalho futuro as seguintes indicações:

- 1) A plataforma que gere os conteúdos desportivos agrega informação proveniente do *Twitter* e da aplicação móvel. Seria interessante testar a recolha de informação proveniente de outras fontes/redes sociais como *Facebook*, *Google Reader*, *Blogs*, entre outros;
- 2) A plataforma foi testada para duas modalidades (futebol e ténis) e seria interessante estender o número de modalidades, como por exemplo ciclismo, basquetebol, desportos motorizados, entre outros;
- 3) A metodologia utilizada na agregação de conteúdos para a plataforma foi baseada na utilização de “*hashtags*”. Também seria interessante explorar a pesquisa baseada em utilizadores específicos. Por exemplo, contas do *Twitter* que sigam determinada equipa desportiva ou jogador;
- 4) A nível da categorização das mensagens seria interessante abordar novas técnicas que permitissem ainda melhores resultados na funcionalidade de diferenciar *tweets* com informação pertinente e *tweets* que embora contenham palavras-chave que simbolizam informação são apenas opiniões pessoais ou o libertar de frustrações. Nesse sentido, e em paralelo com as ideias definidas para o *Twitómetro* (Silva, et al., 2011) onde é feita uma avaliação dos sentimentos revelados nas mensagens tendo por base um conjunto de recursos linguísticos, como léxicos de polaridade ou padrões léxico-sintácticos, seria necessário definir um conjunto de regras que poriam em evidência expressões relativas a sentimentos na mensagem, e nesse caso essas mensagens não seriam categorizadas como informação desportiva credível;
- 5) Em termos de escalabilidade, seria interessante ter uma solução distribuída, onde os índices seriam geridos a partir de diversos *clusters* permitindo obter uma solução mais adaptada ao natural crescimento da plataforma e dos conteúdos

geridos através da mesma. Nesse caso, o *Apache Hadoop*³⁰ seria uma solução fiável capaz de realizar computação e armazenamento distribuído, processando grandes volumes de informação por vários sistemas computacionais;

- 6) Ao nível da aplicação móvel, embora seja possível visualizar conteúdos multimédia mais ricos, como fotografias ou vídeos, existentes numa mensagem apresentada ao utilizador, ainda não é possível carregar esse tipo de multimédia a partir da própria aplicação móvel. Também seria de grande interesse desenvolver esta funcionalidade.

³⁰ <http://hadoop.apache.org/>

7 Bibliografia

140Dev. 2010. [Online] 2010. [Citação: 12 de Fevereiro de 2012.] <http://140dev.com/twitter-api-programming-tutorials/aggregating-tweets-search-api-vs-streaming-api/>.

Apache Software Foundation. 2011. [Online] 2011. [Citação: 28 de Janeiro de 2012.] <http://lucene.apache.org/core/features.html>.

ArticlesBase.com. 2010. [Online] 01 de Dezembro de 2010. [Citação: 08 de Maio de 2011.] <http://www.articlesbase.com/smo-articles/history-and-statisticaldata->.

Asshay, J., et al. 2007. Why We Twitter: Understanding Microblogging Usage and Communities. *WebKDD/SNA-KDD '07 Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*. 2007.

Bada. 2011. [Online] 2011. [Citação: 08 de Julho de 2012.] <http://www.bada.com/whatisbada/index.html>.

Bank, Jacob e Cole, Benjamin. 2008. [Online] 16 de Dezembro de 2008. [Citação: 13 de Fevereiro de 2012.] <http://www.infosci.cornell.edu/weblab/papers/Bank2008.pdf>.

Blei, D. M. 2011. Introduction to probabilistic topic models. *Communications of the ACM*. 2011.

Blei, D. M., Ng, A. Y. e Jordan, M. I. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*. 3, 2003, pp. 993-1022.

Boyd, Danah M. e Ellison, Nicole B. 2008. Social Network Sites: Definition, History, and Scholarship. *Journal of Computer-Mediated Communication*. 2008.

Carvalho, Sérgio e Serrão, Carlos. 2011. Agregação e categorização de informação desportiva baseada em conteúdos gerados por utilizadores. *11ª Conferência da Associação Portuguesa de Sistemas de Informação*. 19-21 Outubro de Outubro de 2011, pp. pp. 73 ISBN: 978-972-8862-12-1.

Charland, Andre e LeRoux, Brian. 2011. Mobile Application Development: Web vs. Native. *Communication of the ACM*. Maio de 2011, Vols. 54, No. 5.

Cohen, William W. 2011. [Online] 2011. [Citação: 13 de Fevereiro de 2012.] <http://secondstring.sourceforge.net/javadoc/com/wcohen/secondstring/Jaccard.html>.

Deerwester, S., et al. 1990. Indexing by latent semantic analysis. 1990, Vol. 41, 6, pp. 391-407.

Eraç, B. e Gülay, Z. 2007. Molecular epidemiology of PER-1 extended spectrum beta-lactamase among gram-negative bacteria isolated at a tertiary care hospital. *Folia Microbiologica*. 2007, Vols. 52(5):535-41.

GSMarena.com. 2012. [Online] 15 de Fevereiro de 2012. [Citação: 05 de Julho de 2012.] http://www.gsmarena.com/gartner_releases_phone_market_share_report_for_2011-news-3832.php.

Gupta, T., et al. 2009. Characterization of friendfeed - a web-based social aggregation service. *Proc. AAAI ICWSM*. 2009.

Hong, L. e Davison, B. D. 2010. Empirical study of topic modeling in Twitter. *Proceedings of the First Workshop on Social Media Analytics*. 2010.

Iana. 2009. [Online] 29 de Julho de 2009. [Citação: 12 de Fevereiro de 2012.] <http://www.iana.org/assignments/language-subtag-registry>.

Jain, A. K., Murty, M. N. e Flynn, P. J. 1999. Data clustering: a review. *ACM Computing Surveys*. 1999, Vols. 31(3):264–323.

Lionbridge. 2012. [Online] 2012. [Citação: 08 de Julho de 2012.] <http://en-us.lionbridge.com/kc/mobile-web-apps-vs-mobile-native-apps.htm>.

Maisfutebol. 2011. [Online] 2011. [Citação: Março de 13 de 2011.] <http://www.maisfutebol.iol.pt/twitter.html>.

Maisfutebol. 2011. [Online] 27 de Fevereiro de 2011. [Citação: 13 de Março de 2011.] http://www.maisfutebol.iol.pt/horas-vagas/noticias-na-hora-futebol-internacional-marca-as-lequipe-ole/1230194-1478.html?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%253A+iol%252Fmaisfutebol+%2528maisfutebol%2529&utm_content=Google+Reader .

Manning, C., Raghavan, P. e Schütze, H. 2008. Introduction to Information Retrieval. s.l. : Cambridge University Press, 2008.

Matteucci, Matteo. 2004. [Online] 2004. [Citação: 10 de Maio de 2012.] http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/index.html.

Meios e Publicidade. 2012. [Online] 2012. [Citação: 7 de Julho de 2012.] <http://www.meiosepublicidade.pt/2012/06/19/twitteuro-no-sapo/>.

Messina, A. e Montagnuolo, M. 2008. Content-based RSS and broadcast news streams aggregation and retrieval. *ICDIM 2008*. Third International Conference on 13-16 Nov, 2008.

Microsoft. 2012. [Online] 20 de Junho de 2012. [Citação: 08 de Julho de 2012.] http://blogs.windows.com/windows_phone/b/windowsphone/archive/2012/06/20/announcing-windows-phone-8.aspx.

Oracle. 2011. [Online] 2011. [Citação: 28 de Janeiro de 2012.] <http://docs.oracle.com/javase/7/docs/api/java/util/regex/package-summary.html>.

Oracle. 2011. [Online] 2011. [Citação: 14 de Fevereiro de 2012.] <http://docs.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html>.

Oracle. 2011. [Online] 2011. [Citação: 14 de Fevereiro de 2012.] <http://docs.oracle.com/javase/6/docs/api/java/util/regex/Matcher.html>.

PCPro. 2010. [Online] 25 de 10 de 2010. [Citação: 20 de Março de 2011.] <http://www.pcpro.co.uk/news/365062/smartphone-shipments-overtake-pcs>.

Ramage, D., Dumais, S. e Liebling, D. 2010. Characterizing microblogs with topic models. *International AAAI ICWM*. 2010.

Ramage, D., et al. 2009. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. 2009, pp. 248-256.

Ramos, J. 2003. Using tf-idf to determine word relevance in document queries. *First International Conference on Machine Learning*. 2003.

Rosen-Zvi, M., et al. 2004. The Author-Topic Model for Authors and Documents. *20th Conference on Uncertainty in Artificial Intelligence*. 2004.

Sarmiento, L., et al. 2009. Automatic Creation of a Reference Corpus for Political Opinion Mining in User-Generated Content. *1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion Measurement*. 2009.

Sencha Docs. 2011. [Online] 2011. [Citação: 28 de Janeiro de 2012.] http://docs.sencha.com/touch/2-0/#!/guide/class_system.

Sencha Inc. 2011. [Online] 2011. [Citação: 28 de Janeiro de 2012.] <http://www.sencha.com/products/touch>.

Sencha Inc. 2011. [Online] 2011. [Citação: 2012 de Janeiro de 2012.] <http://www.sencha.com/products/touch/features/>.

Silva, Mário J. e REACTION TEAM. 2011. [Online] 16 de Maio de 2011. [Citação: 15 de Maio de 2012.] http://xldb.fc.ul.pt/xldb/publications/Silva.etal:NotasSobreA:2011_document.pdf.

Spencer, Ed. 2011. [Online] 11 de Outubro de 2011. [Citação: 28 de Janeiro de 2012.] <http://www.sencha.com/blog/sencha-touch-2-developer-preview/>.

Spencer, Ed. 2010. [Online] 27 de Maio de 2010. [Citação: 21 de Fevereiro de 2012.] <http://www.sencha.com/blog/the-html5-family-web-storage>.

Spencer, Ed. 2011. [Online] 11 de Julho de 2011. [Citação: 14 de Maio de 2012.] <http://www.sencha.com/learn/using-the-data-package-in-sencha-touch/>.

Spencer, Ed. 2011. [Online] 19 de Janeiro de 2011. [Citação: 28 de Janeiro de 2012.] <http://lucene.apache.org/core/features.html>.

Steinbach, Michael, Kumar, Vipin e Tan, Pang-Ning. 2004. [Online] 18 de Abril de 2004. [Citação: 10 de Maio de 2012.] http://www-users.cs.umn.edu/~kumar/dmbook/dmslides/chap8_basic_cluster_analysis.pdf.

Steyvers, M. e Griffiths, T. 2007. Probabilistic topic models. *T. Landauer, D. S. McNamara, S. Dennis, & W. Kintsch. Handbook of Latent Semantic Analysis, 2007.*

Taborda, M., Cardoso, G. e Espanha, R. 2010. A Utilização de Internet em Portugal. *World Internet Project 2010 e LINI. 2010.*

Techopedia. 2010. [Online] 2010. [Citação: 13 de Janeiro de 2012.] <http://www.techopedia.com/definition/25843/regular-expression>.

Twitter. 2011. [Online] 2011. [Citação: 12 de Novembro de 2011.] <https://support.twitter.com/articles/49309-what-are-hashtags-symbols>.

Twitter. 2011. [Online] 2011. [Citação: 13 de Novembro de 2011.] <https://support.twitter.com/articles/14023>.

Twitter. 2011. [Online] 2011. [Citação: 13 de Novembro de 2011.] <http://support.twitter.com/articles/77606-faqs-about-retweets-rt>.

Twitter. 2011. [Online] 2011. [Citação: 12 de Fevereiro de 2012.] <https://dev.twitter.com/docs/history-rest-search-api>.

Twitter. 2011. [Online] 2011. [Citação: 12 de Fevereiro de 2012.] <https://dev.twitter.com/docs/rate-limiting#search>.

Twitter4J. 2007. [Online] 2007. [Citação: 12 de Janeiro de 2012.] <http://twitter4j.org/en/api-support.html>.

Wikipédia. 2007. [Online] 12 de Setembro de 2007. [Citação: 23 de 01 de 2011.] <http://pt.wikipedia.org/wiki/Microblogging>.

Xu, C., et al. 2006. Live sports event detection based on broadcast video and web-casting text. *Proceedings of the 14th annual ACM international conference on Multimedia. 2006.*

Yifan, Z., et al. 2007. Semantic Event Extraction from Basketball Games using Multi-Modal Analysis. *Multimedia and Expo, IEEE International Conference on 2-5 July 2007. 2007.*

Zhao, W., et al. 2011. Comparing twitter and traditional media using topic models. *The 33rd European Conference on Information Retrieval. 2011.*

8 Anexos

Todo o código escrito no desenvolvimento deste projecto encontra-se disponível através do URL: <https://github.com/SergioCarvalho/SocialSports>.