Mestrado em Engenharia Informática
Estágio
Relatório Final

SESE - Search Quality and Analytics

João Fernando Costa Moura
jmoura@student.dei.uc.pt

Orientador da Wizdee:
Mestre Bruno Antunes

Orientador do DEI:
Prof. Dr. Pedro Abreu

Data: 12 de Julho de 2012

FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Aos meus pais,
e a todos os que tiveram algo a ensinar-me.

Resumo

Actualmente com o aumento da informação disponível na internet para o utilizador torna-se necessário que os sistemas de recolha de informação e motores de pesquisa, sejam capazes de responder a todas as necessidades de informação, apresentando resultados relevantes e em tempo útil.

Para tornar este cenário possível, é necessário existir preocupações tanto a nível de disponibilidade de informação bem como a nível de uso e gestão. Neste âmbito surgem sistemas de Qualidade de Pesquisa (QP) e Análise de Pesquisa (AP) que por um lado fornecem informação sobre a qualidade e desempenho contribuindo para melhorar o motor de pesquisa (tornando-o mais rápido e preciso) e por outro lado fornece informação relativa ao modo como um utilizador procura informação e o seu comportamento perante os resultados apresentados. AP pode ainda ser usado como uma mais valia no negócio, caso se aplique, uma vez que apresenta as tendências dos utilizadores e suas necessidades actuais.

Este trabalho tem como objectivo dotar o motor de pesquisa da Wizdee com um sistema de QP e de AP que permita efectuar a monitorização necessária, apresentando relatórios informativos da sua qualidade, desempenho e modo de utilização.

Palavras-chave: Qualidade de Pesquisa, Análise de Pesquisa, Relevância, Precisão, Abrangência, Motor de Pesquisa

Agradecimentos

Quero deixar um agradecimento especial aos meus pais por terem sempre feito os possíveis para me proporcionar uma vida cheia de oportunidades e por me terem sempre incentivado no meu percurso académico. A eles o meu muito obrigado!

À Wizdee por me ter dado a oportunidade de trabalhar numa área tão interessante e tão cheia de desafios, e também a toda a equipa que sempre me apoiou em tudo o que necessitei.

Ao Mestre Bruno Antunes, como orientador e colega de trabalho por todo o conhecimento que me transmitiu, que muito ajudou a desenvolver o meu trabalho, ao Professor Paulo Gomes por me ter sempre ajudado a manter a visão do que era esperado deste estágio e ao Doutor Pedro Abreu pela sua disponibilidade e apoio prestado.

Conteúdo

Lista de Figuras

Lista de Tabelas

Acrónimos

Acrónimo	Descrição
RI	Recuperação de Informação
QP	Qualidade de Pesquisa
AP	Análise de Pesquisa
SEO	*Search Engine Optimization*
SEM	*Search Engine Marketing*
BI	*Business Intelligence*

Capítulo 1

Introdução

Este documento descreve o trabalho efectuado por João Moura durante o ano académico de 2011/2012 no estágio curricular do Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra. O estágio teve lugar na Wizdee[1], empresa criada em 2009, *spin-off* do grupo *Cognitive and Media Systems*[2] do centro de pesquisa CISUC[3]. Esta empresa é especializada no desenvolvimento de soluções para a gestão de conhecimento. A supervisão deste estágio foi efectuada pelo Mestre Bruno Antunes (pela Wizdee) e Prof. Dr. Pedro Abreu (pelo Departamento de Engenharia Informática).

A Wizdee está a desenvolver uma plataforma tecnológica de gestão de conhecimento. Esta plataforma é composta por várias componentes que permitem oferecer diversas funcionalidades, sendo uma delas a capacidade de pesquisa. É sobre esta plataforma que incidiu o estágio de modo a satisfazer uma necessidade da empresa.

1.1 Contexto e Motivação

Actualmente vive-se numa era onde a quantidade de informação disponível na internet é enorme. Este excesso de informação traduz-se num aumento de dificuldade em encontrar a informação de que se necessita. Torna-se necessário gastar bastante tempo à procura de informação, procura esta que engloba a tarefa de encontrar a fonte onde a informação pode estar e depois encontrá-la nessa fonte. Os motores de pesquisa[4] são uma grande ajuda para colmatar essa necessidade. Nesse processo é necessário o motor de pesquisa saber que informação é relevante para a pesquisa, apresentando resultados que satisfaçam as necessidades do utilizador. Contudo, esta tarefa pode tornar-se complexa. Neste contexto surge este estágio que visa resolver este tipo problemas no motor de pesquisa da Wizdee.

Existem ferramentas capazes de ajudar na melhoria e manutenção de um motor de pesquisa. Essas ferramentas, que analisam informação resultante da interacção do utilizador com o motor de pesquisa, geram relatórios e gráficos com informação valiosa para os gestores destes sistemas. Com esta informação acessível, torna-se possível perceber quais as limitações e falhas do sistema e proceder aos respectivos melhoramentos de modo a prestar um melhor serviço ao utilizador final. Torna também possível compreender os hábitos e necessidades actuais dos seus utilizadores possibilitando o ajuste da informação disponível às necessidades, sendo uma mais valia ao nível de negócio. Consegue-se ter

[1] `http://wizdee.com/`.
[2] `http://www.uc.pt/en/fctuc/ID/cisuc/Organization/CMS/`.
[3] `http://www.uc.pt/en/fctuc/ID/cisuc`.
[4] `http://www.searchenginehistory.com/`

também uma panorâmica geral da qualidade de resposta do motor de pesquisa através de métricas como *Hit Rate*[5], *Bounce Rate*[6] e *Refinement Rate*[7].

Apesar das ferramentas existentes, a solução a implementar necessita de ser uma solução à medida. O motor de pesquisa da Wizdee trabalha em mundo fechado, isto é, dentro de um determinado domínio, ao contrário de um motor de pesquisa na *web*. Os dados sobre os quais trabalha são estruturados, sendo essa estrutura vital para o seu funcionamento. Faz uso do Lucene[8], trata-se de uma tecnologia baseada em *Java* que oferece indexação e pesquisa em texto/documentos, facilitando o uso de documentos como recursos. Para além de ser possível fazer uma pesquisa por símbolos, também conhecida como pesquisa por palavra-chave (Baeza-Yates et al., 1999), uns dos pontos fortes do motor da Wizdee está na sua capacidade de pesquisa semântica (Guha et al., 2003). Faz uma análise semântica à pesquisa introduzida pelo utilizador de modo a extrair sentido à pesquisa, oferecendo uma resposta mais completa às necessidades de informação do utilizador.

1.2 Objectivos

Este estágio teve como objectivo o desenvolvimento de um sistema de Qualidade de Pesquisa (QP) e Análise de Pesquisa (AP) que permitiu dotar o motor de pesquisa da Wizdee com capacidades de análise e monotorização de qualidade e desempenho de pesquisa, apresentando de forma fácil a informação recolhida.

Pretende-se assim que o sistema consiga mostrar informação referente a tempos de respostas, pesquisas mais frequentes, pesquisas sem resposta, tendências dos utilizadores ao longo de um determinado periodo de tempo, os recursos mais acedidos, seja capaz de identificar a relevância das respostas dadas tendo para isso em conta métricas como o *hit*, *bounce* e *refinement rate*, entre outras funcionalidades.

Neste âmbito foi necessário numa primeira fase implementar mecanismos capazes de detectar a interacção do utilizador com o motor de pesquisa. Esses dados foram guardados e posteriormente processados de modo a extrair a informação desejada para visualização.

Os resultados deste trabalho são os seguintes:

- Concepção de um processo de recolha e processamento de informação proveninente das pesquisas efectuadas pelos utilizadores;
- Desenvolvimento de uma interface gráfica destinada a visualizar a informação recolhida;
- Execução de testes de desempenho e funcionais sobre o trabalho desenvolvido.

1.3 Metodologia

O projecto seguiu a metodologia de desenvolvimento *Scrum* (Rising and Janoff, 2000), uma metodologia iterativa e incremental utilizada no Desenvolvimento Ágil de Software[9] e que foi adoptada pela Wizdee. O *Scrum* fornece um conjunto de práticas e papéis

[5]Taxa de pesquisas em que foram seleccionados resultados.

[6]Taxa de pesquisas em que não foram seleccionados resultados e de seguida o utilizador abandonou o sistema.

[7]Taxa de pesquisas que são um refinamento da pesquisa anterior.

[8]`http://lucene.apache.org/`.

[9]`http://agilemanifesto.org` – Manifesto do Desenvolvimento Ágil de Software.

predefinidos que podem ser ajustados para melhor integração com a realidade de cada equipa. O conceito chave do *Scrum* é o reconhecimento que os requisitos mudam durante um projecto de *software* e que mudanças imprevisíveis não são endereçadas com facilidade pelas metodologias de desenvolvimento tradicionais. A metodologia *Scrum* utiliza uma abordagem empírica, aceitando que o problema não pode ser totalmente compreendido ou definido, focando-se antes na habilidade para responder a desafios emergentes de uma maneira ágil.

Os principais papéis no *Scrum* são o *Product Owner*, que representa a empresa e os *stakeholders*, o *Scrum Master*, que é responsável por manter a equipa focada e protegida em relação a destabilizações externas e a *Team*, um grupo multi-funcional que executa o trabalho necessário para desenvolver o produto. O processo de desenvolvimento em *Scrum* evolui por iterações, os *Sprints*, tipicamente em períodos de duas a quatro semanas. As tarefas de cada *Sprint*, chamado *Sprint Backlog*, são determinadas no *Sprint Planning Meeting* e escolhidas do *Product Backlog*, uma lista de requisitos a serem desenvolvidos ordenados por prioridade. Os requisitos definidos no *Sprint Backlog* não podem mudar durante o *Sprint* e este tem sempre de terminar a tempo, um requisito que não tenha sido terminado num *Sprint* volta para o *Product Backlog*. O progresso do *Sprint* é acompanhado usando o *Burn Down Chart*, que apresenta a quantidade de trabalho restante para terminar o *Sprint*. O progresso é também discutido em reuniões diárias, chamadas *Daily Scrum Meeting*, com uma duração de, tipicamente, 15 minutos, onde os membros da equipa falam sobre o que fizeram desde a reunião anterior, quais os planos para esse dia e possíveis questões que os estejam a bloquear. Quando um *Sprint* termina a equipa faz uma revisão do trabalho realizado durante o *Sprint* e apresentam-no aos *stakeholders* numa *Sprint Review Meeting*. Finalmente, uma *Sprint Retrospective Meeting* permite à equipa reflectir sobre o último *Sprint* e fazer melhorias ao processo.

Para este projecto a equipa foi composta por Paulo Gomes (*Produt Owner*), Bruno Antunes (*Scrum Master*) e João Moura (*Team*) – o desenvolvimento da plataforma integrou-se numa equipa com mais colaboradores, no entanto neste projecto apenas foram atribuídas tarefas a um elemento. Os *Sprints* tiveram uma duração de três semanas e começaram com um *Sprint Planning Meeting*, onde as tarefas para o *Sprint* foram definidas. O *Scrum Master* e a *Team* reuniram-se nos *Daily Scrum Meetings* para discutir o progresso do *Sprint*. Os *Sprints* terminaram com um *Sprint Meeting Review* onde o trabalho do *Sprint* foi apresentado e seguindo-se um *Sprint Retrospective Meeting* para discutir melhorias ao processo.

O processo *Scrum* foi implementado utilizando o Jira[10], através do *plugin* GreenHopper[11], que providencia uma interface de utilização simples para a gestão de produtos que sigam o processo *Scrum*. Os artefactos, incluindo código fonte e documentação, produzidos ao longo do projecto foram centralizados e versionados utilizando um repositório Subversion[12].

1.4 Planeamento

A figura 1.1 apresenta o diagrama de *Gantt* com o planeamento para o projecto, tanto o primeiro como o segundo semestre.

O diagrama apresenta o plano completo de estágio. Durante o primeiro semestre o tempo alocado ao estágio foi utilizado na pesquisa de informação sobre a área de estágio tanto a nível de estado da arte, como ferramentas e competidores e finalmente a escrita da proposta de estágio.

[10] `http://www.atlassian.com/software/jira/`.

[11] `http://www.atlassian.com/software/greenhopper/`.

[12] `http://subversion.apache.org`.

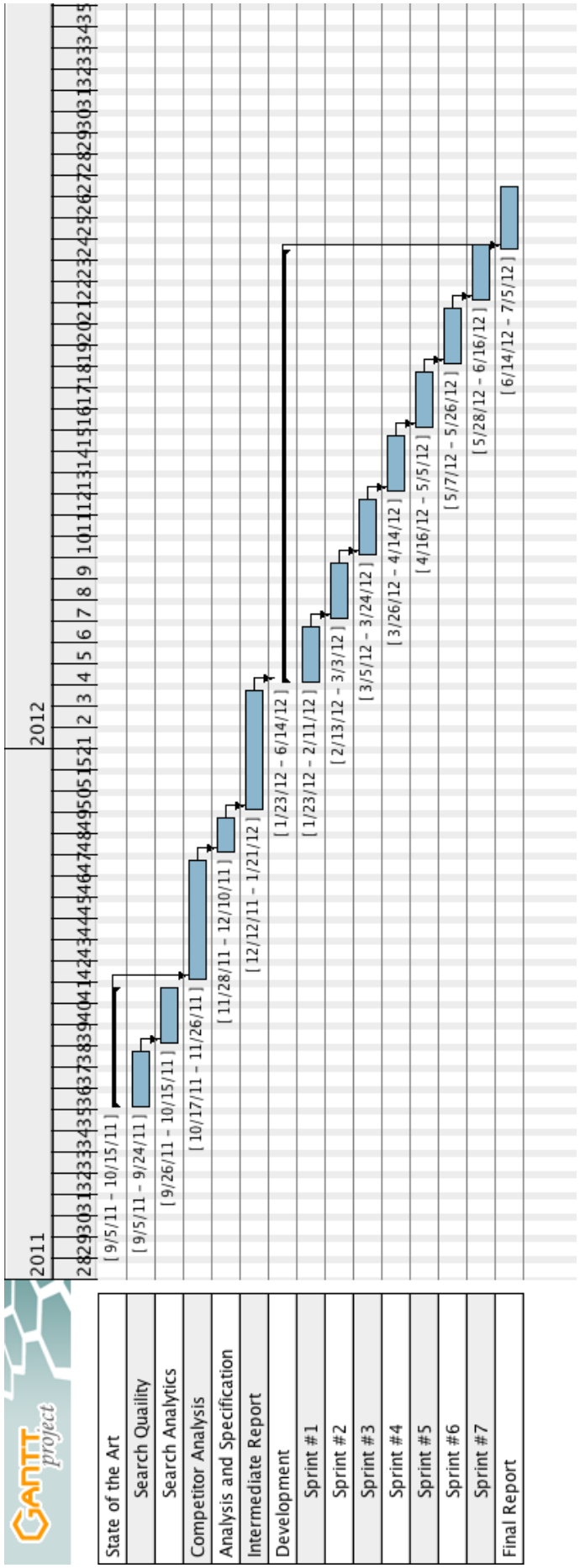


Figura 1.1: Diagrama de *Gantt* do Estágio.

O segundo semestre foi focado no desenvolvimento, sendo este dividido em *Sprints*, de acordo com a metodologia utilizada e integrando-se totalmente no ciclo de desenvolvimento da Wizdee. Cada *Sprint* teve três fases: análise, implementação e testes. Estas fases foram distribuídas ao longo das três semanas de duração de cada *Sprint* e aplicaram-se a cada uma das tarefas atribuídas ao *Sprint*. Em cada *Sprint* tiveram lugar as reuniões previstas pela metodologia adoptada.

Relativamente ao primeiro semestre, de um modo geral correu dentro dos tempos previamente planeados. O Estado da Arte, principalmente a parte de AP, foi das mais difíceis de escrever devido à pouca informação encontrada e a esta estar misturada com outro tipo de conceitos que se desviam um pouco do âmbito do estágio. Também a secção referente à Análise de Competidores foi igualmente díficil devido à pouca informação disponível dos produtos.

Quanto ao segundo semestre, existiram alguns contratempos de implementação normais em desenvolvimento àgil de software, como decisões tomadas inicialmente que tiveram que ser revistas numa fase mais avançada e que teve as suas consequências na implementação já feita. Mas de um modo geral tudo correu dentro dos tempos planeados.

1.5 Trabalho Efectuado

Aqui é apresentada a lista das tarefas executadas em cada um dos *Sprints* definidos.

- ***Sprint* #1 (23/Jan/2012 a 10/Fev/2012)**
  - Preparação do ambiente de desenvolvimento.
  - Criação de *mockups* da interface gráfica.
  - Especificação e criação do modelo de dados.
- ***Sprint* #2 (13/Fev/2012 a 02/Mar/2012)**
  - Processar pesquisas.
  - Implementação do mecanismo que permite recolher e armazenar a informação de pesquisas feitas pelo utilizador.
  - Processar respostas.
  - Implementação do mecanismo que permite recolher e armazenar a informação da resposta dada pelo sistema a uma pesquisa.
  - Correcções sobre o modelo de dados.
- ***Sprint* #3 (05/Mar/2012 a 23/Mar/2012)**
  - Implementação da identificação de um *hit* num resultado de pesquisa.
  - Implementação que permite identificar um *bounce* numa pesquisa.
  - Implementação que permite identificar um *refinement* de pesquisa.
  - Cálculo de *hit rate* por sessão.
  - Cálculo de *bounce rate* por sessão.
  - Cálculo de *refinement rate* por sessão.
  - Cálculo das pesquisas respondidas por sessão.
  - Cálculo das pesquisas sem resposta por sessão.

  - Cálculo do tempo médio de resposta por sessão.
  - Implementação do cálculo da duração de uma sessão.

- ***Sprint* #4 (26/Mar/2012 a 13/Abr/2012)**
  - Cálculo de *hit rate* total.
  - Cálculo de *bounce rate* total.
  - Cálculo de *refinement rate* total.
  - Implementação da contagem de recursos por tipo de recurso.
  - Cálculo do número médio de pesquisas por sessão.
  - Cálculo do tempo médio de duração de sessão.
  - Cálculo do tempo médio de resposta por sessão.
  - Cálculo de tempo médio de resposta por tipo de recurso.
- ***Sprint* #5 16/Abr/2012 a 04/Mai/2012)**
  - Implementação do *top* de recursos com mais *hits*.
  - Implementação do *top* de tipo de recursos.
  - Implementação do *top* de termos de pesquisas mais frequentes.
  - Adicionar filtro para calcular *top* dos termos de pesquisa mais rápidos.
  - Adicionar filtro para calcular *top* dos termos de pesquisa mais lentos.
  - Adicionar filtro para calcular *top* dos termos de pesquisa com mais resultados
  - Implementação do *top* de pesquisas mais frequentes.
  - Adicionar filtro para calcular *top* das pesquisas mais rápidas.
  - Adicionar filtro para calcular *top* das pesquisas mais lentas.
  - Adicionar filtro para calcular *top* das pesquisas com mais resultados
  - Adicionar filtro para calcular *top* das pesquisas por *hit rate*
  - Adicionar filtro para calcular *top* das pesquisas por *bounce rate*
  - Adicionar filtro para calcular *top* das pesquisas por *refinement rate*
- ***Sprint* #6 (07/Mai/2012 a 25/Mai/2012)**
  - Implementação do *top* de tendências de pesquisa.
  - Implementação do processo que permite identificar as tendências *long runners*.
  - Implementação do processo que permite identificar as tendências *novelties*.
  - Implementação do processo que permite identificar as tendências *top movers*.
  - Pesquisa de tendências.
  - Alteração no modelo de dados de forma a simplificar o acesso à informação de evolução de tendências e respectivas correcções de código.
- ***Sprint* #7 (28/Mai/2012 a 15/Jun/2012)**
  - Implementação do mecanismo que permite recalcular diáriamente e de forma automática as tendências e informação evolutiva de sessão.
  - Criação do gráfico de uma tendência ao longo do tempo.
  - Criação do gráfico de evolução de parâmetros de sessão.
  - Desenvolvimento do *design* gráfico.

1.6 Estrutura do Documento

Este documento está estruturado da seguinte forma:

- **Estado da Arte**: onde são descritos os principais conceitos utilizados neste estágio. Estes estão divididos em dois grandes tópicos: Qualidade de Pesquisa e Análise de Pesquisa. São também apresentadas soluções comerciais no âmbito dos motores de pesquisa com AP e QP, assim como soluções exclusivamente dedicadas a este tipo de tarefa sem estarem integradas a um motor de pesquisa em específico. Este enquadramento é mais importante para a parte empresarial do estágio, sendo que é útil para recolher de ideias e avaliar o estado das soluções comerciais.
- **Análise de Requisitos**: onde é feito um enquadramento fazendo um breve resumo do estado em que a plataforma se encontrava antes de iniciado o estágio, apresentando de seguida os requisitos para o projecto.
- **Arquitectura**: onde são apresentadas as componentes da plataforma e o modo como estas se ligam entre si.
- **Especificação e Desenvolvimento**: onde é apresentado todo o trabalho efectuado ao longo do estágio.
- **Testes e Experimentação**: onde são apresentados os testes às funcionalidades implementadas.
- **Conclusões**: onde se finaliza este relatório de estágio fazendo um resumo e apresentando as conlusões.

Capítulo 2

Estado da Arte

Este capítulo apresenta o Estado da Arte nas áreas relevantes para o presente estágio: Qualidade de Pesquisa (QP) e Análise de Pesquisa (AP). Qualidade de Pesquisa e Análise de Pesquisa estão englobadas num grande tópico de investigação - Recuperação de Informação (RI). A Recuperação de Informação (Baeza-Yates et al., 1999), lida com a representação, armazenamento, organização e acesso de informação. Essa informação é geralmente constituída por texto, como por exemplo documentos diversos e páginas web, embora possa também conter imagens e áudio, estando disponível aos utilizadores através de uma consulta num sistema de RI ou motor de pesquisa.

É na utilização de um sistema de RI que entra a QP. Esta incide sobre a informação devolvida pelo sistema à pesquisa do utilizador, analisando a qualidade da resposta. Já a AP recai sobre aspectos de optimização e negócio através da recolha e análise de dados e tendo em conta determinadas métricas.

A plataforma tecnológica da Wizdee serve de base ao desenvolvimento de soluções de gestão de conhecimento, sendo uma delas um motor de pesquisa semântico. Existem no mercado soluções concorrentes que disponibilizam interfaces de apoio ao negócio ou simplesmente de análise de desempenho do sistema. Nesta secção iremos descrever produtos semelhantes ou apenas focados na vertente de negócio. Para a grande maioria dos sistemas disponíveis no mercado não existe informação detalhada sobre os métodos de funcionamento utilizados, mas sempre que possível irá ser dada uma breve explicação sobre as funcionalidades referidas nas descrições comerciais do produto.

2.1 Qualidade de Pesquisa

Actualmente, quando se utiliza um motor de pesquisa é fundamental obter resultados que satisfaçam as necessidades do utilizador de forma eficaz e eficiente. Do ponto de vista do utilizador, este pretende obter resultados no mais curto espaço de tempo possível, reduzindo assim o tempo de resposta entre a introdução da pesquisa e obtenção da resposta e que estes resultados tenham qualidade desejada para satisfazer a sua necessidade de informação.

2.1.1 Tempo de Resposta

Normalmente, o tempo de resposta deverá ser o mais rápido possível (Nielsen, 1993), embora seja possível para o computador reagir tão depressa que o utilizador pode não se

aperceber do tempo de resposta. Por exemplo, no *scrolling*[1] numa lista de resultados, esta pode mover-se tão depressa que o utilizador não consegue parar a tempo de visualizar o resultado desejado. Por esta razão, é necessário efectuar alterações a nível da interface gráfica com o utilizador de modo, a por exemplo, informar do tempo real que demora até aparecerem os resultados.

O tempo de resposta de referência para todas as aplicações foi estabelecido há pouco mais de 40 anos e varia entre 0,1 e 10 segundos (Nielsen, 1993).

O tempo de 0,1 segundos é o limite para os utilizadores sentirem que estão a manipular directamente os objectos da interface gráfica. Por exemplo, 0,1 segundos é o limite de tempo que decorre entre o utilizador seleccionar um resultado e lhe serem apresentados os detalhes desse resultado, ou outro qualquer *feedback*.

Por outro lado, 1 segundo é o limite de tempo aceitável de execução do computador de modo a manter o utilizador focado na sua navegação pela aplicação sem ser interrompida a fluidez das suas interacções.

Finalmente, 10 segundos é o tempo limite máximo para manter a atenção do utilizador na tarefa. Será a partir destes tempos que deverá ser mostrado alguma mensagem de *feedback* com o tempo que demora a executar ou com a percentagem de conclusão da tarefa. A partir dos 10 segundos o utilizador perde a linha de raciocínio, uma vez que este aproveita o tempo de espera para efectuar outras tarefas.

Pode-se verificar que o limite de 1 segundo é a referência para os principais motores de pesquisa que usamos actualmente na *internet*.

2.1.2 Relevância

A qualidade de pesquisa está directamente relacionada com a relevância de pesquisa. Quanto maior a relevância dos resultados, melhor a qualidade de pesquisa, mas conceptualmente representam termos bastante relativos. Facilmente se compreende que um resultado que apresente boa qualidade para um utilizador, poderá não representar a mesma qualidade ou não fazer qualquer sentido para outro utilizador. Como exemplo, imaginando um hipotético cenário em que dois utilizadores, A e B, fazem exactamente a mesma pesquisa, por exemplo "scp". O motor devolve os resultados, suponha-se dez resultados, todos relacionados com o comando *Linux*[2] *Secure Copy* (SCP)[3]. Para o utilizador A estes resultados são bastante relevantes, pois obteve exactamente a informação que procurava, enquanto o utilizador B, que procurava por uma equipa de futebol (Sporting Clube de Portugal), os resultados não apresentam qualquer relevância do ponto de vista das suas necessidades, sendo obrigado a refinar um pouco mais a sua pesquisa.

Pode-se concluir então que um bom resultado de pesquisa é um resultado que apresente o máximo de relevância para o utilizador, isto é, cobre ao máximo a pesquisa feita pelo utilizador, satisfazendo assim as suas necessidades de procura no mais curto de espaço de tempo possível (Baeza-Yates et al., 1999; Nielsen, 1993).

2.1.3 Avaliação da Relevância

Apesar de relevância ser um parâmetro bastante subjectivo, pelas razões já referidas anteriormente, há métodos que permitem avaliar a relevância de um dado resultado ou de um grupo de resultados, ajudando assim os motores de pesquisa a melhorar o seu

[1] Acto de mover o texto ou imagem para cima e para baixo no ecrã do computador no âmbito de visualizar as diferentes partes.

[2] `http://www.linux.org`.

[3] `http://support.real-time.com/linux/web/scp.html`.

desempenho na qualidade das respostas apresentadas. De seguida serão apresentados alguns exemplos.

Pode-se utilizar um sistema de *feedback*, que permite ao utilizador classificar dentro de uma determinada escala, a relevância da resposta devolvida pelo motor de pesquisa. Desta maneira é possível associar as palavras-chave inseridas pelo utilizador com os resultados que foram marcados como relevantes (Büttcher et al., 2010). Um exemplo deste sistema pode ser encontrado no *SearchWiki*[4] implementado pela *Google*.

Também é possível avaliar a relevância de um resultado através da interacção do utilizador com os resultados obtidos do motor de pesquisa. Quando o utilizador acede a um determinado resultado e consome tempo na sua visualização e interacção (por exemplo, acções de *scrolling*), pode-se assumir que esse resultado em específico é relevante para a pesquisa efectuada (Büttcher et al., 2010). Um exemplo deste sistema pode ser encontrado no motor de pesquisa implementado pela *Surf Canyon*[5].

Outro bom exemplo de avaliação é a alocação de recursos humanos para a prévia classificação de termos com resultados de pesquisa. Actualmente é exequível se pensado para um domínio fechado, uma vez que em domínio aberto se torna praticamente impossível devido à quantidade de informação acessível e ao ritmo a que esta cresce (Karen Spärck Jones, 1997).

Existe a hipótese de *clustering* (van Rijsbergen, 1979), defendendo que se dois documentos são similares entre si, então há uma forte probabilidade de serem relevantes para uma mesma pesquisa. Foi tradicionalmente investigado para melhorar os níveis de desempenho dos motores de pesquisa fazendo *pre-clustering* em todos os documentos (van Rijsbergen, 1979). Existe também uma variante de *clustering* mais associado a técnicas de navegação com informação estruturada, permitindo particionar os resultados por grupos de documentos relacionados ou relevância similar (Leouski and Croft, 1996). Por exemplo, uma pesquisa por "java" poderá devolver grupos de resultados para linguagem de programação Java[6], ilha de Java[7] ou mesmo café Java[8].

Em todos os casos apresentados há a intervenção do utilizador para a classificação da relevância dos resultados apresentados pela pesquisa efectuada. Assim, no caso de *feedback* surge novamente o dilema do que é relevante para um utilizador poder não o ser para outro. No caso da avaliação por interacção, nada garante que um utilizador ao ver um certo resultado devolvido, considere interessante e vá explorar mesmo não tendo nada em comum com a sua pesquisa inicial. O caso de alocação de recursos para avaliação é semelhante ao primeiro, com a agravante de se atribuirem termos a um resultado ou conjunto de resultados. Relativamente aos casos de *clustering* é de notar que é necessário haver a classificação de relevância de alguns documentos de modo a ser possível associar um *cluster* com as palavras-chave. Mas por outro lado, se existir uma grande comunidade a contribuir para este tipo de avaliação, torna-se possível fazer um sistema de classificação de relevância bastante preciso, uma vez que a quantidade torna desprezível as excepções (Karen Spärck Jones, 1997).

2.1.4 Precisão e Abrangência

A Precisão e Abrangência são métricas direccionadas para avaliação de sistemas de RI (Manning et al., 2008). Normalmente são usadas para calcular a eficácia de um sistema RI que se traduz directamente na relevância dos resultados apresentados ao utilizador.

[4] `http://googleblog.blogspot.com/2008/11/searchwiki-make-search-your-own.html`.
[5] `http://www.surfcanyon.com/`.
[6] `http://www.oracle.com/technetwork/java/index.html`.
[7] `http://www.javaindonesia.org/`.
[8] `http://www.javacoffee.com/`.

Precisão

A Precisão é a fracção de resultados da pesquisa que são relevantes para o utilizador. Como é ilustrado pela equação 2.1, a precisão é o rácio da intercepção dos resultados relevantes devolvidos pela pesquisa com os resultados devolvidos pela mesma, dividido por todos os resultados devolvidos pela pesquisa (Baeza-Yates et al., 1999).

$$precisão = \frac{\left|\{resultados\ relevantes\} \cap \{resultados\ de\ pesquisa\}\right|}{\left|\{resultados\ de\ pesquisa\}\right|} \tag{2.1}$$

Analisando o seguinte cenário em que um utilizador efectua uma pesquisa, quando um motor de pesquisa devolve os resultados, suponha-se 30 resultados, e desses apenas 20 são relevantes para a pesquisa efectuada, a sua precisão é de 20/30, ou seja, aproximadamente 67%.

Abrangência

Abrangência é a fracção de resultados que são relevantes para o utilizador no total de resultados relevantes possíveis. Observando a equação 2.2, para calcular a abrangência de um resultado, implica saber à partida todos os resultados que são relevantes para uma dada pesquisa. De seguida, fazer a intercepção entre todos os resultados relevantes com os resultados obtidos na pesquisa para obter o número de resultados relevantes obtidos na pesquisa. Divide-se por todos os resultados relevantes e obtém-se assim o rácio da abrangência da pesquisa.(Baeza-Yates et al., 1999)

$$abrangência = \frac{\left|\{resultados\ relevantes\} \cap \{resultados\ de\ pesquisa\}\right|}{|\{resultados\ relevantes\}|} \tag{2.2}$$

Recorrendo ao cenário de exemplo do ponto anterior, admitindo que para além dos 20 resultados relevantes para a pesquisa sabe-se que existem mais 40 que ficaram por devolver, então a abrangência é de 20/60, ou seja, aproximadamente 33%.

Relação entre Precisão e Abrangência

Uma abrangência elevada significa que se tem toda a informação necessária nos resultados obtidos, mas com certeza que se irá encontrar demasiados resultados que não apresentam qualquer tipo de interesse para a pesquisa, implicando assim uma precisão baixa. Por outro lado, ter uma precisão elevada, significa que todos os resultados obtidos pela pesquisa são relevantes, mas com o custo de se estar a deixar de fora alguns resultados também eles relevantes para a pesquisa, implicando assim uma baixa abrangência. É necessário haver um equilíbrio entre ambos, de modo a proporcionar ao utilizador a melhor informação possível (Baeza-Yates et al., 1999).

Uma medida que pode ser interpretada como um equilíbrio entre Precisão e Abrangência é a Medida F, definida pela expressão:

$$F(j) = 2 \cdot \frac{P(j) \cdot A(j)}{P(j) + A(j)} \tag{2.3}$$

onde A(j) e P(j) são respectivamente, a Abrangência e a Precisão para o resultado j da resposta, sendo F(j) a avaliação da Medida F relativamente a A(j) e P(j). A função F assume valores no intervalo [0, 1], sendo 0 quando não existem resultados relevantes na resposta dada e 1 quando todos os resultados apresentados são relevantes. F(j) apresenta valores altos apenas quando a Precisão e Abragência são elevados (Baeza-Yates et al., 1999).

2.2 Análise de Pesquisa

A Análise de Pesquisa é a análise e agregação de estatísticas sobre motores de pesquisa para uso em *marketing* (SEM) e optimização (SEO) para motores de busca. Pode ser compreendido como sendo a medição, recolha e análise de dados com o propósito de compreender e optimizar a utilização de um motor de pesquisa e melhorar o seu desempenho (Moran and Hunt, 2006; Enge et al., 2010).

Existem duas categorias de AP: *offsite* e *onsite*. Análise *offsite* refere-se ao estudo da popularidade por toda a internet de um *website*. Esta análise inclui a medição da potencial audiência do *website* (oportunidade), a sua visibilidade, assim como também a sua relevância para com a comunidade *online* (Chaters, 2011).

Por sua vez, análise *onsite*, estuda a actividade dos visitantes no nosso *website*. Isto é, que documentos/páginas encorajam as pessoas a interagir, medindo assim o seu desempenho num contexto comercial. Esses dados são geralmente comparados com indicadores chave de desempenho e utilizados para melhorar o desempenho do *website* ou verificar a resposta a uma campanha de *marketing* junto do público (Inan, 2006).

2.2.1 SEM e SEO

O SEM, acrónimo de *Search Engine Marketing*, em português *marketing* nos motores de pesquisa, é a actividade que se dedica a promover *websites* nos motores de pesquisa, tornando-os visíveis nos motores de busca de modo a converter essa visibilidade em activos para o site. O SEM funciona como uma extensão do *marketing* da empresa (Moran and Hunt, 2006).

Por sua vez o SEO, do inglês *Search Engine Optimization*, optimização para motores de pesquisa, é um processo que tem como objectivo melhorar a quantidade e a qualidade dos visitantes para um *website*. Ajuda a elevar o *ranking* do *website* perante os motores de pesquisa, colocando-o assim nos resultados visíveis de pesquisa e consequentemente atraindo mais visitantes. O SEO é parte do SEM, que para lá do SEM, se preocupa também em posicionar os *websites* nos resultados pagos. Pode-se por isso falar no SEO como a actividade do *Search Marketing* que atrai visitantes (Enge et al., 2010).

2.2.2 Tendências

Actualmente saber as necessidades actuais e futuras dos utilizadores torna-se importante para se ser competitivo no vasto mercado da internet. Facilmente se compreende que estar um passo à frente da concorrência aumenta as probabilidades de sobrevivência (Gloria J. Miller, 2006). Assim com AP isso torna-se possível. Através dos termos de pesquisa inseridos pelos utilizadores nos motores de busca, consegue-se ter um registo das suas tendências. Analisando a emergência, frequência e variação de novos termos é possível prever as necessidades gerais e preparar-se o serviço para responder às expectativas dos utilizadores (Rosenfeld et al., 2011).

2.2.3 Informação Funcional

Com o crescer de informação disponível e a forte concorrência no mercados de motores de pesquisa, é essencial que este esteja em constantes ajustes para dar respostas que realmente vão de encontro às necessidades de quem o usa e no menor curto espaço de tempo possível. Uma vez mais, a AP vem ajudar nesta tarefa, no sentido em que se torna possível visualizar o comportamento dos utilizadores relativamente às respostas obtidas, mostrando assim a qualidade desta. Permite uma avaliação do tempo gasto a dar uma

resposta final ao utilizador reportando assim o seu desempenho. Assim, com a simples análise de informação gerada pelo uso e funcionamento do motor de pesquisa torna-se também possível ajudar a melhorar e colmatar as falhas ou ausências de informação (Rosenfeld et al., 2011).

2.2.4 Utilidade

Para qualquer empresa é uma mais-valia ter uma página na *internet*, dando-lhe assim uma maior visibilidade e consequentemente aumentando a probabilidade de ganhar novos clientes. É nesta área que a análise de pesquisa pode ajudar a melhorar o negócio. Análise de pesquisa não é apenas uma ferramenta para medir o tráfego de um *website*, pode ser usada também como ferramenta de negócio e análise de mercado (Rosenfeld et al., 2011).

Permite monitorizar quais os tópicos, ou assuntos, mais procurados pelos utilizadores, não só ao nível do *website* mas também em toda a internet, permitindo assim melhorar o desempenho da pesquisa interna do *website* e também as palavras-chave fornecidas aos motores de busca externos pelas quais estes são indexados e posteriormente apresentados nos resultados de pesquisa (Rosenfeld et al., 2011). A nível interno, permite que se fique a saber quais os recursos mais e menos acedidos, saber que recursos são devolvidos por cada pesquisa, pesquisas sem resposta, entre outros. Todo este conjunto de métricas reflecte o desempenho do motor de pesquisa, fornecendo informação importante que pode contribuir para melhoramentos.

Torna-se assim possível criar oportunidades de negócio percebendo o que os utilizadores mais procuram, perceber se uma determinada campanha de *marketing* lançada está a ter ou não o sucesso desejado, entre outros (Rosenfeld et al., 2011).

2.3 Análise de Concorrência

De seguida serão apresentados alguns dos concorrentes do motor de pesquisa semântico da Wizdee. Concorrentes que agregam ao seu produto um sistema de análise do desempenho e qualidade do seu motor de pesquisa, ou informação de *business intelligence*[9]. Por fim será feita uma análise comparativa entre os concorrentes apresentados nesta secção.

2.3.1 Concorrentes

LucidWorks Enterprise (Lucid Imagination)

A *Lucid Imagination*[10] é uma empresa norte-americana, fundada em Agosto de 2007, que fornece a nível empresarial produtos e serviços para pesquisa em *Lucene* (trata-se de uma tecnologia baseada em *Java* que oferece indexação e pesquisa em texto/documentos, bem como correcção ortográfica, realce de ocorrências e capacidades avançadas de análise e tokenização) [11] e *Solr*[12], incluindo suporte, formação, certificação e consultadoria. Fornece também a sua própria plataforma de pesquisa que incorpora ferramentas de pesquisa e indexação de *logs*, *dashboard* com uma visão geral sobre as pesquisas efectuadas, monitorização em interfaces standartizados e segurança no acesso a dados.

[9] *Business intelligence* (BI) é uma vasta categoria de aplicações e tecnologias para recolha, armazenamento, análise e acesso a dados com o objectivo de ajudar os utilizadores empresariais a tomarem melhores decisões de negócio.

[10] `http://www.lucidimagination.com`.

[11] `http://lucene.apache.org/`.

[12] `http://lucene.apache.org/solr/`.

[a]Imagem retirada de `http://www.lucidimagination.com/products/lucidworks-search-platform/enterprise/screenshots` consultado em 22 de Novembro de 2011.

Figura 2.1: Painel de Gestão Disponibilizado pela *LucidWorks*[a]

Na figura 2.1, pode-se visualizar algumas funcionalidades de monitorização. É possível analisar, através de um gráfico, a evolução ao longo do tempo a actividade no motor, tendo filtros para ver a actividade registada na última hora, último dia ou semana passada. Existem também duas secções onde é possível visualizar informação relativa às pesquisas mais frequentes na última hora e no geral, sendo que cada uma das pesquisas indica o número de vezes que foi feita. Existe ainda a possibilidade de ver directamente os *logs* gerados pelo servidor.

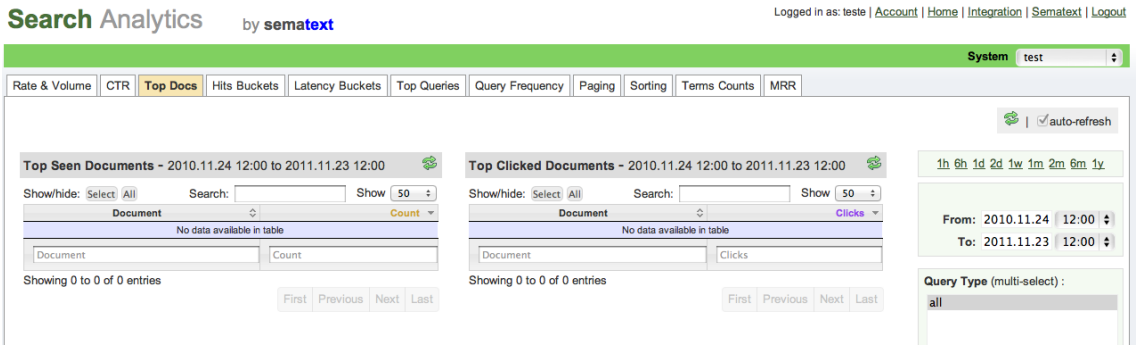
Search Analytics (Sematext)

A *Semantext*[13] é uma empresa norte-americana sediada em Nova Iorque que fornece apoio comercial, consultadoria, desenvolvimento e produtos relacionados com pesquisa *open-source*, Processamento de Linguagem Natural, motores de recomendação e análise de texto e de dados. Destacam-se os seguintes produtos: análise de pesquisa, extractor de chaves de frase, motor de recomendação, pesquisas relacionadas, analisador morfológico.

O produto de análise de pesquisa, *Semantext Search Analytics*[14], recolhe e analisa

[13]`http://sematext.com/`.
[14]`http://sematext.com/search-analytics/index.html`.

o comportamento de pesquisa de dados dos utilizadores, o *clickstream*[15] de dados e de transacções de pesquisas relacionadas, fornecendo uma visão mais exacta aos administradores sobre o comportamento dos utilizadores, a qualidade dos resultados de pesquisa, e um mecanismo para medir qualitativamente qualquer mudança feita no *backend* do motor de pesquisa, entre outros.



[a]Imagem própria obtida a 22 de Novembro de 2011.

Figura 2.2: Painel de *Top* de Pesquisas do *Search Analytics* da *Sematext*[a]

A figura 2.2 foi obtida depois de se criar uma conta de teste no *Search Analytics* que a *Sematext* disponibiliza para uso gratuito durante 30 dias. Verificou-se que este produto permite monitorizar a taxa e o volume de utização do sistema cliente e os volume de sessões de utilizador, isto é, a carga de utilização do sistema com no decorrer do tempo. Possibilita ainda monitorizar o *CTR*[16], ver quais os documentos vistos e os clicados, pesquisas por número de *hits* e de latência, *top* de pesquisas com informação relativa a cada uma, como por exemplo, a percentagem de vezes que foi usada relativamente ao total de pesquisas, latência, entre outros. É ainda disponibilizada a monitorização de frequência de pesquisas, contagem de termos, entre outros.

A cada uma destas funcionalidades podem ser ainda aplicados filtros, que permitem definir o espaço temporal dos gráficos, podendo variar dentro de uma escala definida entre 1 hora e 1 ano, bem como defenir a granularidade e um intervalo de tempo mais específico introduzido pelo cliente.

Enterprise Search 2.0 Platform (Coveo Solutions Inc.)

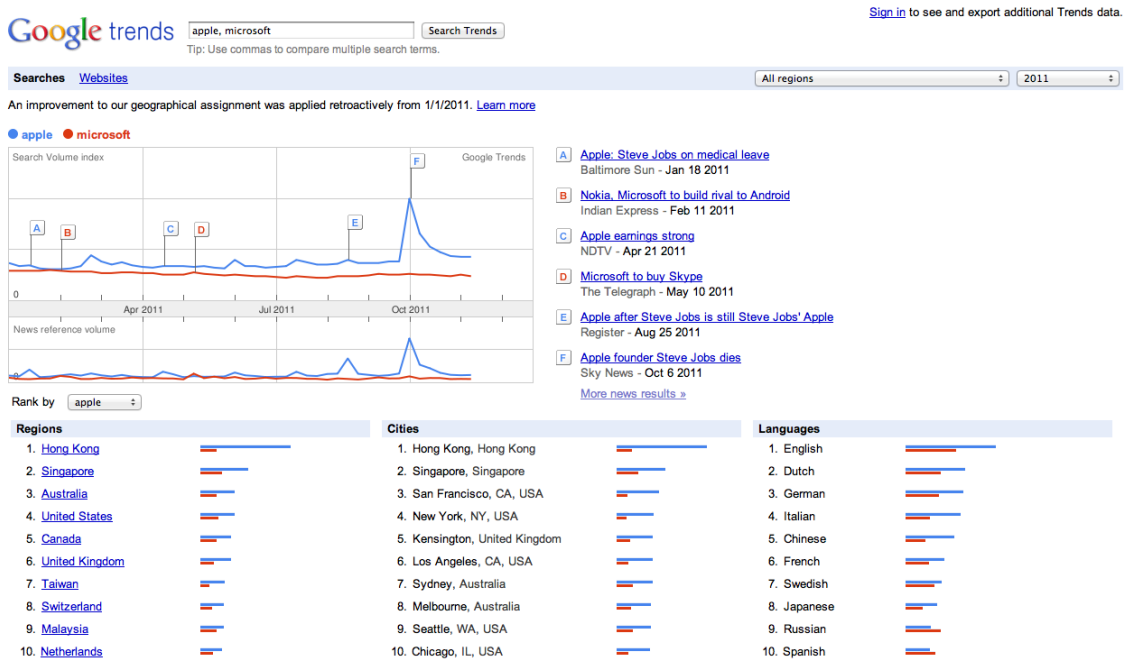
Coveo Solutions Inc.[17] é uma empresa Canadiana fundada em 2004 como uma *spin-off* da *Copernic Technologies Inc*. Dedica-se principalmente à pesquisa empresarial, para a qual desenvolvem a sua própria plataforma, *Coveo Enterprise Search 2.0*.

A plataforma da *Coveo* é uma plataforma modelar e escalável que indexa informação guardada em vários repositórios da empresa cliente de modo a alimentar o seu motor de pesquisa. Contém um módulo de *analytics*, *Coveo Business Analytics*, que fornece aos níveis de administração um panorama geral do negócio, permitindo a monotorização da informação, localizando tendências ou palavras-chave mais em uso pelos utilizadores, podendo assim promover a lealdade e a satisfação das necessidades dos mesmos.

[15]Gravação das partes do ecrã clicadas pelo utilizador de um computador enquanto utiliza uma qualquer aplicação.

[16]Clickthrough Rate - `http://ezinearticles.com/?What-is-CTR-(Click-Through-Rate)?&id=1728303`.

[17]`http://www.coveo.com`.



[a]Imagem obtida em uso da aplicação a 23 de Novembro de 2011.

Figura 2.3: *Google Trends*[a]

Google Inc.

A *Google*[18] é uma multinacional norte-americana fundada a 4 de Setembro de 1998, que investe em pesquisa de Internet, *cloud computing* e tecnologias publicitárias. Faz *hosting*[19] e desenvolve variadíssimos serviços e produtos direccionados para a internet e não só. Entre eles destacam-se os seguintes: o motor de pesquisa *Web*, o *Google Trends* e o *Google Insights*.

Google Trends

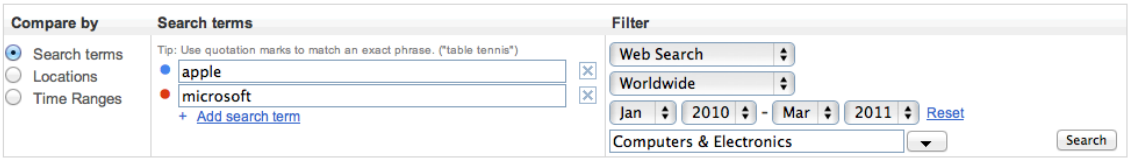
O *Google Trends*[20] é um produto Web desenvolvido pela *Google*, disponibilizado gratuitamente, que permite ver a frequência com que um dado termo de pesquisa, até um limite de cinco termos, aparece relativamente ao volume de pesquisas total.

Observando a figura 2.3, identificam-se várias funcionalidades. É possível filtrar os resultados por região e por ano. Ao escolher a região, pode-se ser mais específicos e indicar uma subregião. Ao lado do gráfico da evolução das frequências dos termos ao longo do tempo, caso exista, é mostrada informação que identifica um momento no gráfico. É também possível analisar os países, cidades e idiomas, nos quais mais vezes foram registados os termos.

Google Insights for Search

Google Insights for Search[21] é mais um produto gratuito da *Google*, trata-se de uma evolução do *Google Trends*, sendo mais sofisticado e avançado que o seu antecessor, é mais

[18]`http://www.google.org`.
[19]`http://searchsoa.techtarget.com/definition/hosting`
[20]`http://www.google.com/trends`.
[21]`http://www.google.com/insights/search`.

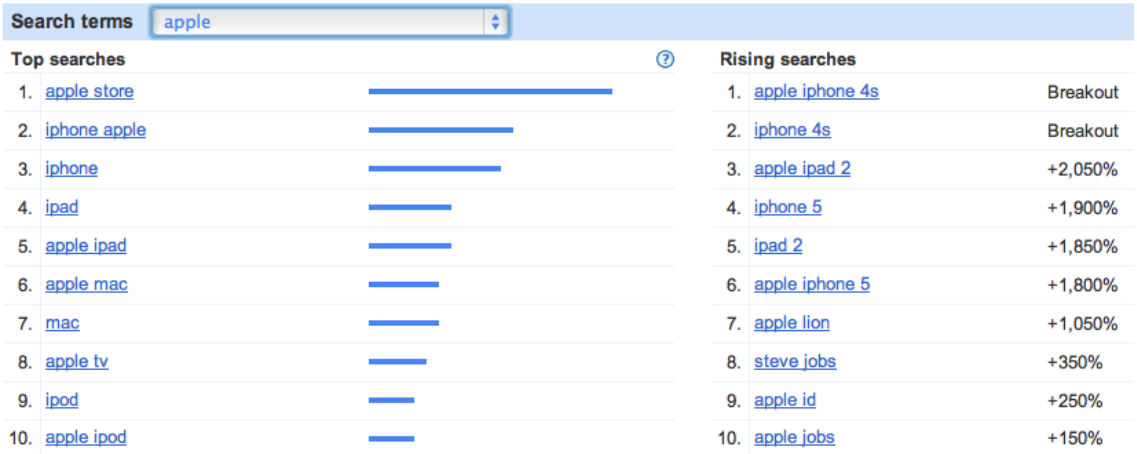


[a]Imagem obtida em uso da aplicação a 23 de Novembro de 2011.

Figura 2.4: Filtro de Pesquisas do *Google Insights*[a]

direccionado para utilizadores como investigadores ou agentes publicitários, que possam tirar melhor partido das funcionalidades avançadas.

Para além das funcionalidades já vistas no *Google Trends*, o *Google Insights for Search* apresenta um filtro melhorado (ver figura 2.4), sendo agora possivel filtrar por tipo de pesquisa, isto é, *web*, imagem, notícias ou produtos. Permite também definir um intervalo de tempo desejado e definir uma categoria ou área de pesquisa, por exemplo, saúde, entertenimento, entre outros.



[a]Imagem obtida em uso da aplicação a 23 de Novembro de 2011.

Figura 2.5: *Top* de Pesquisas do *Google Insights*[a]

Disponibiliza ainda na sua interface um *top* de pesquisas para os tópicos que contêm ou estão relacionados com os termos de pesquisa introduzidos, tendo em consideração os filtros seleccionados, assim como uma tabela que permite visualizar as pesquisas que mais cresceram no intervalo de tempo indicado (ver figura 2.5).

Endeca Latitude (Endeca)

Endeca[22] é uma companhia de *software* sedeada em Cambrige, fundada em 1999 e recentemente, em 2011, adquirida pela *Oracle Corporation*, tendo como negócio a venda de aplicações de *enterprise search* e *business intelligence*.

Um dos seus principais produtos é o *Endeca Latitude*. Trata-se de um produto concebido para *business inteligence*, que oferece aplicações interactivas de análise simples de usar, grande poder de análise como ferramentas de *Discovery*, análise especializada sobre informação não estruturada, entre outros, aleada à pesquisa. Possibilita que a informação possa ser indexada de diversas fontes como repositórios, *fileshares*, *feed* de notícias, *twitter*, entre outros. Á pesquisa alia-se *spell checking*, correspondência semântica e sugestão

[22]`http://www.endeca.com/en/about-us/company-overview.html`.

de pesquisa para ajudar o utilizador no acto de pesquisa, visualização intereactiva que permite ao utilizador ver a informação que pretende e navegação orientada que auxilia o utilizador a encontrar as respostas que precisa.

IBM Content Analytics with Enterprise Search (IBM)

A *International Business Machines*[23], vulgo *IBM*, foi fundada em 1911 com o nome *Computing Tabulating Recording Corporation*, adoptando o actual nome em 1924. Sediada em Armonk, Nova Iorque, trata-se de uma multinacional norte-americana que fabrica e vende o seu próprio *hardware* e *software*, disponibilizando infraestruturas, serviços de *hosting* e consultadoria em áreas desde de *mainframes* a nanotecnologia.

A *IBM* também tem produtos na área de pesquisa empresarial, tendo o seu próprio motor de pesquisa, o *IBM OmniFind Enterprise Edition*. Este motor pode ser acopolado a diversas plataformas e conectado a diferentes repositórios, usa o *Lucene* para indexar os recursos necessários oferecendo assim uma boa escalabilidade.

Na area de *analytics*, a *IBM* tem o produto *IBM Content Analytics*. Este usa a mesma tecnologia de Processamento de Linguagem Natural que o famoso *IBM Watson DeepQA*, a mais avançada máquina de resposta a perguntas do mundo, sendo uma interface robusta para exploração analítica de dados não estruturados.

Estes dois produtos da *IBM* foram agregados num pacote, *IBM Content Analytics with Enterprise Search*, juntando assim o poder de pesquisa com *analytics*, oferecendo a nível e empresarial uma poderosa ferramenta tanto de pesquisa como de negócio. Permite uma profunda e enriquecida análise de texto sobre a informação, descobrir novas percepções de negócio, reduzir tempo e complexidade na construcção de modelos de dados e linguísticos, dicionários ou ontologias. Ajuda a identificar tendências e padrões dentro de um histórico de casos. É focado na vontade do cliente final.

FAST Search Server 2010 for SharePoint (Microsoft)

A *Microsoft*[24] é uma multinacional norte-americana, sediada em Redmond, Washington, estabelecida a 4 de Abril de 1975, que desenvolve, produz, licencia e dá suporte um vasto leque de produtos e serviços relacionados com computadores através das seus vários departamentos de produto.

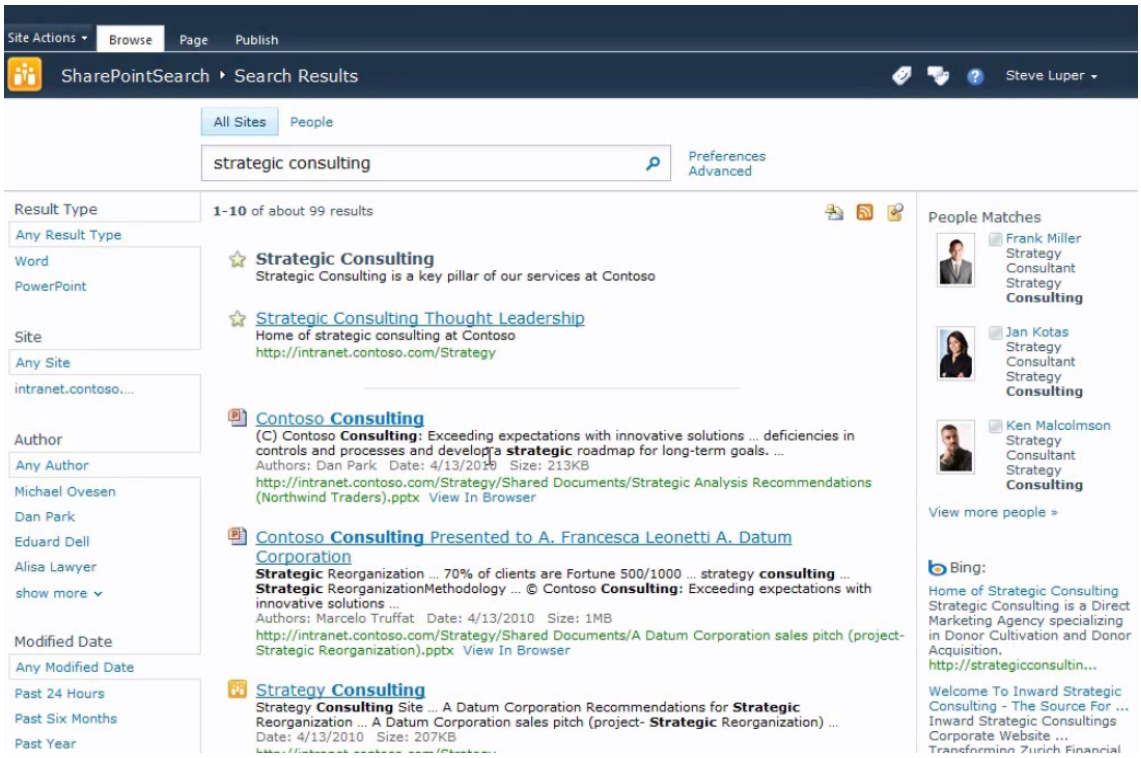
O *FAST* é um motor de busca empresarial da *Innovative Architects*, empresa parceira da *Microsoft*, que oferece uma melhorada precisão de pesquisa, permitindo customizar a pesquisa e aplicar filtros, unifica a pesquisa por toda a empresa. Fornece ainda capacidades de *Discovery*.

Com o crescimento de popularidade do *Sharepoint*[25] da *Microsoft* como portal central para organização e armazenamento de conteúdos dentro de uma *framework* colaborativa, foram integradas as capacidades do *FAST* no *Sharepoint*. Dando aos seus utilizadores um considerável melhoramento de desempenho na pesquisa com todas as funcionalidades do *Sharepoint*.

O *Fast Search Server 2010 for Sharepoint* (ver figura 2.6) permite pesquisa de conteúdos baseada em contexto. Indexa informação de diversas fontes, criando automáticamente *tags* que identificam locais, nomes, companhias, entre outros. Cada utilizador pode também criar as suas próprias *tags*, permitindo assim melhorar a informação disponível. Os resultados são apresentados por ordem de relevância, que é calculada por diversos mecanismos, sendo um deles o *CTR* que apresenta muito bons resultados ao longo do

[23] `http://www.ibm.com/`.
[24] `http://www.microsoft.com`.
[25] `http://sharepoint.microsoft.com`.

tempo. A estes resultados podemos aplicar filtros que ajudam a refinar a pesquisa, como por exemplo, tipo de resultados (*PowerPoint*, PDF, entre outros), por autor e período de modificação (figura 2.6).



[a]Imagem retirada do video disponível em `http://sharepoint.microsoft.com/en-us/Pages/Videos.aspx?VideoID=17` consultado em 25 de Novembro de 2011].

Figura 2.6: Painel de Resultados do *FAST Search Server*[a]

2.3.2 Análise Comparativa

Segue-se uma análise comparativa entre os competidores acima listados. Foi dada especial importância às seguintes características:

Pesquisa Mais Frequente
O sistema apresenta uma listagem das pesquisas mais frequentes.

Palavras-Chave Mais Usadas
O sistema apresenta uma listagem das palavras mais usadas nas pesquisas.

Recursos Mais Consultados
O sistema apresenta uma listagem dos recursos que mais vezes são vistos pelos utilizadores.

Recursos Mais/Menos Devolvidos pela Pesquisa
O sistema disponibiliza uma listagem dos recursos do sistema que mais/menos vezes são devolvidos pela pesquisa.

Motor de Pesquisa Integrado
O sistema encontra-se embebido num motor de pesquisa.

Funcionalidades de *Business Intelligence* (BI)
O sistema disponibiliza um módulo com funcionalidades de BI.

Solução	Pesquisas Mais Frequente	Palavras-Chave Mais Usadas	Recursos Mais Consultados	Recursos Mais/Menos Devolvidos pela Pesquisa	Motor de Pesquisa Integrado	Funcionalidades de BI
LucidWorks Enterprise (Lucid Imagination)	✓	✗	✗	✗	✓	✗
Search Analytics (Sematext)	✓	✓	✓	✓	✗	✗
Enterprise Search 2.0 Platform (Coveo Solutions Inc.)	✗	✓	✗	✗	✓	✗
Google Trends (Google)	✗	✓	✓	✗	✗	✗
Google Insights for Search (Google)	✓	✓	✓	✗	✗	✓
Endeca Latitude (Endeca)	n.d.	n.d.	n.d.	n.d.	✓	✓
IBM Content Analytics with Enterprise Search (IBM)	n.d.	n.d.	n.d.	n.d.	✓	✓
FAST Search Server 2010 for SharePoint (Microsoft)	n.d.	n.d.	n.d.	n.d.	✓	✓

Tabela 2.1: Comparação entre os sistemas concorrentes identificados.

Através da análise da tabela 2.1 é possível verificar que os sistemas concorrentes listados apresentam algumas divergências. Daqueles que disponibilizam ferramentas de BI poucos apresentam ou mencionam, na descrição dos produtos, ferramentas de AP. Se admitirmos que parâmetros como pesquisas mais frequentes ou palavras-chave mais usadas podem de algum modo constituir dados importantes para BI, será pertinente pensar que os concorrentes com ferramentas de BI também englobam as métricas definidas. Infelizmente, por falta de informação disponível não é possível comprovar toda a informação necessária.

O motor de pesquisa da Wizdee é um motor de pesquisa semântico e em mundo fechado, que são duas características que o tornam o sistema bastante diferente dos anteriores. Por exemplo o motor sabe quando não tem resposta para a pergunta ou pesquisa do utilizador, uma coisa que os outros sistemas não sabem, porque fazem apenas uma pesquisa de símbolos. A solução a implementar necessita assim de ser uma solução à medida dado não ser possível incorporar uma das soluções anteriormente apresentadas na plataforma da Wizdee devido às suas características.

Capítulo 3

Análise de Requisitos

Este capítulo apresenta a análise de requisitos para o desenvolvimento deste estágio, começando com um enquadramento e terminando com os requisitos do projecto.

3.1 Enquadramento

Este estágio integrou-se na plataforma tecnológica de gestão de conhecimento em desenvolvimento pela Wizdee. Esta plataforma é composta por várias componentes que permitem oferecer diversas funcionalidades, sendo uma delas a capacidade de pesquisa, fazendo também o respectivo *logging* de uso do sistema - área onde se focalizou o estágio. De seguida será apresentada um panorama geral da plataforma para uma melhor percepção dos requisitos a implementar. Posteriormente serão listadas as funcionalidades e limitações anteriores ao estágio do sistema de *logging*.

O motor de pesquisa da Wizdee permite fazer pesquisa e perguntas. Cada acesso ao motor de pesquisa representa uma sessão no sistema que é persistida numa base de dados e tem um tempo útil até expirar. A cada sessão ficam associadas todas as interacções do utilizador para com o motor de pesquisa.

Em cada pesquisa o motor recorre aos seus recursos de informação para gerar uma resposta. Estes recursos encontram-se divididos em três principais grupos: Factoides, FAQ's e Documentos.

O Factoide é uma classe caracterizada por vários atributos e múltiplas instâncias da referida classe. Esta estrutura de dados pode ser vista como uma tabela tradicional. No entanto, todos os atributos têm um significado associado (tipo) e existe a possibilidade de definir múltiplos termos (grupos de sinónimos) no nome das classes e atributos. A plataforma suporta múltiplos Factoides (ou classes) em simultâneo.

Uma FAQ representa um grupo que contém pares de pergunta-resposta. É possível ter múltiplas FAQ's em simultâneo.

Os documentos, como o próprio nome indica, são documentos que podem estar em diferentes suportes (PDF, DOC, HTML, entre outros). Estes são indexados no Lucene e ficam disponíveis para serem usados pelo sistema.

3.1.1 Funcionalidades Antes do Estágio

O sistema de *logging* antes do estágio estava preparado para as seguintes funcionalidades:

Sessões
São guardadas as sessões numa base de dados, assim como as interacções por cada sessão, onde fica registado a pesquisa feita e a resposta dada pelo motor a essa pesquisa. O início

e fim de sessão são calculados através da data da primeira e última interacção respectivamente. É possível também saber o tempo de resposta por interacção.

Recursos
Relativamente aos recursos, é possível de momento saber quais são mais vezes devolvidos pela pesquisa.

Termos de Pesquisa
Os termos de usados para pesquisar são actualmente tratados e guardados. Sendo possível ainda identificar os termos com resposta e sem resposta.

3.1.2 Limitações

No início do estágio o sistema apresentava as seguintes limitações:

Crescimento da Tabela de *Logs* na Base de Dados
Quando a tabela de *logs* na base de dados cresce o sistema fica mais lento.

Termos de Pesquisa
A extracção dos termos de pesquisa resume-se a retirar as *stopwords*[1] da pesquisa e armazená-la. Termos, como por exemplo tempo verbais, singulares e plurais, são tratados como sendo termos diferentes, isto é, actualmente para o sistema os termos "universidade" e "universidades" são dois termos diferentes. Outra limitação é o facto de não existir referência temporal de entrada de termos no sistema.

Interacções
Actualmente o sistema apenas consegue detectar interacções de pesquisa, representado pela entrada no sistema do texto de pesquisa e saída de uma resposta que fica associado a essa pesquisa.

3.2 Requisitos

Esta secção apresenta os requisitos do projecto que foram implementados no decorrer do estágio.

3.2.1 Requisitos Funcionais

A tabela 3.1 apresenta os requisitos funcionais de alto nível para este projecto.

ID	Nome	Descrição	Dependências
RF.01	Qualidade de Pesquisa	Fornecer informação sobre a qualidade funcional e de desempenho do sistema.	—
RF.02	Análise de Pesquisa	Analisar dados proveniente da interacção dos utilizadores com o sistema e gerar informação útil para o gestor do sistema.	—

Tabela 3.1: Requisitos funcionais.

[1] Lista de palavras sem sentido para o discurso (artigos, pronomes, preposições, entre outros).

RF.01 Qualidade de Pesquisa

Com este requisito prentendeu-se tornar visível o comportamento da plataforma ao nível da qualidade da pesquisa, quer a nível funcional quer a nível de desempenho. Ou seja, que o sistema fosse capaz de fornecer informação que permitisse perceber se o motor de pesquisa está a fornecer respostas adequadas dentro de um espaço de tempo aceitável. A tabela 3.2 apresenta em detalhe os requisitos funcionais para RF.01.

ID	Nome	Descrição	Dependências
RF.01.01	Processar Eventos de *Hit*	Funcionalidade que permita processar os eventos de *Hit* sobre os recursos.	—
RF.01.02	*Hit Rate* Total	Cálculo do *Hit Rate* total.	RF.01.01
RF.01.03	*Hit Rate* por Sessão	Cálculo do *Hit Rate* por sessão.	RF.01.01
RF.01.04	Processar os Eventos de *Bounce*	Funcionalidade que permita processar os eventos de *Bounce*.	—
RF.01.05	*Bounce Rate* Total	Cálculo do *Bounce Rate* total.	RF.01.04
RF.01.06	*Bounce Rate* por Sessão	Cálculo do *Bounce Rate* por sessão.	RF.01.04
RF.01.07	Processar Eventos de *Refinement*	Funcionalidade que permita processar os eventos de *Refinement*.	—
RF.01.08	*Refinement Rate* Total	Cálculo do *Refinement Rate* total.	RF.01.07
RF.01.09	*Refinement Rate* por Sessão	Cálculo do *Refinement Rate* por sessão.	RF.01.07
RF.01.10	Processar Pesquisas	Recolha de toda a informação resultante do acto de pesquisa.	—
RF.01.11	Pesquisa sem Resposta	Funcionalidade que permite identificar as pesquisas efectuadas sem resposta.	RF.01.10
RF.01.12	Média de Pesquisas Respondidas por Sessão	Cálculo do número médio de pesquisas respondidas por sessão.	RF.01.10
RF.01.13	Média de Pesquisas sem Resposta por Sessão	Cálculo do número médio de pesquisas sem resposta por sessão.	RF.01.10
RF.01.14	Processar Tempo de Pesquisa	Funcionalidade que permite calcular o tempo de resposta de cada pesquisa.	RF.01.10
RF.01.15	*Top* de Pesquisas mais Lentas	Funcionalidade que permite identificar as pesquisas mais lentas.	RF.01.14
RF.01.16	*Top* de Pesquisas mais Rápidas	Funcionalidade que permite identificar as pesquisas mais rápidas.	RF.01.14
RF.01.17	Tempo Médio de Resposta por Sessão	Cálculo do tempo médio de resposta por sessão.	RF.01.14
RF.01.18	Tempo Médio de Resposta por Tipo de Recurso	Cálculo do tempo médio de resposta por tipo de recurso.	RF.01.14
RF.01.19	Tempo Médio de Resposta por Pesquisa	Cálculo do tempo médio de resposta por pesquisa.	RF.01.14
RF.01.20	*Top* de Termos de Pesquisa Mais Rápidas	Identificar os termos de pesquisa cujo o tempo de resposta é mais rápido.	RF.01.14
RF.01.21	*Top* de Pesquisas Mais Lentas	Identificar os termos de pesquisa cujo o tempo de resposta é mais lento.	RF.01.14

Tabela 3.2: Requisitos da Qualidade de Pesquisa.

RF.01.01 Processar Eventos de *Hit*

Este requisito tem como objectivo a implementação da funcionalidade que permita identificar e guardar todos os eventos de *Hit* sobre os resultados de pesquisa.

RF.01.02 *Hit Rate* Total

Este requisito tem como objectivo a implementação da funcionalidade que permita calcular a taxa total de pesquisas em que foram seleccionados resultados.

RF.01.03 *Hit Rate* por Sessão

Este requisito é semelhante ao anterior, apenas com a restricção de serem contabilizadas

as pesquisas agrupadas por sessão.

RF.01.04 Processar Eventos de *Bounce*
Este requisito tem como objectivo a implementação da funcionalidade que permita identificar e guardar todos os eventos de *Bounce* sobre pesquisas.

RF.01.05 *Bounce Rate* Total
Este requisito tem como objectivo a implementação da funcionalidade que permita calcular e visualizar a taxa total de pesquisas em que não foram seleccionados resultados e de seguida o utilizador abandonou o sistema.

RF.01.06 *Bounce Rate* por Sessão
Este requisito é semelhante ao anterior, apenas com a restricção de serem contabilizadas as pesquisas agrupadas por sessão.

RF.01.07 Processar Eventos de *Refinement*
Este requisito tem como objectivo a implementação da funcionalidade que permita identificar e guardar todos os eventos de *Refinement* sobre pesquisas.

RF.01.08 *Refinement Rate* Total
Este requisito tem como objectivo a implementação de uma funcionalidade que permita calcular e visualizar a taxa total de pesquisas que são um refinamento da pesquisa anterior.

RF.01.09 *Refinement Rate* por Sessão
Este requisito é semelhante ao anterior, apenas com a restricção de serem contabilizadas as pesquisas agrupadas por sessão.

RF.01.10 Processar Pesquisas
Este requisito tem como objectivo a implementação da funcionalidade que permita identificar e guardar toda a informação resultante dos eventos de pesquisa, incluindo informação das respostas dadas.

RF.01.11 Pesquisa sem Resposta
Este requisito tem como objectivo a implementação da funcionalidade que permita identificar as pesquisa às quais o sistema não conseguiu dar resposta.

RF.01.12 Média de Pesquisas Respondidas por Sessão
Este requisito tem como objectivo a implementação da funcionalidade que permita calcular a média de pesquisas a que o sistema conseguiu dar resposta, por sessão.

RF.01.13 Média de Pesquisas sem Resposta por Sessão
Este requisito tem como objectivo a implementação da funcionalidade que permita calcular a média de pesquisas a que o sistema não consegue dar resposta, por sessão.

RF.01.14 Processar Tempo de Pesquisa
Este requisito tem como objectivo a implementação da funcionalidade que permita guardar o tempo que cada pesquisa demora a devolver obter uma resposta.

RF.01.15 *Top* de Pesquisas mais Lentas
Este requisito tem como objectivo a implementação da funcionalidade que permita iden-

tificar o top de pesquisa mais lentas no sistema.

RF.01.16 *Top* de Pesquisas mais Rápidas
Este requisito é semelhante ao anterior, apenas com a diferença de serem mostradas as pesquisas mais rápidas.

RF.01.17 Tempo Médio de Resposta por Sessão
Este requisito distina-se a calcular o tempo médio de resposta por sessão.

RF.01.18 Tempo Médio de Resposta por Tipo de Recurso
Este requisito distina-se a calcular o tempo médio de resposta de cada tipo de recurso do sistema.

RF.01.19 Tempo Médio de Resposta por Pesquisa
Este requisito distina-se a calcular o tempo médio de resposta por pesquisa.

RF.01.20 *Top* de Termos de Pesquisa Mais Rápidas
Este requisito tem como objectivo identificar os termos de pesquisa cujo o tempo médio de resposta das pesquisas associadas é mais rápido.

RF.01.21 *Top* de Termos de Pesquisa Mais Lentas
Este requisito tem como objectivo identificar os termos de pesquisa cujo o tempo médio de resposta das pesquisas associadas é mais lento.

RF.02 Análise de Pesquisa

Com os dados obtidos através do sistema de recolha de informação, o sistema sintetiza essa informação e gera quadros informativos, gráficos, entre outros, de modo a facilitar a visualização de informação útil e que de outro modo seria de difícil acesso. A tabela 3.3 apresenta em detalhe os requisitos funcionais para RF.02.

ID	Nome	Descrição	Dependências
RF.02.01	*Top* de Pesquisas por *Hit Rate*	Identificar e visualizar os termos de pesquisa com mais *Hit Rate*.	RF.01.01, RF.01.10
RF.02.02	*Top* de Pesquisas por *Bounce Rate*	Identificar e visualizar os termos de pesquisa com mais *Bounce Rate*.	RF.01.04, RF.01.10
RF.02.03	*Top* de Pesquisas por *Refinement Rate*	Identificar e visualizar os termos de pesquisa com mais *Refinement Rate*.	RF.01.07, RF.01.10
RF.02.04	*Top* de Pesquisas Mais Frequentes	Identificar e visualizar os termos de pesquisa mais frequentes.	RF.01.10
RF.02.05	*Top* de Pesquisas com Mais Resultados	Identificar os termos de pesquisa que devolvem mais resultados.	RF.01.10
RF.02.08	*Top* de Termos de Pesquisa Mais Frequentes	Identificar os termos de pesquisa mais frequentes.	RF.01.10
RF.02.09	*Top* de Termos de Pesquisa com Mais Resultados	Identificar os termos de pesquisa que obtêm mais resultados de resposta.	RF.01.10
RF.02.07	*Top* de Recursos com mais *Hits*	Identificar os recursos com maior número de *Hits*.	RF.01.01, RF.01.10
RF.02.08	*Top* de Tipos de Recursos	Identificar os tipos de recursos mais vezes devolvidos nas pesquisas.	RF.01.10
RF.02.09	*Trends* de Pesquisa	Identificar os termos mais frequentemente usados nos últimos 30 dias.	RF.01.10
RF.02.10	*Top Movers*	Identificar os termos com maior variação nas tendências.	RF.02.09
RF.02.11	*Long Running*	Identificar os termos com maior permanência nas tendências.	RF.02.09
RF.02.12	*Novelty*	Identificar os termos novos nas tendências.	RF.02.09
RF.02.13	Quantidade de Recursos por Tipo de Resultado	Manter uma contagem de recursos por tipo de resultado.	—
RF.02.14	Número Médio de Pesquisas por Sessão	Cálculo da média das pesquisas efectuadas por sessão.	RF.01.10
RF.02.15	Tempo Médio por Sessão	Cálculo do tempo médio de duração de sessão.	RF.01.10

Tabela 3.3: Requisitos da Análise de Pesquisa.

RF.02.01 *Top* de Termos de Pesquisa por *Hit Rate*
Neste requisito pretende-se que o sistema seja capaz de identificar as pesquisas com maior *Hit Rate*. Para tal é necessário implementar um sistema que permita recolher essa informação e permitir a sua visualização.

RF.02.02 *Top* de Termos de Pesquisa por *Bounce Rate*
Neste requisito pretende-se que o sistema seja capaz de identificar as pesquisas com maior *Bounce Rate*. Para tal é necessário implementar um sistema que permita recolher essa informação e permitir a sua visualização.

RF.02.03 *Top* de Termos de Pesquisa por *Refinement Rate*
Neste requisito pretende-se que o sistema seja capaz de identificar as pesquisas com maior *Refinement Rate*. Para tal é necessário implementar a funcionalidade que permita recolher essa informação e permitir a sua visualização.

RF.02.04 *Top* de Termos de Pesquisa
Este requisito tem como objectivo a implementação da funcionalidade que permita recolher as pesquisas ou termos usados e mostrar aqueles que são usados mais frequente-

mente.

RF.02.05 *Top* de Pesquisas com Mais Resultados
Este requisito tem como objectivo a implementação da funcionalidade que permita recolher o número de resultados devolvidos por pesquisa e mostrar as pesquisas ou termos que mais resultados devolvem.

RF.02.06 *Top* de Recursos mais Acedidos
Este requisito tem como objectivo a implementação da funcionalidade que permita recolher e visualizar os recursos que são devolvidos nas pesquisa mais frequentemente.

RF.02.07 *Top* de Recursos com mais *Hits*
Este requisito tem como objectivo a implementação da funcionalidade que permita recolher e visualizar os recursos que mais vezes são vistos.

RF.02.08 *Top* de Tipos de Recursos
Este requisito tem como objectivo a implementação da funcionalidade que permita recolher e visualizar os tipos de recursos que mais vezes são devolvidos na pesquisa.

RF.02.09 *Trends* de Pesquisa
Neste requisito pretende-se que o sistema seja capaz de identificar as tendências da semana através das pesquisas efectuadas e guardar um histórico das mesmas.

RF.02.10 *Top movers*
Neste requisito pretende-se que o sistema seja capaz de identificar as pesquisas com maior variação ao longo do tempo.

RF.02.11 *Long Running*
Neste requisito pretende-se que o sistema seja capaz de identificar as pesquisas que mais tempo permanecem como tendência actual.

RF.02.12 *Novelty*
Neste requisito o sistema deverá identificar a entrada de novas pesquisas para as tendências.

RF.02.13 Quantidade de Recursos por Tipo de Resultado
Neste requisito o sistema deverá manter uma contagem dos recursos disponíveis por tipo de resultado.

RF.02.14 Número Médio de Pesquisas por Sessão
Neste requisito pretende-se que o sistema seja capaz de guardar o número de pesquisas feito em cada sessão e apresentar uma média das mesmas.

RF.02.15 Tempo Médio por Sessão
Neste requisito o sistema deve calcular o tempo de duração das sessões e mostrar uma média destes.

3.2.2 Requisitos Tecnológicos

A tabela 3.4 apresenta os requisitos tecnológicos (ferramentas, recursos e desenvolvimento) para o projecto.

ID	Nome	Descrição	Dependências
TR.01	Linguagem de Programação	Toda a programação foi desenvolvida utilizando a linguagem de programação Java e Javascript.	—
TR.02	Licenças	As licenças de todo o software utilizado permitem a comercialização sem encargos adicionais. As licenças permitidas são: LGPL, BSD License, Apache License, MIT License ou Common Public License.	—

Tabela 3.4: Requisitos Tecnológicos.

Capítulo 4

Arquitectura

Neste capítulo é apresentada a arquitectura da plataforma da Wizdee e quais as camadas e componentes que a constituem. Também se descrevem as componentes que foram necessárias adaptar ou criar de raiz para o desenvolvimento do projecto.

4.1 Camadas

Esta secção apresenta a arquitectura actual da plataforma e as suas diversas camadas (ver Figura 4.1, os nomes das componentes são apresentados com o seu nome original na plataforma.).

A plataforma divide-se em três camadas: camada de dados, camada de negócio e camada de apresentação. A camada de dados trata da persistência de dados em estruturas adequadas à pesquisa e armazenamento. A camada de negócio contém toda a lógica de funcionamento da plataforma e apresenta quatro tipos de componentes: *Managers*, *Handlers*, *Engines* e *Modules*. Em paralelo existem as *Libraries*, ferramentas abstractas que são usadas pelo sistema de um modo transversal. Finalmente a camada de apresentação fornece a interface ao utilizador, neste caso em formato de aplicação Web.

De seguida vai ser apresentado em detalhe cada uma das camadas da plataforma.

4.1.1 Camada de Dados

A camada de dados do sistema permite persistir os dados necessários ao seu funcionamento, bem como a optimização de tarefas específicas, tais como a pesquisa e a representação de relações entre dados.

Base de Dados
A base de dados permite a persistência dos dados necessários ao funcionamento do sistema. Actualmente o sistema usa o motor de base de dados PostgreSQL[1] mas a migração para um RDBMS[2] diferente é uma tarefa de baixa complexidade dada a separação em camadas que a plataforma oferece.

Repositório de Documentos
O repositório de documentos tem como base a plataforma de pesquisa Solr, que por sua vez é implementada sobre motor de pesquisa Lucene. Esta estrutura é utilizada para que o sistema possa executar pesquisas de texto de forma eficiente, algo impossível de

[1] `http://www.postgresql.org`.
[2] Relational Database Management System - Sistema de base de dados relacional.

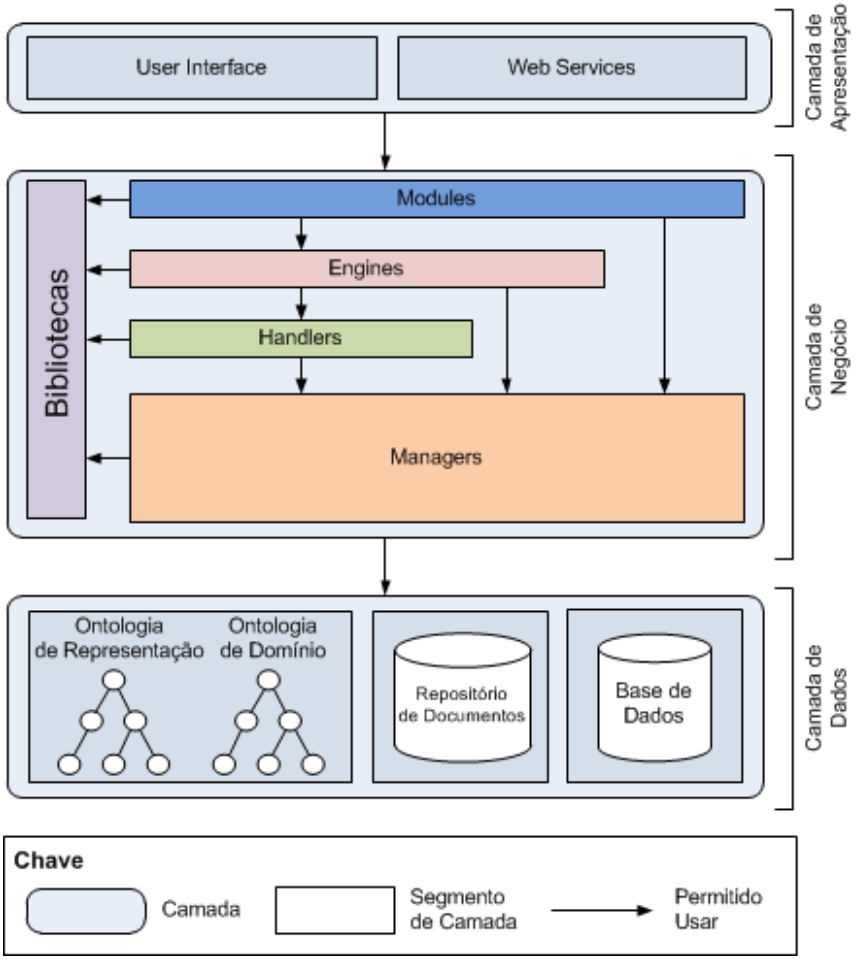


Figura 4.1: Arquitectura da Plataforma.

realizar numa base de dados relacional dado que está não está preparada para lidar de forma eficiente com procura de texto. Neste repositório é possível guardar qualquer tipo de dados para que se possa realizar uma pesquisa sobre toda a base de conhecimento disponível.

Ontologia
São utilizadas duas Ontologias[3]: uma Ontologia de Representação e uma Ontologia de domínio.

A Ontologia de representação tem a função de manter as relações entre dados. Esta Ontologia é importante para definir relações complexas entre dados que não possam estar definidas noutro local da camada de dados. Qualquer tipo de dados pode tirar partido desta estrutura.

A Ontologia de domínio (ou lexical) tem a função de agrupar termos com o mesmo sentido. Esta fornece informação que permite relacionar dados que estejam associados com base num mesmo sentido.

4.1.2 Camada de Negócio

A camada de negócio está dividida em quatro tipos de componentes que interagem entre si e que serão descritas de seguida.

[3] `http://semanticweb.org/wiki/Ontology`.

Managers
Os *Managers* permitem a interacção com a camada de dados, executando todas as operações básicas (CRUD[4]) sobre os mesmos. A sua principal função é o acesso abstracto aos dados, de modo a que a persistência de dados seja transparente para as restantes componentes.

Handlers
Os *Handlers* têm como função processar o texto de entrada num pedido (seja linguagem natural ou pesquisa) e determinar uma resposta para o mesmo. Cada *Handler* anota parte do pedido que deu entrada com uma possível resposta sendo possível que para um pedido sejam geradas múltiplas respostas (uma por *Handler*). Neste caso será o engine que invocou os handlers a decidir qual ou quais as respostas a serem apresentadas com base num mecanismo de prioridade, natureza de cada handler e a certeza que estes têm na resposta gerada.

Engines
Os *Engines* funcionam como agregadores de *Handlers*. Coordenam o processo de geração de resposta para um dado pedido. É esta componente que contém a lógica de selecção de resposta, gestão da sessão e gravação de estatísticas relacionadas com a geração da resposta. Um *Engine* é criado para fornecer uma funcionalidade de alto nível ao sistema como suporte de conversação ou pesquisa.

Modules
Os *Modules* servem de API da camada de negócios e agregam vários *Managers* e *Engines* em operações complexas. Estes serão usados pela camada de apresentação e API externa.

4.1.3 Camada de Apresentação

Esta é a camada que fornece Web Services para o exterior, interfaces de gestão e *Front-Ends* de utilização.

Interfaces Web
A camada de apresentação (*Front-End*) são aplicações *Web* que permite aos utilizadores interagirem com o sistema, tanto a nível de administração e gestão como utilização comum.

Web Services
O sistema fornece uma API REST[5] que permite integração de qualquer sistema fornecendo as funcionalidades de conversação presentes no sistema. Esta API tem sido utilizada para desenvolver *widgets* e clientes móveis.

4.2 Componentes

Nesta secção serão apresentadas todas as componentes com as quais se interagiu e que foram alvo de alterações quando necessário. Posteriormente serão apresentadas as componentes que foram criadas de raíz para o desenvolvimento deste estágio (ver Figura

[4]CRUD – Create, Read, Update e Delete (criar, ler, actualizar e apagar).

[5]Representational State Transfer - `https://www.ibm.com/developerworks/webservices/library/ws-restful/`.

4.2).

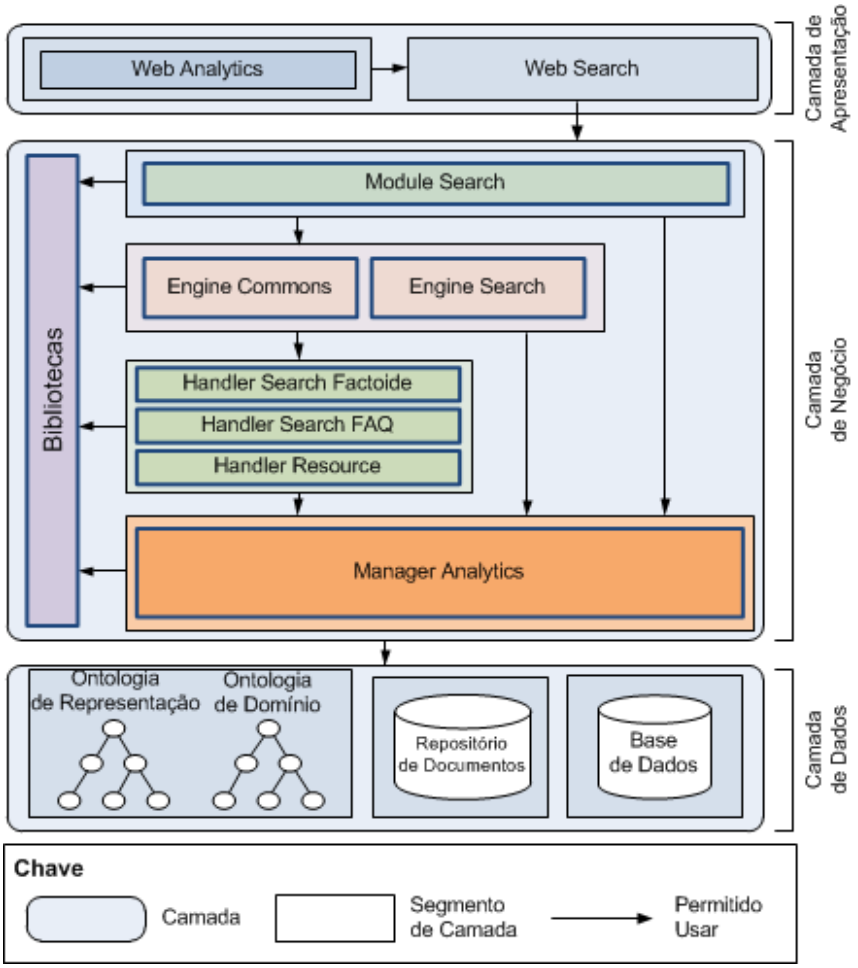


Figura 4.2: Componentes de Interacção.

4.2.1 Componentes Relevantes

Nesta secção são apresentadas as componentes que interagem com as componentes que foram implementadas e qual a natureza da interacção: simples utilização da API ou implementação de funcionalidades.

Handler Search Factoide
O *Handler Search Factoide* é a componente responsável pela geração da resposta a um pedido de *Factoide*. De momento, este *Handler* efectua todas as operações na base de dados e índice de pesquisa, assim como nas ontologias. Desta componente é utilizado a sua API para obter a informação correspondente ao tipo de recursos.

Handler Search Faq
O *Handler Search Faq* é a componente responsável pela geração da resposta a um pedido de *Faq*. De momento, este *Handler* efectua todas as operações na base de dados. Desta componente é utilizado a sua API para obter a informação correspondente ao tipo de recursos.

Handler Resource
O *Handler Resource* é a componente responsável pela geração da resposta a um pedido de

Resource. De momento, este *Handler* efectua todas as operações na base de dados. Desta componente é utilizado a sua API para obter a informação correspondente ao tipo de recursos.

Engine Commons
Esta componente é a base dos Engines, fornecendo toda a estrutura lógica para a sua implementação. Desta componente é utilizado a sua API para obter a informação.

Engine Search
Esta componente está orientada especificamente à gestão de uma pesquisa introduzida pelo utilizador e controla o fluxo de execução dos Handlers necessários para a geração da resposta. Desta componente é utilizado a sua API para obter uma resposta às pesquisas efectuadas.

Module Search
Componente responsável pela ligação entre o *Front-End* de pesquisa da Wizdee e a camada de negócio. Este módulo sofreu algumas alterações a nível da sua estrutura de modo a adicionar suporte para acesso às novas funcionalidades.

Web Search
Componente responsável pela comunicação entre o Module Search e a interface de visualizaçao de dados desenvolvida. Trata-se de um *webservice* REST que disponibiliza uma API de pesquisa para o motor. Este *webservice* foi alterado de modo a suportar a novas funcionalidades

Portal Search
Trata-se da componente gráfica de pesquisa da Wizdee. É através desta que os utilizadores podem fazer as suas pesquisas no motor e obter as respostas. Este módulo foi alterado de modo possibilitar a recolha de informação proveniente da interacção do utilizador.

4.2.2 Componentes Desenvolvidas

Nesta secção são apresentadas as componentes que foram criadas.

Manager Analytics
Esta componente é responsável pelo processamento e persistência da informação recolhida e pela gestão de pedidos sobre essa mesma informação por parte da camada superior, neste caso *Module Search*, através de uma API. Faz também a ligação entre a camanda de negócio e a camada de dados.

Web Analytics
Trata-se da componente gráfica que permite aos utilizadores vizualizarem a informação já processada em forma de texto, gráficos, entre outros.

Capítulo 5

Especificação e Desenvolvimento

Este capítulo descreve o trabalho efectuado ao longo estágio. Trabalho esse que teve origem na lista de requisitos definidos na Secção 3.2 e focou-se em duas áreas: recolha e processamento de informação e visualização de informação. De seguida será apresentada uma análise detalhada de cada uma dessas áreas.

5.1 Recolha e Processamento de Informação

A recolha de informação proveniente da interacção do utilizador com o portal de pesquisa da Wizdee é fundamental para o sistema a implementar. É desses dados que se vão extrair as informações necessárias para as funcionalidades a desenvolver. Nesta secção serão detalhados os processos de recolha de dados e respectiva base de dados. Serão ainda apresentadas algumas especificações de conceitos adquiridos na implementação.

5.1.1 Modelo de Dados

Para armazenar os dados recolhidos usou-se uma base de dados relacional em PostgreSQL[1], que já é usada pela restante plataforma. Após identificação dos dados a serem recolhidos, desenhou-se um modelo de dados que posteriormente sofreu algumas alterações para facilitar o acesso a dados. O esquema final pode ser visto da Figura 5.1.

Seguidamente será detalhado mais a pormenor cada tabela da base de dados para se perceber melhor como os dados são organizados.

Analytics Abstract Search
Nesta entidade são guardadas todas as pesquisas feitas pelos utilizadores. Caso seja uma pesquisa repetida, é incrementado um contador no registo correspondente à pesquisa, caso contrário, um novo registo é adicionado.

Analytics Abstract Search Term
Uma pesquisa é composta por termos, sendo que cada termo representa uma palavra da pesquisa, exemplo: "clientes almedina" representa uma pesquisa com 2 termos. Estes termos são separados e guardados. Não há termos repetidos na tabela. Um contador do associado a registo do termo é incrementado ou iniciado a 1, consoante se o termo já

[1] `http://www.postgresql.org/`

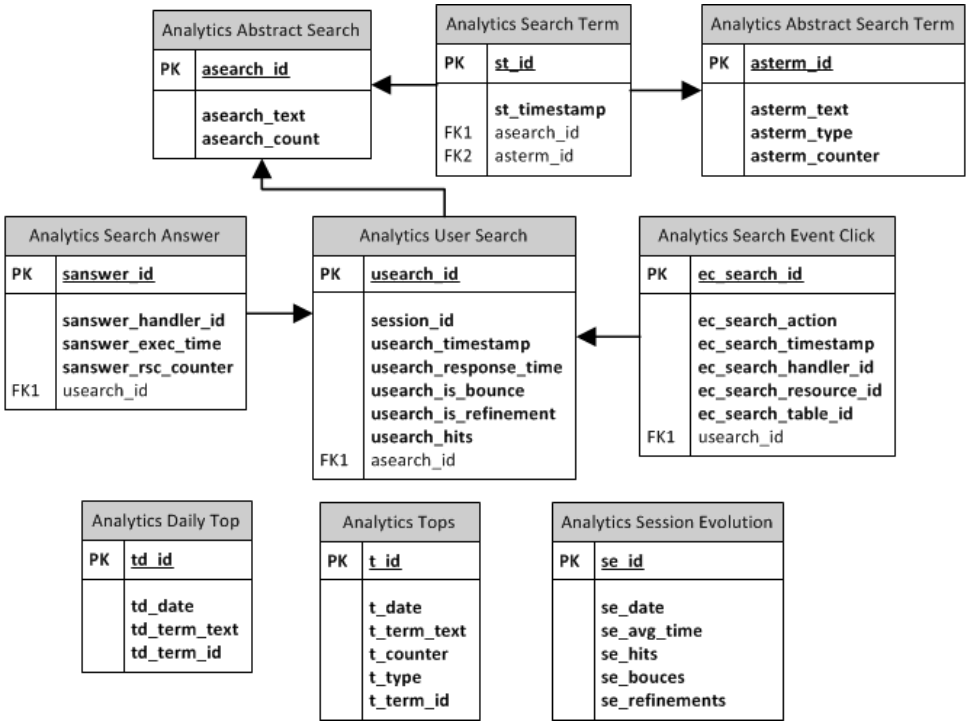


Figura 5.1: Diagrama da base de dados.

existir na tabela ou não, respectivamente.

Analytics Search Term
Esta tabela faz a ligação entre os termos e as pesquisas. Mantém um registo temporal do uso dos termos, isto é, sempre que há uma pesquisa há uma entrada nesta tabela dos termos usados.

Analytics User Search
Nesta tabela ficam associadas todas as pesquisas que são feitas numa determinada sessão. Representa as interacções de pesquisa do utilizador.

Analytics Search Answer
Cada resposta dada a uma pesquisa pelo motor, caso haja resposta, pode ter diversos tipos de recursos associados com diversos recursos cada. Nesta tabela ficam guardadas todas as informações que dizem respeito a essa parte.

Analytics Search Event Click
As acções de *hit* sobre os resultados de pesquisa são guardados nesta tabela. Fica registado que recurso sofreu o *hit*, assim como também a que pesquisa pertence.

Analytics Daily Top
Guarda os *top* diários das tendências que servem posteriormente para calcular os *novelties*, *long runners* e *top movers*.

Analytics Tops
Tabela que contém os *novelties*, *long runners* e *top movers* para o dia. Esta informação é calculada a partir dos registos da tabela anterior.

Analytics Session Evolution
Contém informação de evolução de determinados indicadores de sessão ao longo do tempo. Esta informação é calculada diariamente.

5.1.2 Processamento Primário dos Dados

São recolhidos dados relativos a pesquisas inseridas pelo utilizador, dados provenientes da resposta do sistema à pesquisa introduzida e dados de acção do utilizador após a pesquisa. Esses dados são tratados e agrupados no Module Search e enviados posteriormente para o Manager Analytics, onde está centralizada toda a lógica deste projecto, sendo guardados na base de dados para uso futuro. Pode-se ver na Figura 5.2 o fluxo de informação.

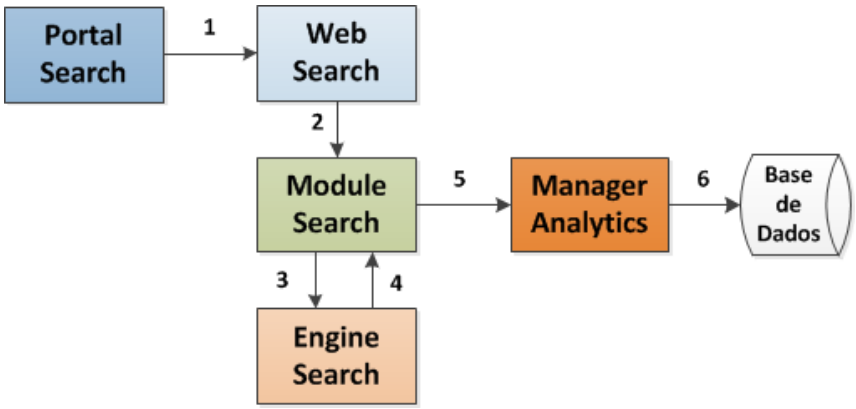


Figura 5.2: Fluxo de informação.

Relativamente aos dados, apresenta-se uma descrição detalhada da forma como são tratados e guardados para responder às necessidades requeridas.

Pesquisa

Da pesquisa introduzida é enviado para o Manager Analytics o texto escrito pelo utilizador juntamente com a informação a que sessão pertence. Esse texto é separado em palavras e anotado por ferramentas da plataforma para identificar verbos, *stopwords* entre outros.

De seguida é inserida a informação de pesquisa na base de dados, mais concretamente, na tabela Analytics Abractract Search com as regras referidas anteriormente para esta tabela. Após este passo, é a vez de guardar os termos de pesquisa na tabela Analytics Abstract Search Term uma vez mais seguindo as regras referidas anteriormente para esta tabela.

Um passo importante é ligar os termos com as pesquisas, para tal são inseridos na tabela Analytics Search Term as relações entre ambos, sendo adicionado um *timestamp* em cada registo. É criada uma entrada na tabela Analytics User Search, representando uma pesquisa de utilizador, ficando associado a uma sessão do sistema e ao registo da pesquisa correspondente da tabela Analytics Abstract Search, sendo também marcada com um *timestamp*.

Resposta

Após processados e guardados os dados de pesquisa, passa-se a guardar os dados da resposta dada pelo motor. É extraído o número de recursos devolvidos por cada tipo de recurso, assim como o tempo de execução de cada um. Esta informação é guardada na tabela Analytics Search Answer sendo inserido um registo por cada tipo de recurso com o respectivo identificador de pesquisa (Analytics User Search). O tempo de resposta de cada tipo de recurso é somado, contabilizando assim o tempo total de resposta, sendo actualizado no registo da pesquisa correspondente da tabela Analytics User Search.

Interacção

Resultados
Relativamente aos dados de interacção sobre os resultados, considera-se apenas os *clicks* dados sobre os resultados de pesquisa apresentados. Para este efeito, ao clicar num resultado, deve ser enviado para o servidor que acção foi executada, neste caso um *hit*, que recurso foi clicado e a que tipo de recurso pertence, assim como também de que pesquisa de utilizador teve origem. Essa informação é inserida na tabela Analytics Search Event Click ficando associado ao registo da pesquisa correspondente da tabela Analytics User Search, sendo ainda esse registo incrementado no campo correspondentes ao número de *hits*.

Pesquisa
Quando na mesma sessão é feita mais do que uma pesquisa, existem duas acções feitas pelo sistema que são transparentes para o utilizador - verificação de se a pesquisa anterior representa um *bounce* ou *refinement*.

São considerados *bounces*, pesquisas em que nenhum resultado sofreu uma acção de *hit* sendo feita uma nova pesquisa ou a janela do *browser* fechada ou em caso da sessão expirar.

Para a verificação de um *refinement*, é necessário comparar a pesquisa anterior com a nova. De ambos os textos de pesquisa são removidas as *stopwords* sendo depois comparadas as restantes palavras entre si. Se a nova pesquisa contiver alguma palavra da pesquisa anterior, então considera-se essa nova pesquisa como sendo um refinamento da anterior.

Esta informação é guardada na tabela Analytics User Search, no registo correspondente à pesquisa em causa, tendo em atenção que se for um *bounce*, é guardado na pesquisa antiga, se for um *refinement* é guardado na nova pesquisa. É de notar que é possível existir um *bounce* e um *refinement* na mesma interacção.

5.1.3 Outros Processamentos

Uma vez os dados de informação de pesquisa recolhidos e armazenados nas respectivas tabelas é necessário proceder a pré-cálculos de modo a que o acesso à informação seja mais rápido caso fossem executados em tempo real.

Tendências

Com origem nos termos de pesquisa que são guardados, reflectem as tendências de pesquisa dos utilizadores num dado período de tempo. Diariamente são identificados um top 10 das tendências do dia anterior, sendo esses dados armazenado na tabela Analytics Daily Top. De notar que o top 10 diário é calculado pelos termos de pesquisa e tendo por base a informação temporal tabela Analytics Search Term. Para o cálculo

diário usa-se o Quartz Scheduler[2] que é uma *framework* que permite agendar trabalhos e executá-los no tempo definido. Para facilitar a implementação, definiu-se uma janela de tempo movível de 30 dias, ou seja, sempre que um dia passa a janela avança um dia também.

Dentro desse período é possível identificar 3 tipos de tendências que ajudam a perceber melhor a variação de procura de conhecimento no sistema, sendo cada uma detalhada de seguida.

Novelties
Com os *novelties* pretende-se identificar as novas tendências de pesquisas nos últimos dias. Foi definido extrair os termos dos últimos 3 dias que não aparecem durante os restantes 27 dias. Os termos são guardados na tabela Analytics Tops, juntamente com identificação do *top* a que pertencem, neste caso *novelty*, e número de ocorrências.

Long Runners
Identifica as tendências que se mantêm constantes ao longo do tempo. Foi definido identificar os termos que aparecem mais que 86% das vezes nos *tops*. Os 86% corresponde à não ocorrência desta tendência cerca de 4 vezes por mês. Com a restrição de ter obrigatoriamente de aparecer pelo menos 3 dias consecutivos semanais, obriga que as não aparências mensais sejam repartidas por semana, eliminando a possibilidade de ocorrerem todas consecutivas. Os termos são guardados na tabela Analytics Tops, juntamente com identificação do *top* a que pertencem, neste caso *long runner*, e número de ocorrências.

Top Movers
Identifica as tendências que estão constantemente a entrar e a sair dos tops diários. Foi definido identificar os termos que apareçam entre 30% e 86% das vezes nos *tops* e que nunca apareçam 3 dias consecutivos. Desta forma consegue-se diferenciar dos *long runners* e excluir termos que surgem esporadicamente, assegurando uma distribuição uniforme no período de tempo definido. Os termos são guardados na tabela Analytics Tops, juntamente com identificação do *top* a que pertencem, neste caso *top mover*, e número de ocorrências.

Evolução de Indicadores de Sessão

Como indicadores de sessão são agrupados por dia o número de pesquisas com *hit*, pesquisas marcadas com *bounce*, *refinement* e tempo de sessão. À semelhança do *top* diário de tendências, esta informação é também recolhida diariamente sendo definida uma tarefa no Quartz Scheduler para esse efeito. Os dados são guardados na tabela Analytics Session Evolution.

5.2 Visualização de Informação

Nesta secção irá ser apresentado em detalhe que informação é mostrada e como pode ser obtida na base de dados.

Foi criado uma interface gráfica, Web Analytics, responsável por apresentar os dados ao utilizador. Foi decidido dividir a apresentação de informação em duas secções, Informação Geral e Informação de Sessão, para facilitar a consulta da informação. Ambas as secções são detalhadas em seguida.

[2] `http://quartz-scheduler.org/`

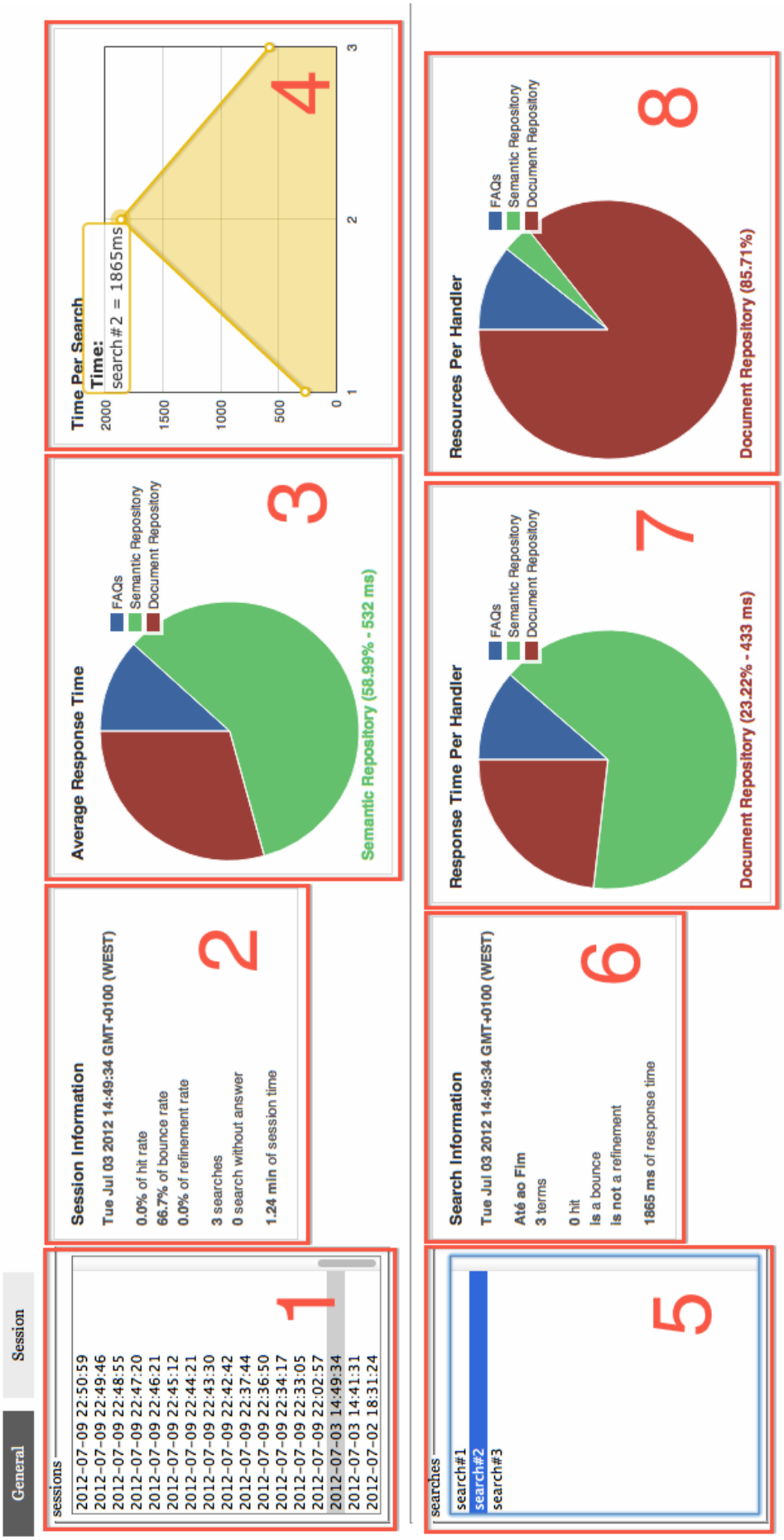


Figura 5.3: Ecrã da secção Sessão.

5.2.1 Informação de Sessão

Na informação de sessão (Figura 5.3) pode-se ver o detalhe de uma sessão em particular. É apresentado uma lista com as sessões (Figura 5.3 - quadro 1) todas ordenadas por data, onde o utilizador selecciona a sessão de que pretende ver detalhes. Essa lista de sessões é obtida através da consulta da tabela Analytics User Search, agruparando as pesquisas por sessão, obtendo-se desta forma as sessões distintas guardadas. Para a interface, para além do identificador de sessão, é enviado a data em que ocorreu a sessão. Na lista, a sessão é identificada pela data. Ao seleccionar uma sessão da lista, é enviado para o servidor o identificador de sessão, sendo retornado a informação dessa sessão em específico. Essa informação foi dividida em 4 quadros: dados de sessão, tempo médio de resposta, tempo de resposta por pesquisa e lista de pesquisas.

Dados de Sessão

Neste quadro (Figura 5.3 - quadro 2) podemos ver a seguinte informação da sessão seleccionada: *hit rate*, *bounce rate*, *refinement rate*, número de pesquisas, número de pesquisas sem resposta e tempo de sessão.

O *hit rate* é calculado através do quociente do número de pesquisa em que existiram *hit* pelo número total de pesquisas. O *bounce rate* é calculado de forma semelhante sendo o quociente do número de pesquisas marcadas com *bounce* pelo número total de pesquisas. O *refinement rate* é o quociente do número de pesquisas marcadas como *refinement* sobre o número total de pesquisas. Estes cálculos são efectuados tendo apenas em consideração as pesquisas que pertencem à sessão seleccionada. Esta informação é obtida através de consultas à tabela Analytics User Search.

O número de pesquisas mostra a quantidade de pesquisas feitas na sessão, sendo facilmente calculado com uma contagem à tabela Analytics User Search restringindo ao id de sessão. Quanto ao número de pesquisa sem resposta, que mostra a quantidade de pesquisas para o qual o motor não encontrou resposta, é calculado através de uma contagem relacionando as tabelas Analytics User Search e Analytics Search Answer através da chave primária (usearch_id) da primeira tabela.

Finalizando, o tempo de sessão, que representa a duração da sessão, é calculado a partir da diferença entre o máximo valor do máximo entre a data da última pesquisa efectuada e da última acção registada, pela data da primeira pesquisa da sessão. A data da primeira e última pesquisa da sessão, podem ser obtidas na tabela Analytics User Search. Já a data da última acção da sessão pode ser obtida relacionando a tabela Analytics User Search, através da sua chave primária (usearch_id), com a tabela Analytics Search Event Click.

Tempo Médio de Resposta

O tempo médio de resposta (Figura 5.3 - quadro 3) mostra o tempo médio gasto por cada tipo de recurso para apresentar resposta a uma pesquisa. Esta informação é obtida relacionando as tabelas Analytics User Search e Analytics Search Answer através da chave primária (usearch_id) da primeira tabela, agrupando em seguida, por tipo de recurso, os registos da tabela Analytics Search Answer, fazendo a média dos tempos de resposta de cada tipo de recurso.

Tempo de Resposta por Pesquisa

Apresenta o tempo de resposta para cada uma das pesquisas efectuadas na sessão (Figura 5.3 - quadro 4). Pode-se obter esta informação através de uma consulta, restringida ao

id de sessão, à tabela Analytics User Search que já contém o tempo total da pesquisa em cada registo.

Lista de Pesquisas

Apresenta uma lista das pesquisas que foram feitas na sessão (Figura 5.3 - quadro 5). Essa lista é obtida através da consulta, restringida pelo id de sessão, da tabela Analytics User Search, obtendo-se assim as pesquisas da sessão correspondente. Para a interface, para além do identificador de pesquisa (usearch_id), é enviado o texto de pesquisa que é apresentado na lista, sendo obtido relacionando as tabelas Analytics Abstract Search e Analytics User Search através da chave primária (asearch_id) da primeira tabela.

Ao seleccionar uma pesquisa da lista, é enviado para o servidor o identificador de pesquisa, sendo retornado a informação dessa pesquisa em específico. Essa informação foi dividida em 3 quadros: informação de pesquisa, tempo de resposta por tipo de recurso e recursos por tipo de recurso.

Informação de Pesquisa
Neste quadro (Figura 5.3 - quadro 6) pode-se ver a seguinte informação da pesquisa seleccionada: data de pesquisa, número de termos de pesquisa, número de *hits*, indicação de *bounce*, indicação de *refinement* e tempo de resposta.

A data de pesquisa indica quando a pesquisa foi efectuada. O número de *hits* é a contagem de resultados de pesquisa que foram clicados. A indicação de *bounce* e *refinement* mostra se uma pesquisa foi marcada como *bounce* e/ou *refinement*. O tempo de resposta indica o tempo total de resposta. Esta informação pode ser toda acedida directamente através de uma consulta à tabela Analytics User Search com a restrição ao identificador da respectiva pesquisa. O número de termos indica quantos termos foram usados na pesquisa, sendo necessário relacionar as tabelas Analytics Abstract Search e Analytics User Search através da chave primária (asearch_id) da primeira tabela, acedendo ao número de termos da pesquisa.

Tempo de Resposta por Tipo de Recurso
Este quadro (Figura 5.3 - quadro 7) mostra o tempo com que cada tipo de recurso contribui para o tempo total de resposta. Esta informação é acedida através do relacionamento entre as tabelas Analytics User Search e Analytics Search Answer através da chave primária (usearch_id) da primeira tabela, restringida pelo identificador da respectiva pesquisa.

Recursos por Tipo de Recurso
Este quadro (Figura 5.3 - quadro 8) mostra a quantidade de recursos com que cada tipo de recurso contribuiu para a resposta à pesquisa. Esta informação é acedida através do relacionamento entre as tabelas Analytics User Search e Analytics Search Answer através da chave primária (usearch_id) da primeira tabela, restringida pelo identificador da respectiva pesquisa.

5.2.2 Informação Geral

Nesta secção pode-se ver a informação geral do sistema (Figura 5.4). É a informação que é mostrada quando se acede ao Web Analytics. A informação está agrupada em 10 quadros: informação dos recursos do sistema (3 quadros), tipo de recursos mais vezes devolvidos nas pesquisas, recursos com mais *hits*, informação geral de sessão, *top* de pesquisa, *top* de termos de pesquisa, tendências e evolução de métricas de sessão.

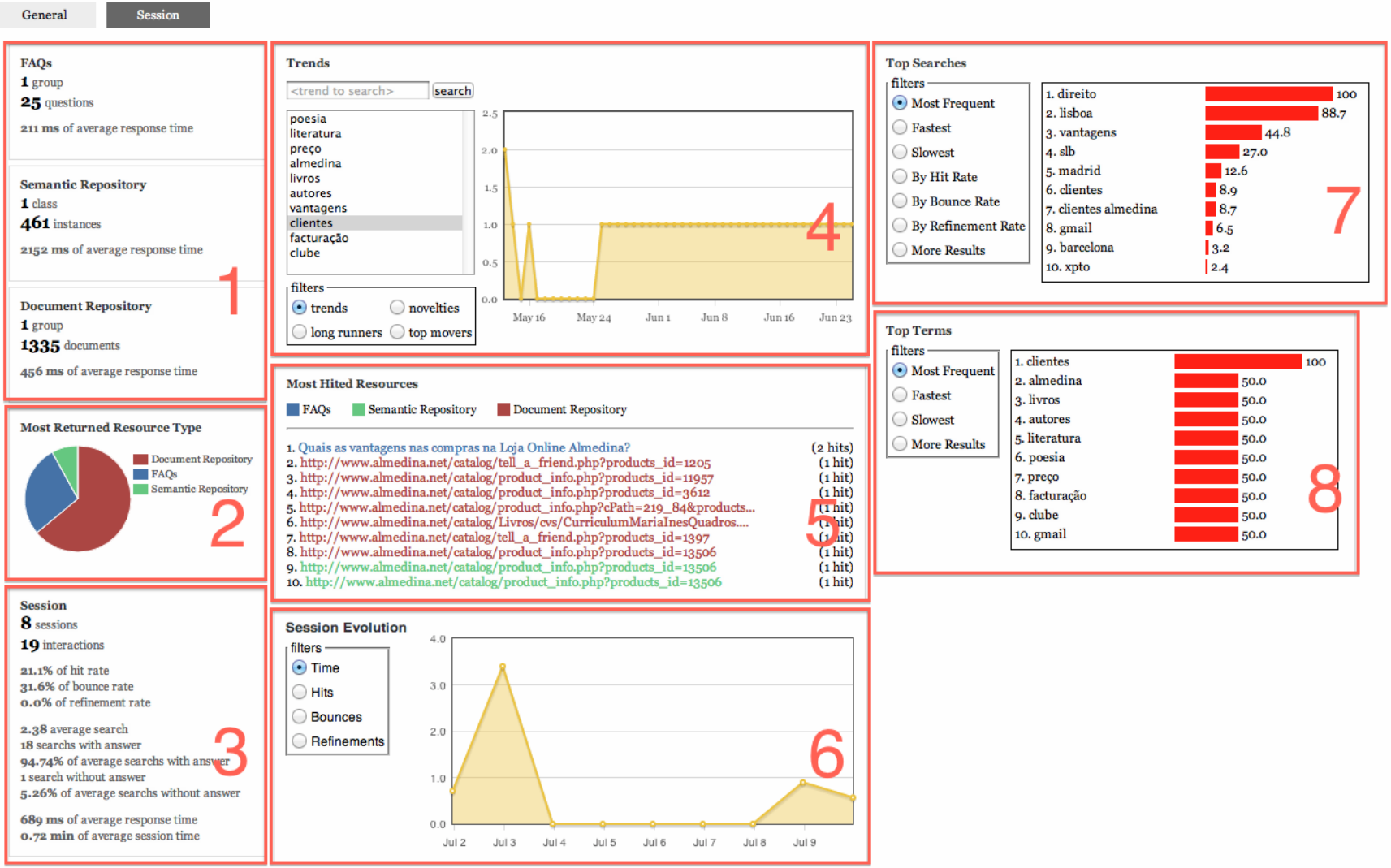


Figura 5.4: Ecrã da secção Geral.

Informação dos Recursos do Sistema

Mostra os recursos disponíveis para o sistema (Figura 5.4 - quadro 1). São estes recursos que o motor usa para dar resposta às pesquisas. Existem 3 tipos de recursos disponíveis: FAQs, Repositório Semântico e Repositório de Documentos.

As FAQs são compostas por grupos e cada grupo tem n-pares de pergunta-resposta. O quadro mostra o número total de grupos, perguntas e o tempo médio de resposta. A quantidade de grupos é obtida com uma consulta de contagem à tabela FAQ. O número total de perguntas é obtido através de uma consulta de contagem à tabela Faq Question. O tempo médio de resposta é obtido através de consulta à tabela Analytics Search Answer restringida pelo tipo de recurso (sanswer_handler_id), neste caso "FAQ", calculando a média de tempo de resposta.

O Repositório Semântico é composto por classes e cada classe tem as suas instâncias. O quadro mostra o número total de classes, instâncias e o tempo médio de resposta. A quantidade de classes é obtida com uma consulta de contagem à tabela Factoide. O número total de perguntas é obtido através de uma consulta de contagem à tabela Factoide Instance. O tempo médio de resposta é obtido através de consulta à tabela Analytics Search Answer restringida pelo tipo de recurso (sanswer_handler_id), neste caso "FACTOIDE", calculando a média de tempo de resposta.

O Repositório de Documentos é composto por grupos e cada grupo tem documentos. O quadro mostra o número total de grupos, documentos e o tempo médio de resposta. A quantidade de grupos é obtida com uma consulta de contagem à tabela Repository Group. O número total de documentos é obtido através de uma consulta de contagem à tablema Repository Resource. O tempo médio de resposta é obtido através de consulta à tabela Analytics Search Answer restringida pelo tipo de recurso (sanswer_handler_id), neste caso "RESOURCE", calculando a média de tempo de resposta.

Tipo de Recursos Mais Vezes Devolvidos

Apresenta uma comparação percentual das vezes que cada tipo de recurso é devolvido em pesquisas (Figura 5.4 - quadro 2). Essa informação pode ser obtida consultando a tabela Analytics Search Answer, agrupando os registos por tipo de recurso (sanswer_handler_id) e contando o número de vezes que cada um aparece nos registos. Esses dados são devolvidos para a interface gráfica, onde é criado um gráfico que permite comparar o uso de cada tipo de recurso pelo sistema.

Recursos Com Mais Hits

Neste quadro (Figura 5.4 - quadro 5) pode-se ver o top 10 de recursos com mais *hits*. Os dados são obtidos através de uma consulta à tabela Analytics Search Event Click, fazendo uma contagem de *hits* agrupados por identificador de recurso (ec_search_resource_id) obtendo assim os recursos com mais *hits*. Em seguida, a cada um dos 10 registos seleccionados é necessário ir buscar o recurso à tabela correspondente, usando o tipo de recurso (ec_search_handler_id) que permite identificar a origem do recurso ("FAQ", "Factoide", "Resource"). Consulta-se à tabela de origem do recurso restringida ao seu identificador (ec_search_resource_id), extraindo assim o nome do recurso.

Informação de Sessão

Este quadro (Figura 5.4 - quadro 3) permite visualizar a informação global das sessões. A seguinte informação pode ser consultada: número total de sessões, número total de interacções, *rates*, média de pesquisas, número total de pesquisas com e sem resposta,

percentagem de pesquisas com e sem resposta, tempo médio de resposta e tempo médio de sessão.

O número total de sessões obtêm-se com uma consulta de contagem à tabela Analytics User Search, contando os distintos identificadores de sessão. O número total de interacções no sistema, conta o número total de pesquisas e pode-se obter com uma consulta de contagem à tabela Analytics User Search, contando assim todos os seus registos.

Os rates - *hit rate*, *bounce rate* e *refinement rate* - já foi explicado anteriormente como eram calculados para cada sessão. Aqui o processo é o mesmo, apenas é feito uma média desses valores calculados, sendo então essas médias apresentadas.

A média de pesquisas por sessão é obtida consultando a tabela Analytics User Search, fazendo a média da contagem dos registos agrupados por sessão.

O número de pesquisas sem resposta já foi visto anteriormente como proceder ao seu cálculo. A diferença é que em vez de ser calculada agora para uma sessão em específico, é calculado no global. O número de pesquisas com resposta é calculado aproveitando o número de pesquisas total calculado anteriormente, fazendo a diferença com o número de pesquisas sem resposta. Desta forma obtém-se o valor desejado.

As percentagem de pesquisas com e sem resposta são calculadas usando dados já obtidos em consultas anteriores. Assim, a percentagem de pesquisas com resposta é calculada pelo quociente entre o número de pesquisas com resposta e o número total de pesquisas, multiplicado por 100. A percentagem de pesquisa sem resposta é calculada pelo quociente entre o número de pesquisas sem resposta e o número total de pesquisas, multiplicado por 100.

O tempo médio de resposta é obtido achando a média dos tempos de resposta de cada pesquisa. Consultando a tabela Analytics User Search, cada registo representa uma pesquisa tendo o tempo total de resposta associado.

O tempo médio de sessão, apresenta a média de duração de sessão. Anteriormente foi visto como calcular o tempo de duração de uma sessão. O processo é o mesmo, calculando para todas as sessões, fazendo uma média dos valores de cada sessão.

Top de Pesquisas

Apresenta uma lista com o *top* 10 das pesquisas mais feitas no sistema (Figura 5.4 - quadro 7). Pode-se filtrar a informação através de 7 filtros: mais frequentes, mais rápidas, mais lentas, maior *hit rate*, maior*bounce rate*, maior *refinement rate* e mais resultados devolvidos.

O filtro mais frequentes é a selecção por defeito, apresentando as pesquisas mais frequentes. Essa informação é obtida consultando a tabela Analytics Abstract Search, ordenando por ordem decrescente o registos por quantidade de pesquisas (asearch_count) e limitando aos 10 primeiros.

O filtro mais rápidas lista as 10 pesquisas mais rápidas enquanto o filtro mais lentas é o inverso, ou seja, lista as 10 pesquisas mais lentas. Estes dados são obtidos relacionando as tabelas Analytics Abstract Search e Analytics User Search através da chave primária (asearch_id) da primeira tabela, fazendo uma média dos tempos de resposta e ordenando-os por ordem crescente, caso das pesquisas mais lentas, ou ordem decrescente no caso das pesquisas mais rápidas.

Os filtros de rates - maior *hit rate*, maior *bounce rate* e maior *refinement rate* - lista as 10 pesquisas com os valores da métrica seleccionada mais altos. Já foi explicado anteriormente como eram calculados para cada sessão. Aqui o processo é semelhante, tendo o cuidado de relacionar as tabelas Analytics Abstract Search e Analytics User Search pela chave primária (asearch_id) da primeira tabela, de modo a calcular os valores não por sessão, mas por pesquisa geral.

O filtro de mais resultados devolvidos lista as 10 pesquisas que obtiveram mais resulta-

dos na resposta. Os dados são obtidos relacionando as tabelas Analytics Abstract Search e Analytics User Search através da chave primária (asearch_id) da primeira tabela e relacionando a segunda tabela através da sua chave primária (usearch_id) com a tabela Analytics Search Answer, obtendo-se assim a soma de recursos devolvidos (sanswer_rsc_counter) para cada pesquisa geral.

Top de Termos

Apresenta uma lista com o *top* 10 dos termos mais usados em pesquisas no sistema (Figura 5.4 - quadro 8). Pode-se filtrar a informação através de 4 filtros: mais frequentes, mais rápidas, mais lentas e mais resultados devolvidos.

O filtro mais frequentes é a selecção por defeito, apresentando os 10 termos mais frequentes. Essa informação é obtida consultando a tabela Analytics Abstract Search Term, ordenando por ordem decrescente o registos por número de vezes utilizado (asterm_counter) e limitando aos 10 primeiros. Os termos do tipo (asterm_type) *stopword* são ignorados.

O filtro mais rápidas lista os 10 termos de pesquisa mais rápidas enquanto o filtro mais lentas é o inverso, ou seja, lista os 10 termos de pesquisa mais lentas. Para obter os dados é necessário primeiro encontrar todas as pesquisas em que o termo é usado, relacionando as tabelas Analytics Abractract Search Term e Analytics Search Term através da chave primária (asterm_id) da primeira tabela, obtêm-se os identificadores correspondentes aos registos da tabela Analytics Abstract Search. Com esses dados e relacionando as tabelas Analytics Abstract Search e Analytics User Search, obtem-se todas as pesquisas e respectivos tempos de resposta em que o termo aparece. Fazendo uma média dos valores dos tempos de resposta por cada termo e ordenado por ordem crescente, caso filtro mais lento seleccionado, ou por ordem decrescente caso filtro mais rápido seleccionado, obtêm-se a lista de termos desejados.

O filtro de mais resultados devolvidos lista os 10 termos de pesquisa que obtiveram mais resultados na resposta. Para obter os dados é necessário primeiro encontrar todas as pesquisas em que o termo é usado, relacionando as tabelas Analytics Abractract Search Term e Analytics Search Term através da chave primária (asterm_id) da primeira tabela, obtêm-se os identificadores correspondentes aos registos da tabela Analytics Abstract Search. Com esses dados e relacionando as tabelas Analytics Abstract Search e Analytics User Search, obtêm-se todas as pesquisas em que o termo aparece. Fazendo um relacionamento entre as tabelas Analytics User Search e Analytics Search Answer através da chave primária (usearch_id) da primeira tabela, obtêm-se as resposta dadas às pesquisas em que o termo surge. Somando o contador de número de resultados devolvidos (sanswer_rsc_counter) obtêm-se o valor desejado, que ordenado por ordem decrescente devolve a lista esperada.

Tendências

Apresenta o top 10 das tendências dos utilizadores no último mês (Figura 5.4 - quadro 4). Pode-se filtrar a informação através de 4 filtros: *trends*, *novelties*, *long runners* e *top movers*.

O filtro *trends* é a selecção por defeito, apresentando as tendências no último mês como referido. Essa informação é obtida relacionando as tabelas Analytics Abstract Search Term e Analytics Search Term através da chave primária (aster_id) da primeira tabela, sendo a consulta restringida pelo periodo temporal de 1 mês, os termos não podem ser do tipo *stopwords* e são só seleccionados os 10 termos mais usados. Para a interface gráfica é devolvido o texto do termo e o respectivo identificador.

O filtro de *novelties* mostra as novas tendências ocorridas exclusivamente nos últimos

3 dias do intervalo de tempo (1 mês). Os dados são obtidos através da consulta da tabela Analytics Tops restringida pelo tipo de registo, neste caso "NOVELTY". Para a interface gráfica é devolvido o texto do termo e o respectivo identificador.

O filtro de *long runners* mostra as tendências mais constantes durante o intervalo de tempo (1 mês). Os dados são obtidos através da consulta da tabela Analytics Tops restringida pelo tipo de registo, neste caso "LONG RUN". Para a interface gráfica é devolvido o texto do termo e o respectivo identificador.

Por fim, o filtro de *top movers* mostra as tendências que mais variam ao longo do intervalo de tempo (1 mês). Os dados são obtidos através da consulta da tabela Analytics Tops restringida pelo tipo de registo, neste caso "TOP MOVER". Para a interface gráfica é devolvido o texto do termo e o respectivo identificador.

Ao seleccionar uma tendência da lista, é enviado para o servidor o identificador do termo sendo que na resposta é devolvido o histórico de uso desse termo no sistema desde o seu primeiro uso até à actualidade. Essa informação é acedida através de uma consulta à tabela Analytics Search Term restringida apenas pelo identificador de termo. Com essa informação é criado um gráfico que permite visualizar a evolução.

Existe ainda a opção de pesquisa, que permite encontrar uma tendência caso esta não apareça em nenhum dos*tops*.

Evolução de Sessão

Neste quadro (Figura 5.4 - quadro 6) pode-se ver a evolução de métricas de sessão ao longo do tempo. Pode-se filtrar a informação através de 4 filtros: tempo, *hits*, *bounces* e *refinements*. Ao seleccionar um filtro, é enviado para o servidor o tipo de filtro seleccionado, sendo devolvido na resposta os dados pretendidos sendo criado um gráfico que permite visualizar a evolução temporal da respectiva métrica.

O filtro tempo, seleccionado por defeito, mostra a evolução do tempo de duração de sessão ao longo do tempo. Os dados podem ser acedidos com uma consulta na tabela Analytics Session Evolution restringida ao tipo do filtro, neste caso "TIME".

O filtro *hits*, mostra a evolução de *hits* ao longo do tempo. Os dados podem ser acedidos com uma consulta na tabela Analytics Session Evolution restringida ao tipo do filtro, neste caso "HIT".

O filtro *bounces*, mostra a evolução de *bounces* ao longo do tempo. Os dados podem ser acedidos com uma consulta na tabela Analytics Session Evolution restringida ao tipo do filtro, neste caso "BOUNCE".

O filtro *refinements*, mostra a evolução de *refinements* ao longo do tempo. Os dados podem ser acedidos com uma consulta na tabela Analytics Session Evolution restringida ao tipo do filtro, neste caso "REFINEMENT".

Capítulo 6

Testes e Experimentação

Neste capítulo são apresentados os testes efectuados sobre a nova versão com as novas funcionalidades implementadas. São apresentados testes funcionais e testes de desempenho.

6.1 Testes Funcionais

Os testes funcionais permitem aferir se o sistema, processa e apresenta correctamente a informação recolhida. Esta secção apresenta os testes à recolha e processamento de informação e testes à visualização de informação.

6.1.1 Recolha e Processamento de Informação

Com estes testes pretende-se verificar se os dados recolhidos estão a ser processados e armazenados correctamente na base de dados.

Recolha de Dados de Pesquisa

O *input* de testes usado é o seguinte:

- "até ao fim"
- "leis de direito"
- "em nome da terra"

Vai ser executado duas vezes, criando duas sessões na base de dados. Após a primeira sessão os dados armazenados irão ser verificados e terão que coincidir com os resultados apresentados a seguir.

- **Analytics Abstract Search**
  Três novas entradas correspondentes a cada uma das pesquisas do *input*. Os contadores de cada pesquisa terão que estar a 1.
- **Analytics Abstract Search Term**
  Dez novas entradas correspondentes aos termos de cada uma das pesquisas. Os contadores de cada termo terão que estar a 1.
- **Analytics Search Term**
  Dez novas entradas com a respectiva data da realização do teste terão que estar registadas na tabela ligando uma pesquisa abstracta com os respectivos termos.

- **Analytics User Search**
  Três novas entradas correspondentes a cada uma das pesquisas do *input*.
- **Analytics Search Answer**
  Nove novas entradas com a seguinte distribuição:
  - "até ao fim" - 3 entradas.
  - "leis de direito" - 3 entradas.
  - "em nome da terra" - 3 entradas.

De seguida é executado uma vez mais o *input* de testes, criando uma nova sessão. Após esta sessão os dados irão ser verificados uma vez mais e terão que coincidir com os seguintes resultados.

- **Analytics Abstract Search**
  Não há novas entradas. Os contadores de cada pesquisa da primeira sessão terão que estar a 2.
- **Analytics Abstract Search Term**
  Não há novas entradas. Os contadores de cada termo anteriores terão que estar a 2.
- **Analytics Search Term**
  Dez novas entradas com a respectiva data da realização do teste terão que estar registadas na tabela ligando uma pesquisa abstracta com os respectivos termos.
- **Analytics User Search**
  Três novas entradas correspondentes a cada uma das pesquisas do *input*.
- **Analytics Search Answer**
  Nove novas entradas com a seguinte distribuição:
  - "até ao fim" - 3 entradas.
  - "leis de direito" - 3 entradas.
  - "em nome da terra" - 3 entradas.

No final dos testes os dados armazenados foram comparados com os dados esperados verificando-se que os testes foram executados com sucesso. Toda a informação se encontrava armazenada correctamente. Pode-se assim concluir que a recolha de informação está a funcionar como previsto.

Eventos

Entende-se por eventos acções de *hit*, *bounce* e *refinements* ocorridos nos sistema.

O input de testes usado é o seguinte:

- "leis de direito"
- "em nome da terra"
- "em nome da terra santa"

Para testar a acção *hit*, será feita uma pesquisa do *input*, "leis de direito". A resposta à pesquisa terá os três tipos de recurso do sistema: FAQs, repositório de documentos e repositório semântico. Será clicado um recurso em cada tipo de recurso. Serão verificados os dados na base de dados e terão que coincidir com os seguintes.

- **Analytics Search Event Click**
  Três novas entradas correspondente a cada um dos recursos clicados e associados à pesquisa feita.

Para facilitar os testes, os *bounces* e *refinements* serão testados no mesmo teste. Será feita uma primeira pesquisa, "em nome da terra", onde nenhum resultado será clicado. De seguida é introduzida a pesquisa seguinte "em nome da terra santa". Serão verificados os dados na base de dados e terão que coincidir com os seguintes.

- **Analytics User Search**
  As duas entradas correspondentes às pesquisas do teste, a primeira têm que estar marcada como *bounce* - dado que nenhum resultado foi visto. A segunda pesquisa tem que estar marcada como *refinement*.

No final dos testes os dados recolhidos foram comparados com os dados esperados verificando-se que os testes foram executados com sucesso. Toda a informação se encontrava armazenada correctamente. Conclui-se assim que os eventos estão a ser processados correctamente.

Tendências

Para se poder testar o processamento das tendências, principalmente para os *novelties*, *top movers* e *long runners*, é necessário haver pelo menos um intervalo temporal de 30 dias, como já foi visto no Capítulo 5. Para tal, vão ser introduzidos na tabela Analytics Abstract Search Term alguns termos. Na tabela Analytics Search Term, serão introduzidos registos de forma a combinar as datas e os termos até existir um intervalo de 30 dias. Note-se que os termos e a distribuição pelos dias serão conhecidos e podem-se ver ambos na tabela 6.1.

Por dia será calculado o *top* 10 diário de tendências e armazenado na tabela Analytics Daily Top. Os resultados esperados podem ser vistos na tabela 6.2 do Anexo **??**. A partir desses *tops* diários serão calculados os *novelties*, *top movers* e *long runners* que são armazenados na tabela Analytics Tops. Os resultados esperados podem ser consultados na tabela 6.3.

Após a execução do teste comparou-se os resultados obtidos na base de dados com os resultados esperados, constatando que estes coincidiam, podendo-se concluir assim que os cálculos de novelties, long runners e top movers funcionam como esperado.

Termos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
clientes	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
almedina	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
livros	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
mensagem	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	1
garrafa	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
leis	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
direito	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
comboio	2	2	1	2	2	1	2	2	1	2	2	1	2	2	1	2	2	1	2	2	1	2	2	1	2	2	1	1	1	1
seguro	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
mundo	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
contos	1	1	6	1	1	6	1	1	6	1	1	6	1	1	6	1	1	6	1	1	6	1	1	6	1	1	6	1	1	6
estrela																											5	5		5
empresa																														5
eclipse																													5	
contabilidade																													5	

Tabela 6.1: Distruibuição dos termos por dia

Termos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
clientes	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
almedina	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
livros	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
mensagem	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	
garrafa	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
leis	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
direito	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
comboio	2	2		2	2		2	2		2	2		2	2		2	2		2	2		2	2		2	2				
seguro	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
mundo	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2		
contos			6			6			6			6			6			6			6			6						6
estrela																											5	5		5
empresa																														5
eclipse																													5	
contabilidade																													5	

Tabela 6.2: Top diário dos termos

Long Runners
clientes
almedina
livros
mensagem
garrafa
leis
direito
seguro
mundo

Top Movers
comboio
contos

Novelties
empresa
eclipse
contabilidade

Tabela 6.3: Resultados esperados para as tendências

6.2 Testes de Desempenho

Os testes de desempenho permitem aferir quais os tempos médios que o sistema demora a gerar uma resposta. Com estes testes pretende-se verificar se as funcionalidades implementadas prejudicam os tempos de resposta do sistema. Para efectuar os testes foi usada uma máquina com as seguintes características:

- Máquina utilizada:
  - Intel Core i5 2.3GHz (2 Cores, 4 Threads, 4 MB Cache)
  - 4GB RAM

6.2.1 Recolha de Informação

O sistema de recolha de informação implementado neste trabalho é à partida o ponto que pode fazer com que o tempo de resposta aumente. O *input* de teste teve origem na base de dados de conhecimento usada para alimentar o motor de pesquisa da Wizdee no decorrer deste projecto. Trata-se de *strings* com nomes de livros, contabilizando um total de 440 pesquisas distintas, que serão feitas de modo sequencial no motor. Este conjunto de pesquisas será executado 30 vezes, sendo obtido assim o tempo médio de resposta (TM) e o respectivo desvio padrão (DP).

Na Tabela 6.4 são apresentados os resultados dos testes de desempenho efectuados sobre três cenários diferentes de quantidade de informação na base de conhecimento. Desta forma testa-se o sistema implementado e verifica-se também se o tamanho da base de conhecimento influência o seu desempenho.

Os cenários são os seguintes:

- Cenário A - os recursos são usados na sua totalidade, perfazendo um total de 1921 recursos estando distribuídos pelos tipos de recurso:
  - FAQs: 25 questões;
  - Repositório Semântico: 461 instâncias;
  - Repositório de Documentos: 1335 documentos.
- Cenário B - os recursos foram reduzidos aproximadamente para metade em cada tipo de recurso:

  - FAQs: 12 questões;
  - Repositório Semântico: 230 instâncias;
  - Repositório de Documentos: 667 documentos.

- Cenário C - não existem recursos.

Cada um destes cenários é testado com o sistema de recolha de informação inactivo, activo e activo com remoção dos registos das tabelas que armazenam os dados recolhidos por cada vez que é executado o *input* de teste.

Versão	Cenário A		Cenário B		Cenário C	
	TM (ms)	DP (ms)	TM (ms)	DP (ms)	TM (ms)	DP (ms)
Sem Recolha de Informação	83	1	77	0,6	74	0,5
Com Recolha de Informação	89	0,8	82	0,5	79	0,9
Com Recolha de Informação - 1ª inserção	96	1,6	90	1,4	89	1

Tabela 6.4: Tempos médios de resposta

É possível verificar pelos resultados obtidos que o desempenho do sistema piorou em todos os cenários com o sistema de recolha activo. No entanto são diferenças mínimas - 5 a 6 milissegundos. Pode-se verificar que na 1ª inserção dos dados, a diferença é maior - 13 a 15 milissegundos - devendo-se ao facto do número de inserções ser maior do que nos testes posteriores.

Relativamente ao tamanho da base de conhecimento, verifica-se que este não influência o sistema de recolha de informação, mantendo aproximadamente a diferença de 5 a 6 milissegundos entre o cenário com o sistema inactivo e o cenário com o sistema activo. O gráfico da Figura 6.1 mostra a curva do tempo de pesquisa nos diferentes cenários, tornando mais perceptíveis os resultados obtidos.

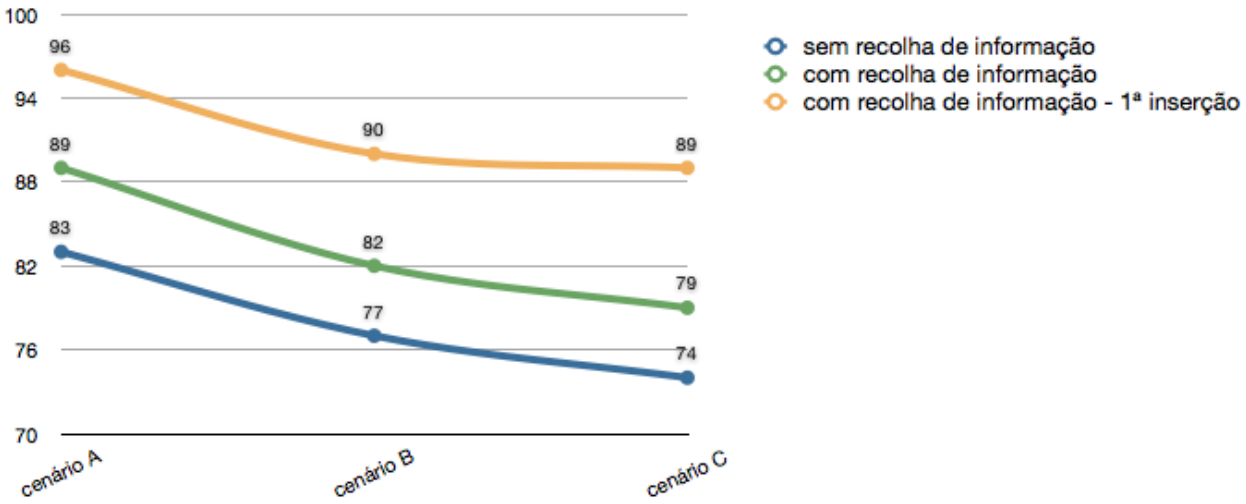


Figura 6.1: Gráfico de curva de pesquisa nos diferentes cenários.

Foi também testado o limite de registos nas tabelas com maior tendência a crescer, Analytics User Search e Analytics Search Term, prejudicial aos tempos de novas inserções. Após cerca de 100 mil registos inseridos na primeira tabela e aproximadamente 320 mil da segunda tabela, verificou-se que os tempos de resposta se mantinham inalterados. Apesar de saber-se que esta limitação existe devido a experiências passadas com uma tabela de *logs* da plataforma, não foi possível realizar o teste com sucesso em tempo útil.

Capítulo 7

Conclusões

Este estágio veio complementar um sistema já existente mas com algumas limitações existentes a nível de extracção de informação proveniente da interacção do utilizador com o motor de pesquisa da Wizdee e de análise dessa mesma informação. Tornando possível processar informação que de outra maneira seria de difícil compreensão e visualização e criação de relatórios, gráficos, entre outros, que mostrem essa informação de forma útil e compreensível.

Como foi possível ver no *Estado da Arte*, QP é um conceito bastante relativo dado que depende da relevância que um resultado tem para a necessidade de informação. Mas no entanto é possível afinar um motor de pesquisa de modo a que essas necessidades sejam preenchidas. Por sua vez AP incide sobre dados recolhidos através da interacção do utilizador para com a informação apresentada, variações de comportamentos, tendências, entre outros, ajudando a compreender necessidades que permite melhorar a disponibilidades de informação, efectuar melhorias quer a nível de desempenho do motor de pesquisa quer a nível de visualização de informação.

A *Análise de Concorrência* veio ajudar a conhecer melhor as soluções concorrentes que importantes companhias estão a desenvolver. Informações essas que nem sempre estavam explicitas o que dificultou um pouco o trabalho. No entanto foi uma mais valia para obter ideias para o trabalho a desenvolver.

Outros aspectos importantes a salientar foram a definição de requisitos e a análise de arquitectura da plataforma, identificando os componentes relevantes e a desenvolver, que irão interagir entre si.

Apesar do *Planeamento* adoptado pela Wizdee, um processo *Scrum* normalmente não obedece à criação de um diagrama de *Gantt*. No entanto a sua criação ajudou a planear o trabalho desenvolvido ao longo do ano.

As contribuições deste trabalho são as seguintes:

- Estado da arte e análise de competidores em QP e AP;
- Modelo computacional do processo de recolha e processamento da informação das pesquisas, bem como o modelo de dados que o suporta;
- Desenvolvimento e implementação do modelo computacional;
- Desenvolvimento de uma interface gráfica destinada à visualização de informação;
- Testes funcionais e experimentação do sistema desenvolvido.

O processo de recolha e processamento de informação veio enriquecer a plataforma da Wizdee com um valioso conhecimento sobre o modo como o utilizador usa a pesquisa, das suas necessidades de informação e do próprio desempenho do motor de pesquisa. Esta

representa um ponto importante para que se possam efectuar melhorias a nível interno do motor de modo a torná-lo mais eficaz, como também ajudar a colmatar ausências de informação relativamente a determinados tópicos. Este sistema disponibiliza uma API interna que pode ser usada para estender as suas funcionalidades a outros produtos que façam uso da plataforma. O facto de ser usado um *webservice* para comunicação entre a plataforma e a interface gráfica, abre portas a que a informação possa ser apresentada em vários ambientes diferentes que não sejam só o da plataforma tecnológica da Wizdee.

Pelos testes de desempenho observou-se que apesar de haver um processo complexo de recolha de informação entre a pesquisa e a obtenção de resposta, os tempos de resposta apenas sofreram uma alteração na ordem dos 5 milissegundos nas condições óptimas e 13 milissegundos no pior cenário. Com a variação do tamanho da base de conhecimento do motor de pesquisa, constatou-se que a apesar de haver variação do tempo total de resposta, o tempo de execução do sistema de recolha de informação era aproximadamente o mesmo para os diferentes cenários.

O trabalho apresenta algumas limitações, tais como:

- Não há informação sobre os recursos que constituem uma resposta no seu total. Esta funcionalidade permite verificar que recursos estão a ser usados para responder a determinadas pesquisas, podendo permitir afinações no sentido de melhorar os recursos usados para determinados temas de pesquisas.

- A forma como as tendências *novelties*, *long runners* e *top movers* são calculadas, não permite variação temporal, estando restrito a 1 mês.

Relativamente a trabalho futuro, identificam-se as seguintes tarefas:

- Implementar um sistema de alerta para variações abrutas de métricas que possa ser configurável relativamente às métricas a monitorizar e valores para os quais devem ser emitidos os alertas;

- Ter mais informação pré-calculada, como por exemplo *hit rate*, *bounce rate* e *refinement rate* de forma a minimizar o tempo de resposta no pedido de informação;

Referências

Baeza-Yates, R., Ribeiro, B., and Ribeiro-Neto, B. (1999). *Modern information retrieval*. ACM Press books. ACM Press.

Büttcher, S., Clarke, C. L. A., and Cormack, G. V. (2010). *Information retrieval : implementing and evaluating search engines*. MIT Press, Cambridge, Mass.

Chaters, B. (2011). *Mastering Search Analytics - Measuring SEO, SEM, and Site Search.* O'Reilly.

Enge, E., Spencer, S., Fishkin, R., and Stricchiola, J. C. (2010). *The Art of SEO - Mastering Search Engine Optimization.* O'Reilly.

Gloria J. Miller, Dagmar Bräutigam, S. V. G. (2006). *Business Intelligence Competency Centers: A Team Approach to Maximizing Competitive Advantage*. John Wiley & Sons, Inc.

Guha, R., McCool, R., and Miller, E. (2003). Semantic search. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 700–709, New York, NY, USA. ACM Press.

Inan, H. (2006). *Search Analytics: A Guide to Analyzing and Optimizing Website Search Engines*. BookSurge, LLC.

Karen Spärck Jones, P. W. (1997). *Readings in information retrieval*. Morgan Kaufmann Publishers Inc.

Leouski, A. and Croft, W. (1996). An evaluation of techniques for clustering search results.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Moran, M. and Hunt, B. (2006). *Search Engine Marketing Inc.* IBM Press.

Nielsen, J. (1993). *Usability engineering.* Academic Press.

Rising, L. and Janoff, N. (2000). The Scrum Software Development Process for Small Teams. *IEEE Software*, 17:26–32.

Rosenfeld, L., Krug, S., and Kaushik, A. (2011). *Search Analytics for Your Site: Conversations with Your Customers*. Rosenfeld Media, LLC.

van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworths, London, 2 edition.