INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Área Departamental de Engenharia de
Eletrónica e Telecomunicações e de Computadores

Construção de uma Base de Conhecimento para Geração de Conteúdos

Hélio Pedro Miranda
(Licenciado)

TRABALHO DE PROJETO PARA OBTENÇÃO DO GRAU DE MESTRE
EM ENGENHARIA INFORMÁTICA E DE COMPUTADORES

Presidente

Professor Doutor Manuel Martins Barata

Orientador

Professor Doutor Paulo Manuel Trigo Cândido da Silva

Arguente

Professor Doutor Porfírio Pena Filipe

SETEMBRO DE 2013

Agradecimentos

Um agradecimento especial para o Dr. Paulo Trigo, pelo tempo empenhado e pela disponibilidade que apresentou sempre que algum problema era apresentado.

Um outro agradecimento especial para o Eng. Paulo Soares Marques, pela ajuda prestada e disponibilidade.

Também não posso esquecer a minha família que durante largos meses me acompanhou neste desenvolvimento.

Resumo

No decorrer de um trabalho de investigação, ou de desenvolvimento, deparamos com inúmeros artigos e documentação técnica que em geral catalogamos (ou classificamos) atribuindo (eventualmente) um determinado grau de relevância. Essa classificação é muitas vezes armazenada de modo 'ad-hoc' (e.g., num *bookmark* ou como uma anotação num *post-it*) e com o passar do tempo vamos acumulando referências que temos dificuldade em pesquisar e compilar para daí extrair uma síntese que possa servir como esboço para a redação do relatório técnico desse trabalho de investigação (ou de desenvolvimento).

Neste projeto é descrita e implementada uma ferramenta que permite ao utilizador catalogar cada uma das suas referências e registar excertos de texto (dessas referências) associando-lhes significado (semântica) extraído de uma taxonomia (axiomas de inclusão de uma ontologia). Esta associação, entre os excertos de texto e os conceitos da taxonomia, é usada para implementar um sistema de pesquisa integrado (sintaxe e semântica).

A taxonomia suporta-se na noção de ontologia e permite construir referências cruzadas sobre os documentos e os excertos que vão sendo registados.

O suporte de uma taxonomia permite focar o motor de pesquisa fornecendo, aos utilizadores, os resultados que pretendem. A pesquisa pode também ser feita de forma não estruturada a partir de parcelas dos excertos de texto relevante. A pesquisa pode ser estruturada combinando critérios que recorram à taxonomia definida.

Após a obtenção dos resultados, é apresentado um modo onde o utilizador pode visualizar os vários excertos num formato (previamente) configurado através de um modelo que contém vários estilos de modo a que cada excerto associado a determinado conceito (da taxonomia) seja apresentado de acordo com o nível hierárquico a que pertence (nessa taxonomia).

Com a utilização deste sistema é possível de forma simples agrupar e catalogar um conjunto de informação, assim como recuperar informação relevante que tenha sido registada de forma não estruturada.

Índice do Conteúdo

Índice de Imagens

1 – Introdução

Ao desenvolver um trabalho de projeto, ou algum tipo de investigação, vamos acumulando diversos artigos e informação relevante para esse trabalho. Com o passar do tempo todos esses artigos se vão acumulando perdendo muitas vezes o contexto necessário. Ou seja, com o passar do tempo vamos apenas ter uma enorme quantidade de informação espalhada no computador ou simplesmente em *bookmarks*.

Ao aceder à informação previamente guardada, vamos notando que não está estruturada, e por isso não temos forma de pesquisar nem de apresentar, essa informação, de forma simples.

Este é o contexto que motiva o desenvolvimento deste projeto onde se descreve e implementa uma ferramenta que permite, a cada utilizador e de forma simples, realizar as seguintes tarefas: a) catalogar (usando uma taxonomia) excertos de informação relevante, b) de seguida proceder a uma pesquisa estruturada (sobre essa informação) e, c) por último recompor todos os excertos relevantes num único documento com um formato que se garante consistente com a taxonomia (usada durante a catalogação).

1.1 - Objetivo

Estão disponíveis, a nível de mercado, algumas soluções capazes de responder apenas parcialmente às ideias previamente apresentadas, mas nenhuma das soluções consegue abranger a sua totalidade.

Por essa razão foi criado um sistema que pudesse integrar um modo simples e intuitivo de catalogação para todos os utilizadores, uma interface simples de pesquisa e um modo de visualização da informação relevante.

Como principais objetivos deste sistema temos: a) construção de uma árvore de conceitos, que permita aos utilizadores caracterizar a sua informação de forma mais precisa possível, permitindo ao mesmo tempo que esta seja facilmente editável pelos administradores do sistema, b) possibilidade de definir a origem da informação, permitindo assim que a origem da informação seja sempre preservada,

e c) indexação da informação armazenada, possibilitando assim uma eficaz recuperação e d)criação de uma apresentação pré formatada e com estilos configuráveis.

Com base nestes objetivos identificam-se três fases na utilização no sistema.

A fase inicial onde o utilizador poderá inserir um texto ou documento relevante, tendo apenas de preencher alguns meta-dados, associando ao seu texto uma ou mais etiquetas. São estas etiquetas que atribuem uma noção de semântica aos textos, conseguido assim garantir que todos os textos têm significado para manipulação automática.

Após catalogar o texto, o utilizador necessita de um sistema que lhe permita encontrar a informação desejada de forma orientada.

Com recurso a uma taxonomia e uma simples zona pesquisável o utilizador poderá pesquisar por parcelas de textos ou ainda por textos dentro de um documento previamente indexado. Com esta possibilidade um utilizador poderá facilmente encontrar os melhores resultados para a sua questão, podendo ainda acrescentar filtros sobre as etiquetas a pesquisar.

Finalmente, e utilizando os resultados obtidos anteriormente, os utilizadores podem visualizar os mesmos escolhendo um modelo de apresentação previamente configurado. Estes modelos são configurados por administradores do sistema, oferecendo assim uma forma formatada de apresentar resultados.

1.2 - Organização do Documento

Nos próximos capítulos iremos abordar cada uma das diferentes fases da aplicação assim como relacionar o ambiente de suporte (*SharePoint 2010* [9]) com outros sistemas.

Capítulo 2 - serão apresentados alguns sistemas existentes e onde estes se podem enquadrar com a nossa aplicação. Da mesma forma iremos comparar os diferentes sistemas analisados com o sistema escolhido para implementação da nossa aplicação.

Capítulo 3 - é explicado a primeira fase da aplicação, ou seja, a fase onde o utilizador irá introduzir os seus dados escolher etiquetas e ainda colocar anexos nos seus dados.

Capítulo 4 - apresenta o modelo de pesquisa que foi parametrizado assim como este funciona e todos os passos que implicam a catalogação correta (extração de palavras, redução à forma canónica, entre outros passos).

Capítulo 5 - é explicado com a aplicação interage com os resultados das pesquisa possibilitando ao utilizador obter um modo pré definido para apresentar os seus resultados.

Capítulo 6 - são apresentadas algumas conclusões relativas ao trabalho desenvolvido assim como apresentados alguns objetivos futuros.

Capítulo 7 - são apresentados alguns scripts de instalação e configuração da aplicação.

2 - Trabalho Relacionado

Ao analisar os requisitos funcionais da aplicação desenvolvida foram também explorados diversos sistemas com capacidades de oferecer algumas respostas ao problema, mas nenhum conseguia responder inteiramente à necessidade.

O sistema necessitava de responder aos seguintes requisitos:

a) Gestão de permissões, ou seja possibilidade de definir quem são os utilizadores e quem são os administradores do sistema, um sistema de recuperação de informação não permite esta gestão;
b) Motor de pesquisa configurável, de forma a permitir escolher qual o algoritmo de Ranking a utilizar, um sistema de Gestão documental não permite este tipo de opções;
c) Fácil desenvolvimento de User Inferface, de todos os sistemas analisados apenas o SharePoint oferecia uma forma simples de criação de sites e páginas.

De forma a responder aos requisitos previamente apresentados, foi desenvolvida uma aplicação, que se divide em três fases: catalogação, pesquisa e visualização.

Para cada uma das fases foram analisados diferentes sistemas, nomeadamente sistemas de gestão documental, sistemas de recuperação de informação e sistemas de gestão de conteúdos.

2.1 - Gestão Documental

A gestão documental seria a primeira hipótese a ponderar nesta aplicação, pois iremos depender em grande parte em armazenamento de ficheiros e respetivos meta-dados, mas para o nosso caso a gestão documental oferece funcionalidade que não iremos necessitar, por exemplo os mecanismos de *check-in* e de publicação.

Um sistema de gestão documental assume a existência de um documento e foca-se em suportar a gestão (acompanhamento) de todo o seu ciclo de vida. No nosso caso o foco não está na gestão do (ciclo de vida do) documento mas na

geração de um documento a partir da catalogação de diversos excertos de informação considerada relevante.

Por outro lado um sistema de gestão documental permite o arquivo organizado da informação, e isso seria uma possível vantagem para esta aplicação.

No entanto, um puro sistema de gestão documental não oferece suporte para pesquisa de informação não-estruturada, nem permite a criação de modos de composição de um novo documento a partir de excertos de outros.

Resumindo, um sistema de Gestão documental oferece algumas funcionalidades interessantes, mas que não são essências para a nossa aplicação, tais como: a) *check-in / check-out*, b) controlo de versões, c) *audit trail*, d) anotações.

Um conhecido sistema de gestão documental é o *Alfresco[8]*. Este sistema tem diversas características entre as quais se salientam:

- Controlo de Versões, o Alfresco armazena todas as versões que um documento pode ter, sendo estas versões finais ou intermédias. Uma versão intermédia consiste num documento que ainda está em fase de desenvolvimento, ou seja, uma versão ainda não finalizada e que poderá ser editada a qualquer momento.

  O Alfesco permite que assim que todas as versões do documento sejam passiveis de reversão, possibilitando assim que caso um erro seja detectado numa versão seja possível saber quem originou o erro e reverter para uma versão anterior,

- Pré-visualização Online, com esta opção os utilizadores podem visualizar documentos diretamente pelo browser, sem a necessidade de os abrir, isto é possível utilizando ferramentas online, como o caso do Google Docs,

- Workflows, o Alfresco possibilita a construção de workflows sem a necessidade da existência de um programador. Os workflows podem ser construídos de forma fácil e intuitiva utilizado para isso apenas o browser e algumas ações pré definidas,

- Integração com aplicações MS Office, com esta integração é possível o utilizador abrir um documento numa qualquer aplicação MS Office, editar o seu conteúdo e simplesmente guardar o documento sendo que este é automaticamente guardado no Alfesco. O Alfresco possibilita também uma integração com a ferramenta Google Docs,
- Meta-dados, tal como todos os sistemas de gestão documental o Alfresco possibilita a criação de meta-dados de forma a enriquecer a informação de cada documento.

2.2 - Recuperação de Informação

Um sistema de recuperação de informação suporta o essencial da segunda fase do projeto (pesquisa), ou seja, a existência de um motor que permite a recuperação automática de informação não-estruturada (i.e., reposta a interrogações sobre texto livre).

No nosso caso iremos necessitar de um motor que permita ao utilizador colocar texto livre, e apresentar-lhe um conjunto de resultados válidos e por isso foram analisados sistemas que oferecem estas capacidades, tal como o *Lucene[7]*.

Um sistema de recuperação de informação oferece as seguintes funcionalidades:

- Preparação dos documentos
- Indexação
- Armazenamento
- Recuperação

Apesar de oferecer um excelente motor de pesquisa um sistema de recuperação de informação também não consegue responder à totalidade dos requisitos, pois não oferece um repositório organizado para o utilizador, não oferece uma interface gráfica agradável e apesar de ser altamente configurável é algo complexo de parametrizar. Para além disso, em geral, não suporta catalogação a partir de uma taxonomia nem permite compor um documento a partir de excertos.

O *Lucene* é um sistema de recuperação de informação, este sistema puramente desenvolvido em Java tem como principais atributos o facto de oferecer diversos modelos de ordenação, *Ranking*, todos eles configuráveis.

Outros atributos principais do *Lucene* consistem no seu elevado desempenho no processo indexação assim como os baixos requisitos de processamento que possui.

O *Lucene* é apropriado para aplicações que necessitem de indexação e pesquisa customizada.

Pelo facto de ser uma aplicação open-source, a sua utilização é gratuita tanto para aplicações de teor pessoal como comercial.

2.3 - Gestão de Conteúdos

Foram também analisados os sistemas de gestão de conteúdos, nomeadamente os CMS[1], pois um dos objetivos da aplicação é que seja facilmente gerida e mantida, por essa razão os CMS também foram incluídos nesta análise.

Um sistema de CMS oferece habitualmente uma versão *web* simplificada de controlo de versões, indexação, pesquisa e recuperação de informação, sendo ainda utilizados para armazenamento de documentos. Existem diversos sistemas de CMS, mas iremos aprofundar o SharePoint 2010.

O SharePoint não é um sistema puro de CMS, mas oferece grande parte das componentes que procuramos para solucionar os nossos problemas.

O SharePoint incorpora um motor de indexação e pesquisa que vem de acordo com uma das fases do projeto (pesquisa). O motor de pesquisa é extensível e configurável, permitindo ajustar diversos parâmetros para refinar a pesquisa.

Para a fase de catalogação, o SharePoint permite a criação de controlos configuráveis (*webparts*) que permitem ajudar o utilizador a catalogar a sua informação de uma forma simples e eficaz, sendo que com a utilização das listas

[1] CMS – Content Management System

adjacentes ao SharePoint temos toda a informação armazenada com garantia de integridade.

Por fim, para a fase de visualização o SharePoint permite que esta possa ser construída de uma forma customizadas, utilizando para isso as referidas componentes (*webparts*).

O SharePoint oferece um modelo de dados simples de utilizar e que se baseia em listas: Cada lista pode ter associada diversas colunas sendo que cada coluna representa os meta-dados dos objetos da lista.

O SharePoint possibilita também a criação de controlos dinâmicos, *webparts* estes controlos são desenvolvidos em ASP.Net e podem ser desde simples controlos assentes em HTML até controlos dinâmicos com interação com o utilizador e apresentação de resultados diretamente do servidor; e isto responde aos requisitos da aplicação desenvolvida.

O SharePoint disponibiliza também componentes que podem ser instaladas na aplicação, designadas por *features*, e que permitem estender funcionalidade para o utilizador. É recorrendo às *features* que a nossa aplicação é instalada no servidor.

Por fim o SharePoint permite definir e gerir permissões, garantindo que cada utilizador apenas tem acesso ao que necessita.

Muitas das componentes do SharePoint podem ser utilizadas utilizando o *browser* apenas, sendo que para a grande maioria das suas funcionalidades não é necessário um conhecimento técnico para funcionar, apenas componentes como *webparts* necessitam de um conhecimento mais técnico.

Resumindo, um sistema de CMS como o SharePoint, tem características que podem ser exploradas e estendidas de modo a responder à totalidade dos requisitos do projeto, sendo por isso o sistema escolhido para a implementação.

2.4 – Comparação entre Sistemas

De modo a resumir os diferentes sistemas apresentados podemos analisar a tabela seguinte.

	Gestão de Permissões	Motor de Pesquisa configurável	User Interface configurável	Fácil inserção de dados
Gestão Documental	Sim	Não	--	Sim
Recuperação de Informação	Não	Sim	Não	Sim
Gestão de Conteúdos (SharePoint 2010)	Sim	Sim	Sim	Sim

Tabela 1 - Comparação de Sistemas

Para além de apresentar a melhor resposta para os desafios existentes o SharePoint é um sistema com o qual já existe experiência prévia (à deste projeto), originando assim uma vantagem, pois a curva de aprendizagem desta tecnologia é acentuada.

3 - Catalogação

Nesta primeira fase, o utilizador irá introduzir as suas referências/dados na aplicação, associando a estes meta-dados, nomeadamente nos campos (da interface gráfica) designados por *Source*, *Tag* e *Attach* e que são ilustrados na Imagem 1.

A Imagem 1 apresenta a interface gráfica que concretiza a primeira fase da aplicação. Aqui é de salientar: a) árvore de conceitos, b) capacidade de anexar ficheiros.

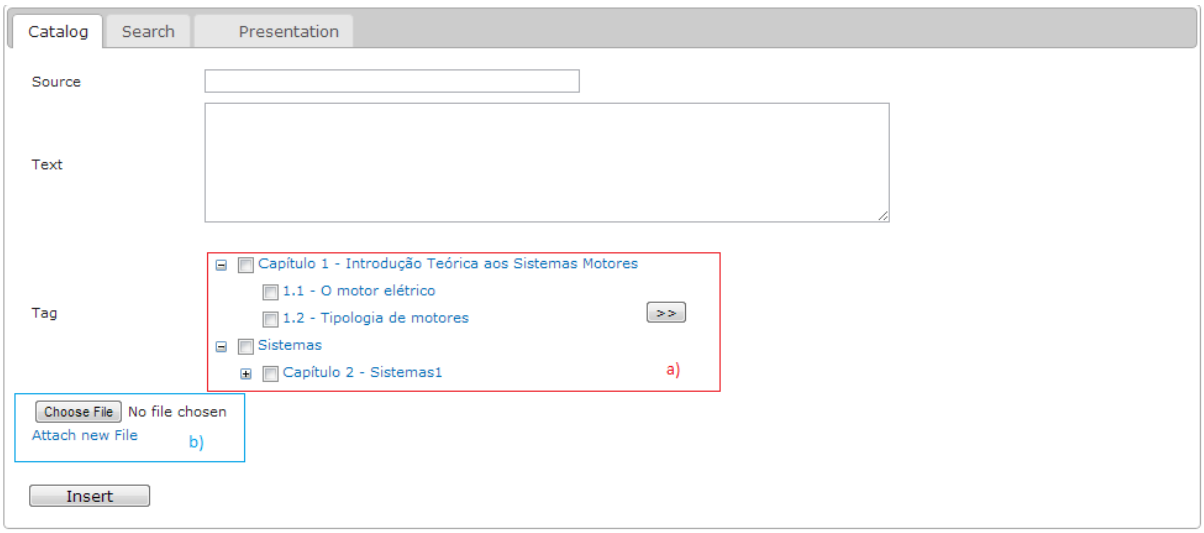

Imagem 1 - Aspecto de Módulo de Catalogação

3.1 - Texto

O principal campo que o utilizador deverá preencher é o campo *Text*, é aqui que o utilizador deverá introduzir todas as suas transcrições ou excertos que deseja manter como registo para futuras análises.

É sobre este campo que será incidido maior importância pois é ele que será apresentado na fase final da aplicação.

Neste campo deverá já ser introduzido o texto final a apresentar, ou seja, o tipo de letra a utilizar, os pedaços de texto a realçar, as numerações necessárias e ainda os *bullets* necessários. Na Imagem 2 - Exemplo de Texto, podemos observar

um exemplo de um excerto de texto, já com alguma indentação, e o tipo de letra previamente escolhido.

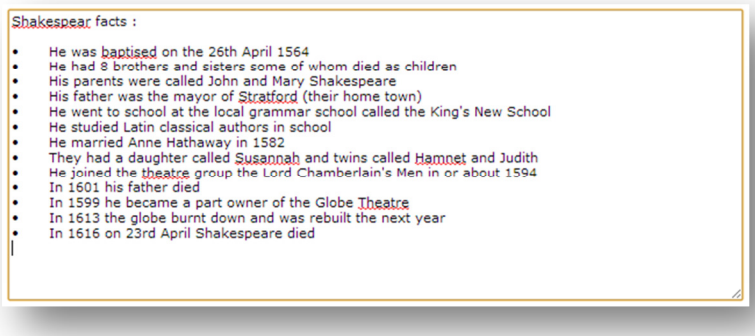

Imagem 2 - Exemplo de Texto

3.2 - Meta-dados

O campo *Source* permite ao utilizador identificar a origem da sua referência. Este campo é aberto e não obrigatório permitindo ao utilizador decidir se deseja ou não colocar o local de onde a referência é originária, seja este um website, um livro ou mesmo um artigo *online*.

Por sua vez o campo *Tag* permite aos utilizadores selecionarem uma ou mais etiquetas com o objetivo de associar os textos introduzidos a uma noção de semântica, garantindo assim que todos os textos têm contexto. São estas etiquetas que irão permitir a organização e pesquisa dos dados introduzidos. As etiquetas são criadas e configuráveis pelos administradores do sistema, estes têm de criar as etiquetas antes de os utilizadores começarem a introduzir as suas referências. Esta atribuição de semântica (associação de etiquetas) deverá ser feita de modo consistente, pois dela depende a correta catalogação do sistema.

Por fim podemos associar a cada *Text* um ou mais anexos, oferecendo assim a possibilidade de enriquecer cada um dos textos inseridos com imagens ou ainda artigos complementares.

Na Imagem 3 - Modelo Entidade Associação de Catalogação, podemos ver que a tabela de *Catalog* tem associado a si as propriedades *Text* e *Source*, da mesma forma tem uma ligação para a tabela de *Attach*, para permitir que cada

informação possa ter associada a si um ou mais anexos, de igual forma possui múltiplas ligações para a tabela de *Tags*, permitindo assim que cada informação seja catalogada da forma que o utilizador ache mais apropriada.

Pelo facto de ser utilizado o SharePoint, o modelo EA apresenta como chave primária sempre um campo *identity* denominado de ID. Este campo está presente em todas as tabelas utilizadas no SharePoint e oferece assim um identificador único para cada elemento da mesma.

No caso das tabelas *Catalog* e *Attach* apenas a coluna ID é chave, pois todas as outras colunas não são passiveis de ser chaves candidatas.

No caso da tabela *Tag* a coluna *Title* pode ser considerado uma chave candidata, pois foi definido que esta coluna não podia ter valores repetidos.

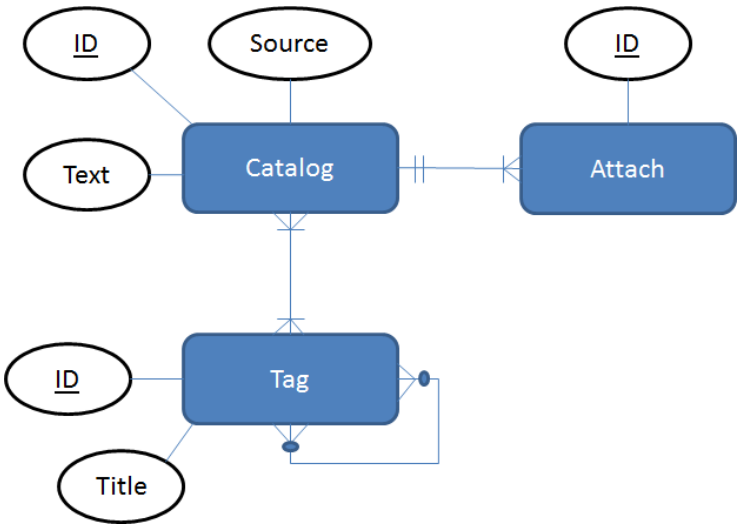

Imagem 3 - Modelo Entidade Associação de Catalogação

3.3 - Guardar Catalogação

De forma a armazenar toda a informação catalogada foram utilizadas duas listas [2] de SharePoint, a lista *Catalog* e a lista *Tags*.

A lista *Catalog* irá armazenar todas referências introduzidas e os respetivos meta-dados. É sobre esta lista que a pesquisa irá incidir, e é sobre o seu conteúdo que os mecanismos de *stremming* e *word breaking*[3] irão incidir.

[2] Uma lista de SharePoint pode ser vista como uma tabela no modelo EA

A lista *Catalog* está também associada à lista de *Tags*, ou seja, cada vez que uma referência é catalogada com uma determinada *Tag* é adicionada uma referência à lista de *Tags.*

A lista *Tags* irá armazenar o contexto de semântica da aplicação, ou seja, esta lista é constituída por diversos conceitos associados entre si numa relação de Pai-Filho, possibilitando assim a construção de uma árvore semântica, estas ligações podem ser vistas na Imagem 4.

Tal como explicado anteriormente a coluna *Title* é uma chave candidata, pois engloba apenas valores únicos e não possíveis de repetir.

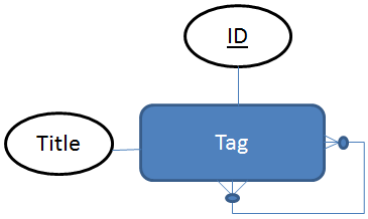

Imagem 4 - Modelo Tags

Este conceito de taxonomia pode-se definir como um conjunto de conceitos que mantêm entre si relações de inclusão, o que iria resultar numa estrutura do tipo árvore em que cada nó representa um conceito relacionado com o seu nó pai. No entanto, é possível associar um nó a diversos conceitos, tornando a nossa estrutura num grafo.

Na Imagem 5 é possível ver que o Janeiro tem duas relações de pai-filho, no caso de a) Janeiro tem como seu nó pai o 1º Trimestre, por sua vez na ligação b) tem como nó pai o 1º Semestre. Por sua vez na ligação c) apenas vemos que Dezembro tem apenas uma ligação para o 2º Semestre. Com o objetivo de facilitar a visualização do grafo este é apresentado ao utilizador na forma de uma árvore.

Para melhor caracterizar a referência introduzida é possível selecionar diversos conceitos, assim como selecionar uma combinação de conceitos. Por exemplo, poderíamos ter uma referência que se englobaria no 1º Trimestre e no 1º Semestre em simultâneo.

[3] Ver Capitulo sobre Pesquisa para mais informação

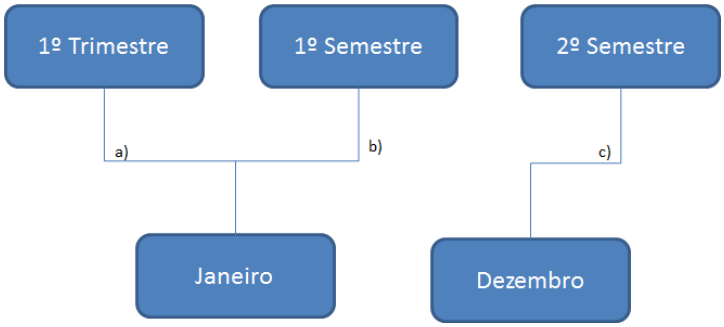

Imagem 5 - Exemplo de Grafo

De forma a manter o sistema atualizado foi também adicionado um evento que é evocado quando um conceito é eliminado. Nestes casos, é removido o conceito do grafo, e é colocada a informação que este conceito foi apagado numa variável interna. Não são removidas as ligações para o conceito, garantindo assim que se alguém apagar um conceito por engano, é armazenada essa informação, possibilitando assim a recuperação da informação, bastando apenas recriar o mesmo e todas as referências continuarão válidas.

Esta opção tem como consequência guardar, em cada referência, conceitos já apagados. Isto permite a criação de um serviço que ocorra num determinado período de tempo para eliminar todas as referências que não estão já em uso.

Foi também criado um evento que aquando a atualização de um conceito, irá percorrer todas as referências e atualizar o valor do conceito, permitindo assim que as referências estejam sempre atualizadas.

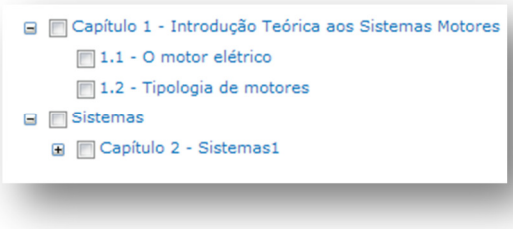

Imagem 6 - Exemplo de Tag

Esta taxonomia necessita de ser definida antes do início da catalogação de dados, pois sem uma taxonomia não será possível inserir dados.

A opção de guardar a taxonomia numa lista de SharePoint foi tomada pelo facto do SharePoint oferecer uma forma simples de controlo de permissões a cada lista e suportar diretamente o preenchimento de dados, assim como das suas associações Pai-Filho.

Para guardar os conceitos e ser possível diferenciar quando uma referência tem associada múltiplas vezes o mesmo conceito mas o conceito tem nós pais diferentes, foi criado um algoritmo que permite diferenciar essas catalogações, ou seja, para além de ser guardado o valor do conceito é também guardado o conceito pai. Desta forma sabemos sempre o caminho completo de todos os conceitos guardados.

Na Imagem 5 caso selecionássemos **Janeiro** iriamos armazenar: **{1º Trimestre, Janeiro} , {1º Semestre, Janeiro}**.

Na Imagem 7 podemos ver os seguintes passos:

1. O utilizador abre a aplicação e introduz os seus dados
2. Após uma primeira validação dos dados, estes são submetidos e inseridos na Lista *Catalog*
3. Caso não exista nenhum erro na inserção é apresentada uma mensagem de sucesso ao utilizador
4. Após a inserção dos dados é iniciado o processo de *crawl* de forma a atualizar o motor de pesquisa

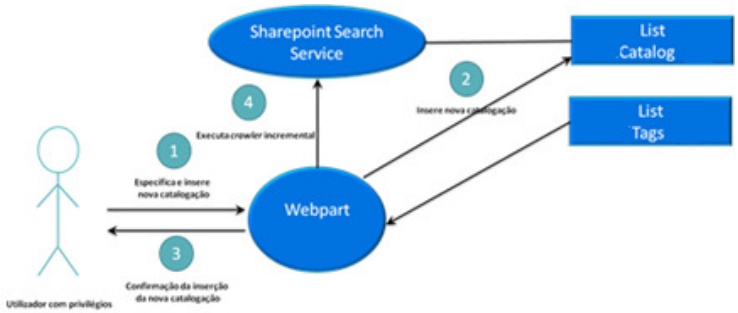

Imagem 7 - Processo de Catalogação

4 - Pesquisa

Após catalogar a sua informação, o principal objetivo do sistema é oferecer uma forma simples e eficaz de pesquisa de informação

Para garantir que todos os conteúdos são passiveis de ser é necessário iniciar um mecanismo de *crawl*, que permitirá a disponibilização da referência.

Foi utilizado o motor de pesquisa e indexação do SharePoint que garante que todo o conteúdo é indexado e os algoritmos de relevância são corretamente aplicados.

Ao utilizar o motor de pesquisa e indexação do SharePoint devemos começar por entender como este funciona.

4.1- Modelo Vetorial

O motor de pesquisa do SharePoint utiliza o algoritmo de pesquisa baseado num algoritmo de classificação de documentos, a esta base foi adicionada a capacidade de aprendizagem ao modelo, ou seja, os resultados são influenciados pelos comportamentos dos utilizadores. A quantidade de ligações para um documento, ou mesmo a quantidade de vezes que um documento é acedido irá influenciar a sua classificação final.

O algoritmo que o SharePoint utiliza (apesar da pouca informação disponível) tem como base todos os conceitos do modelo vetorial. Ou seja, cada referência é particionada em *t* termos distintos, estes termos são agrupados no que se chama um espaço de sectores, são esses vetores que são utilizados aquando de uma interrogação ao sistema.

A cada termo é atribuído um peso calculado com base na frequência em cada documento e na coleção. Cada interrogação é transformada num vetor, e é feita a distância entre esse vetor e cada uma das linhas da matriz de termos existente; são retornados os resultados onde a distância entre vetores constitui a métrica para ordenação (*ranking*).

4.2 - Extração de Palavras

Ao particionar as palavras o *crawler* utiliza um algoritmo de *word breaker* para saber quais as palavras de deverá ignorar, como por exemplo espaços, vírgulas, ou outro tipo de separadores comuns.

O SharePoint disponibiliza um conjunto de palavras pré-definidas a ignorar para cada idioma, ou seja, o que para um idioma pode ser uma palavra a ignorar noutra tem um significado completamente distinto.

4.3 - Redução à forma canónica

Quando é iniciado o processo de *crawl* o SharePoint começa por analisar cada texto contido nos elementos, particionando todas as palavras de forma a mapearem uma certa zona no vetor de palavras (o particionamento das palavras varia de acordo com a linguagem utilizada), após esta fase é feito um *stemming* do conteúdo, ou seja cada palavra é reduzida ao seu elemento principal (remoção de plurais, de género nas palavras, entre outras operações). São estes elementos que após nova verificação contra o índex permitem a criação de um mapa de todas as palavras usadas e sua relevância.

Todo este mecanismo é feito internamente pelo SharePoint o que simplifica substancialmente o trabalho a realizar, tendo apenas a aplicação que introduzir os resultados no motor de pesquisa e acionar o mecanismo de *crawl*.

4.3 - Pesquisa Booleana

Uma outra opção para a pesquisa é a utilização de uma pesquisa booleana invés da pesquisa vetorial.

A pesquisa booleana tem como principal vantagem o fato de não ser necessário esperar que a indexação seja feita, ou seja, utilizando a pesquisa booleana os resultados ficam disponíveis de forma imediata.

De forma a ser possível utilizar este modelo de pesquisa é usada a linguagem de interrogação CAML (**C**ollaborative **A**pplication **M**arkup Language).

Esta linguagem de interrogações, baseada em XML, permite definir um conjunto de condições de forma a encontrar os resultados desejados.

Existem algumas *tags* específicas para utilizar como por exemplo: *fields*, permite definir um campo a pesquisar, *if/else*, define condições de pesquisa, entre outras tags.

Uma CAML query deverá ser executada contra uma lista específica, retornando assim os resultados.

No nosso caso são definidas varias CAML query que interagem com a lista *Catalog*, oferecendo assim os resultados de forma imediata.

Esta pesquisa foi utilizada na fase inicial do projeto, tendo sido substituída pela pesquisa vetorial, no entanto é possível escolher qual a pesquisa a utilizar.

4.4 - Interação da Aplicação com o Modelo de Pesquisa

De forma a manter a pesquisa sempre atualizada é iniciado uma indexação incremental logo após a inserção de uma referência, garantindo assim que toda a informação seja indexada e passível de ser pesquisável.

Para poder fazer a sua pesquisa o utilizador tem ao seu dispor uma simples caixa de texto. Ao inserir o texto desejado é feita uma interrogação (*query*) ao motor de pesquisa, e este irá apresentar os resultados que melhor se adequam, na Imagem 8 é possível ver o aspeto gráfico do módulo de Pesquisa.

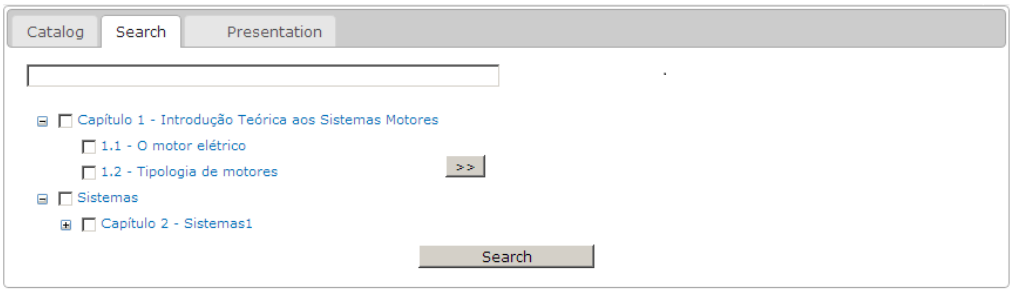

Imagem 8 - Aspecto de Pesquisa

É também disponibilizado ao utilizador uma sugestão de pesquisa, ou seja, caso o utilizador tente pesquisar por um termo que não existe, o sistema apresenta-lhe algumas sugestões, estas sugestões podem ser vistas na Imagem 9.

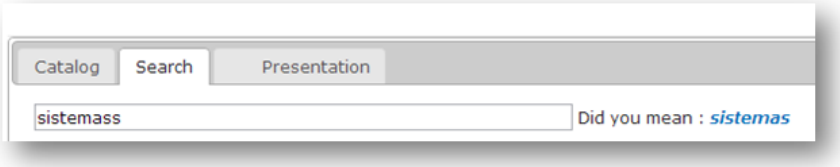

Imagem 9 - Sugestão de Pesquisas

Para permitir que a resposta vai de acordo com o pedido do utilizador é apresentado um mecanismo de filtragem, onde o utilizador poderá escolher um conjunto de *Tags* para assim refinar os seus resultados.

A filtragem com base nos conceitos selecionados utiliza o mecanismo referido no capítulo anterior, ou seja, utiliza o caminho de seleção dos conceitos para assim apresentar apenas os conceitos corretos.

A apresentação dos resultados indica ao utilizador as principais informações sobre cada um dos resultados, ou seja, apresenta a *fonte, as tags,* os *anexos* e ainda um excerto da referência introduzido. Existe ainda a possibilidade de o utilizador visualizar a totalidade da referência, bastando para isso escolher a opção ***more***.

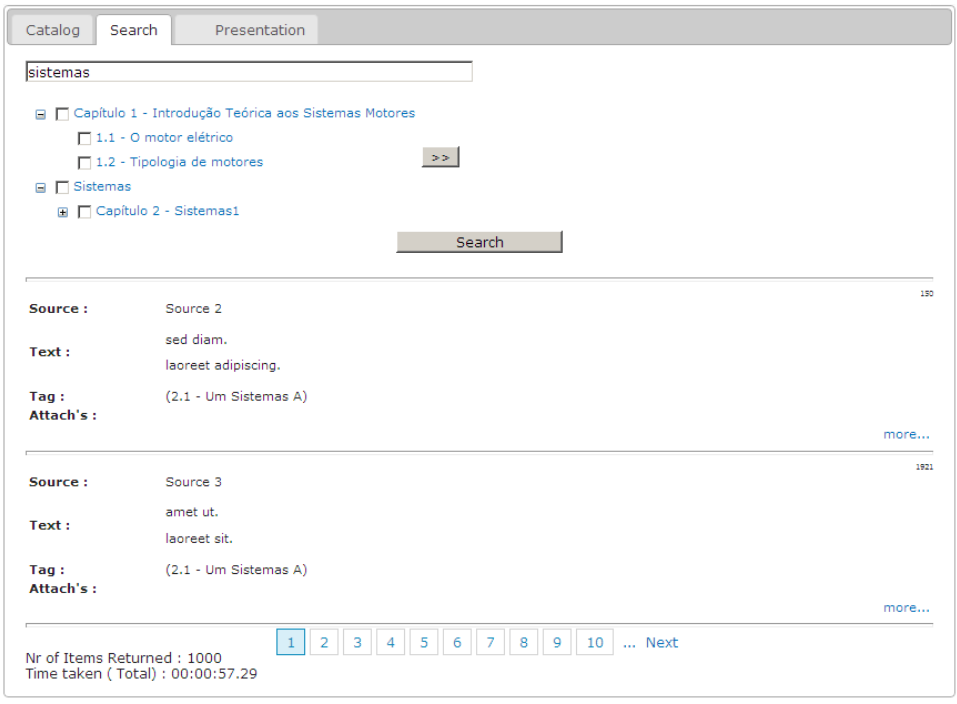

Imagem 10 - Exemplo de pesquisa com resultados

De forma a apresentar o conceito de forma correta, foi necessário recorrer a um algoritmo que removesse o caminho completo até chegar ao conceito final, ou seja, um algoritmo que consiga diferenciar os conceitos "pai" dos conceitos "filhos" permitindo assim a apresentação apenas do conceito que o utilizador escolheu, o essencial deste algoritmo pode ser visto na Imagem 11.

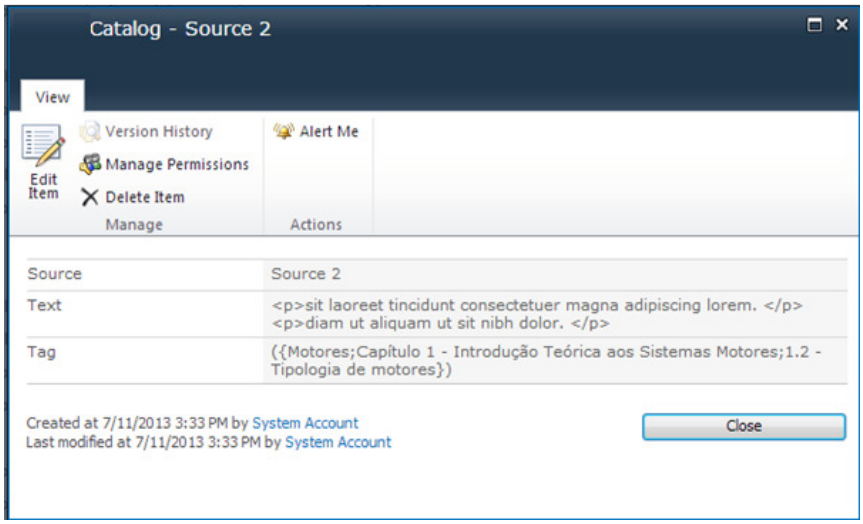

Imagem 11 - Exemplo de armazenamento de Catalogação

Como podemos ver na Imagem 11, no campo ***Tag*** é guardado todo o caminho de um conceito, mas para apresentar é utilizado o seguinte algoritmo:

1. Adquirir os conceitos associados a uma referência
2. Remover o carácter ' (' e ')'
3. Dividir o conteúdo em substrings com base no carácter ';'
4. Por cada substring remover caracteres '{' e '}'
5. Dividir o conteúdo com base no carácter ','
6. Adquir apenas o último elemento (Conceito final)

Para resumir a forma como a aplicação interage com o motor de SharePoint podemos ver a Imagem 12.

1. O utilizador especifica os termos de pesquisa
2. A aplicação envia os termos da pesquisa para o SharePoint Search Service

3. O serviço retorna todos os resultados que satisfazem o pedido
4. A aplicação, se necessário, aplica uma filtragem de acordo com os conceitos selecionados pelo utilizador
5. São apresentados os resultados finais ao utilizador

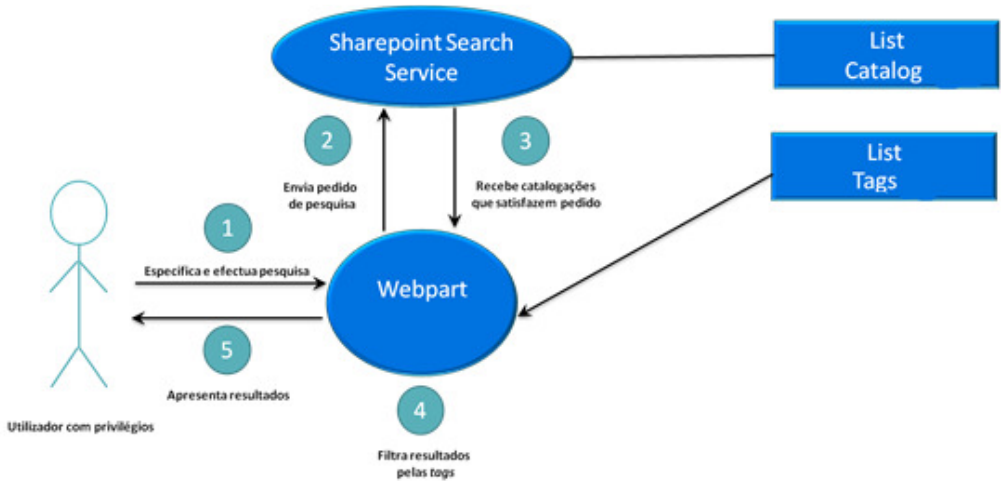

Imagem 12 - Interação utilizador e SharePoint

5 - Apresentação

Após encontrarem a informação pretendida os utilizadores pretendem apresentar os seus resultados de uma forma coerente e previamente formatada.

O módulo de apresentação permite aos utilizadores visualizarem toda a sua informação de uma forma simples, podendo estes escolher um dos diferentes *templates* configurados.

5.1 - Modelo Utilizado

Este módulo utiliza três listas de SharePoint para suportar todas as suas funcionalidades, nomeadamente a lista *Macros*, *Templates* e *Headings*.

Na lista *Headings* iremos encontrar o código CSS que será aplicado a cada parcela de texto, ou seja, é aqui que será definido todo o aspeto a aplicar a cada parcela de texto.

A lista *Templates* por sua vez define quatro zonas de texto: *Tag*, *Source*, *Text* e *Attach*. A cada uma destas zonas é possível adicionar uma referência para um dos itens existentes na lista de *Headings*, ou seja, a lista de *Templates* possibilita que se tenha aspeto de CSS diferente para cada zona apresentável - Imagem 13.

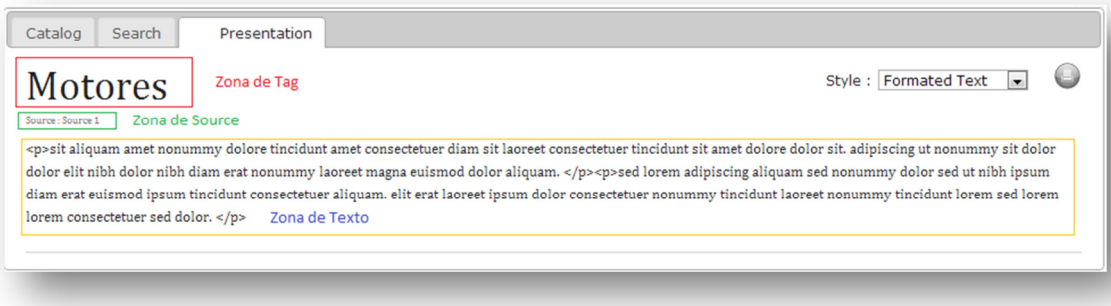

Imagem 13 - Apresentação

A lista *Macros* possibilita a escolha de múltiplas referências para a lista *Templates*, permitindo assim a possibilidade de atribuir um estilo diferente para cada nível de apresentação.

Na Imagem 14 é possível ver o modelo para esta fase da aplicação, nestes modelos temos de notar as ligações entre *Templates* e *Headings*. Existem quatro ligações entes estas listas cada ligação permite, tal como o nome indica, associar código CSS a cada uma das diferentes zonas de texto.

Associados aos Conceitos existe a opção de não apresentar estes no modo apresentação, devido a esta opção é feita uma verificação aquando de um Conceito a apresentar, e é verificado se este deve ou não ser apresentado.

Esta opção permite que algum conteúdo possa ser indexado e passível de ser pesquisável, mas que não seja apresentado.

Nestas tabelas pode se concluir que a coluna *Title* é chave candidata pois foi definida como coluna onde cada valor deve ser único.

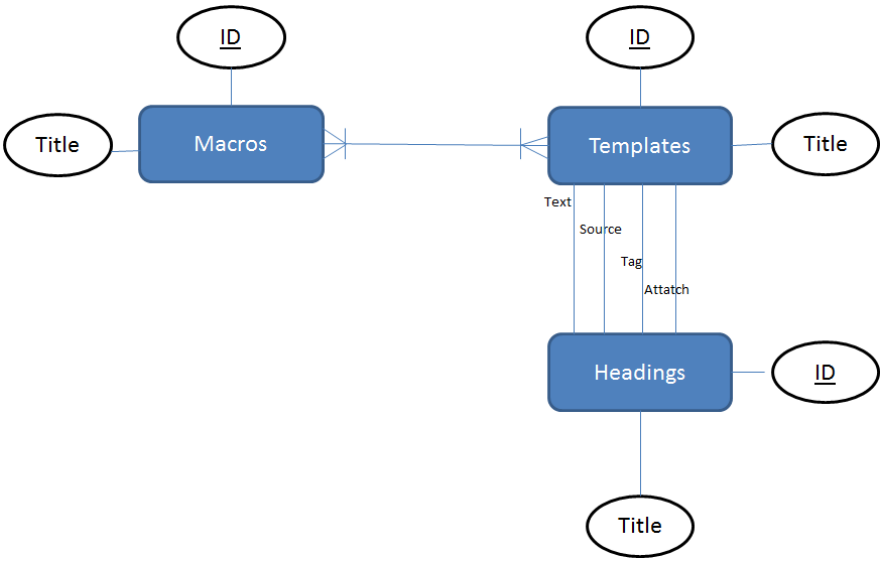

Imagem 14 - Esquema Entidade Associação Apresentação

5.2 - Ordem de Apresentação

A ordem de apresentação é definida pela taxonomia, ou seja, no primeiro nível será apresentada a taxonomia inicial, seguida pelos seus nós filhos.

A cada nível de apresentação é atribuída uma referência para um item na lista de *Templates*, como previamente explicado.

Existe ainda a possibilidade de o utilizador visualizar os seus resultados em diferentes formatos, tendo para isso apenas de selecionar as diferentes referências para a lista de ***Macros*** existentes, esta opção é possível alterando dinamicamente as classes atribuídas a cada conteúdo.

Por fim é ainda disponibilizado ao utilizador uma opção de impressão, que permite ao utilizador imprimir ou simplesmente salvar os seus dados num formato aceite pelo seu computador (pdf, word, entre outros), um exemplo de Apresentação pode ser visto na Imagem 15.

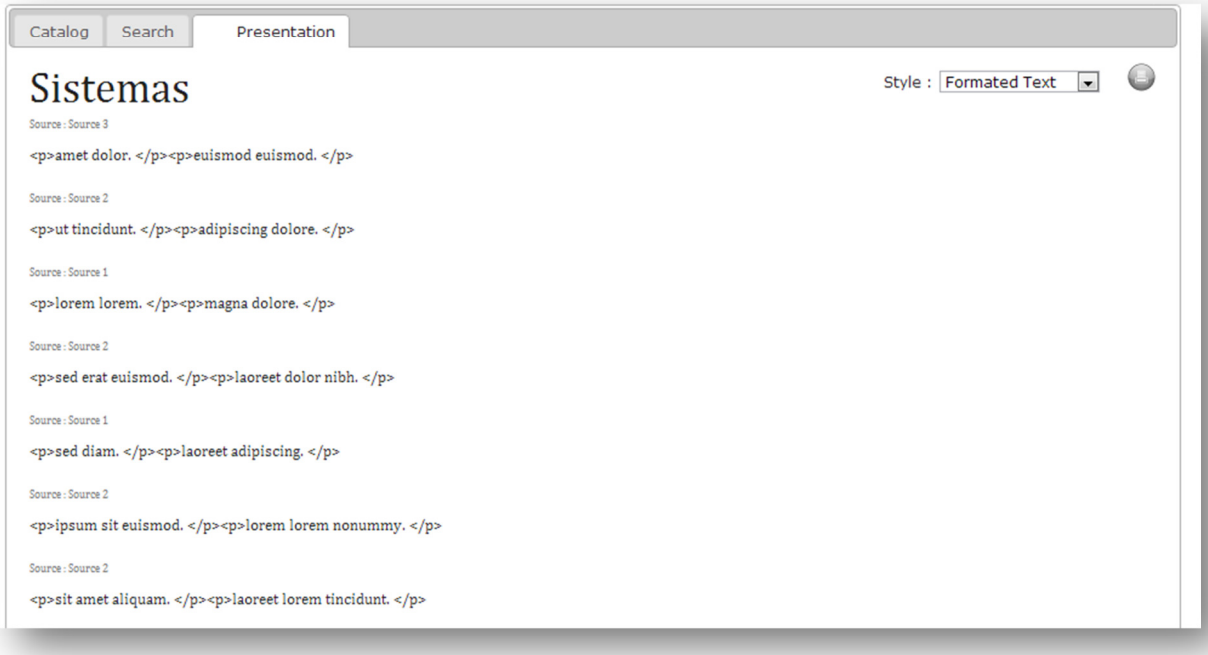

Imagem 15 - Exemplo de Apresentação

6 – Conclusões

Neste trabalho foi desenvolvida uma aplicação que oferece ao utilizador uma forma de catalogar e recuperar a sua informação. A catalogação oferece ao utilizado uma forma simples e intuitiva de colocar os seus dados e caracteriza-los utilizando as suas etiquetas. Após uma catalogação correta o utilizador pode pesquisar e encontrar os seus resultados. Por fim é oferecido ao utilizador uma forma de apresentar os seus conteúdos, forma essa previamente configurada e uniformizada.

Tendo como base uma ferramenta como o SharePoint, este trabalho conseguiu tirar partido das suas funcionalidades nativas, oferecendo assim uma aplicação de fácil uso e com funcionalidades que não existiam noutros sistemas.

Como trabalho futuro estão alinhados os seguintes tópicos:

a. Criação de um algoritmo de *ranking* de SharePoint customizado, com pesos diferentes para cada um dos atributos, oferecendo assim resultados mais precisos aos utilizadores

b. Criação de mais modelos de apresentação, os modelos criados oferecem apenas um modo muito simples de apresentar a informação, de futuro devem ser criados novos modelos com aspeto diferente, por exemplo para impressão ou para apresentação mobile

c. Integração entre os resultados da pesquisa vetorial com os resultados da pesquisa booleana

Destes tópicos, o que iria oferecer uma maior valia seria a criação do novo algoritmo de *ranking* possibilitando a definição de pesos diferentes para cada um dos campos das referências, atribuindo maior qualificação a alguns campos.

7 - Anexos

7.1 - Instalação e ativação do pacote de instalação (WSP)

Para facilitar a instalação da aplicação foram criados dois scripts *PowerShell* que instalam e ativam a aplicação.

O script de instalação efetua o registo do nosso pacote de instalação no SharePoint, para tal utiliza o método *Add-SPSolution*. Este método não instala a aplicação apenas a torna disponível na consola de administração.

De seguida é chamado o método *Install-SPSolution* sendo este o método que instala toda a nossa aplicação nos servidores possibilitando agora que esta seja ativada.

Estes primeiros métodos podem ser executados utilizando o script *01 - Feature Instalation* disponibilizado nos anexos do trabalho.

De seguida devemos ativar a nossa aplicação, para tal usamos o método *Enable-SPFeature*, este método recebe por parâmetro o url do site onde pretendemos instalar a nossa aplicação.

A aplicação necessita que duas componentes que sejam ativadas por isso é necessário chamar este método novamente com o identificador da segunda componente a instalar.

A ativação da aplicação pode ser executada utilizando o script *02 - Feature Activate* disponibilizado nos anexos do trabalho.

7.2 - Configuração de Pesquisa

De forma a utilizar a pesquisa do SharePoint é necessário criar um **Content Source** e um **Scope** de pesquisa.

Um **Content Source** consiste num conjunto de regras que especificam qual o tipo de conteúdo que deverá ser mapeado, assim como define quais os URL a mapear e o nível de profundidade associado.

Por sua vez um **Scope** permite restringir a informação a apresentar num search índex. O **Scope** tem como objetivo restringir ainda mais a informação a apresentar, por exemplo criar um **Scope** que retorna a informação financeira de um Site, e outro **Scope** que apenas retorne informação de recursos humanos.

Para criar o **Content Source** necessário temos de proceder a alguns passos.

1. Abrir o Central Administration do SharePoint 2010
2. Selecionar a opção **Manage Service Aplications**
3. Selecionar a opção **Search Service Aplication**
4. Escolher opção **Content Source**
5. Escolher opção **Criar novo Content Source**
6. Dar nome a **Content Source**(este nome será usado pela aplicação)
7. No tipo de **Content Source** escolher : SharePoint Site
8. No **Start Address**, colocar o URL da lista ***Catalog*** da aplicação
9. Nas Crawl Settings escolher a opção **: Only Crawl the Site Collection of each address**

Completando estas configurações é necessário proceder a um *Full Crawl* a partir desse momento o motor de pesquisa está ativo e preparado para retornar resultados para a nossa aplicação.

Após a criação do **Content Source** vãos criar o **Scope**, para isso temos de percorrer os seguintes passos.

1. Abrir o Central Administration do SharePoint 2010
2. Selecionar a opção **Manage Service Aplications**
3. Selecionar a opção **Search Service Aplication**
4. Escolher opção **Scopes**
5. Escolher **Criar new Scope**
6. Definir nome : **In2P – Scope**
7. Criar uma regra para o **Scope**
8. Definir a regra como **Web Address**
9. Definir o **Web address** como o URL para a lista de **Catalog** do In2P
10. No **Behavior** escolher a opção **Required**

Após a criação deste **Scope**, é necessário uma compilação do mesmo, o que poderá demorar aproximadamente 15 minutos, quando terminado, o **Scope** está disponível para uso.

A utilização do **Scope** poderia se ver como opcional, pois se o **Content Source** já mapeia apenas a lista de **Catalog** a utilização do **Scope** parece desnecessária, mas a criação do **Scope** permite que no caso de o **Content Source** não possa ser criado exclusivamente para a lista o **Scope** filtrará a informação.

Referências

[1] SharePoint 2010 Search , http://www.cmswire.com/cms/information-management/sharepoint-2010-search-relevance-refinement-people-015955.php

[2]SharePoint 2010 Search Model , http://www.microsoft.com/en-us/download/details.aspx?id=20066

[3] jQuery , http://jquery.com/

[4] Programming SharePoint 2010 Search, http://msmvps.com/blogs/windsor/archive/2011/09/29/how-to-programmatically-read-best-bets-for-sharepoint-2010-search.aspx

[5] Gestão documental, http://pt.wikipedia.org/wiki/Gest%C3%A3o_documental

[6] Lucene, http://en.wikipedia.org/wiki/Lucene

[7] Lucene, http://lucene.apache.org

[8]Alfresco , http://www.alfresco.com/

[9]SharePoint,http://office.microsoft.com/en-us/microsoft-sharepoint-collaboration-software-FX103479517.aspx