



Poi engine - Sistema automático de pesquisa e recolha de pontos de interesses turísticos relativos a formação na área da saúde

HÉLDER MIGUEL SOARES NUNES

novembro de 2016

Poi engine

***Sistema automático de pesquisa e recolha de pontos de interesse
turísticos relativos a formação na área da saúde***

Hélder Miguel Soares Nunes

***Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Sistemas Computacionais***

Orientador: Ana Maria Neves Almeida Baptista Figueiredo

Coorientador: António Constantino Lopes Martins

Júri:

Presidente:

[Nome do Presidente, Categoria, Escola]

Vogais:

[Nome do Vogal1, Categoria, Escola]

[Nome do Vogal2, Categoria, Escola] (até 4 vogais)

Porto, outubro 2016

Resumo

Sendo o turismo um dos grandes fatores económicos e sociais de alguns países, e também uma percentagem de 2.9% do PIB mundial, faz com que a área de planeamento de rotas turísticas, de grande importância (Silva 2014).

Muitos desses turistas, planeiam as suas viagens através da internet escolhendo os pontos turísticos que pretendem visitar. Esta atividade torna-se stressante devido à quantidade excessiva de informação que se obtém ao pesquisar por informação turística em motores de pesquisa convencionais (Ciurana Simó 2012). Uma forma de minimizar este esforço é utilizar sistemas de recomendação que permitem filtrar a informação, minimizando a quantidade de informação irrelevante (Ciurana Simó 2012).

Surgem com regularidade novos interesses e categorias de turismo, como, o turismo desportivo e o turismo medicinal (Mun et al. 2015). Uma categoria de turismo em crescimento neste momento é o turismo profissional, sendo que este, normalmente, é de curto prazo. Os sistemas de recomendação, em geral, recomendam nesta categoria, sugestões de restauração, transportes e museus, o que torna a categoria demasiado generalista.

O objetivo do presente projeto tem como foco uma subcategoria de turismo profissional, o turismo para profissionais de saúde. Os profissionais de saúde necessitam de se manter informados e formados nas melhores práticas e opções disponíveis para o tratamento de doenças e lesões dos seus pacientes. Para isso, começam a não ter o tempo disponível para assistir a eventos formativos dentro da área profissional em que trabalham (Smith et al. 2009). Muitos desses eventos decorrem em localizações turísticas, sendo aqui que este projeto contribui.

Neste projeto, foi criado um protótipo que recolhe a informação existente sobre estes eventos formativos, através de pesquisas avançadas. Disponibiliza essa informação a sistemas de recomendação turísticos sob a forma de ponto de interesse. Com esta informação acessível, os profissionais de saúde podem assim a planear as suas rotas turísticas nas suas férias de uma forma mais agradável.

Palavras-chave: *Sistemas de recomendação, turismo, turismo profissional, pontos de interesse*

Abstract

Being tourism a major economic and social factor in some countries, with a percentage of 2.9% of world GDP, makes planning tourism routes of great importance (Silva 2014).

Many of these tourists use the Internet to plan their trips and to pick points of interest to visit, this quickly becomes a stressful activity due to the excessive amount of information obtained while searching for touristic information on conventional search engines (Ciurana Simó 2012). A way to minimize this effort is by using recommendation systems applications, which filter information minimizing the amount of unwanted data (Ciurana Simó 2012).

There are new interests and tourism categories emerging like sports tourism and the medicinal tourism (Mun et al. 2015). A tourism category emerging is the professional tourism, and is usually short-term tourism. In general, recommendation systems, recommend for this category, suggestions in food industry, transportation, museums, treating this category as a general one. The objective of this thesis is focused on a subcategory of professional tourism, tourism for health care professionals. These professionals need to be up to date with late knowledge and practices to better treat their patients. To do so, they must attend to lectures and professionals practice courses, which most times they don't have the time available to attend (Smith et al. 2009). Many events in the health care area emerge in tourism locations, this is where the topic of this thesis is centered and where it contributes.

In this project, an application prototype was developed to collect existing information on these academic events. Providing it to recommendation systems has point of interest, this will help the health care professionals to plan their tourism routes in a more pleasing way.

Keywords: *Recommendation system, tourism, professional tourism, points of interest*

Agradecimentos

O meu maior obrigado à minha namorada Cátia Sousa e aos meus pais, Conceição Nunes e Augusto Nunes por todo o apoio dado durante meu percurso académico e profissional. Não tenho como agradecer o suficiente toda a motivação, conselhos, confiança, paciência, sacrifício pessoal.

Agradeço aos Engenheiros Ana Maria Neves Almeida Baptista Figueiredo e António Constantino Lopes Martins pela enorme disponibilidade, dedicação e a imprescindível ajuda prestada neste projeto.

A todos, muito obrigado.

Índice

1	Introdução	1
1.1	Contextualização	2
1.2	O problema	2
1.3	Objetivos propostos	3
1.4	Objetivos a atingir	3
1.5	Contributos	4
1.6	Abordagem preconizada.....	4
1.7	Abordagem metodológica	5
1.7.1	Design Science Research	5
1.7.2	Metodologia agile / scrum	7
1.8	Estrutura do documento	9
2	Estado da arte	11
2.1	Turismo.....	11
2.2	Sistemas de recomendação	13
2.2.1	Modelação do utilizador.....	14
2.2.2	Técnicas de modelação de utilizador.....	14
2.2.3	Técnicas de filtragem	15
2.2.4	Vantagens e desvantagens das técnicas de filtragem	17
2.3	Sistemas de recomendação turísticos	18
2.3.1	Desvantagens das aplicações estudadas.....	21
2.4	Aplicações e ferramentas de pesquisa e classificação.....	21
2.4.1	Motores de pesquisa Google e Bing	22
2.4.2	Aplicações de web scrapping.....	23
2.4.3	Serviço Web	25
2.4.4	Bases de dados.....	27
2.5	Conclusão	28
3	Análise de valor	29
3.1.1	Proposta de valor.....	29
3.1.2	Valor para o cliente (VC)	32
3.1.3	Contributo para a comunidade	33
3.1.4	Cenário de negócio	34
3.1.5	Modelação de Valor	34
3.1.6	Conclusão	39
4	Avaliação	41
4.1	Medidas de avaliação	41

4.2	Hipóteses e metodologias de avaliação	42
4.2.1	Condições do ambiente de testes	42
4.2.2	Ambiente de teste	42
4.2.3	Casos de teste	43
4.2.4	Resultados de teste	47
4.3	Testes estatísticos e resultados.....	47
5	Design da solução.....	49
5.1	Design Conceptual.....	49
5.2	Arquitetura	51
5.2.1	Web crawler.....	52
5.2.2	Motor de feeds.....	53
5.2.3	Web Scrapper	55
5.2.4	Gerador de POI's	56
5.2.5	Processo de criação de poi's	56
5.2.6	Portal de internet	57
5.3	Base de dados	60
5.3.1	Regras	61
5.4	Conclusão	64
6	Desenvolvimento do protótipo.....	65
6.1	Poi engine.....	65
6.1.1	Portal Web.....	65
6.1.2	Mecanismo automático de pesquisa e classificação de informação	74
7	Avaliação de resultados	87
7.1	Grandezas a avaliar	87
7.1	Hipóteses de teste e metodologias de avaliação	87
7.2	Testes estatísticos.....	88
8	Conclusões	91
8.1	Objetivos atingidos.....	91
8.2	Trabalho futuro	92
8.3	Apreciação final	92
Anexos	99	
Anexo 1	100	
Anexo 2	105	
Anexo 3	109	
Anexo 4	114	
Anexo 5	122	

Lista de Figuras

Figura 1 - Modelo de ciclo de processos <i>DSR</i> (Vaishnavi & Kuechler 2004)	6
Figura 2 – Quadro de <i>sprint</i> (Brandstetter 2011).....	8
Figura 3 – Mapa mental da <i>sprint</i> (Brandstetter 2011)	9
Figura 4 – Pedido Serviço <i>web REST</i>	27
Figura 5 – Estudo comparativo de motores de base de dados <i>SQL</i> (McColl et al. 2014)	28
Figura 6 – Diagrama componentes da aplicação	51
Figura 7 – Diagrama de componentes do <i>web crawler</i>	52
Figura 8 – Diagrama de componentes motor de <i>feeds</i>	54
Figura 9 – Diagrama de componentes <i>web scrapper</i>	55
Figura 10 – Diagrama de sequência – geração de pontos de interesse.....	56
Figura 11 - <i>API</i> pedido de pontos de interesse por categoria.....	59
Figura 12 - Registo de pontos de interesse através do portal <i>web</i>	60
Figura 13 – Modelo de dados <i>POI engine</i>	61
Figura 14 – Diagrama de classes interface gráfica do portal de Internet	66
Figura 15 – Página de entrada do portal <i>web</i>	66
Figura 16 – Funcionalidade <i>poi manager</i>	67
Figura 17 – Funcionalidade de submissão de <i>POI's</i> por <i>CSV</i>	68
Figura 18 – diagrama de classes Serviço de internet	70
Figura 19 – Resultado pedido <i>API</i> de ponto de interesse específico	71
Figura 20 - Resultado pedido <i>API</i> de todos os pontos de interesse	72
Figura 21 - Resultado pedido <i>API</i> de todas as categorias	73
Figura 22 - Resultado pedido <i>API</i> de categoria específica	74
Figura 23 - Diagrama de classes - mecanismo automático de pesquisa e classificação de informação	75
Figura 24 - Fluxograma de pesquisa sem sequência de elementos.....	83
Figura 25 – Registos sequência elementos <i>HTML</i>	84
Figura 26 – Fluxograma de pesquisa com sequência de elementos.....	85
Figura 27 – Gráfico de contexto (<i>context graph</i>) (Pant et al. 2004)	103
Figura 28 – Esquema de comunicação <i>RabbitMQ</i> (Edelson 2011).....	106
Figura 29 – Filas de comunicação <i>ZeroMQ</i> (Amazon et al. 2012).....	106
Figura 30 – Estrutura de comunicação do <i>ActiveMQ</i> (Inc. 2012).....	107

Lista de Tabelas

Tabela 1 – vantagens e desvantagens de técnicas de pesquisa.....	17
<i>Tabela 2 - Comparativo dos sistemas de recomendação</i>	<i>20</i>
Tabela 3 - Comparativo tecnológico de base de dados	28
Tabela 4 - Balanço sacrifícios e benefícios para valor do cliente	33
<i>Tabela 5 – Poi engine- modelo de negócio de Canvas</i>	<i>38</i>
Tabela 6 – Recolha de tempos de testes a motores de base de dados	47
Tabela 7 – Regras tabela enderecosURL	61
Tabela 8 – Regras tabela poi	62
Tabela 9 – Regras tabela poiEngine.....	62
Tabela 10 – Regras tabela poiCategories	62
Tabela 11 – Regras tabela categories.....	63
Tabela 12 – Regras tabela user.....	63
Tabela 13 – Regras tabela parserSequence	63
Tabela 14 – Dados obtidos para análise estatística	88
Tabela 15 – Valores obtidos para processar o teste	89
Tabela 16 – Resultados obtidos teste <i>Wilcoxon signed rank test for paired samples</i>	90
Table 17 – database tests results (in seconds).....	110
Tabela 18 - Ficheiro Csv para inserção de pontos de interesse	122

Lista de Código

Código 1 – Representação de envelope <i>SOAP</i> (Nahon 2011)	26
Código 2 – construção de endereço de pesquisa por HTML e RSS	53
Código 3 - Exemplo de feed de um curso (Formacaosaude.com 2016).....	53
Código 4 – Leitura de <i>Feeds</i> com <i>SyndicationFeed</i> (Levesque 2009).....	54
Código 5 – Atributos classe <i>POI</i>	56
Código 6 – Método de envio de ficheiro <i>CSV</i>	69
Código 7 – Método de recolha de conteúdos <i>Feed</i>	78
Código 8 – Função de recolha de data de um elemento <i>HTML</i>	79
Código 9 – Função de extração do atributo <i>href</i> de elementos âncora.....	80
Código 10 – Pedido de HTML através da biblioteca <i>HtmlAgilityPack</i>	80
Código 11 – Remoção de código <i>HTML</i> não relevante	81
Código 12 – Recolha de objetos através do tipo de elemento <i>HTML</i>	82

Acrónimos e Símbolos

Lista de Acrónimos

WTTC	<i>World Travel & Tourism Council (texto não português em itálico, se tese escrita em português)</i>
DSR	<i>Design Science Research</i>
PIB	<i>Produto interno bruto</i>
GDP	<i>Gross Domestic Product</i>
DOM	<i>Document Object Model</i>
STOMP	<i>Streaming Text Oriented Messaging Protocol</i>
AMQP	<i>Advanced Messaging Queuing Protocol</i>
HTML	<i>HyperText Markup Language</i>
API	<i>Application program interface</i>
DAL	<i>Data Access Layer</i>
BLL	<i>Business Logic Layer</i>
MVC	<i>Model View Controller</i>
UI	<i>User Interface</i>
UM	<i>User modeling</i>
Poi	<i>Point of interest</i>
URL	<i>Uniform Resource Locator</i>
CSS	<i>Cascading style sheets</i>
JS	<i>JavaScript</i>
GIS	<i>Geographical Information System</i>
CSV	<i>Comma Separated Values</i>
ACL	<i>Access Control List</i>

1 Introdução

Nos últimos anos tem-se visto um enorme crescimento do turismo profissional nas suas diversas áreas (Swarbrooke & Horner 2001). Este tipo de turismo engloba todos os acontecimentos e experiências ocorridas numa viagem de negócios (Swarbrooke & Horner 2001), como conferências, congressos ou formações profissionais.

Um problema com que os profissionais das áreas científicas se deparam, é a constante exigência formativa a que têm de se submeter, sendo necessário manterem-se atualizados por forma a progredir na carreira, e melhorando também os seus conhecimentos para melhor realizarem os seus trabalhos (Smith et al. 2009).

Com estas duas necessidades, surge um novo segmento de turismo profissional em que os pontos de interesse abrangem áreas formativas (Nunes et al. 2016).

Este projeto de mestrado recai sobre um segmento turístico específico, o turismo profissional para profissionais de saúde e visa providenciar pontos de interesse que abranjam eventos, ações de formação e congressos na área da saúde.

1.1 Contextualização

Hoje em dia, muitos profissionais em diversas áreas, são submetidos a um elevado nível de ansiedade e a demasiada exigência, o que acaba por lhes consumir a maior parte do pouco tempo livre que tem disponível (Bhattacharya & Basu 2007). Assim sendo, o planeamento de umas merecidas férias, deveria ser uma atividade calmante e relaxante mas, no entanto, com o excesso de informação disponível na internet, torna-se difícil a pesquisa e escolha dos pontos de interesse a visitar durante as mesmas (Ciurana Simó 2012; Nunes et al. 2016). Mais stressante ainda, se houver intenção de conciliar o tempo de férias com uma atualização profissional através da obtenção de algum tipo de formação. Este tipo de planeamento torna a pesquisa mais criteriosa e de mais difícil concretização, sendo que grande parte dos sistemas de recomendações turísticos contemplam apenas pontos de interesse para turismo de lazer (Adomavicius et al. 2011). Assim, os turistas incluídos no grupo de turismo profissional ou formativo, necessitam de pesquisar por eventos em turismo profissional (i.e. congressos, conferências, cursos) de forma manual, uma vez que os sistemas de recomendação não contemplam, na generalidade, este tipo de ponto de interesse.

1.2 O problema

Tendo em conta o elevado nível de stress que a sociedade apresenta no dia a dia, existe a necessidade geral de refúgio do ambiente habitual por forma a quebrar rotinas e aliviar o estado de espírito e níveis de stress com que os profissionais vivem (Gomes & Cruz 2004).

A rotina, a pressão e a falta de motivação são algumas das causas do stress a que os profissionais de saúde estão sujeitos. Uma das formas de aliviar este estado é o afastamento do ambiente habitual de rotina, sob a forma de turismo, sendo este profissional ou de lazer (Gomes & Cruz 2004).

No entanto, sendo a formação académica uma constante preocupação e exigência das carreiras profissionais, isso obriga a que os profissionais tenham que despende demasiado tempo pessoal para colmatar esta necessidade fundamental na sociedade atual. Por exemplo, os profissionais de áreas científicas necessitam cada vez mais de acesso a conhecimento atualizado, uma vez que este se encontra em constante evolução, bem com o uso de tecnologias de comunicação e de informação (Smith et al. 2009).

A tarefa de planejar quando, e o que fazer através de itinerários turísticos adaptados ao seu perfil, de modo a rentabilizar o pouco tempo disponível de que se dispõe conjuntamente com a necessidade de formação, não são contemplados atualmente, pelo menos na revisão de literatura efetuada no âmbito deste trabalho.

A presente dissertação, irá dar resposta a esta exigência de mercado, fornecendo pontos de interesse a sistemas de recomendação com interesse na mesma área.

1.3 Objetivos propostos

O objetivo principal deste projeto é providenciar a sistemas de recomendação, pontos de interesse de turismo profissional.

Para o conseguir, criou-se de um sistema informático que pesquise, recolha e forneça informação sobre eventos formativos na área da saúde, sob a forma de ponto de interesse. Esta sistema é designado de *POI engine*.

1.4 Objetivos a atingir

O *POI engine*, deverá ser um sistema automático de pesquisa e recolha de dados relativos a formação na área da saúde. Este sistema automático deverá recolher a informação necessária para gerar pontos de interesse e disponibilizá-los a sistemas de recomendação.

Para isso, deverá ser composto por um *Web crawler* por forma a pesquisar de uma lista de sítios de internet com informação considerada relevante à criação dos pontos de interesse, e também *à posteriori* um algoritmo de *scrapping* que utilizará parte dos dados recolhidos para gerar os pontos de interesse. Irá também conter um portal de internet, onde utilizadores autorizados poderão inserir manualmente alguns pontos de interesse.

Para o bom funcionamento deste sistema, os sistemas de recomendação parceiros do *POI engine* terão que conseguir fornecer pontos de interesse adaptados ao perfil do utilizador, nomeadamente com necessidade de formação profissional na área da saúde.

Por forma a demonstrar a utilidade do projeto numa perspetiva comercial, será também implementada uma prova de conceito que demonstre a validade da abordagem e a utilidade da ferramenta a desenvolver, complementada por um estudo bibliográfico que se encontra no

Abordagem preconizada

capítulo 3 com o fim de definir o valor de negócio do projeto, o impacto que este terá nos clientes, e uma proposta de negócio do projeto. Será também necessário diferenciar os valores tangíveis e intangíveis do serviço que se pretende fornecer, por forma identificar corretamente o valor do serviço.

1.5 Contributos

A contribuição deste projeto à comunidade de turismo profissional, é tornar mais simples o acesso a informação de ações de formação e eventos formativos na área de saúde. Juntando toda esta informação, e disponibilizando-a, providencia-se ao turista, a possibilidade de incluir alguns pontos de interesse profissionais durante o seu período de férias, ou em viagens profissionais. O tipo de pesquisa efetuado para encontrar esta informação é feito através de um sistema automático de classificação e pesquisa, sendo-lhe sugeridas propostas formativas sob a forma de ponto de interesse na subcategoria de turismo profissional.

Com este tipo de planeamento, o stress e tempo de planeamento de uma estadia que inclua objetivos profissionais, deverá ser reduzido e facilitado. Outro grande contributo para a comunidade, passa por ajudar na pesquisa de soluções de atualização dos conhecimentos dos profissionais de saúde para melhor desempenharem as suas funções.

1.6 Abordagem preconizada

Um sistema de recomendação turístico é também um sistema de apoio ao planeamento de rotas turísticas. O objetivo final deste projeto, passa por tornar acessível a turistas, pontos de interesse em eventos formativos na área da saúde. A solução passará pelo desenvolvimento de um sistema que deverá pesquisar, recolher e filtrar da internet por informações relativas a atividades formativas ou académicas relevantes à criação de pontos de interesse, tornando-os acessíveis a sistemas de recomendação.

Este módulo após o seu desenvolvimento, deverá ser composto por um sistema automático de pesquisa de informação sobre eventos formativos na área da saúde no Porto, uma vez que o sistema de recomendação inicial, o *ToursPlan*, contempla apenas pontos de interesse nesta cidade.

Este sistema automático, deverá ser construído em camadas, havendo uma camada de geração de pontos de interesse e um serviço web para comunicação com os sistemas de recomendação.

A pesquisa de dados relativos aos pontos de interesse, será efetuada através de um motor de *feeds*, que irá recolher a informação fornecida por sites da especialidade. E um outro módulo de pesquisa automática (i.e *web crawler*) que irá varrer uma lista de sites conhecidos, com o intuito de recolher o código fonte dessas páginas. Após a recolha deste código, um outro módulo (i.e., *scraper*) deverá retirar apenas a informação relevante à criação dos pontos de interesse. No final, a informação necessária à criação de pontos de interesse deverá estar disponível através de um serviço de internet por forma a ser obtido pelos sistemas de recomendação.

1.7 Abordagem metodológica

Neste subcapítulo apresenta-se a metodologia utilizada no desenvolvimento do projeto, tanto na parte de investigação, tanto na parte de desenvolvimento do software.

1.7.1 Design Science Research

Esta dissertação irá aplicar a metodologia de pesquisa conhecida como *Design Science Research (DSR)*.

O *DSR* dita um conjunto de paradigmas de resolução de problemas que apoiam as pesquisas tecnológicas (Cheong Christopher, Cheong France 2013).

A utilização de *DSR* nesta dissertação ajuda, segundo Vijay Vaishnavi (Vaishnavi & Kuechler 2004), na criação de novo conhecimento através da conceção de artefactos inovadores e análise da utilização ou desempenho dos mesmos, juntamente com reflexão e abstração para melhorar e entender o comportamento de um sistema. Esses artefactos incluem, algoritmos (i.e. recolha de informação), interfaces de utilizador e metodologias de sistema ou linguagens de programação (Vaishnavi & Kuechler 2004).

A metodologia *DSR* segue normalmente um ciclo de processos de investigação. Alguns desses ciclos tem momentos que obrigam o investigador a voltar a ciclos anteriores e rever ou

Abordagem metodológica

adquirir novos conhecimentos por forma a continuar com a sua pesquisa. A Figura 1, abaixo apresentada, demonstra o ciclo de um trabalho de investigação seguindo a metodologia *DSR*.

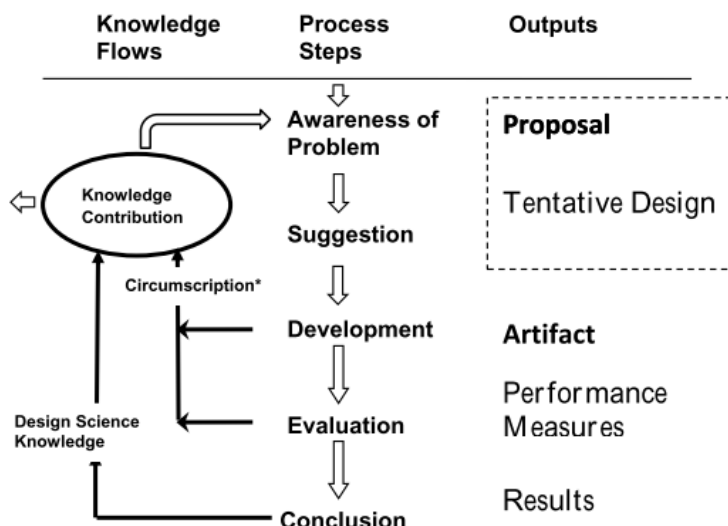


Figura 1 - Modelo de ciclo de processos *DSR* (Vaishnavi & Kuechler 2004)

Como se pode verificar, o processo de investigação tem um processo que vai ser detalhado de seguida no âmbito deste trabalho, com base no artigo "*Design Science Research in Information Systems Section*" de Vijay Vaishnavi e Bill Kuechler (Vaishnavi & Kuechler 2004).

Consciencialização do problema, é a consciência por parte do investigador para um problema de investigação do seu interesse, podendo este ser de qualquer fonte, mesmo que diferente da sua área profissional. O objetivo desta fase é uma proposta para uma nova investigação.

Sugestão, é a fase seguinte, aparece após a proposta e está interligada com ela, uma vez que a sugestão é feita com base no resultado da proposta, e com um design inicial da aplicação, ou pelo menos, uma ideia do possível resultado final.

Desenvolvimento, surge após o estudo de uma possível solução de design da aplicação, onde temos a possibilidade de começar com o desenvolvimento da aplicação, seguindo os padrões escolhidos no design da aplicação. Nesta fase temos um ponto de saída em que, caso a solução apresentada no Design não nos tenha permitido obter tanto conhecimento do problema quanto o necessário para o desenvolvimento da aplicação, devemos voltar às fases anteriores e recolher mais informações que permitam alterar o design.

Avaliação, é feita após o desenvolvimento da aplicação, esta deverá ser avaliada de acordo com os critérios definidos na proposta. Caso os resultados não estejam de acordo com o

proposto, deve ser atribuído um grau de tolerância em função com o resultado esperado e caso a avaliação não obtenha resultados satisfatórios aos requisitos propostos, deve-se voltar às fases iniciais e alterar por forma a atingir os objetivos.

A fase de conclusões, deverá remeter para o final de um ciclo de investigação ou para a conclusão do projeto. No ciclo de conclusão de uma investigação ou projeto, os resultados, não só em termos de desenvolvimento e escrita, mas também em termos de conhecimento adquirido ou criado, deverão ser satisfatórios mediante os objetivos pressupostos.

1.7.2 Metodologia agile / scrum

Scrum é o processo da metodologia *Agile* mais praticado, tem sido usado em desenvolvimento de software com sucesso há 20 anos. Como outros métodos *Agile*, o *scrum* otimiza recursos limitados por forma a criar eficiência. Também potencia a organização da equipa e o trabalho ritmado, sem interferências externas. Isto ajuda as equipas de desenvolvimento a atingir maturidade e potencial máximo, libertando os gestores de equipa para ações de gestão estratégica da empresa, em vez gestão de equipa.

1.7.2.1 Divide and conquer

O *scrum* segue um princípio de *divide and conquer*, divide tarefas complexas em tarefas pequenas e simples, equipas grandes em equipas mais pequenas e eficientes, e projetos com tempos de desenvolvimento elevados em séries de objetivos de curta duração, denominados de sprints.

A ideia de separar trabalho complexo em tarefas mais pequenas e simples, torna mais fácil mapear o que precisa de ser feito, criando assim um mapa de tarefas (*roadmap*) claro e conciso, o que permite à equipa começar o desenvolvimento.

Ao dividir equipas grandes em equipas mais pequenas, o *scrum* transforma a organização em pequenas empresas, tornando assim as equipas especializadas em determinadas áreas (i.e., *front end*, *back end*, *middleware*). Outra vantagem é melhorar o foco das equipas, pois com equipas mais pequenas perde-se menos tempo a comunicar detalhes.

O princípio *divide and conquer*, também particiona projetos grandes em séries de sprints pequenas, tornando assim mais simples o planeamento para períodos de tempo pequenos.

Abordagem metodológica

1.7.2.2 Inspeccionar e adaptar

Ao partir o trabalho em tarefas mais simples, torna-se mais rápido de entregar. Acelerando o processo e colocando o produto em utilização, mais rapidamente se obtém *feedback*. E o *feedback* ajuda a equipa a melhorar o produto.

1.7.2.3 Transparência

Na metodologia *scrum*, não há segredos quanto às tarefas que existem para fazer, quem as está a fazer e como serão atingidas. Se uma organização tem este tipo de conhecimento, possíveis erros que ocorram podem ser corrigidos quase de imediato.

Neste projeto e, uma vez que não existe uma equipa de desenvolvimento, a metodologia *scrum* tem como vantagem o planeamento do trabalho em tarefas pequenas e simples. Para isso está a ser usada uma plataforma de *scrum* na internet disponível gratuitamente, *work-2-together* da empresa *2plan* (Brandstetter 2011).

Esta aplicação permite o mapeamento das tarefas num quadro de *sprint*, com atribuição de tarefas (tickets) e planeamento temporal, com alguns estados de desenvolvimento.

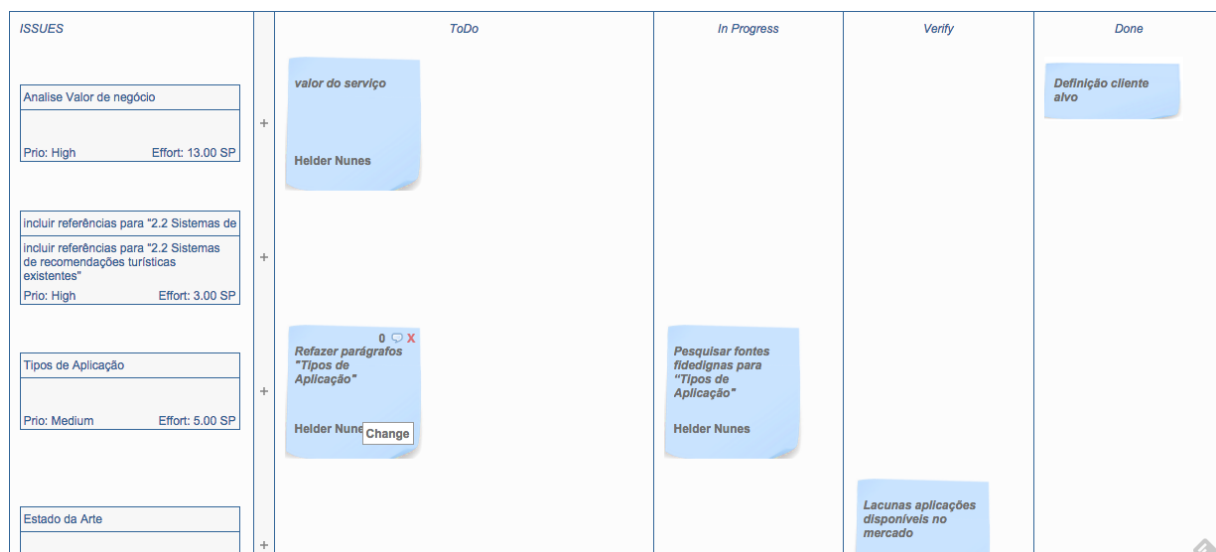


Figura 2 – Quadro de *sprint* (Brandstetter 2011)

A Figura 2 mostra, os requisitos (*user stories*) na coluna à esquerda, e as tarefas (*tasks*) necessárias à conclusão do requisito sob a forma de post-it azul nas colunas da direita. Estas tarefas podem ser atribuídas a pessoas diferentes. O estado de desenvolvimento das tarefas vai avançando mediante o desenvolvimento da mesma, até que chega à sua conclusão.

Esta aplicação também providencia uma ferramenta de mapeamento mental, o que ajuda, mais a administração da equipa, a organizar as tarefas de forma temporal e por prioridades. Como se pode ver na Figura 3.

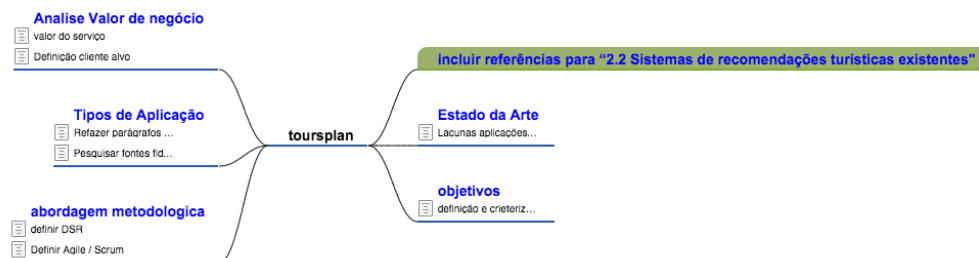


Figura 3 – Mapa mental da *sprint* (Brandstetter 2011)

Com a utilização da metodologia *scrum* no desenvolvimento da aplicação, deverá tornar-se mais simples e rápida a conclusão da mesma.

1.8 Estrutura do documento

Este documento está dividido em sete capítulos, descritos de seguida:

O capítulo 1 descreve a introdução ao projeto, com o objetivo de contextualizar de o numa perspetiva global do problema. São sumariamente apresentados o problema e os principais objetivos a atingir. São também apresentadas as abordagens a seguir, tanto na investigação como desenvolvimento do projeto.

O capítulo 2 pretende contextualizar a problemática do projeto. Aqui apresenta-se o estudo do estado de arte realizado, mostrando-se também todas as necessidades inerentes ao projeto e fazendo uma revisão bibliográfica de projetos existentes no mercado.

O capítulo 3, apresenta uma proposta de valor pormenorizada do projeto, contendo o valor do produto com base na sua definição, valor para o cliente, cliente alvo e contribuição para a comunidade.

O capítulo 4 apresenta o estudo estatístico das tecnologias estudadas como hipótese de escolha.

O capítulo 5 por sua vez, apresenta uma análise detalhada referente aos componentes

Estrutura do documento

inerentes à implementação do projeto. Refere pormenores sobre a base de dados que contém toda a informação referente ao mecanismo de geração de pontos de interesse, bem como os dados de utilizador do portal de internet;

O capítulo 6 apresenta o desenvolvimento conseguido do projeto, e também uma prova de conceito de trabalho a realizar no futuro;

Os capítulos 7 e 8 apresentam estudo estatístico com base nos resultados obtidos, e algumas conclusões do projeto, referindo-se um conjunto de melhorias a desenvolver em trabalho no futuro no projeto;

No final deste documento encontra-se a lista de referências bibliográficas, e uma secção com anexos de apoio ao documento.

2 Estado da arte

Neste capítulo são apresentados os conceitos relacionados com o tema da tese, nomeadamente o tipo de turismo em foco, tipo de sistemas alvo para fornecer os pontos de interesse gerados pelo sistema construído, e as técnicas que utilizam.

Estas aplicações diferenciam-se sob várias perspetivas, como: tipo, funcionalidades ou publico alvo. No decorrer deste capítulo serão descritos alguns destes sistemas e o tipo de aplicação e usabilidade que fornecem.

Será também apresentado neste capítulo um estudo relativo a tecnologias e conceitos disponíveis, relacionados com a implementação de sistemas de recomendação, com o objetivo de avaliar quais as mais vantajosas face ao problema em consideração.

2.1 Turismo

Denomina-se de turismo, segundo Emili Ciurana Simó (Ciurana Simó 2012), o conjunto de atividades levadas a cabo por pessoas no decorrer de viagens feitas pelas mesmas, em localizações diferentes do seu ambiente habitual. É um período de tempo consecutivo com duração entre um dia e um ano, e pode ter base em questões de lazer ou profissionais.

Sendo que o turismo é um dos grandes fatores económicos e sociais de alguns países, e tendo também uma percentagem de 2.9% do PIB mundial, isto faz com que a área de planeamento

Turismo

de rotas turísticas e apresentação de pontos de interesse turísticos a possíveis clientes, seja muito importante (Silva 2014).

Este subcapítulo tem como objetivo diferenciar os tipos de turismo abordados pelo *POI engine*, sendo eles o turismo profissional e o turismo formativo.

Turismo profissional

Segundo Rob Davinson no livro *Business Travel* de 1994, turismo profissional é toda a viagem que alguém realiza com pressupostos relacionados com o seu trabalho (Swarbrooke & Horner 2001). Ou seja, definido como o ato de viajar com objetivos relacionados com o trabalho. Este é um dos tipos de turismo mais antigos existentes, sendo a famosa rota da seda e o caminho marítimo para a Índia descoberto por Vasco da Gama um exemplo disso.

Turismo Formativo

Turismo formativo, acontece quando um turista se desloca a uma localização pré-definida com o intuito de comparecer num curso de formação.

Inicialmente, o propósito deste trabalho, seria adicionar uma subcategoria do turismo profissional a um sistema de recomendação de pontos de interesse já existente: o *Tours Planning Support System* também conhecido como *ToursPlan*.

O *ToursPlan* é um sistema de recomendação de turismo, criado no grupo de investigação GECAD do Instituto Superior de Engenharia do Porto em parceria com a Câmara Municipal do Porto e tem o objetivo de ajudar a encontrar um plano turístico personalizável ao seu período de estadia (Almeida 2009).

No entanto devido à inexistência de acesso ao sistema, optou-se por implementar um serviço de Recomendação de pontos de interesse em turismo profissional.

2.2 *Sistemas de recomendação*

Com o crescimento da internet, aumentou também exponencialmente a quantidade de informação existente. Com este aumento de informação, surgiu a necessidade de filtrar os resultados das pesquisas mais eficientemente. Uma forma de resposta a esta necessidade, são os sistemas de recomendação, que surgem com o objetivo de minimizar o tempo consumido em pesquisas de internet (Luz et al. 2010), e de apoiar na escolha da informação que se pretende encontrar.

Os sistemas de recomendação, são ferramentas de software que apresentam sugestões em benefício do utilizador. A sugestão apresentada é escolhida com base em vários processos de decisão, como que itens comprar, que musica ouvir, que site de noticias ler. As sugestões apresentadas ao utilizador são, em geral, designadas pelo termo item (Bogers & Van Den Bosch 2009; Ciurana Simó 2012; Ricci et al. 2011).

Os sistemas de recomendação são, em geral, especializados num domínio específico (i.e., musica, livros, turismo) e, a sua aparência, funcionalidades e as técnicas utilizadas, são construídas por forma a facilitar o mecanismo de gerar sugestões para apresentar. Todas estas características do sistema de recomendação são customizadas para fornecer sugestões úteis ao utilizador de forma eficaz, no domínio em que é especializado. Na sua essência um sistema de recomendação, pode ser considerado um sistema de apoio à decisão com uma vertente mais comercial (Coelho 2010).

Sendo o contexto desta tese ligado ao turismo, o âmbito deste capítulo foca-se em sistemas de recomendação turísticos e, mais propriamente, na pesquisa e classificação dos pontos de interesse. A função de um sistema de recomendação turístico, passa por facilitar, a um potencial turista, a escolha dos locais a visitar e o planeamento de uma estadia numa determinada localização. Para isso, o sistema de recomendação apresenta uma série de pontos de interesse na localização pretendida, baseada em parâmetros de pesquisa, e potencialmente no seu perfil de utilizador ou em perfis de utilizador semelhantes (Luz et al. 2010).

2.2.1 Modelação do utilizador

Os sistemas de recomendação são, na realidade, um conjunto de técnicas de filtragem de informação, utilizadas para seleccionar qual a melhor ou a mais relevante informação que o utilizador pesquisa ou, muitas vezes, apenas tem interesse (Coelho 2010). Para isso, um tópico muito importante para o bom funcionamento de um sistema de recomendação é o processo de recolha de conhecimento das interações que o utilizador vai tendo com as aplicações. Inicialmente a utilização de formulários de satisfação e questionários de interesses eram utilizados como forma de recolher informação para traçar o perfil do utilizador (Luz et al. 2010). No entanto esta técnica era baseada na boa vontade dos utilizadores despendarem do seu tempo para os responder e submeter. Isso provoca uma má experiência de utilizador, o que leva a que muitos desses utilizadores não gostem da aplicação. Com isto, e tendo em conta que os sistemas de recomendação têm normalmente uma vertente comercial, e com o aparecimento de novas e muitas vezes melhores aplicações, fidelizar os clientes é cada vez mais uma preocupação. Assim sendo, a recolha de informações para traçar os perfis de utilizador, deve ser o mais transparente possível, sem prejudicar em demasia a experiência de utilização. Este tipo de funcionamento ajuda a aumentar a confiança que os utilizadores têm na aplicação, fidelizando-os (Luz et al. 2010). Esse tipo de informação é recolhido num componente de *user modelling*, que hoje em dia é muito utilizado em conjunto com os sistemas de recomendação (Coelho 2010).

2.2.2 Técnicas de modelação de utilizador

Existem várias técnicas e metodologias de modelação de utilizador que tem vindo a ser utilizadas nos últimos anos para criar perfis de utilização. Neste documento iremos descrever algumas das mais importantes, uma vez que existem muitas variantes criadas com base nestas (Coelho 2010).

Modelos lineares (*linear models*):

Sendo esta uma das técnicas mais utilizadas, estes modelos tem uma implementação simples e de fácil compreensão. São bastante eficientes e baseiam-se em dados probabilísticos (Zukerman et al. 2001).

Árvores de decisão (*decision trees*):

Esta é também uma técnica de fácil compreensão e implementação simples. Consiste em árvores com nós, em que cada nó representa diferentes valores ou escolhas de vários atributos (Coelho 2010).

Redes neuronais (*neural networks*):

Sendo uma das técnicas mais recentes a ser utilizada em modelação de utilizadores, a sua metodologia faz analogia COM a rede neurológica de um cérebro humano. Uma rede neuronal é composta por um conjunto de nós (neurónios por analogia), interligados e relacionados entre si. Estas relações definem uma métrica em cada nó (neurónio). Essa métrica define a importância da informação guardada no nó, e as relações mais cotadas, definem sugestões propícias ao utilizador (Coelho 2010).

Existem outras técnicas de modelação de utilizadores, no entanto o foco desta tese não recai sobre este tema. Este tipo de recolha de informação tem interesse para sistemas de recomendação. No caso do *POI engine* teria se fosse recomendar em tempo real os seus pontos de interesse.

2.2.3 Técnicas de filtragem

Por norma, o funcionamento de um sistema de recomendação é baseado em paradigmas de filtragem de informação. Existem vários, mas os mais importantes, segundo vários autores (Berka, T., & Plößnig 2004; Luz et al. 2010; Coelho 2010) , são os 3 seguintes: filtragem baseada em conteúdos (*content-based filtering*), filtragem colaborativa (*collaborative filtering*), filtragem baseada em conhecimento de domínio (*knowledge-based filtering*). Existem também sistemas de recomendação que misturam estas técnicas dando assim a existência a uma quarta técnica denominada de filtragem híbrida (*hybrid filtering*) que mistura dois ou mais dos paradigmas anteriores (Berka, T., & Plößnig 2004).

Filtragem de informação baseada em conteúdo (*Content-based filtering*):

Esta técnica é utilizada por sistemas de recomendação que se baseiam na informação dos itens da sua base de conhecimento para apresentar recomendações ao utilizador. Assim sendo, as recomendações são baseadas nos atributos dos itens e em pontuações e revisões para calcular se o item satisfaz as necessidades do utilizador, ou seja, o sistema de

Sistemas de recomendação

recomendação contém uma base conhecimento com itens estruturados ou não estruturados, e pesquisa neles por itens similares às preferências do utilizador, traçando assim um perfil dos gostos do cliente. A recolha dos itens neste paradigma, é baseada na comparação textual dos atributos dos itens da base de conhecimento por algoritmos de *text mining* (Pazzani & Billsus 2007), normalmente de duas formas:

- pesquisa e extração por palavras ou frases chave de conteúdos de texto e, comparando estes conteúdos com os atributos do perfil de utilizador ou com outros itens de informação através de cálculos probabilísticos;
- comparar a similaridade entre excertos de textos com as palavras chave.

Se conseguir alguns resultados nessa pesquisa, apresenta os itens mais semelhantes ao utilizador.

Filtragem de informação colaborativa ou social (Collaborative filtering):

Esta técnica de pesquisa de recomendações, é baseada em perfis de utilizadores com características semelhantes. Com o forte crescimento da recolha de conteúdos sociais por parte das páginas de internet, esta técnica tornou-se muito utilizada para os sistemas de recomendação (Berka, T., & Plöβnig 2004; Coelho 2010). O funcionamento deste tipo de pesquisa passa pelo processamento de funções de cálculo de similaridade entre perfis de utilizadores. Em teoria este método entende que, se dois ou mais utilizadores tem perfis semelhantes, então os seus gostos também serão semelhantes, podendo assim criar grupos de utilizadores. Assim, sempre que um utilizador de um destes grupos visitar e validar o seu gosto por um item, este mesmo item passa a ser um item a recomendar a utilizadores do mesmo grupo. Para definir utilizadores com perfis semelhantes, são comparados alguns traços de personalidade e outras características tais como, sexo, idade, orientação sexual entre outros.

Filtragem de informação baseada em conhecimento (Knowledge-based filtering):

Esta técnica baseia-se em toda a base de conhecimento do sistema de recomendação e, assim sendo, é muito aconselhável a sua utilização. Aqui, o objetivo passa por inserir nas propriedades e atributos dos itens da base de conhecimento, algum tipo de característica que diferencie o item num determinado domínio. Por exemplo, saber que um monumento só pode pertencer a um de X estilos de arquitetura, distingue esse item dos itens de outros

estilos arquitetônicos. O funcionamento deste paradigma passa por processar funções de similaridade entre os itens. Com esses atributos como parâmetro de comparação, consegue-se recolher assim itens com características similares aos itens já visitados ou utilizados pelo utilizador. Os itens recolhidos para sugestão, são vistos pelo sistema como itens com potencialidade de o utilizador gostar. No entanto, este paradigma é mais profundo que o descrito, para além desta similaridade, o sistema também verifica se as características dos itens coincidem com as características do perfil do utilizador (Coelho 2010).

Filtragem híbrida (Hybrid filtering):

A filtragem híbrida existe quando o sistema de recomendação aplica mais do que uma das técnicas descritas, ou outras não descritas. Por exemplo, um sistema híbrido poderá utilizar na pesquisa de itens os métodos colaborativos e baseado em conteúdos (Coelho 2010).

2.2.4 Vantagens e desvantagens das técnicas de filtragem

Este subcapítulo tem como objetivo destacar as vantagens e desvantagens de cada uma das principais técnicas de filtragem. Esta avaliação é baseada no estudo feito que pode ser encontrado em (Coelho 2010), e segundo este estudo, a avaliação das técnicas de pesquisa foi feita de forma imparcial. A Tabela 1 refere as vantagens e desvantagens de cada uma das técnicas abordadas.

Tabela 1 – vantagens e desvantagens de técnicas de pesquisa

Técnica de filtragem	Vantagens	Desvantagens
Content-based	<ul style="list-style-type: none"> • De fácil implementação e entendimento; • Não requer que a informação esteja estruturada. 	<ul style="list-style-type: none"> • Não contém características de domínio.
Collaborative	<ul style="list-style-type: none"> • Simula a recomendação humana; • Utiliza informação recolhida em interações sociais utilizando a arquitetura da web 2.0. 	<ul style="list-style-type: none"> • Necessita de uma enorme quantidade de informação, e requer muito tempo para alimentar inicialmente a base de conhecimento.
Knowledge-based	<ul style="list-style-type: none"> • Distingue itens por domínio; • Utiliza o perfil do utilizador para validar as recomendações; • A sua utilização é intemporal. 	<ul style="list-style-type: none"> • Exige muito processamento para conseguir um bom modelo de itens.

Como demonstra a Tabela 1, existem vários fatores a ter em conta na escolha das técnicas de filtragem de informação. Sendo a técnica baseada em conteúdos mais fácil de entender e aplicar, no entanto, não permite a utilização de domínios, o que implica processar toda a informação para recolher sugestões.

A técnica colaborativa tem como vantagem simular a recomendação humana, ou seja, utiliza informação recolhida através de interações sociais entre utilizadores, e recomenda com base nessa informação. A desvantagem desta técnica, é que no seu início requer um período de tempo considerável para construir a sua base de conhecimento.

Por fim, a técnica baseada em conhecimento, utiliza domínios como fator de eliminação, ou seja, elimina itens que não tenham relação de domínio com o contexto da pesquisa, e compara os resultados com as preferências do perfil do utilizador. Como desvantagem exige demasiado processamento para obter sugestões e construir a base de conhecimento.

2.3 Sistemas de recomendação turísticos

Existem muitas aplicações de planeamento turístico disponíveis no mercado, com diferentes objetivos e funcionalidades vantajosas ao turista. Neste subcapítulo vão-se abordar diversas destas aplicações e descrever as suas vantagens.

- ***Personal Travel Assistant (PTA)***: esta aplicação é orientada à área de reserva de voos. Este sistema traça o perfil dos utilizadores, caracterizando-os pelas suas preferências e usa depois essa informação quando recolhem itens para sugerir. Esta aplicação é um exemplo da técnica de pesquisa baseada em conteúdo (Coyle & Cunningham 2003).
- ***Entreé***: esta aplicação utiliza a técnica de conhecimento de domínio (*Knowledge-based filtering*). A sua área de atuação é a restauração, sugerindo restaurantes e gastronomia aos seus utilizadores (Ciurana Simó 2012).
- ***TIP (Tourist Information Provider)***: esta aplicação móvel utiliza técnicas híbridas para recomendar vistas ou pontos de interesse ao utilizador bem como, informação sobre o que o está a visitar. Para além disso sugere outros pontos de interesse na localização. Tudo isto é calculado em similaridade com as características do perfil do utilizador, *feedback* de, e comparação com utilizadores

de perfil semelhante, histórico de visitas, contexto do utilizador e do item (Anacleto et al. 2010).

- **CATIS:** esta aplicação fornece informação sobre pontos de interesse baseados na localização do dispositivo móvel, hora atual, direção de deslocamento, e preferências do utilizador. Por exemplo, se o utilizador tiver nas suas preferências recomendação na área de restauração, a aplicação recomendará restaurantes nas imediações dentro do período normal de refeições (Anacleto et al. 2010; Luz et al. 2010).
- **M-ToGuide Project:** esta aplicação móvel, atua no mercado turístico europeu e oferece informações multimédia sobre pontos de interesse turísticos nas redondezas do dispositivo. Esta aplicação apresenta apenas itens na localização do dispositivo (Luz et al. 2010).
- **Deep Map:** esta aplicação tem o propósito de apoiar a navegação com objetivo turístico, na cidade de Heidelberg, fornecendo tours guiadas. As tours recomendadas, são baseadas nos interesses e necessidades pessoais do utilizador, estado sócio cultural (i.e., idade, sexo), e o meio de mobilidade disponível. Os itens apresentados nesta aplicação são do tipo *Geographical Information System (GIS)*, ou seja, apresenta itens baseados na localização atual (Luz et al. 2010).
- **GoGoBot, Tripit e Trippy:** estas 3 aplicações, são sistemas de recomendação geográfico-sociais, e utilizam a técnica de filtragem colaborativa, uma vez que funcionam por grupos de utilizadores, baseando-se no *feedback* dado pelos utilizadores, e nos seus perfis (de Graaff et al. 2012).

Existem muitas outras aplicações de turismo, sendo cada uma delas orientada a vários tipos de domínio da área de turismo. As aplicações acima apresentadas foram estudadas com base na investigação de vários autores, e também, tendo em conta os principais tipos de filtragem de informação apresentadas no capítulo 2.2.3. Outro fator da escolha destas aplicações é o tipo de item que fornecem podendo também ser opções de interesse para o cliente alvo que se pretende atingir (i.e., restauração, monumentos, vistas). No decorrer deste estudo bibliográfico, não foi encontrada nenhuma aplicação que sugira eventos em áreas profissionais.

A *Tabela 2* descreve resumidamente as aplicações descritas anteriormente. Faz referência ao tipo de itens em que é especializada, se utiliza alguma técnica de modelação de utilizador, o

Sistemas de recomendação turísticos

tipo de técnica de filtragem que utiliza e se recomenda pontos de interesse em eventos formativos.

Tabela 2 - Comparativo dos sistemas de recomendação

Aplicação	Tipo	UM	Técnica de filtragem	Eventos formativos
Personal Travel Assistant (PTA)	<i>Voos</i>	<i>Sim</i>	<i>Content-based</i>	<i>Não</i>
Entreé	<i>Restauração</i>	<i>sim</i>	<i>Knowledge-based</i>	<i>Não</i>
TIP (Tourist Information Provider)	<i>Informação localização</i>	<i>Sim</i>	<i>Hybrid</i>	<i>Não</i>
CATIS	<i>Informação localização</i>	<i>Sim</i>	<i>Hybrid</i>	<i>Não</i>
M-ToGuide Project	<i>Informação localização</i>	<i>Sim</i>	<i>Hybrid</i>	<i>Não</i>
Deep Map	<i>Informação localização</i>	<i>Sim</i>	<i>Hybrid</i>	<i>Não</i>
Trippy	<i>Planeamento de viagens de estrada, tours</i>	<i>Sim</i>	<i>Hybrid</i>	<i>Não</i>
Gogobot	<i>Alojamento, voos e tours</i>	<i>Sim</i>	<i>Hybrid</i>	<i>Não</i>
Tripit	<i>Planeamento de voos</i>	<i>não</i>	<i>Hybrid</i>	<i>Não</i>

Das aplicações estudadas, a que mais se aproximada ao objetivo do projeto é a *GoGoBot*, pois contém a categoria profissional e académica como opção de escolha de grupo de utilizadores. As sugestões fornecidas nesse domínio, são pontos de interesse turísticos gerais como museus, monumentos, vida noturna, e não eventos formativos. Tendo isto em conta, não foi

encontrada, até ao momento, uma aplicação que satisfaça as necessidades académicas dos profissionais de saúde.

2.3.1 Desvantagens das aplicações estudadas

Os sistemas de recomendações apresentados no ponto anterior, têm como desvantagem, face ao propósito desta tese, não segmentar o turismo profissional. Na sua grande maioria, os sistemas de recomendações turísticos, apenas sugerem, para a categoria de turismo profissional, alojamento, opções de restauração, e alguns pontos de interesse para lazer nas horas vagas. Outra desvantagem, no turismo académico, subcategoria não existente em muitos deles, sugerem opções para divertimento noturno e algumas opções culturais. Em suma, sugerem pontos de interesse genéricos.

Uma vez que profissionais habilitados necessitam de atualização constante das suas competências profissionais, o turismo profissional e o turismo formativo estão interligados. Assim podem-se apresentar propostas formativas em ambos, surgindo uma oportunidade para os sistemas de recomendação de adquirirem novos clientes (Nunes et al. 2016).

2.4 Aplicações e ferramentas de pesquisa e classificação

Sendo a quantidade de informação disponível na internet extensa, torna-se difícil filtrar as pesquisas sem se obter informação desnecessária. Com isto, para criar pontos de interesse com base nestes resultados, será necessário fazer pesquisas mais detalhadas, e fazer uma triagem à informação retornada, uma vez que mesmo em pesquisas detalhadas também se obtém informação desnecessária (Nunes et al. 2016).

São várias, as aplicações disponíveis na internet para pesquisar por informação específica e, neste capítulo, vamos abordar algumas como, *web crawlers*, *feeds RSS* e *scrappers*.

Inicialmente pensou-se a implementação de um algoritmo de pesquisa na internet (*web crawler*). Foi feito um estudo sobre diversos algoritmos deste tipo, no entanto, chegou-se à conclusão de que não seria a melhor abordagem para o problema. O estudo realizado pode ser visto no Anexo 1.

2.4.1 Motores de pesquisa Google e Bing

Uma vez que existem já soluções viáveis e gratuitas de pesquisa de informação disponíveis, foi elaborado um estudo sobre o funcionamento de motores de pesquisa acessíveis na internet, nomeadamente o Google.pt e o Bing.pt. Sendo que a necessidade do projeto passa por recolher endereços URL que contenham no seu conteúdo informação relevante à criação de pontos de interesse, será necessário recolher o resultado dessas pesquisas para recolher os dados pretendidos. Sendo que os motores de pesquisa funcionam pelo envio de um pedido GET, com um conjunto de parâmetros configuráveis, pode-se manipular esses parâmetros por forma a conseguir filtrar os resultados. Sendo o Google um dos motores de pesquisa mais utilizados atualmente, foram analisados que parâmetros utiliza, e que conteúdos aceitam na página de internet de google *advanced search* (Google 2016), sendo os parâmetros:

- as_q: pesquisa todas as palavras atribuídas a este parâmetro;
- as_epq: pesquisa exatamente as palavras atribuídas a este parâmetro;
- as_oq: pesquisa por qualquer uma das palavras atribuídas a este parâmetro;
- as_eq: pesquisa por resultados que não contenham as palavras atribuídas a este parâmetro
- q: pesquisa pela palavra ou conjunto de palavras atribuídas a este parâmetro;
- pesquisa por Intervalo de tempo:
 - as_nlo: desde a data ou hora atribuída a este parâmetro;
 - as_nhi: até a data ou hora atribuída a este parâmetro;

Por exemplo, se a intenção for pesquisar exatamente a frase “formação profissional em saúde no Porto”, o URL resultante será algo como:

```
https://www.google.pt/search?  
as_q=as_epq=forma%C3%A7%C3%A3o+profissional+sa%C3%BAde  
as_oq=  
as_eq=  
as_nlo=  
as_nhi=  
lr=  
cr=
```

```

as_qdr=all
as_sitesearch=
as_occt=any
safe=images
as_filetype=
as_rights=#as_qdr=all
q=%22forma%C3%A7%C3%A3o+profissional+em+sa%C3%BAde%22++Porto

```

Outro motor de pesquisa muito utilizado hoje em dia, é o motor de pesquisa da Microsoft Bing. Esta plataforma faz também a pesquisa através de parâmetros enviados por URL e permite a pesquisa de endereços com tecnologia RSS (Microsoft Developer Network (Microsoft) 2016).

```

http://www.bing.com/search?
q=feed:formação+medicina+porto+-alegre
go=Procurar
qs=ds
form=QBRE

```

Este tipo de funcionalidade de pesquisa vai permitir alimentar o motor de *feeds* do sistema.

2.4.2 Aplicações de web scrapping

Web scrappers são algoritmos utilizados para extrair dados de páginas de internet, através de uma análise a estas páginas e extraíndo dados relevantes, como por exemplo, o interesse em recolher automaticamente o calendário de eventos culturais de uma localidade, por forma a disponibilizá-los numa outra página de internet (Cording 2011). O aspeto e código fonte de uma página de internet é normalmente volátil, o que obriga a uma atualização manual do *scraper* (Penman et al. 2009).

Este subcapítulo apresenta algumas soluções existentes para *web scrapping*.

2.4.2.1 Chickenfoot

A aplicação *chickenfoot* é uma extensão disponível para o explorador de internet Mozilla Firefox, e baseia-se numa biblioteca que estende o motor de *javascript* do explorador de internet. Esta aplicação permite ao utilizador utilizar uma série de comandos para automação de internet, e também permite desenvolver alguns scripts no próprio browser em *javascript*. O funcionamento de pesquisa do *chickenfoot* utiliza a capacidade do *javascrit* de manipular ou moldar conteúdos de páginas de internet, através de *Document Object Model (DOM)*.

Esta aplicação funciona através de comparação de padrões. O *chickenfoot* aceita dois tipos de padrões: palavra chave (*keyword patterns*) e constantes de texto (*text constraint patterns*).

Aplicações e ferramentas de pesquisa e classificação

O padrão palavra chave, consiste numa lista de palavras que são pesquisadas na página com o objetivo de localizar o componente da *DOM*, recolhendo assim a informação do componente, como o tipo e a descrição do mesmo.

O padrão constante de texto, combina a biblioteca de padrões primitivos do *javascript* (i.e., *link*, *textbox*), *strings* e operadores relacionais (i.e., *in*, *contains*, *just before*, *just after*, *starts*, *ends*). Este tipo de padrões são normalmente usados para identificar partes da página para modificação e extração (Bolin et al. 2005).

2.4.2.2 Piggy Bank

Esta aplicação é também uma extensão do explorador de internet *Mozilla Firefox* e utiliza semântica de internet. A ideia nesta aplicação é permitir aos utilizadores introduzirem *scripts* de *scrapping* juntamente com expressões regulares em endereços URL que sejam relevantes. Depois quando o utilizador navegar para uma página de internet semelhante, o *Piggy Bank* corre os *scripts* de comparação de padrões e apresenta a informação retirada dos dados semânticos da página (Web et al. 2013).

2.4.2.3 Scrubyt

Esta aplicação utiliza bibliotecas de Ruby e funciona por pesquisa e comparação de *strings* na página de internet. O utilizador ao introduzir uma *string* no *Scrubyt* este irá pesquisar pela mesma na página e, caso a encontre, irá recolher todos os objetos similares dessa mesma página. Este tipo de aplicação pretende extrair listas de informação (i.e., listas de produtos) mas no entanto, permite controlar que grupos de extração e não se limita apenas a links (Web et al. 2013).

2.4.2.4 WWW-Mechanize

Esta aplicação é uma biblioteca de Perl que simula uma sessão de internet num explorador de internet. Suporta a utilização de cookies, preenchimento de formulários, navegação de links, no entanto como não permite *Javascript*, não suporta recolha de alto nível. É, no entanto, uma boa solução para recolha parcial de informação de páginas de internet, devido à capacidade de navegação entre links (Bolin et al. 2005).

2.4.2.5 TemplateMaker

Esta aplicação é uma biblioteca de *Python* e tem uma aproximação diferente das ferramentas enumeradas anteriormente. O *TemplateMaker* testa através de um conjunto de páginas de internet exemplo, e examina as diferenças entre o HTML de cada página de internet para

determinar que conteúdos são estáticos e dinâmicos. O conteúdo dinâmico é assumido como de interesse na página de internet logo, este deve ser recolhido. É desta forma que o *TemplateMaker* gera um modelo de recolha de dados (Penman et al. 2009).

O *TemplateMaker* é uma ferramenta interessante que tem uma abordagem nova ao *web scrapping*. Necessita apenas de um conjunto de endereços URL de exemplo para aprender o que recolher (criação do modelo de recolha), o que torna simples e automático uma nova aprendizagem se atualizarmos a lista de endereços exemplo. No entanto, o *TemplateMaker* não tem grandes desempenhos em recolha de páginas de internet muito pesadas, criando problemas de performance (Penman et al. 2009).

2.4.3 Serviço Web

Nos anos 80 e 90, com o desenvolvimento de tecnologias de comunicação, surgiu a possibilidade de sistemas independentes poderem comunicar entre si através de infraestruturas de rede informática. Esta possibilidade impulsionou o aparecimento do *Service Oriented Architecture (SOA)*.

SOA é uma infraestrutura de software conectado através de serviços web de vínculos dinâmicos e com baixo acoplamento (Nahon 2011), ou seja, esta infraestrutura permite que dois ou mais sistemas informáticos possam trabalhar em conjunto. No entanto, a ligação entre eles não é exclusiva, qualquer um dos sistemas pode desvincular-se do outro ou substituí-lo, manualmente ou automaticamente.

Atualmente o *SOA* é dominado por dois protocolos *Simple Object Access Protocol (SOAP)* e *Representational Stateless Transfer (REST)*, não sendo este um protocolo, mas sim uma arquitetura baseada em *HTTP*.

2.4.3.1 SOAP – Simple Object Access Protocol

O *SOAP* é um protocolo de comunicação baseado em *XML*. Permite o envio de mensagens entre dois ou mais sistemas conectados entre si com uma estrutura conhecida por ambas as partes. As mensagens *SOAP* tem uma estrutura bem definida, e são encapsuladas num "envelope" *SOAP* que contém um cabeçalho que pode ainda conter informações de encriptação ou roteamento. O *SOAP* é usualmente transmitido por *HTTP*, mas permite que a comunicação seja feita por outros protocolos como *FTP*, *SMTP* entre outros.

Aplicações e ferramentas de pesquisa e classificação

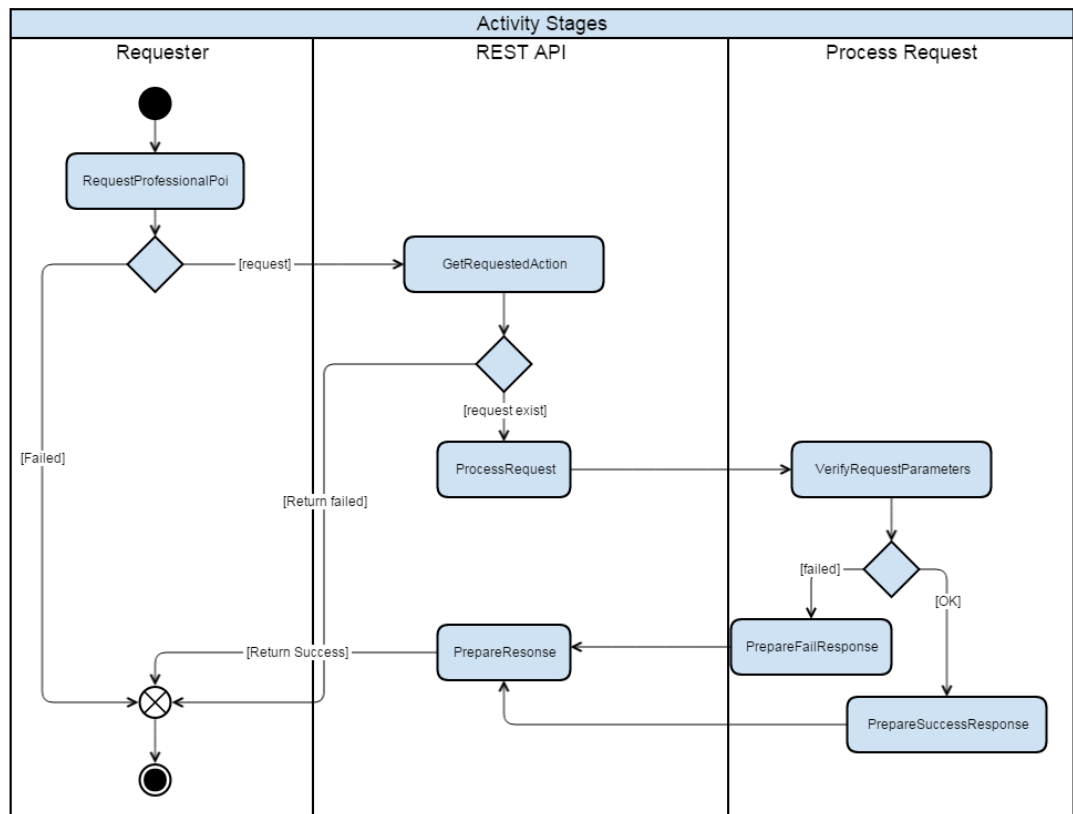
Uma das maiores características do *SOAP* é a utilização de *Web Service Definition Language (WSDL)*. *WSDL* define, em formato *XML*, os serviços web como *endpoints*. Em *SOAP* o *WSDL* é utilizado para descrever o conteúdo que o serviço web *SOAP* fornece, ou seja, nome dos métodos disponíveis, parâmetros de entrada e tipo e estrutura da resposta (Nahon 2011). O Código 1 que se segue, representa o *standard* de um envelope *SOAP*.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://schemas.conversesolutions.com/xsd/dmticta/v1">
  <soapenv:Header/>
  <soapenv:Body>
    <v1:GetProfessionalPoi>
      <v1:poiCategory>?</v1: poiCategory >
      <v1:date>?</v1: date >
    </v1: GetProfessionalPoi >
  </soapenv:Body>
</soapenv:Envelope>
```

Código 1 – Representação de envelope *SOAP* (Nahon 2011)

2.4.3.2 *REST – Representational Stateless Transfer*

Embora o *REST* utilize *XML*, utiliza-o de uma forma completamente diferente. O *REST* é uma arquitetura baseada exclusivamente no protocolo *HTTP* sendo este o meio utilizado para comunicar e navegar. A navegação num serviço web *REST* é feita com base em atalhos fornecidos nas mensagens de resposta. Contrariamente ao *SOAP* com o *WSDL*, o *REST* não tem uma definição ‘oficial’ de mensagens estruturada (Nahon 2011). A Figura 4 representa o processo de um pedido a um serviço web *REST*.

Figura 4 – Pedido Serviço *web REST*

2.4.4 Bases de dados

Para motor de bases de dados, uma vez que inicialmente o projeto iria integrar-se com o sistema de recomendação *ToursPlan*, sendo que o seu motor de bases de dados utiliza LINQ ligado a uma base de dados SQL (Almeida 2006). Assim sendo, por forma a manter compatibilidade em caso de se pretender no futuro uma migração para o motor de gestão de base de dados do *ToursPlan*, decidiu-se optar por um motor de base de dados relacional. Sendo os motores escolhidos, *SQLite*, *Mysql*, *Oracle* e *Microsoft SQL Server*.

Segundo o estudo feito pelos autores do documento científico “*A performance evaluation of open source graph databases*” as características principais destes motores de bases de dados são apresentadas na Figura 5.

Conclusão

	Owner / Maintainer	License	Platform	Language	Distribution	Cost	Transactional	Memory-based	Disk-based	Single-Node	Distributed	Graph Algorithms	Text-Based Query Language	Embeddable Software	Data Store	Type	
MySQL	Oracle	GPL/Proprietary	x86	C/C++	Bin	Free	X	-	X	X	-	-	SQL	X	X	X	SQLDB
Oracle	Oracle	Proprietary	x86	C/C++	Bin	\$180-\$950	X	-	X	X	X	-	SQL	X	X	X	SQLDB
SQL Server	Microsoft	Proprietary	x86-Win	C++	Bin	\$898-\$8592	X	-	X	X	-	-	SQL	-	X	X	SQLDB
SQLite	Richard Hipp	Public Domain	x86	C	Src/Bin	Free	X	X	X	X	-	-	SQL	X	X	X	SQLDB

Figura 5 – Estudo comparativo de motores de base de dados SQL (McColl et al. 2014)

Tendo em conta os custos, licenças e plataforma optou-se pela utilização de SQLite ou MySQL.

Tabela 3 - Comparativo tecnológico de base de dados

BD	Licença	Custo	Baseado memória	Escrita disco	Nó único	Distribuído	Tipo
<i>MySql</i>	<i>GPL</i>	<i>gratuito</i>	<i>Não</i>	<i>Sim</i>	<i>Sim</i>	<i>Não</i>	<i>SQLDB</i>
<i>SQLite</i>	<i>publico</i>	<i>gratuito</i>	<i>Sim</i>	<i>Sim</i>	<i>Sim</i>	<i>Não</i>	<i>SQLDB</i>

Comparando então ambos os motores de bases de dados na Tabela 3, temos que qualquer uma destas opções cumpre os requisitos que pretendemos para o projeto. Sendo que outras características como plataforma, distribuição, linguagem de programação, entre outras apresentadas na Figura 5, são semelhantes entre ambos, a escolha do motor de bases de dados irá recair sobre um estudo de performance. Que será efetuado no capítulo 4 desta tese.

Após alteração de requisitos devido à falta de acesso a inserção de dados no *ToursPlan*, e o desenvolvimento estar a decorrer, optou-se por manter o motor de base de dados.

2.5 Conclusão

Este capítulo teve como objetivo pesquisar soluções viáveis e para o objetivo proposto. nomeadamente a pesquisa de soluções existentes que cumpram as exigências dos requisitos do presente projeto. Teve também o propósito de reunir várias tecnologias existentes por forma a cumprir os mesmos requisitos em caso da necessidade do desenvolvimento da aplicação. Um estudo do funcionamento dos parâmetros de pesquisa de alguns motores de pesquisa conhecidos. O interesse de se inserir um *web scrapper*, que permite recolher informação necessária à criação de pontos de interesse de eventos formativos, de endereços de páginas de internet previamente definidos. Um estudo de tecnologias existentes à criação de serviços de internet, e também um estudo de tecnologias de motores de bases de dados relacionais. Todas essas técnicas, aplicações e tecnologias foram descritas com base em documentação científica.

3 Análise de valor

Este subcapítulo tem o propósito de definir o valor de produto e definir o contributo do mesmo ao mercado. Irá também existir neste subcapítulo o modelo conceptual de *canvas*.

3.1.1 Proposta de valor

Segundo o autor do documento científico “*Assets management - A conceptual model decomposing value for the customer and quantitative model*”, uma proposta de valor é “uma visão global do conjunto de produtos e serviços que providenciam valor para o cliente” (Nicola et al. 2013). No entanto, nem todos os produtos ou serviços fornecidos por uma empresa, tem o mesmo significado ou importância do ponto de vista dos clientes. Então é de extrema importância, na construção de uma proposta de valor, identificar quais os fatores que determinam importância para o cliente e como é este valor percebido, de acordo com os benefícios que este obtém, e os sacrifícios que tem que fazer para o obter (Nicola et al. 2014).

Também segundo o mesmo autor, o propósito de uma proposta de valor é “definir uma estratégia específica para competir por novos clientes”, tendo esta a função de definir o valor para o cliente, baseado num balanço entre benefícios do produto e sacrifícios que o cliente terá que fazer para o obter (Nicola et al. 2013).

Sendo um dos principais objetivos de uma boa proposta de valor, creditar um produto ou serviço no mercado. Assim sendo, uma proposta de valor bem conseguida, deve evidenciar por forma de escrita, imagem ou som, uma sensação de confiança e bem-estar ao utilizar o

Conclusão

produto/serviço fornecido. Uma forma de o conseguir passa por responder a 4 questões que definem valor, sendo estas:

- O que é o produto?
- Quem é o cliente alvo? Para quem iremos fornecer o produto?
- Qual o valor que fornecemos?
- O que torna o nosso produto único? (Nicola 2016)

3.1.1.1 Necessidade de uma proposta de valor

O processo de desenvolver uma proposta de valor é a melhor forma de uma organização definir valor, e o seu cálculo é feito através da diferença entre benefícios e sacrifícios para o cliente.

Por forma a desenvolver uma boa proposta de valor, o fornecedor deverá ter bem definido o que é o produto/serviço fornecido, cliente alvo, benefícios e sacrifícios para o cliente, contributo para a comunidade. Tendo estes componentes definidos, é possível desenvolver uma proposta de valor sustentável ao cliente, e por consequência, a principal forma de obter valor é através de valor proposto (Barnes et al. 2009).

3.1.1.2 O produto

O produto proposto nesta dissertação, inicialmente era a adição de uma subcategoria de tipo de turismo, no sistema de recomendação turístico *ToursPlan*. O do turismo profissional. Por falta de acesso ao Sistema de recomendação, o nosso produto deixou de inserir pontos de interesse no *ToursPlan*, mas permite a sistemas de recomendação obterem os pontos de interesse através de um serviço web. Este tipo de turismo é normalmente feito por funcionários ou empresários, e consiste em comparecer em eventos orientados às suas áreas profissionais, por forma a adquirir novo conhecimento que necessitam para melhor exercerem as suas funções.

No entanto, o segmento de turismo profissional que se pretende atingir nesta tese é o turismo profissional para profissionais de saúde. Assim sendo, esta subcategoria, deverá conter pontos de interesse baseados em eventos na área da saúde, como congressos, conferências, lançamentos de novos produtos ou formações.

Estes pontos de interesse deverão ser apresentados em sistemas de recomendação, através de configurações de pesquisa por parte do turista.

3.1.1.3 O cliente alvo

Sendo o produto final desta tese o turismo profissional para profissionais de saúde, o cliente que se pretende atingir com o serviço serão os profissionais ou estudantes da área de saúde, não descartando possíveis entusiastas da área que pretendam adquirir conhecimento na mesma.

No entanto, sendo o conteúdo prático do serviço desta tese fornecer pontos de interesse à subcategoria de turismo profissional para profissionais de saúde, os clientes serão os sistemas de recomendação.

3.1.1.4 Valor do produto

O valor de um produto pode ser quantificado em vários fatores, sendo eles: qualidade, design, usabilidade, customização, conveniência, entre outros (Nicola et al. 2013).

O produto que se pretende oferecer tem um valor intangível para os clientes dos sistemas de recomendação, uma vez que o objetivo dos seus clientes ao comparecerem nos pontos de interesse fornecidos, é adquirir conhecimento ou informações científicas na área da saúde. Outra grande vantagem para esses clientes, será a facilidade de planejarem rotas turísticas simultaneamente com atividades de turismo profissional, com acesso a eventos na área da saúde sob a forma de pontos de interesse. Sendo este o objetivo do serviço, serão fornecidos aos sistemas de recomendação turísticos interessados, pontos de interesse baseados em eventos da área da saúde, como congressos, conferências, lançamentos de novos produtos e formações.

A qualidade do nosso produto será calculada através da relação custo benefício das propostas apresentadas ao cliente final dos sistemas de recomendação e pelo balanço entre qualidade, prestígio, preço, durabilidade da formação, tempo despendido na pesquisa por pontos de interesse, inclusão social e se a formação tem algum componente adaptável às necessidades do cliente. Isto irá definir o valor dos conteúdos fornecidos, uma vez que os mesmos terão que causar interesse por forma a serem consumidos.

Uma vez também que o propósito é a formação em período de férias, estas formações deverão ser de curta duração, por forma a não retirar demasiado tempo ao cliente.

Conclusão

3.1.1.5 Valor percebido

Valor percebido é medido em função da utilidade do produto para o cliente, e por rácio de benefícios esperados face aos sacrifícios necessários, pela qualidade e esforço de aquisição.(Nicola et al. 2013)

A definição de valor percebido, varia de acordo com o tipo de produto ou serviço. No entanto, a definição de valor percebido é mensurável sobre perspetivas comportamental e utilitária. Sobre uma perspetiva comportamental, a definição de valor percebido é a " avaliação global da utilidade de um produto com base em perceções do que é recebido face ao que é fornecido". De uma perspetiva utilitária , valor percebido é calculado em função da utilidade, aquisição, e transação (Yang et al. 2014).

O serviço fornecido, oferece informação sobre eventos formativos na área da saúde, com possibilidade de planeamento dos mesmos. Sendo este serviço ainda raro, potencia ainda mais o valor percebido por parte do cliente.

3.1.1.6 O que destaca o produto dos restantes

O que destaca o nosso serviço dos restantes, é o enfoque nos pontos de interesse do turismo profissional em saúde. Os sistemas de recomendação terão disponível para o turista profissional de saúde uma vasta gama de eventos da área de saúde para escolher como ponto de interesse. Sendo que após um estudo de algumas aplicações turísticas semelhantes, estas não contemplam este tipo de ponto de interesse.

3.1.2 Valor para o cliente (VC)

Conforme já dito anteriormente, o valor do produto oferecido é quantificado num balanço entre benefícios e sacrifícios para o cliente antes, durante e após na aquisição do produto ou serviço (Nicola et al. 2013). A forma como um cliente avalia um produto ou serviço varia de cliente para cliente. Esta variação de valor percebido é tão frágil que a perceção de valor desse produto ou serviço varia mediante uma perspetiva longitudinal. A perceção de valor que o cliente tem do produto ou serviço sofre alterações ao longo do processo de aquisição e utilização. Um cliente tem uma ideia do valor do item antes de o comprar, a mesma pode variar no momento que a compra, por diversos fatores, como o preço ou a construção. A perceção de valor pode também alterar após a compra e também após a utilização do produto ou serviço.

A Tabela 4 define os valores de benefícios e sacrifícios para o cliente, identificando os elementos necessários para definir valor de acordo com a perspectiva do cliente.

Tabela 4 - Balanço sacrifícios e benefícios para valor do cliente

<i>Benefícios</i>		<i>Sacrifícios</i>
<i>Atributos</i>	<i>Resultados</i>	
<i>Qualidade do serviço</i>	<i>Utilidade</i>	<i>Tempo despendido</i>
<i>Performance</i>	<i>Função</i>	<i>Energia despendida</i>
<i>Usabilidade</i>	<i>Design</i>	<i>Esforço</i>
<i>Customização</i>	<i>Benefícios emocionais</i>	<i>Custos não monetários</i>
<i>Raridade</i>	<i>Conveniência</i>	
	<i>Apreciação por utilizadores</i>	
	<i>Benefícios Sociais</i>	
	<i>Segurança</i>	

Como demonstra a Tabela 4, estão identificados os atributos e resultados dos benefícios com vista ao cliente final, e também os seus sacrifícios. A coluna de atributos apresenta as qualidades que o serviço oferece ao cliente, como vantagem.

Sendo a qualidade do serviço um fator de peso na escolha do cliente por determinados, é importante fornecer pontos de interesse de centros de congressos e formação de renome e conceituados.

3.1.3 Contributo para a comunidade

A contribuição deste projeto à comunidade, passa por facilitar o acesso a formações ou eventos na área de saúde, como ponto de interesse, ou seja, possibilita ao cliente planear alguns pontos de interesse em formação durante o seu período de férias, ou em viagens profissionais. O tipo de pesquisa feito para encontrar este tipo de ponto de interesse passa por aceder a um dos sistemas de recomendação e escolher a subcategoria de turismo formativo na área da saúde numa localização à escolha, e ser-lhe-ão apresentadas propostas formativas nessa área. Isto irá reduzir em muito o stress da pesquisa manual, bem como o tempo de planeamento. Outro contributo para a comunidade passa por fornecer informação sobre aos profissionais de saúde para melhor desempenharem as suas funções.

Conclusão

3.1.4 Cenário de negócio

Tendo em conta o objetivo desta tese, o fornecimento de pontos de interesse formativos, o tipo de negociação que se adequa é a negociação integrativa, que aborda um cenário de negócio *win – win*. Este tipo de negociação adequa-se ao modelo de negócio, uma vez que o que se pretende é apenas recolher informações sobre os conteúdos dos eventos formativos e fornecer pontos de interesse aos Sistemas de recomendação. É um valor derivado do consumo, intrínseco, não monetário.

O cenário *win – win* adequa-se, uma vez que os custos de recolha de dados relevantes à geração dos pontos de interesse, são nulos, e ao mesmo tempo, para os fornecedores adquirem uma forma nova e gratuita de promover os seus eventos, com informação presente nas suas páginas de internet.

3.1.5 Modelação de Valor

Por forma a modelar corretamente a criação de valor, fez-se um pequeno estudo de métodos quantitativos existentes, chegando aos seguintes:

- Teoria dos conjuntos aproximativos– TCA de RoMan Slowinski (Pawlak 1982)
- Teoria dos conjuntos nebulosos (Fuzzy Set Theory) de Lotfi Zadeth (Harris 1999)
- Teoria dos jogos (Erhun & Keskinocak 2003) (Erhun & Keskinocak 2003)
- Apoio Multicritério – análise multicritério de apoio à decisão: (AMD) (Greco et al. 2001)

3.1.5.1 Teoria dos conjuntos aproximativos

A Teoria dos conjuntos aproximativos deriva da teoria dos conjuntos, e a sua função é o tratamento da imprecisão de dados, por aproximação de um conjunto de dados (Pessoa & Simões 2003). Foi inicialmente desenvolvida por Zdzislaw Pawlak no início da década de 80, e tem como ponto inicial, a relação de "indiscernibilidade" (indiscernibility), isto é, aquela que identifica os objetos com a mesma propriedade (Pessoa & Simões 2003).

3.1.5.2 Teoria dos conjuntos nebulosos (Fuzzy Theory)

A teoria tem como finalidade processar informações subjetivas, de natureza vaga e incerta, da linguagem natural, possibilitando a modelação de conceitos subjetivos. Assim, como a Teoria Clássica dos Conjuntos (Crisp Set Theory) é o fundamento da Matemática Clássica, a Teoria dos Conjuntos Nebulosos (TCN) é o fundamento de toda Teoria de Sistema Nebuloso (TSN) e consiste numa extensão da Teoria Clássica de Conjuntos (Harris 1999).

Na teoria dos conjuntos clássica, a associação de elementos em um conjunto é avaliada em termos binários, um elemento ou pertence ou não pertence a um conjunto. A teoria dos conjuntos nebulosos pode ser vista como uma extensão da teoria clássica de conjuntos. Ela usa um lógica difusa que calcula o grau de pertença a um conjunto, por exemplo, num conjunto “pessoas altas”, um elemento com 180cm pode ou não ser considerado alto, sendo que na teoria dos conjuntos clássica, o elemento era [0,1], não pertence ou pertence, na Teoria dos conjuntos nebulosos pode ser 0,75 (Tanscheit 2012).

3.1.5.3 Teoria dos jogos

Este método, tem como função ajudar a analisar e compreender o comportamento de vários agentes que interagem entre si enquanto tomam decisões. É uma ferramenta de análise poderosa em situações onde os agentes pretendem maximizar os seus ganhos, através de estratégias de negócio, e onde os seus ganhos dependem das decisões dos outros agentes (Erhun & Keskinocak 2003).

3.1.5.4 Análise multicritério de apoio à decisão

Os métodos de decisão multicritério são ferramentas de comparação que recorrem a técnicas numéricas para auxiliar a tomada de decisão através de um conjunto de alternativas. O processo de decisão é baseado nas alternativas encontradas em cruzamento com os critérios levantados para a resolução do problema (Costa et al. 2012).

O método Análise Multicritério de Decisão permite a priorização de alternativas numa situação de critérios conflituosos, procurando satisfazer as restrições, com objetivos conflituantes, ou seja, uma solução de compromisso. Assim sendo, o AMD pode fornecer métodos para o apoio à negociação e/ou decisão em grupo (Buchanan & Gardiner, 2003). Alguns métodos de análise multicritério conhecidos são (Costa et al. 2012):

Conclusão

- AHP (Analytic hierarchy process)
- MACBETH (Measuring attractiveness by a categorical based evaluation technique)
- PROMETEE (Preference ranking organization method for enrichment evaluations)
- ELECTRE (Elimination et choix traduisant la réalité)

3.1.5.5 Análise de valor modelo conceptual

O modelo *Conceptual Model for Decomposing the Value for the Customer (CMDVC)*, é uma ferramenta de análise de valor percebido pelo cliente, baseada no modelo quantitativo de benefícios e sacrifícios por parte do cliente e da empresa fornecedora. O modelo conceptual distingue a relação entre ativos tangíveis e intangíveis endógenos e exógenos utilizado pela empresa para fornecer valor ao cliente. Este modelo faz um levantamento de valor percebido para o cliente nestas 4 fases temporais, baseadas em formas quantitativas de valor (Nicola et al. 2012):

- Net VC – balanço entre benefícios e sacrifícios que irá fornecer os bons ou maus resultados em termos de valor para o cliente;
- Marketing VC – atributos percebidos pelo cliente face ao produto ou serviço;
- Sale VC – relacionado com o preço do produto ou serviço;
- Rational VC - valor racional que o cliente tem do produto ou serviço;
- Derived VC – experiência do cliente na interação com o produto ou serviço
- Estas formas de valor serão aplicadas no modelo conceptual em 4 fases temporais distintas: (Nicola et al. 2012)
- Ex Ante VC - definição de valor do cliente antes da compra do serviço ou produto;
- Transaction VC - definição de valor para o cliente no momento da aquisição do produto ou serviço;
- Ex Post VC - definição de valor do cliente após a compra do serviço ou produto;

- Disposition - definição de valor do cliente após experimentar o serviço ou produto;(Nicola et al. 2012)

3.1.5.6 Modelo de negocio Canvas

A Tabela 5 contém o modelo de negócio de Canvas que serve de ferramenta de gestão estratégica, que permite desenvolver e esboçar modelos de negócio novos ou existentes (Osterwalder 2004).

Canais de acesso:

Sendo o objetivo desta tese a geração de pontos de interesse de turismo profissional, os clientes interessados em aceder a esta informação necessitam de uma interface de acesso aos mesmos. Essa interface serão os sistemas de recomendação parceiros, que providenciam pontos de interesse turísticos na cidade do Porto.

A forma de adquirir informação, será através de um algoritmo de pesquisa e recolha de dados na Internet, com consequente criação de pontos de interesse da categoria de turismo profissional.

Tabela 5 – Poi engine- modelo de negócio de Canvas

		Projeto: Poi engine		Data: 02/01/2016
<i>Parcerias</i> <ul style="list-style-type: none"> Universidades; Centros de formação; Centro de congressos; Sítios da internet; 	<i>Atividades</i> <ul style="list-style-type: none"> Programação; Componente de pesquisa e recolha de informação referente a eventos formativos; Tratamento de dados e consequente geração de pontos de interesse. Fornecimento dos pontos de interesse a Sr's 	<i>Proposta de valor</i> <ul style="list-style-type: none"> Sistema de recomendação de eventos formativos na área da saúde; Pontos de interesse baseados em pesquisa; Planeamento dos eventos, em conjunto com reservas em restauração e hotelaria; Planeamento de pontos de interesse turísticos; 	<i>Relacionamento com Clientes</i> <ul style="list-style-type: none"> Um portal de pesquisa por pontos de interesse em eventos formativos na área da saúde; Pontos de interesse de acordo com a pesquisa pretendida; Boa experiência de utilização; 	<i>Segmento de mercado</i> <ul style="list-style-type: none"> Profissionais das áreas da saúde; Entusiastas de formação em áreas da saúde; Clientes interessados em medicinas alternativas;
	<i>Recursos</i> <ul style="list-style-type: none"> Informação digital referente a cursos e congressos na área da saúde. Estudo e conhecimentos adquiridos para implementação da solução proposta; Equipa de programação 		<i>Canais</i> <ul style="list-style-type: none"> Através da utilização do sistema de recomendação Eventos formativos na área da saúde; Algoritmo de pesquisa e recolha de informação; Algoritmo de tratamento de dados e geração de POI's; 	
<i>Custos estruturais</i> <ul style="list-style-type: none"> Tempo de pesquisa científica; Tempo de desenvolvimento; Custo de manutenção de servidores; 		<i>Fontes de receita</i> <ul style="list-style-type: none"> Pontos de interesse formativos na área pesquisada; Congressos na área de medicina com oradores de renome; Eventos formativos em estabelecimentos de ensino de renome; 		

3.1.6 Conclusão

Este capítulo teve como objetivo fazer uma análise do valor do projeto para a comunidade de turismo profissional. A informação pretendida à criação de sugestões, é pública e encontra-se disponível em páginas de internet, podendo ser recolhida de forma gratuita. No entanto, essas páginas de internet têm, na sua maioria, um interesse comercial nos eventos que querem promover. Esta aplicação irá também ajudar a publicitar estes eventos a uma maior quantidade de possíveis clientes.

Neste capítulo, foi também feito um estudo do valor do projeto para os clientes, sendo que estes são os sistemas de recomendação. Esta aplicação vai beneficia-los, uma vez que passam a abranger um novo mercado de turismo com informação acessível de forma gratuita.

Uma aplicação com a potencialidade deste projeto irá ajudar a facilitar o planeamento e a pesquisa de pontos de interesse de turismo profissional, ajudando assim, indiretamente, os profissionais das áreas abordadas a adquirir conhecimentos para melhor executarem as suas funções profissionais.

4 Avaliação

Neste capítulo apresentam-se e justificam-se as tecnologias e abordagens que irão ser usadas. Será feito um estudo estatístico de algumas das tecnologias abordadas no estado da arte, por forma a determinar quais mais vantajosas para o projeto em desenvolvimento.

O estudo a fazer neste capítulo cairá sobre a escolha da base de dados, que no caso deste projeto, será uma base de dados SQL para manter compatibilidade com a base de dados do sistema de recomendação *Tours Plan*, visto que inicialmente havia a possibilidade de uma possível integração do componente POI engine com o sistema de recomendação. Das bases de dados SQL existentes as escolhidas foram MySQL e SQLite, devido a serem gratuitas e compatíveis com o motor de bases de dados SQL implementado no Sistema de recomendação.

4.1 Medidas de avaliação

A grandeza a avaliar neste estudo, será a nível de performance de escrita em bases de dados. O estudo apresentado neste capítulo é baseado no estudo feito e publicado na página oficial do SQLite (SQLite 2016). O referido estudo foi efetuado para três motores de bases de dados, embora, para o propósito deste trabalho apenas nos interessem os valores do MySQL e o SQLite.

Neste estudo a grandeza estudada é o tempo de escrita numa base de dados.

4.2 Hipóteses e metodologias de avaliação

O estudo em análise foi feito na pela SQLite e documentado na sua página de internet (SQLite 2016). Para o propósito deste trabalho iremos analisar os resultados obtidos em testes de escrita nos motores de bases de dados *SQLite 2.7.6* e *MySQL 3.23.41*.

4.2.1 Condições do ambiente de testes

Os testes realizados neste estudo não tiveram como objetivo testar performance de escrita em múltiplos acessos em simultâneo. As *queries* realizadas à base de dados não têm elevado grau de complexidade e não pressupõem a utilização de vários *join* ou *subqueries*.

A base de dados usada neste estudo é relativamente pequena com aproximadamente 14 MB.

O ambiente de testes, não prevê o desempenho do motor de bases de dados em condições extremas.

4.2.2 Ambiente de teste

Os testes foram executados num servidor com processador 1.6GHz *Athlon* com 1GB de memória e um disco IDE. O sistema operativo utilizado foi o *RedHat Linux 7.2* com um stock *Kernel*.

Os servidores de base de dados *MySQL* utilizados foram os instalados com as configurações por defeito para o sistema operativo utilizado, sendo a versão usada já referida. Uma característica interessante da configuração por defeito do servidor *MySQL* no *RedHat 7.2* é não suportar transações, o que lhe confere uma vantagem ao nível da velocidade. A configuração do *SQLite* foi mantida por defeito por forma a não manipular o resultado do teste. Foi compilado com otimização `-O6` e com a configuração `-DNDEBUG a1`, o que desabilita as excessivas instruções de *assert* do *SQLite* a sincronização no *SQLite* com o disco foi desligada para estar em pé de igualdade com o servidor *MySQL*.

Todos os tempos recolhidos durante o teste, foram medidos em segundos.

4.2.3 Casos de teste

Neste capítulo serão apresentados os testes e, no final, os resultados obtidos.

4.2.3.1 Teste 1 – 1000 INSERTs

Neste teste, cada instrução SQL é uma transação independente, o que vai obrigar a um acesso por transação à base de dados, e consequentemente a uma ação de escrita e tratamento de cache por transação, demonstrando assim o desempenho dos motores de bases de dados em acessos contínuos.

```
CREATE TABLE t1(a INTEGER, b INTEGER, c VARCHAR(100));
INSERT INTO t1 VALUES(1,13153,'thirteen thousand one hundred fifty three');
... 998 linhas omitidas
INSERT INTO t1 VALUES(1000,94142,'ninety four thousand one hundred forty
two');
```

4.2.3.2 Teste 2 – 25000 INSERTs por transação

Neste teste, as instruções SQL são aglomeradas numa única transação, o que vai obrigar apenas a um acesso à base de dados, e consequentemente a uma ação de escrita e tratamento de cache, demonstrando assim o desempenho de uma transação de escrita de tamanho considerável.

```
BEGIN;
CREATE TABLE t2(a INTEGER, b INTEGER, c VARCHAR(100));
INSERT INTO t2 VALUES(1,59672,'fifty nine thousand six hundred seventy
two');
... 24998 linhas omitidas
INSERT INTO t2 VALUES(25000,94666,'ninety four thousand six hundred sixty
six');
COMMIT;
```

4.2.3.3 Teste 3 – 25000 INSERTs para tabelas indexadas

Neste teste, demonstra-se a performance de adição de valores à base de dados com a utilização de índices de tabelas.

```
BEGIN;
CREATE TABLE t3(a INTEGER, b INTEGER, c VARCHAR(100));
CREATE INDEX i3 ON t3(c);
... 24998 linhas omitidas
INSERT INTO t3 VALUES(24999,88509,'eighty eight thousand five hundred
nine');
INSERT INTO t3 VALUES(25000,84791,'eighty four thousand seven hundred
ninety one');
COMMIT;
```

4.2.3.4 Teste 4 – 100 SELECTs sem índices

Neste teste, são feitos 100 pedidos de dados sem índices a uma tabela com 25000 entradas, o que obriga a uma pesquisa completa da tabela por pedido.

```
BEGIN;
SELECT count(*), avg(b) FROM t2 WHERE b>=0 AND b<1000;
SELECT count(*), avg(b) FROM t2 WHERE b>=100 AND b<1100;
... 96 linhas omitidas
SELECT count(*), avg(b) FROM t2 WHERE b>=9800 AND b<10800;
SELECT count
```

4.2.3.5 Teste 5 – 100 SELECTs com comparação de texto

Neste teste, são feitos 100 pedidos de dados sem índices, obriga a uma pesquisa completa da tabela por pedido, mas utiliza comparação de texto em vez de comparações numéricas como o teste anterior, demonstrando assim a diferença de performance entre os motores de bases de dados, em comparações por tipo de dados.

```
BEGIN;
SELECT count(*), avg(b) FROM t2 WHERE c LIKE '%one%';
SELECT count(*), avg(b) FROM t2 WHERE c LIKE '%two%';
... 96 linhas omitidas
SELECT count(*), avg(b) FROM t2 WHERE c LIKE '%ninety nine%';
SELECT count(*), avg(b) FROM t2 WHERE c LIKE '%one hundred%';
COMMIT;
```

4.2.3.6 Teste 6 – Criação de índice

Este teste, demonstra o desempenho dos motores de bases de dados na criação de índices.

```
CREATE INDEX i2a ON t2(a);
CREATE INDEX i2b ON t2(b)
```

4.2.3.6.1 Teste 7 – 5000 SELECTs com índice

Neste teste, são feitos 5000 pedidos de dados sem índices, o que obriga a uma pesquisa completa da tabela por pedido, para demonstrar a performance de pesquisa de dados em tabelas.

```
SELECT count(*), avg(b) FROM t2 WHERE b>=0 AND b<100;
SELECT count(*), avg(b) FROM t2 WHERE b>=100 AND b<200;
SELECT count(*), avg(b) FROM t2 WHERE b>=200 AND b<300;
... 4994 linhas omitidas
SELECT count(*), avg(b) FROM t2 WHERE b>=499700 AND b<499800;
SELECT count(*), avg(b) FROM t2 WHERE b>=499800 AND b<499900;
SELECT count(*), avg(b) FROM t2 WHERE b>=499900 AND b<500000;
```

4.2.3.6.2 Teste 8 – 1000 UPDATEs sem índice

Neste teste, são feitos 1000 pedidos de atualização de dados, sem a utilização de índices, para demonstrar a performance dos motores de bases de dados neste tipo de ação.

```
BEGIN;
UPDATE t1 SET b=b*2 WHERE a>=0 AND a<10;
UPDATE t1 SET b=b*2 WHERE a>=10 AND a<20;
... 996 linhas omitidas
UPDATE t1 SET b=b*2 WHERE a>=9980 AND a<9990;
```

```
UPDATE t1 SET b=b*2 WHERE a>=9990 AND a<10000;
COMMIT;
```

4.2.3.6.3 Teste 9 – 25000 UPDATEs com índice

Neste teste, são feitos 25000 pedidos de atualização de dados, com a utilização de índices, para demonstrar a performance dos motores de bases de dados neste tipo de ação comparativamente ao teste anterior.

```
BEGIN;
UPDATE t2 SET b=468026 WHERE a=1;
UPDATE t2 SET b=121928 WHERE a=2;
... 24996 linhas omitidas
UPDATE t2 SET b=35065 WHERE a=24999;
UPDATE t2 SET b=347393 WHERE a=25000;
COMMIT;
```

4.2.3.6.4 Teste 10 – 25000 UPDATEs de Texto com índice

Neste teste, são feitos 1000 pedidos de atualização de dados em texto, com a utilização de índices, para demonstrar a performance dos motores de bases de dados em atualização com tratamento de texto.

```
BEGIN;
UPDATE t2 SET c='one hundred forty eight thousand three hundred eighty two'
WHERE a=1;
UPDATE t2 SET c='three hundred sixty six thousand five hundred two' WHERE
a=2;
... 24996 linhas omitidas
UPDATE t2 SET c='three hundred eighty three thousand ninety nine' WHERE
a=24999;
UPDATE t2 SET c='two hundred fifty six thousand eight hundred thirty' WHERE
a=25000;
COMMIT;
```

4.2.3.6.5 Teste 11 – INSERTs de um SELECT

Este teste, demonstra a performance de inserção de dados de uma tabela numa outra.

```
BEGIN;
INSERT INTO t1 SELECT b,a,c FROM t2;
INSERT INTO t2 SELECT b,a,c FROM t1;
COMMIT;
```

4.2.3.6.6 Teste 12 – DELETE sem índice

Este teste, demonstra a performance dos motores de bases de dados em ações de eliminação sem a utilização de índices.

```
DELETE FROM t2 WHERE c LIKE '%fifty%';
```

4.2.3.6.7 Teste 13 – DELETE com índice

Este teste, demonstra a performance dos motores de bases de dados em ações de eliminação com a utilização de índices.

```
DELETE FROM t2 WHERE a>10 AND a<20000;
```

4.2.3.6.8 Teste 14 – um INSERT grande após um DELETE grande

Este teste, é realizado após a eliminação de uma grande quantidade de dados de uma tabela, por forma a comparar ao teste 11.

```
INSERT INTO t2 SELECT * FROM t1;
```

4.2.3.6.9 Teste 15 – um DELETE grande seguido de múltiplos INSERTs pequenos

Este teste, demonstra a performance dos motores de bases de dados em ações de eliminação de grandes quantidades de dados, seguida de várias instruções de dados. Alguns motores de bases de dados tem falhas de performance neste tipo de instruções (SQLite 2016).

```
BEGIN;  
DELETE FROM t1;  
INSERT INTO t1 VALUES(1,10719,'ten thousand seven hundred nineteen');  
... 11997 linhas omitidas  
INSERT INTO t1 VALUES(11999,72836,'seventy two thousand eight hundred  
thirty six');  
INSERT INTO t1 VALUES(12000,64231,'sixty four thousand two hundred thirty  
one');  
COMMIT;
```

4.2.3.6.10 Teste 16 – DROP TABLE

Este teste, demonstra a performance dos motores de bases de dados em ações de eliminação de tabelas. Sendo que os motores de bases de dados usam técnicas diferentes de armazenamento dos dados, faz com que a eliminação completa de tabelas, seja mais penosa a uns que a outros.

```
DROP TABLE t1;  
DROP TABLE t2;  
DROP TABLE t3;
```

4.2.4 Resultados de teste

A Tabela 6 apresenta os tempos recolhidos ao efetuar os testes apresentados.

Tabela 6 – Recolha de tempos de testes a motores de base de dados

	<i>MySQL</i>	<i>SQLite</i>
<i>Teste 1</i>	0.114	0.223
<i>Teste 2</i>	2.184	0.757
<i>Teste 3</i>	3.197	1.402
<i>Teste 4</i>	2.760	2.526
<i>Teste 5</i>	4.640	3.372
<i>Teste 6</i>	0.318	0.659
<i>Teste 7</i>	1.270	1.162
<i>Teste 8</i>	8.410	0.638
<i>Teste 9</i>	8.134	3.104
<i>Teste 10</i>	6.982	1.725
<i>Teste 11</i>	1.537	1.599
<i>Teste 12</i>	0.975	0.560
<i>Teste 13</i>	2.262	0.752
<i>Teste 14</i>	1.815	1.485
<i>Teste 15</i>	1.704	0.406
<i>Teste 16</i>	0.015	0.254

4.3 Testes estatísticos e resultados

Neste estudo, não foram apresentados estudos estatísticos, no entanto apresentam algumas conclusões sobre os tempos obtidos.

SQLite 2.7.6 é normalmente mais rápido que o *MySQL* 3.23.41 na maior parte das operações comuns, por vezes com atinge o dobro da rapidez.

SQLite não executa *queries* de CREATE INDEX ou DROP TABLE tão rápido como o *MySQL*, no entanto estes tipos de operações não são tão frequentes.

SQLite tem melhor desempenho em múltiplas operações numa única transação.

5 Design da solução

O projeto proposto é um componente auxiliar a sistemas de recomendação turísticos, denominado de *POI engine*. Neste capítulo definem-se as abordagens a seguir no desenvolvimento, que padrões usar, nos diversos módulos necessários à construção deste protótipo.

5.1 *Design Conceptual*

O conceito do projeto proposto passa por construir um motor de pesquisa e classificação de pontos de interesse em ações formativas da área da saúde. Sendo o sistema proposto independente do sistema de recomendação, será um serviço externo e será constituído por vários componentes, sendo eles:

- ***web crawler***, componente responsável por requisitar endereços ao motor de pesquisa.
- ***web scrapper***, componente responsável por recolher o conteúdo dos endereços recolhidos no componente anterior, e retirar a informação não relevante.
- **motor de *feeds***, componente responsável por recolher a informação de páginas com tecnologia RSS.
- **gerador de pontos de interesse**, componente responsável por agregar a informação relevante recolhida pelos componentes anteriores, e armazena-la sob a forma de ponto de interesse.

- **portal de internet**, este componente contém funcionalidades para inserir e validar os pontos de interesse criados nele próprio e no componente anterior.

Cada um destes componentes terá uma função independente, com o objetivo de ajudar a criar pontos de interesse formativos na área da saúde. Devido à impossibilidade de se inserir pontos de interesse no sistema de recomendação inicialmente proposto, surgiu a necessidade de se implementar um serviço *web* que permite o acesso aos pontos de interesse.

O funcionamento do *POI engine* deverá começar por utilizar o componente *web crawler*, por forma a pesquisar através de um motor de pesquisa, endereços *URL* com possível informação de eventos formativos na área da saúde. Esta pesquisa será feita tanto para páginas com conteúdo *HTML*, bem como para páginas com tecnologia *RSS* disponível para download. Após recolher esses endereços, os mesmos serão armazenados na base de dados por tipo, para posteriormente poderem ser rastreados para recolha da dita informação.

Após esta recolha de endereços *URL*, a recolha de informação para gerar pontos de interesse é feita de duas formas separadas, no motor de *feeds* e no *web scrapper*.

O motor de *feeds* irá recolher da base de dados os endereços do tipo *RSS*, para poder recolher essa informação. Após essa recolha, o conteúdo *XML* devolvido é processado, e os itens recolhidos serão então enviados para o gerador de pontos de interesse, por forma a criar e armazenar na base de dados a informação como ponto de interesse.

O componente *web scrapper*, faz um processo semelhante, recolhe os endereços de páginas do tipo *HTML*, e tenta retirar, do conteúdo recolhido, toda a informação relevante à criação de pontos de interesse.

Por forma a gerar pontos de interesse com sucesso, o gerador de pontos de interesse terá que recolher a informação dos vários tipos, *HTML* e *RSS*, e estruturar essa informação para um objeto comum. Esse objeto será então armazenado na base de dados, para posteriormente poder ser fornecido aos sistemas de recomendações. Este componente será, sem dúvida, desafiante, uma vez que reconhecer a informação importante de informação *HTML* não estruturada de forma automática, irá requerer muito processamento de funções de comparação.

Outro dos problemas previstos neste projeto passa por arranjar forma de providenciar os pontos de interesse gerados a sistemas de recomendação. Neste projeto como referido no parágrafo anterior, será implementado um serviço *web* com uma *API*, que dará resposta aos vários pedidos de pontos de interesse.

Esta *API* terá o conhecimento necessário para receber e processar pedidos de aplicações externas, recolhendo assim a informação pretendida, caso a mesma exista, e retornando os resultados obtidos numa estrutura documentada.

5.2 Arquitetura

Neste subcapítulo irão ser descritos os diversos componentes a implementar no sistema *POI engine* do ponto de vista técnico. Cada um deles é composto por vários micro serviços que efetuam as tarefas do componente de forma independente.

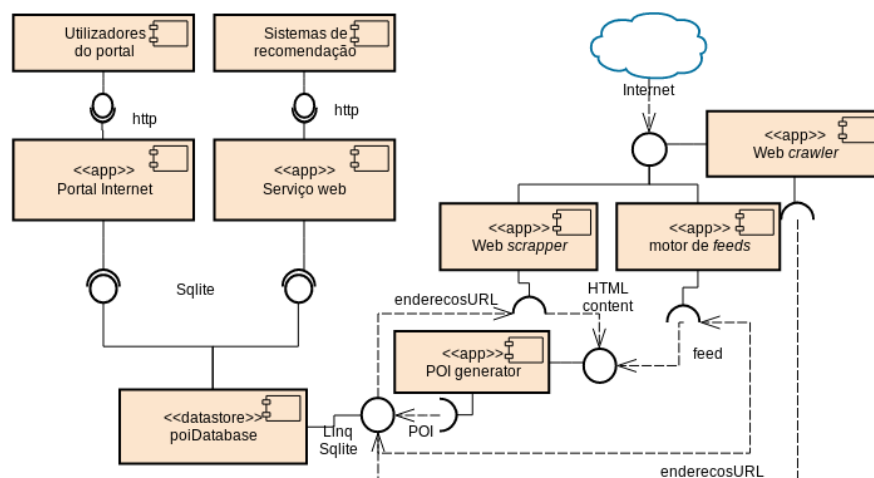


Figura 6 – Diagrama componentes da aplicação

A Figura 6 mostra a arquitetura proposta para o protótipo, sendo que, os detalhes técnicos de cada um dos componentes nele propostos são descritos nos seguintes subcapítulos.

O sistema *POI engine* é composto por vários módulos, sendo eles, o “motor de *feeds*”, o “*web crawler*”, o “*web scrapper*”, o “gerador de *POI*’s”, o “portal de internet” e o “serviço de internet”.

5.2.1 Web crawler

Este componente, conforme foi já descrito, tem a função de pesquisar na internet por endereços U com potencial informação para a criação dos pontos de interesse.

Sendo que se irá utilizar um motor de pesquisa convencional, a opção mais vantajosa para desenvolver, será a utilização do motor de pesquisa Bing, uma vez que contém uma API que permite a pesquisa por endereços de com tecnologia RSS, o que favorece diminui o processamento inerente a ter que diferenciar os conteúdos das páginas HTML ou XML. Assim sendo o funcionamento deste componente irá recolher separadamente os endereços do tipo HTML ou XML, e irá armazenar os resultados na base de dados para se analisar os seus conteúdos posteriormente. Este processo irá utilizar a biblioteca System.Net da Microsoft, que permite através de componente Webrequest, recolher o conteúdo de um endereço URL. A Figura 7 demonstra o processo do componente web crawler.

Component Diagram - Web crawler

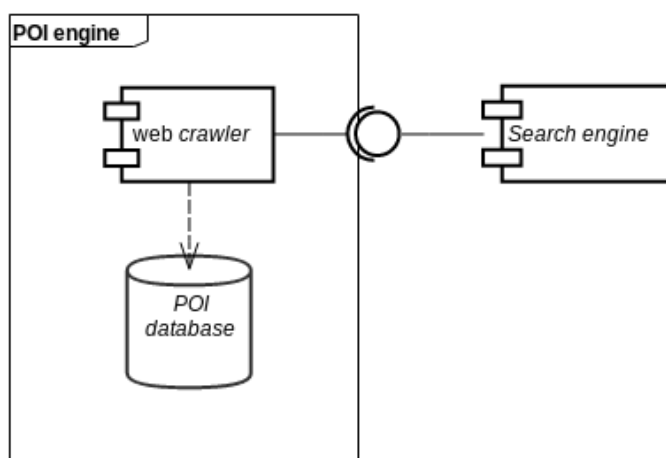


Figura 7 – Diagrama de componentes do web crawler

No entanto este pedido será diferente mediante o tipo de informação que se pretender obter, HTML ou RSS, assim sendo, a parametrização do endereço do motor de pesquisa será diferenciada mediante o tipo de pesquisa que se pretende. Sendo que para se realizar uma pesquisa por endereços com tecnologia RSS, o parâmetro q da pesquisa terá que conter no seu início o parâmetro feed seguido da query de pesquisa. No caso de se pesquisar por páginas de HTML genérico, o parâmetro de pesquisa q apenas necessitará da query, conforme demonstra o excerto de Código 2 seguinte.

```

string queryString = row.searchQuery.Replace(" ", "+");
string query = "";
switch (type)
{
    case "rss":
        query = "?q=feed%3A" + queryString + "&go=Submit&q=ds&form=QBLH";
        break;
    case "html":
    default:
        query = "?q=" + queryString + "&go=Submit&q=ds&form=QBLH";
        break;
}
string requestUrl = pageUrl + query;
HttpRequest request = (HttpRequest)WebRequest.Create(requestUrl);
HttpWebResponse response = (HttpWebResponse)request.GetResponse();

```

Código 2 – construção de endereço de pesquisa por HTML e RSS

5.2.2 Motor de feeds

Este módulo, conforme o nome indica, tem a responsabilidade de recolher informação através de um serviço RSS. Existem páginas de internet com esta tecnologia disponível, assim sendo, e uma vez que a estrutura de feeds é conhecida os dados são fornecidos numa estrutura de XML standard.

```

<item>
  <title>Curso de Auxiliar de Ação Médica</title>
  <link> http://www.formacaosaude.com/2015/02/curso-de-auxiliar/</link>
  <comments>
    http://www.formacaosaude.com/2015/02/curso-de-aux...
  </comments>
  <pubDate>Wed, 11 Feb 2015 12:56:54 +0000</pubDate>
  <dc:creator><![CDATA[ formacaosaude ]]></dc:creator>
  <category>
    <![CDATA[ Formação ]]>
  </category>
  <guid isPermaLink="false">http://www.formacaosaude.com/?p=1249</guid>
  <description>
    <![CDATA[A I9PROJECT - Formação e eLearning encontr...]]>
    <![CDATA[As com competências na intervenção ..]]>
  </description>
  <content:encoded>
    <![CDATA[<p>A I9PROJECT - Formação e eLearning encontra-se a]]>
  </content:encoded>
  <wfw:commentRss>
    http://www.formacaosaude.com/2015/02/curso-de-auxiliar-de-acao-
    medica/feed/
  </wfw:commentRss>
  <slash:comments>0</slash:comments>
</item>

```

Código 3 - Exemplo de feed de um curso (Formacaosaude.com 2016)

Arquitetura

Com este tipo de funcionalidade, pode facilmente ser recolhida a informação necessária à criação de pontos de interesse.

Sendo que a framework .Net contempla uma classe que trata este tipo de dados, a classe *SyndicationFeed*. Esta classe possui já métodos de tratamento e leitura deste tipo de dados, o que facilita a implementação deste módulo. De seguida observa-se um exemplo da leitura deste tipo de dados.

```
string url = " http://www.formacoasaude.com/categoria/formacao/feed";  
using (XmlReader reader = XmlReader.Create(url))  
{  
    SyndicationFeed feed = SyndicationFeed.Load(reader);  
    Console.WriteLine(feed.Title.Text);  
    Console.WriteLine(feed.Links[0].Uri);  
    foreach(SyndicationItem item in feed.Items)  
    {  
        Console.WriteLine(item.Title.Text);  
    }  
}
```

Código 4 – Leitura de *Feeds* com *SyndicationFeed* (Levesque 2009)

Como se pode ver nos exemplos Código 3 e Código 4 o nível de simplicidade com que se recolhe dados para a criação de pontos de interesse, é alto. Assim sendo o passo seguinte será a criação de uma mensagem no componente “gerador de POI’s” por forma a enviar o objeto para o sistema de recomendação. A Figura 8 mostra o funcionamento do componente motor de feeds.

Component Diagram - motor de feeds

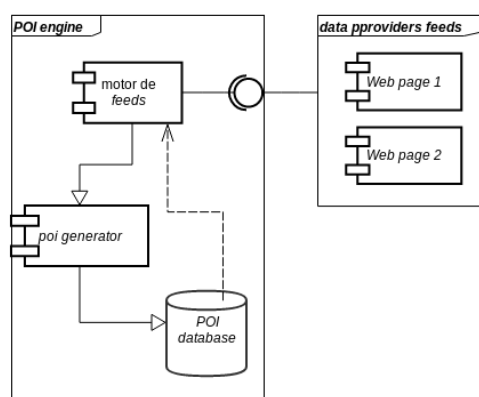


Figura 8 – Diagrama de componentes motor de *feeds*

5.2.3 Web Scrapper

Este módulo é o maior desafio deste projeto, uma vez que ao tentar recolher dados de diferentes páginas de internet, será necessário criar um motor suficientemente dinâmico para que consiga trabalhar com dados diferentes.

O *scraper* terá a função de recolher o conteúdo *HTML* das páginas de internet cujo endereço está guardado na base de dados, diferenciado por tipo. Após obter o código *HTML*, este conteúdo será analisado com funções de comparação, ou seja, serão pesquisados excertos de texto na código fonte, como datas, palavras chave (i.e., curso, formação, congresso). Caso se encontre este tipo de texto, o elemento *HTML* desse texto irá ser recolhido, bem como o elemento *HTML* pai da mesma, e será pesquisado no conteúdo do elemento *HTML* pai e filhos por toda a informação necessária à criação de um item.

Em caso de sucesso, cria-se o objeto POI para ser guardado na base de dados, e uma outra análise irá verificar os elementos *HTML* que continham a informação tentando encontrar id's, classes CSS, e o tipo de elemento *HTML* (i.e, *a*, *div*, *span*), por forma a identificar padrões para melhorar a performance do *scraper* para a página em análise. Este padrão, caso seja detetado com sucesso, irá, também ele, ser armazenado na base de dados, e será feita uma verificação inicial pelo *scraper* pela existência de um padrão conhecido antes de iniciar o processo de pesquisa. Na Figura 9 observa-se o processo de recolha de conteúdos de páginas *HTML*.

Component Diagram - Web scrapper

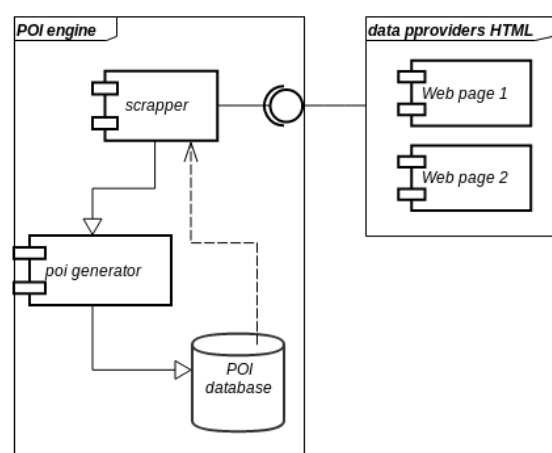


Figura 9 – Diagrama de componentes *web scrapper*

Arquitetura

5.2.4 Gerador de POI's

Este módulo é responsável por criar e armazenar na base de dados a informação que constitui um ponto de interesse. Este componente irá conter atributos que definem o ponto de interesse, sendo eles os atributos demonstrados no Código 5:

```
class Poi
{
    private string title; //titulo do ponto de interesse
    private string comments; //comentários existentes
    private DateTime date; //data de inicio do item
    private string url; //url para mais informações
    private string[] category; //categorias em que se insere (i.e., medicina)
    private string descricao; // descrição do item
    private string content; // descrição alargada do item
    private string type; // tipo de item
    ...
}
```

Código 5 – Atributos classe *POI*

5.2.5 Processo de criação de poi's

Neste subcapítulo mostra-se e explica-se o processo de criação de pontos de interesse com base na informação recolhida da internet. É abordado o exemplo de criação de pontos de interesse com base no motor de *feeds*, sendo o processo mais simples de compreender.

Diagrama de sequencia - geração de Poi's

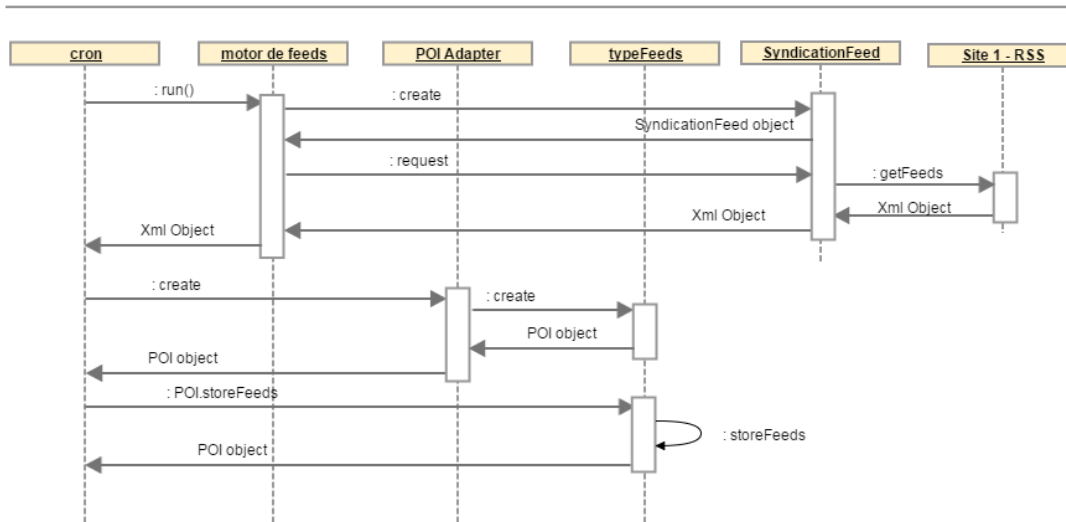


Figura 10 – Diagrama de sequência – geração de pontos de interesse

A Figura 10 representa o processo de geração de pontos de interesse pelo componente gerador de *POI's*. Como se pode ver, existe um processo inicial que irá utilizar a biblioteca *SyndicationFeed*, que permite a ligação a páginas de internet com fornecimento de *feeds RSS*. Após a recolha dos *feeds* num objeto XML, este é tratado dentro de um objeto do tipo *POI*, criado através do padrão *adapter*, que irá tratar a informação e armazena-la.

Os sistemas de recomendação turísticos que pretendam aceder aos pontos de interesse gerados no *POI engine* poderão fazê-lo através da *Api* do sistema.

5.2.6 Portal de internet

Após uma análise aos pontos de acesso ao sistema de recomendação *ToursPlan*, chegou-se à conclusão da inviabilidade de integrar a solução apresentada e nos termos propostos inicialmente, devido à falta de métodos de inserção de pontos de interesse e categorias de turismo nos serviços web do *ToursPlan*. Com o aparecimento desta incompatibilidade, foi necessária uma solução para fornecer os pontos de interesse do *POI engine*, a sistemas externos. Como solução, optou-se para evoluir o *POI engine* num *webservice* fornecedor de pontos de interesse ao invés de parte integrante do *ToursPlan*. O componente “portal de internet” contém um portal onde utilizadores conseguem interagir com a aplicação, e um serviço de internet, para permitir o acesso exterior.

5.2.6.1 web services

São aplicações informáticas desenvolvidas em diferentes tecnologias como, XML ou Json, que estão disponíveis num servidor web permitindo ser acedidos por outras aplicações através de protocolos, como SOAP, HTTP, Jms. Os serviços Web são independentes de plataforma e geralmente baseados em mensagens de texto que contém informações de desenvolvimento, processamento ou acesso em diferentes tecnologias. Isto torna os serviços Web interoperáveis (Dudhe & Sherekar 2014).

Atualmente existem duas tecnologias de serviços web conhecidas, Simple Object Access Protocol (SOAP) e Representational State Transfer (REST) (Carneiro 2013).

- SOAP é definido como um protocolo de comunicação padrão para troca de informação estruturada, segundo as normas Web Service Description Language (WSDL), entre sistemas ligados a uma infraestrutura de rede informática (Potti

Arquitetura

2011). Em termos gerais o *SOAP* é permite a troca de mensagens estruturadas entre diversos sistemas, e sobre diversos protocolos como o *HTTP*, *JMS* e *SMTP*.

Para o projeto em questão será abordado o protocolo *HTTP* por tornar o processo comunicação mais simples, evitando questões relacionadas com *proxies* e *firewalls*. O *SOAP* acrescenta um elevado nível de descrição de objetos e serviços em formato *XML* permitindo assim interoperabilidade entre plataformas (Carneiro 2013).

- REST é uma arquitetura para desenvolvimento de serviços web, baseada no protocolo *HTTP* (Potti 2011). O REST não adiciona nenhuma camada à já existente pelo protocolo. Os serviços são disponibilizados nos endereços de internet e a utilização dos métodos GET, POST, PUT, DELETE e HEAD do protocolo *HTTP* define as ações que esses serviços vão efetuar (Carneiro 2013).

No entanto, uma das principais diferenças entre o REST e o SOAP, e ao mesmo tempo um fator de escolha importante é rastreamento dos pedidos. Enquanto o SOAP mapeia um histórico das comunicações entre sistemas, permitindo o processamento de pedidos em várias etapas recolhendo a informação necessária mesmo em momentos separados, o REST obriga a que toda a informação necessária para processar o mesmo seja enviada (Nahon 2011).

A aplicação irá conter um serviço web de acesso aos pontos de interesse guardados na base de dados, e uma vez que se pretende que a implementação do mesmo seja rápida e a comunicação de pontos de interesse seja feita com o menor custo de performance, optou-se pela implementação de um serviço web REST. A Figura 11 mostra o processo de um pedido de pontos de interesse por categoria, em que é feito um pedido ao *CategoryController* da API, e o mesmo reencaminha o pedido ao modelo para recolha dos dados da base de dados.

Sequence Diagram - API getPoisByCategory

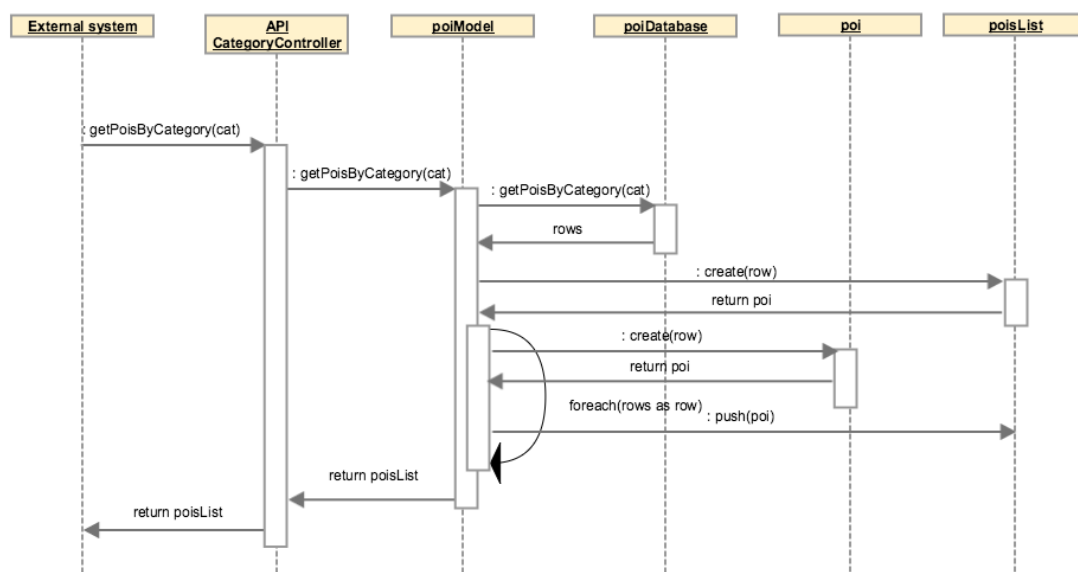


Figura 11 - API pedido de pontos de interesse por categoria

Sendo este serviço *web* do tipo *REST*, foi decidida a sua implementação numa linguagem de programação e *framework* orientada a *web*. Para isso, foi escolhida a *framework laravel* que utiliza PHP como linguagem de programação. Esta decisão é baseada na rápida implementação de páginas de internet e serviços *web Rest* em metodologia *Model View Controller (MVC)* que a *framework laravel* oferece.

A utilização da *framework laravel*, permite de uma forma acessível, a implementação de um portal *web* de inserção manual de pontos de interesse, abrindo assim a possibilidade de um novo método de criação de pontos de interesse.

5.2.6.2 Portal de internet

A criação de um portal *web* orientado à inserção de pontos de interesse profissionais, torna o método de criação de pontos de interesse mais acessível a utilizadores humanos, criando assim a possibilidade de inserir manualmente os pontos de interesse na base de dados.

A Figura 12 representa o processo de criação de pontos de interesse através do portal *web* do sistema *Poi engine*. Conforme se pode observar, após o login o utilizador tem acesso a uma funcionalidade de criação de pontos de interesse onde lhe é permitido, através do preenchimento de um formulário, criar um ponto de interesse.

Use Case - registo de Poi no portal web

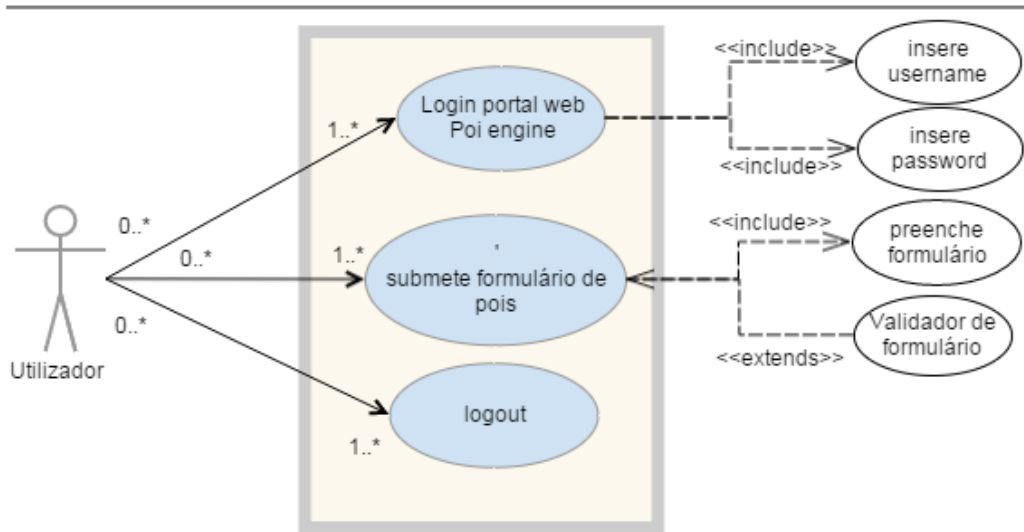


Figura 12 - Registo de pontos de interesse através do portal *web*

5.3 Base de dados

Neste capítulo apresenta-se o modelo de dados que irá constituir a base de dados de POI's. Esta base de dados irá ser usada pelo POI engine e terá o objetivo de guardar a informação pontos de interesse e tipos de dados e endereços URL a pesquisar.

Sendo a utilização da base de dados apenas para guardar algumas informações relevantes na criação de pontos de interesse, esta terá apenas algumas tabelas. Como demonstrado na Figura 13 existem 6 tabelas, sendo as tabelas siteType, enderecosURL, usadas como apoio na criação de objetos do tipo POI através do motor de geração de pontos de interesse, uma vez que este se baseia no tipo de dados a recolher para criar o objeto, e também a lista de endereços URL dos sites onde essa informação será recolhida. As restantes tabelas POI, poiEngine, poiCategories e categories servem para guardar os dados recolhidos em forma de poi para posteriormente não existir informações duplicadas de pontos de interesse na base de dados.

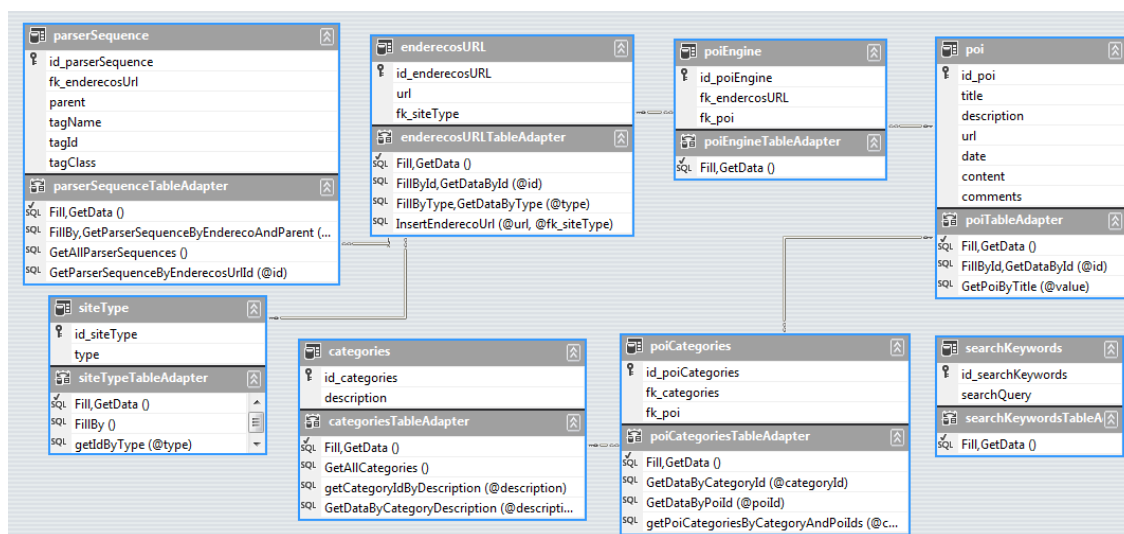


Figura 13 – Modelo de dados POI engine

5.3.1 Regras

Neste subcapítulo serão enumeradas regras e tipos de campos guardados por tabela.

5.3.1.1 Tabela enderecoURL

Esta tabela tem como função guardar os endereços URL das páginas de internet de onde serão recolhidos os dados. Esta informação é importante no momento de criação do objeto, vai recolher o código HTML das páginas de internet definidas. As regras dos campos são apresentadas na Tabela 7.

Tabela 7 – Regras tabela enderecoURL

Campo	Tipo	Nulo	Tamanho	Default	PK	Unique	FK
<code>id_enderecoURL</code>	<i>inteiro</i>	<i>não</i>	-	-	<i>sim</i>	<i>sim</i>	<i>não</i>
<code>siteType</code>	<i>Enum(HTML,RSS,search_engine)</i>	<i>não</i>	-	<i>HTML</i>	<i>não</i>	<i>não</i>	<i>não</i>
<code>url</code>	<i>varchar</i>	<i>não</i>	<i>10000</i>	-	<i>não</i>	<i>sim</i>	<i>não</i>

5.3.1.2 Tabela poi

Esta tabela tem como função guardar os dados necessários à criação de pontos de interesse para o *ToursPlan*. Esta informação é importante no momento de criação do objeto POI que será enviado para o *ToursPlan*. As regras dos campos são apresentadas na Tabela 8.

Tabela 8 – Regras tabela poi

Campo	Tipo	nulo	Tamanho	Default	PK	Unique	FK
<i>Id_poi</i>	<i>inteiro</i>	<i>não</i>			<i>sim</i>	<i>sim</i>	<i>não</i>
<i>title</i>	<i>text</i>	<i>não</i>			<i>não</i>	<i>não</i>	<i>não</i>
<i>description</i>	<i>text</i>	<i>não</i>			<i>não</i>	<i>não</i>	<i>não</i>
<i>url</i>	<i>varchar</i>	<i>não</i>	<i>10000</i>		<i>não</i>	<i>não</i>	<i>não</i>
<i>date</i>	<i>date</i>	<i>não</i>			<i>não</i>	<i>não</i>	<i>não</i>
<i>content</i>	<i>text</i>	<i>não</i>			<i>não</i>	<i>não</i>	<i>não</i>
<i>comments</i>	<i>text</i>	<i>sim</i>			<i>não</i>	<i>não</i>	<i>não</i>
<i>active</i>	<i>int</i>	<i>não</i>		<i>0</i>	<i>não</i>	<i>não</i>	<i>não</i>

5.3.1.3 Tabela poiEngine

Esta tabela tem como função fazer a ligação entre as restantes tabelas. Esta informação é importante para se ter conhecimento se um POI já existe, que tipo de dados forma recolhidos e de que site. As regras dos campos são apresentadas na Tabela 9.

Tabela 9 – Regras tabela poiEngine

Campo	Tipo	nulo	PK	Unique	FK
<i>Id_poiEngine</i>	<i>inteiro</i>	<i>não</i>	<i>sim</i>	<i>sim</i>	<i>não</i>
<i>fk_siteURL</i>	<i>inteiro</i>	<i>não</i>	<i>não</i>	<i>sim</i>	<i>sim</i>
<i>fk_poi</i>	<i>inteiro</i>	<i>não</i>	<i>não</i>	<i>sim</i>	<i>sim</i>

5.3.1.4 Tabela poiCateopries

Esta tabela tem como função estabelecer uma relação de muitos para um com a tabela POI. Esta informação é importante para se poder atribuir vários POI's a uma categoria. As regras dos campos são apresentadas na Tabela 10.

Tabela 10 – Regras tabela poiCategories

Campo	Tipo	nulo	PK	Unique	FK
<i>Id_poiCategories</i>	<i>inteiro</i>	<i>não</i>	<i>sim</i>	<i>sim</i>	<i>não</i>
<i>fk_categories</i>	<i>inteiro</i>	<i>não</i>	<i>não</i>	<i>sim</i>	<i>sim</i>
<i>fk_poi</i>	<i>inteiro</i>	<i>não</i>	<i>não</i>	<i>sim</i>	<i>sim</i>

5.3.1.5 Tabela categories

Esta tabela tem como função armazenar os tipos de categorias encontradas. Esta informação é importante para se poder atribuir categorias aos pontos de interesse. As regras dos campos são apresentadas na Tabela 11.

Tabela 11 – Regras tabela categories

Campo	Tipo	nulo	PK	Unique	FK
<i>Id_categories</i>	<i>inteiro</i>	<i>não</i>	<i>sim</i>	<i>sim</i>	<i>não</i>
<i>description</i>	<i>inteiro</i>	<i>não</i>	<i>não</i>	<i>sim</i>	<i>sim</i>

5.3.1.6 Tabela user

Esta tabela tem como função armazenar os tipos de utilizador do portal web do *POI engine*. Esta informação é importante para se poder definir que tipo de acesso o utilizador tem, e quais as funções a que tem acesso. As regras dos campos são apresentadas na Tabela 12.

Tabela 12 – Regras tabela user

Campo	Tipo	nulo	Tamanho	Default	PK	Unique	FK
<i>Id_user</i>	<i>inteiro</i>	<i>não</i>			<i>sim</i>	<i>sim</i>	<i>não</i>
<i>username</i>	<i>varchar</i>	<i>não</i>	<i>20</i>		<i>não</i>	<i>sim</i>	<i>não</i>
<i>type</i>	<i>Enum (admin,manager,user)</i>	<i>não</i>		<i>user</i>	<i>não</i>	<i>não</i>	<i>não</i>
<i>active</i>	<i>inteiro</i>	<i>não</i>		<i>0</i>	<i>não</i>	<i>não</i>	<i>não</i>

5.3.1.7 Tabela parserSequence

Esta tabela tem como função armazenar a sequência de elementos *HTML* encontrados na criação com sucesso de um ponto de interesse derivado de código *HTML*. Esta informação é importante para melhorar o processo de criação de pontos de interesse deste tipo, uma vez que se a estrutura *HTML* com o conteúdo pretendido for conhecida, pode-se quase de imediato extrair os conteúdos necessários, algumas características do elemento *HTML* poderão também ser importantes, como *id* do elemento ou uma classe *CSS* específica. As regras dos campos são apresentadas na Tabela 13.

Tabela 13 – Regras tabela parserSequence

Campo	Tipo	nulo	PK	Unique	FK
<i>id_parserSequence</i>	<i>inteiro</i>	<i>não</i>	<i>sim</i>	<i>sim</i>	<i>não</i>
<i>fk_enderecosUrl</i>	<i>inteiro</i>	<i>não</i>	<i>não</i>	<i>sim</i>	<i>não</i>
<i>parent</i>	<i>inteiro</i>	<i>sim</i>	<i>não</i>	<i>não</i>	<i>não</i>
<i>tagName</i>	<i>varchar</i>	<i>sim</i>	<i>não</i>	<i>não</i>	<i>não</i>
<i>tagId</i>	<i>varchar</i>	<i>sim</i>	<i>não</i>	<i>não</i>	<i>não</i>
<i>tagClass</i>	<i>varchar</i>	<i>sim</i>	<i>não</i>	<i>não</i>	<i>não</i>

5.4 Conclusão

O presente capítulo definiu a arquitetura escolhida para o desenvolvimento do sistema, sendo que irá conter serviços separados, um portal de internet que permite inserir e validar pontos de interesse de forma manual, e contém também um serviço de internet que permite obter externamente os pontos de interesse. Este capítulo é parte integrante do sistema *POI engine*, no entanto está numa infraestrutura separada dos restantes componentes “*web crawler*”, que é responsável por requisitar resultados de pesquisas ao motor de pesquisa Bing. O “*motor de feeds*”, responsável por adquirir os dados de páginas de internet com tecnologia RSS. O “*web scrapper*”, que é responsável por recolher a informação necessária das páginas de Internet sem tecnologia RSS, e por remover a informação não relevante dos conteúdos. E o gerador de pontos de interesse, que é responsável por aglomerar a informação numa estrutura conhecida e armazená-la.

6 Desenvolvimento do protótipo

Neste capítulo irá ser apresentada a solução do protótipo desenvolvido para o problema apresentado. Serão apresentados excertos de código e representações gráficas do produto a implementar.

6.1 *Poi engine*

O sistema *POI engine*, foi implementado em várias camadas sendo seguida uma metodologia *Data Access Layer (DAL)*, *Business Logic Layer(BLL)* para o mecanismo automático de pesquisa e classificação de informação e uma *User Interface (UI)*, implementada separadamente, com uma metodologia *Model View Controller (MVC)*.

6.1.1 *Portal Web*

Este componente é constituído por duas camadas, uma interface de utilizador onde podem aceder utilizadores autorizados, e uma API para providenciar acesso autorizado a sistemas externos.

6.1.1.1 *Interface de utilizador*

A decisão de se criar uma interface de utilizador foi devido a ser uma forma de fácil acesso e inserção de pontos de interesse. Esta camada seguiu uma metodologia *MVC*, tendo a estrutura apresentada na Figura 14.

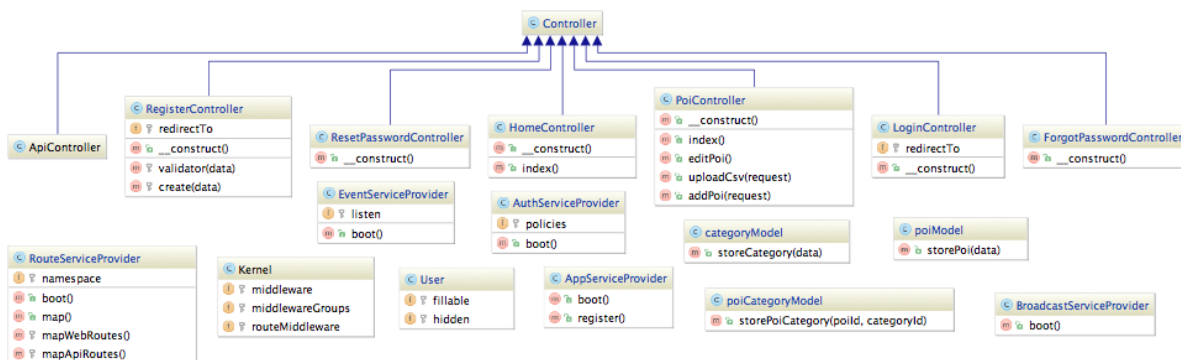


Figura 14 – Diagrama de classes interface gráfica do portal de Internet

Como demonstra a Figura 14, existem controladores para gerir a conta de utilizador, sendo eles o “ResetPasswordController”, o “LoginController”, o “RegisterController” e o “ForgotPasswordController”. Cada um destes controladores tem a função que lhes foi atribuída ao nome, por ordem, repor a palavra passe, autorizar acesso ao utilizador, registar um novo utilizador e repor a palavra passe.

Sendo o acesso à interface reduzido a utilizadores registados e ativos, será necessário, para além dos controladores de gestão de utilizadores mencionados, implementar um módulo lista de controlo de acessos (ACL) para diferenciar o tipo de utilizador e definir os acessos a funcionalidades de cada um. Assim sendo, haverá vários níveis de acesso ao portal, sendo estes definidos pelo tipo de utilizador. A Figura 15 mostra a página de entrada do portal web, com as funcionalidades disponíveis. Esta funcionalidade provém do controlador “HomeController”.

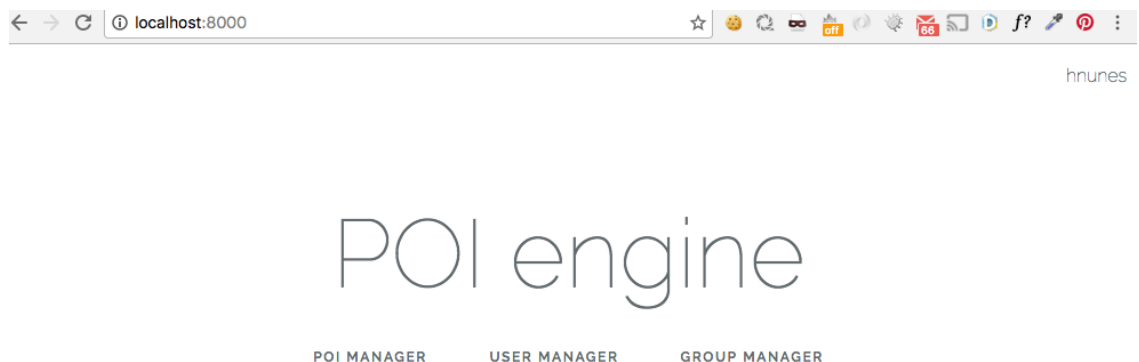


Figura 15 – Página de entrada do portal web

6.1.1.1.1 Gestão de utilizadores

Esta subcapítulo irá descrever os tipos de utilizador, as funções que lhes estão atribuídas e que nível de acesso têm. Irão ser 3 níveis de acesso ao portal: utilizador, manager e administrador, tendo cada um deles acesso a funcionalidades diferentes. As características dos utilizadores são detalhadas de seguida.

6.1.1.1.2 Utilizador normal

O *user* ou utilizador, terá acesso apenas a criação de pontos de acesso, tendo, após autenticação no portal, acesso a uma funcionalidade de criação e edição de pontos de interesse. Este tipo de utilizador irá pertencer a um grupo de utilizadores.

6.1.1.1.3 Utilizador gestor de grupo

O gestor de grupo, tem um acesso de gestão a todos os dados inseridos pelos utilizadores do grupo a que pertence. As funções de um gestor de grupo passam por validar e ativar os pontos de interesse criados pelos utilizadores, acumulando também com as funcionalidades disponíveis aos utilizadores. Um grupo poderá ter um ou mais gestores.

6.1.1.1.4 Utilizador administrador

O administrador do portal terá como funções dar apoio a utilizadores ou gestores de contas, validando a criação de grupos e respetivas contas de gestores.

6.1.1.1.5 Funcionalidade de gestão de pontos de interesse

Esta funcionalidade irá permitir aos utilizadores do portal web criar pontos de interesse de forma manual. Estará disponível a todos os utilizadores, e irá transportar o utilizador para uma nova página, onde poderá escolher que tipo de ação pretende efetuar conforme mostra a Figura 16. Esta funcionalidade provém do controlador "*PoiController*".

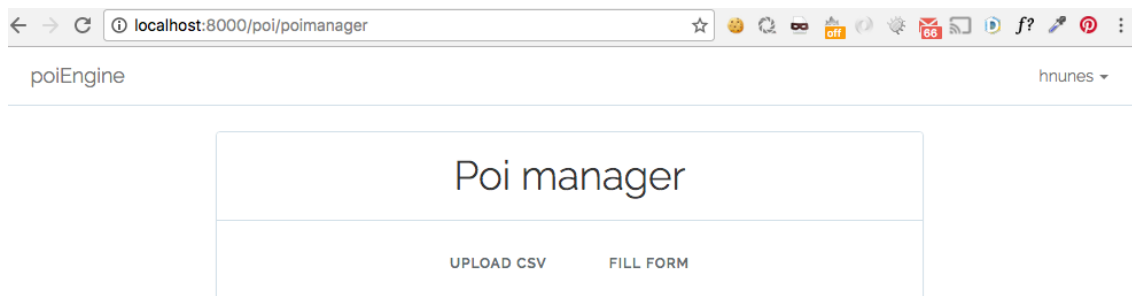


Figura 16 – Funcionalidade *poi manager*

A funcionalidade gestor de *POI's*, permite inserir *POI's* de duas formas, carregando um ficheiro do com extensão CSV, ou por submissão de formulário.

- **Upload CSV**

Esta funcionalidade permite aos utilizadores inserir vários pontos de interesse em simultâneo. Ao aceder a esta funcionalidade, o utilizador é transportado para um novo ecrã onde pode escolher e submeter um ficheiro. A Figura 17 mostra o ecrã com o formulário de submissão. Esta funcionalidade provém do controlador "*PoiController*".

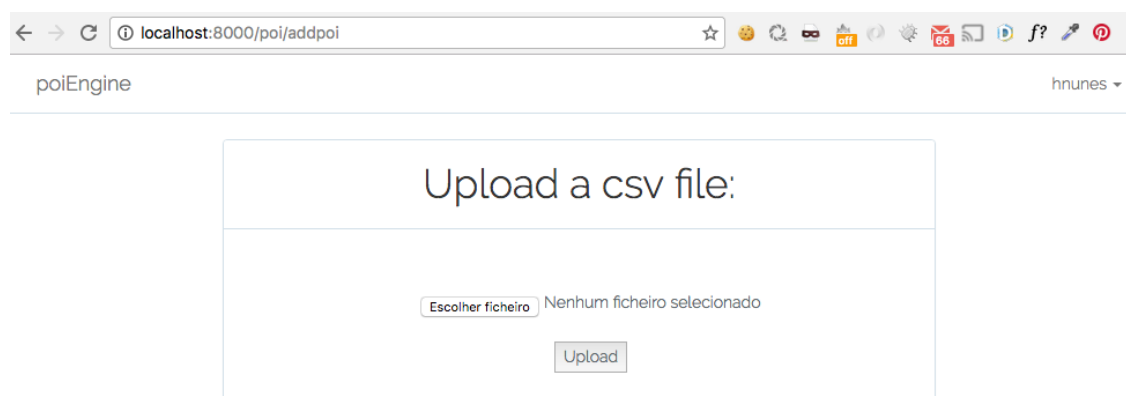


Figura 17 – Funcionalidade de submissão de *POI's* por CSV

Ao submeter este ficheiro, o motor de geração de pontos de interesse guarda-o numa localização conhecida, e posteriormente recolhe a informação do mesmo através da biblioteca *ddeboer/data-import* disponível através do PHP *composer* e que permite facilmente ler um ficheiro CSV, como mostra o excerto de Código 6.

```
public function uploadCsv(Request $request)
{
    if ($request->hasFile('file')) {
        try {
            $file = $request->file('file');
            $filename = $file->getClientOriginalName();
            $destinationPath =
                config('app.fileDestinationPath').'/' . $filename
            ;
            $uploaded = Storage::put(
                $destinationPath,
                file_get_contents($file->getRealPath())
            );
            if ($uploaded) {
                $path = resource_path('app') . '/' . $destinationPath;
                $file = new \SplFileObject($path);
                $reader = new CsvReader($file, ';');
                $headerFlag = false;
                $header = [];
                $poi = [];
            }
        }
    }
}
```

```

$poiModel = new poiModel();
$poiCategoryModel = new poiCategoryModel();
$categoriesModel = new categoryModel();
$categoryId = false;
$poiId = false;
foreach ($reader as $row) {
    if (empty($headerFlag)) {
        $header = $row;
        $headerFlag = true;
        continue;
    }
    foreach ($row as $key => $value) {
        if ($key == 4) {
            $categoryId = $categoriesModel->storeCategory(
                ['description' => $value]
            );
            continue;
        }
        $poi[$header[$key]] = $value;
    }
    $poiId = $poiModel->storePoi($poi);
}
if ($categoryId && $poiId) {
    $poiCategoryModel->storePoiCategory(
        $poiId, $categoryId
    );
}
return redirect()->to('poi/successpoi');
}
} catch (\Exception $e) {
    return redirect()->to('poi/errorpoi');
}
}
return redirect()->to('poi/errorpoi');
}
}

```

Código 6 – Método de envio de ficheiro CSV

- **Submissão de formulário**

Esta funcionalidade permite aos utilizadores inserir pontos de interesse através do preenchimento de um formulário. Para isso o utilizador é transportado para um novo ecrã onde pode preencher e submeter o formulário. Esta funcionalidade está em desenvolvimento neste momento, e provém do controlador “*Poicontroller*”.

6.1.1.1.6 Gestão de conta

Esta funcionalidade tem o propósito de permitir aos utilizadores *managers* ou administradores ativarem ou desativarem contas de utilizador. Será apresentada uma grelha com a lista de utilizadores onde os utilizadores com permissões de edição poderão ativar ou desativar utilizadores da aplicação. Esta funcionalidade permitirá também aos utilizadores habilitados a validação e ativação dos pontos de interesse. Esta funcionalidade está em desenvolvimento neste momento.

6.1.1.1.7 Gestão de grupos e gestores

Esta funcionalidade estará disponível apenas ao administrador do portal, por forma a validar utilizadores *manager* e grupos de utilizadores, e assim, permitir o acesso ao motor de pontos de interesse através do portal web. Para tal, o administrador irá ter duas grelhas, uma de utilizadores e outra de grupos, onde poderá editar, validar e ativar ou desativar utilizadores e grupos. Esta funcionalidade está em desenvolvimento neste momento.

6.1.1.2 Serviço web de acesso a dados (Api)

Esta funcionalidade assenta num serviço de internet *Restfull*, que permitirá a aplicações externas comunicarem com o *POI engine* e obterem os pontos de interesse para apresentar aos seus clientes. Esta camada segue uma metodologia MVC, tendo a estrutura apresentada na Figura 18.

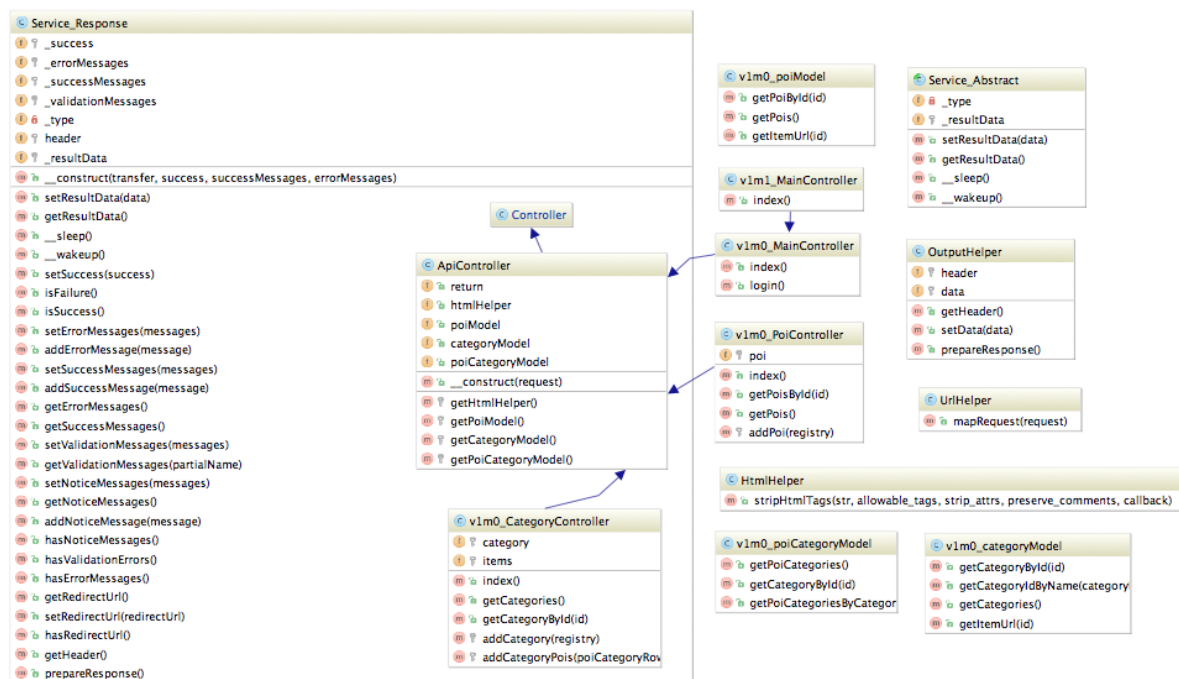


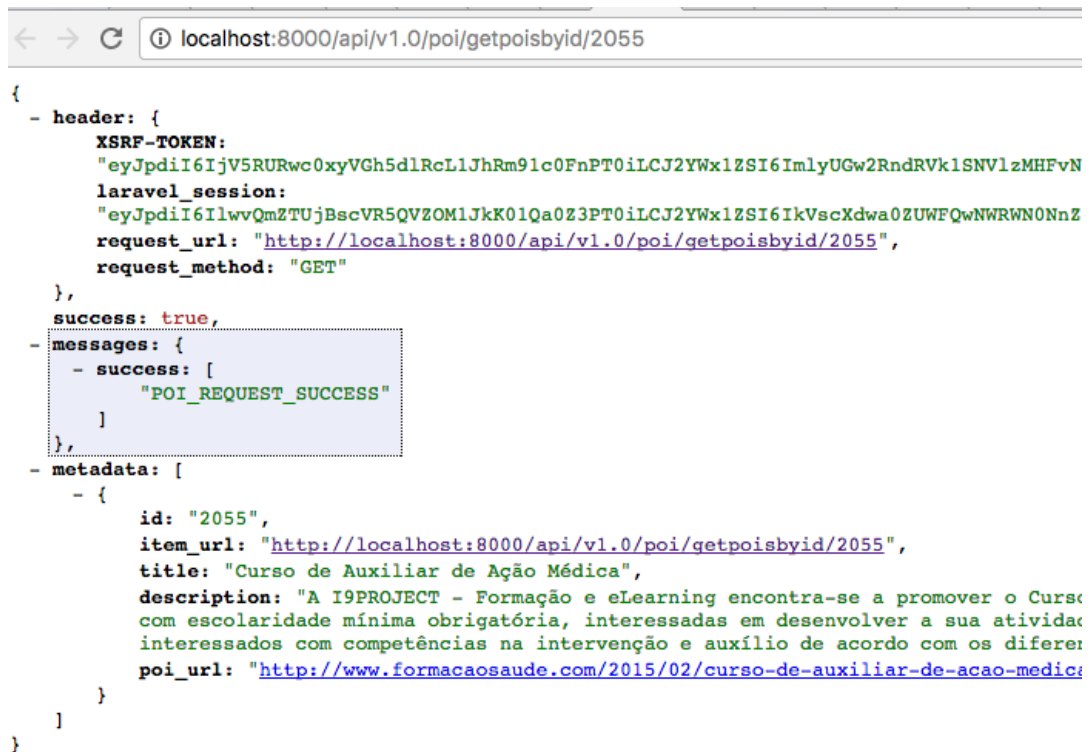
Figura 18 – diagrama de classes Serviço de internet

Este serviço tem a capacidade de ser versionado, permitindo assim alterações à estrutura de resposta em versões diferentes da API. De seguida irão ser definidos os métodos de acesso disponíveis para obter pontos de interesse, sendo os implementados na versão 1.0.

- **Pedido API *getPoisById***

Esta função encontra-se no controlador “*v1m0_PoiController*”, e tem o propósito de obter um ponto de interesse específico. Recebe como parâmetro o valor do registo *id_poi* da

Tabela 8 (POI) e devolve o conteúdo do ponto de interesse numa estrutura *Json*. A Figura 19 mostra o resultado de um pedido.



```

{
  - header: {
    XSRF-TOKEN:
      "eyJpdii6IjV5RURwc0xyVGh5dGlRcLlJhRm9lc0FnPT0iLCJ2YWx1ZSI6ImlyUGw2RndRVk1SNVlzMHFvN
    laravel_session:
      "eyJpdii6IlwvQmZTUjBscVR5QVZOMlJkK01Qa0Z3PT0iLCJ2YWx1ZSI6IkVscXdwa0ZUWFQwNWRWN0NnZ
    request_url: "http://localhost:8000/api/v1.0/poi/getpoisbyid/2055",
    request_method: "GET"
  },
  success: true,
  - messages: {
    - success: [
      "POI_REQUEST_SUCCESS"
    ]
  },
  - metadata: [
    - {
      id: "2055",
      item_url: "http://localhost:8000/api/v1.0/poi/getpoisbyid/2055",
      title: "Curso de Auxiliar de Ação Médica",
      description: "A I9PROJECT - Formação e eLearning encontra-se a promover o Curso
      com escolaridade mínima obrigatória, interessadas em desenvolver a sua atividade
      interessados com competências na intervenção e auxílio de acordo com os diferen
      poi_url: "http://www.formacaosaude.com/2015/02/curso-de-auxiliar-de-acao-medica
    }
  ]
}

```

Figura 19 – Resultado pedido API de ponto de interesse específico

- **Pedido API getPois**

Esta função encontra-se no controlador “*v1m0_PoiController*”, e tem o propósito de devolver todos os pontos de interesse existentes na base dados. Este método não recebe parâmetros. A Figura 20 representa a resposta a um pedido de todos os pontos de interesse.

```

localhost:8000/api/v1.0/poi/getpois/
{
  - header: {
    XSRF-TOKEN:
      "eyJpdiI6IjFCRHhcL3VGyTFiVjAybXFnYmZOUFRPT0iLCJ2YWx1ZSI6InFHQUI1TFZzSG1YT0YySEpROWx0NmJOYmRXVj
    laravel_session:
      "eyJpdiI6InNPa09TRytWRDRzUjN2ak5DVEc2OGc9PSIsInZhbnVlIjoY0I1RGJURmlnMU9cL0NwRkFveU5rUTJYZnMrT%
    request_url: "http://localhost:8000/api/v1.0/poi/getpois/",
    request_method: "GET"
  },
  success: true,
  - messages: {
    - success: [
      "POI_REQUEST_SUCCESS"
    ]
  },
  - metadata: [
    - {
      id: "2055",
      item_url: "http://localhost:8000/api/v1.0/poi/getpoisbyid/2055",
      title: "Curso de Auxiliar de Ação Médica",
      description: "A I9PROJECT - Formação e eLearning encontra-se a promover o Curso de Auxiliar
        pessoas com escolaridade mínima obrigatória, interessadas em desenvolver a sua atividade pr
        os interessados com competências na intervenção e auxílio de acordo com os diferentes tipos
      poi_url: "http://www.formacao.saude.com/2015/02/curso-de-auxiliar-de-acao-medica/"
    },
    - {
      id: "2056",
      item_url: "http://localhost:8000/api/v1.0/poi/getpoisbyid/2056",
      title: "Cursos de Inglês (Nível)",
    }
  ]
}

```

Figura 20 - Resultado pedido API de todos os pontos de interesse

- **Pedido API getCategories**

Esta função encontra-se no controlador “v1m0_CategoryController”, e tem o propósito de devolver todas as categorias existentes na base dados, este método não recebe parâmetros. A Figura 20 representa a resposta a um pedido de todas as categorias.

```

{
  - header: {
    XSRF-TOKEN:
      "eyJpdiI6IkNSak54c3VkdXh4Z1pMQ2ZDZm9rdkE9PSIsInZhbnVlIjoic1R0OTJiUjM1MlhISnc3ZHpj
    laravel_session:
      "eyJpdiI6InREU29WWDZtTDB4SXRzZVJoQ3AxUVE9PSIsInZhbnVlIjoic1R0OTJiUjM1MlhISnc3ZHpj
    request_url: "http://localhost:8000/api/v1.0/category/getcategories",
    request_method: "GET"
  },
  success: true,
  - messages: {
    - success: [
      "POI_REQUEST_SUCCESS"
    ]
  },
  - metadata: [
    - {
      id: "39",
      item_url: "http://localhost:8000/api/v1.0/category/getcategorybyid/39",
      - items: {
        - redirect: [
          - {
            id: "2055",
            item_url: "http://localhost:8000/api/v1.0/poi/getpoisbyid/2055"
          }
        ]
      },
      description: "Formação"
    },
    - {
      id: "40",
      item_url: "http://localhost:8000/api/v1.0/category/getcategorybyid/40",
      - items: {
        - redirect: [
          - {
            id: "2055",
            item_url: "http://localhost:8000/api/v1.0/poi/getpoisbyid/2055"
          },
          - {
            id: null,
            item_url: "http://localhost:8000/api/v1.0/poi/getpoisbyid/"
          }
        ]
      }
    }
  ]
}

```

Figura 21 - Resultado pedido API de todas as categorias

- **Pedido API getCategoryById**

Esta função encontra-se no controlador “*v1m0_CategoryController*”, e tem o propósito de obter uma categoria específica. Recebe como parâmetro o valor do registo `id_categories` da Tabela 11 (*categories*) e devolve o conteúdo da categoria numa estrutura *Json*. A Figura 19 demonstra o resultado de um pedido.

```

{
  - header: {
    XSRF-TOKEN:
      "eyJpdiI6InZwVzB1V011U2RLTXQ1UEtQTkkrQVE9PSIsInZhbHVlIjoicFBhc1Nyd05yODZuanhEYVlu
    laravel_session:
      "eyJpdiI6InBLc2Y0bDh0VjZyY1FQRzVLUmNDT2c9PSIsInZhbHVlIjoic1lwd0VIXC9ueGZqVXVPb3di
    request_url: "http://localhost:8000/api/v1.0/category/getcategorybyid/39",
    request_method: "GET"
  },
  success: true,
  - messages: {
    - success: [
      "CATEGORY_REQUEST_SUCCESS"
    ]
  },
  - metadata: [
    - {
      id: "39",
      item_url: "http://localhost:8000/api/v1.0/category/getcategorybyid/39",
      - items: {
        - redirect: [
          - {
            id: "2055",
            item_url: "http://localhost:8000/api/v1.0/poi/getpoisbyid/2055"
          }
        ]
      },
      description: "Formação"
    }
  ]
}

```

Figura 22 - Resultado pedido API de categoria específica

6.1.2 Mecanismo automático de pesquisa e classificação de informação

Este componente foi implementado em duas camadas distintas, a DAL e a BLL. Foi criado um diagrama de classes, por forma, a melhor se entender a estrutura deste componente e que atributos contém. O diagrama de classes está representado na Figura 23.

Diagrama de classes - Mecanismo automático de pesquisa e classificação de informação

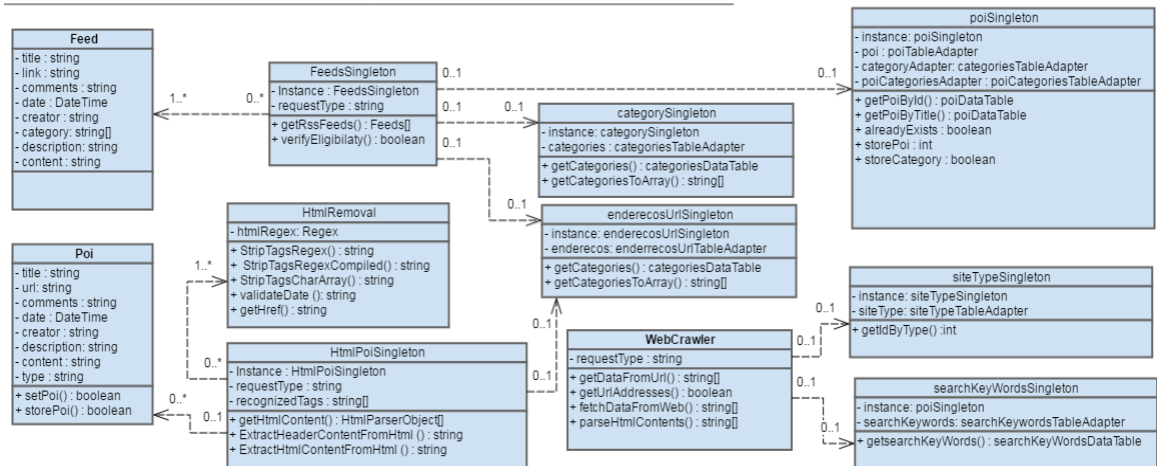


Figura 23 - Diagrama de classes - mecanismo automático de pesquisa e classificação de informação

6.1.2.1 Camada de abstração de dados - Data Access Layer (DAL)

A camada de abstração de dados, contém todas as classes de acesso à base de dados. Estas classes foram implementadas com base no padrão *singleton*, uma vez que se pretende que exista apenas uma instância de ligação a cada tabela.

Classe *enderecoUriSingleton*: A classe *endereçosUriSingleton*, permite o acesso a todos os endereços URL guardados na base de dados. Esta classe dispõe de alguns métodos públicos que permitem obter o conteúdo da tabela *endereçosUri* com alguma parametrização.

O método *getUrlsByType* tem como parâmetro um objeto do tipo alfabético *type* e devolve todas as entradas existentes na tabela do tipo passado por referência. Este tipo está identificado na tabela *siteType*.

O método *getUrlsById* recebe como parâmetro um objeto do tipo inteiro *id* que é o identificador do endereço pretendido.

O método *storeUrlsArray* recebe dois parametros um objeto do tipo alfanumérico *link* e um objeto do tipo inteiro *typeId* que é o identificador do tipo de endereço. Este método tem como função guardar endereços URL na tabela.

Classe *siteTypeSingleton*: A classe *siteTypeSingleton*, permite o acesso aos conteúdos da tabela *siteType* que contém os tipos de endereço URL conhecidos do *POI engine*. Esta classe dispõe de um método público que recebe um parâmetro do tipo alfanumérico *type* contendo o tipo de endereço pretendido, e retorna o identificador do mesmo.

Classe *categorySingleton*: A classe *categorySingleton*, permite o acesso a todas as categorias guardadas na base de dados. Esta classe dispõe de dois métodos públicos que permitem obter o conteúdo da tabela *categories*.

O método *getCategories* devolve uma instância da tabela *categories* com a totalidade do seu conteúdo.

O método *getCategoriesToArray*, recebe como parâmetro um objeto booleano *tolowercase*, e devolve a totalidade do conteúdo da tabela *categories*. No entanto o conteúdo é devolvido num *array*, e mediante o conteúdo do parâmetro *tolowercase*, devolve o conteúdo em letras minúsculas se for verdadeiro ou maiúsculas se for falso.

Classe *poiSingleton*: A classe *poiSingleton*, permite o acesso aos conteúdos da tabela POI guardados na base de dados que contém os pontos de interesse. Esta classe dispõe de cinco métodos públicos que permitem obter o conteúdo da tabela *poi*.

O método *getPoiById* recebe como parâmetro um objeto do tipo inteiro *id* que é o identificador do ponto de interesse pretendido. Este método retorna o conteúdo da linha da tabela *poi* com o identificador do parâmetro.

O método *getPoiByTitle* recebe como parâmetro um objeto do tipo alfanumérico *title* que é o título do ponto de interesse pretendido. Este método retorna o conteúdo da linha da tabela *poi* com o título recebido por parâmetro.

O método *alreadyExists* recebe como parâmetro um objeto do tipo alfanumérico *title* que é o título do ponto de interesse que se pretende guardar na tabela *poi*. Este método verifica se o ponto de interesse já existe na tabela *POI*, e devolve verdadeiro no caso de já existir e falso caso ainda não exista.

O método *storePoi* recebe como parâmetros seis objetos do tipo alfanumérico: *title*, *comments*, *url*, *description* e *requestType*, um objeto do tipo data tempo *date* e um outro objeto do tipo inteiro longo *urlId*. Este método tem o propósito de guardar na tabela *poi*, o conteúdo dos parâmetros recebidos sob a forma de ponto de interesse.

O método *storeCategory* recebe como parâmetros um objeto do tipo *array* de alfanuméricos *categories* e um objeto do tipo inteiro *poId*. Este método tem o propósito de guardar na tabela *categories*, as categorias encontradas ao criar pontos de interesse.

Classe *searchKeywordsSingleton*: A classe *searchKeywordsSingleton*, permite o acesso aos conteúdos da tabela *keywords* que contém as palavras chave utilizadas na pesquisa por pontos de interesse. Esta classe dispõe de um método público, sem parâmetros adicionais e retorna todas as palavras chave guardadas.

6.1.2.2 Camada de lógica de negócio - Business Layer (BL)

A camada de abstração de negócio, contém as classes com lógica de negócio desenvolvidas, ou seja, os mecanismos de criação de pontos de interesse e recolha de informação para a criação dos mesmos. Estes mecanismos são o motor de *feeds*, *web crawler* e *scraper* e o gerador de pontos de interesse.

6.1.2.2.1 Gerador de pontos de interesse

Por forma a criar e fornecer pontos de interesse, foi necessária a criação de uma estrutura de dados que representasse o objeto a fornecer. Para isso foi criado um tipo de dados *poi*, que detém a responsabilidade e a estrutura necessária para armazenar a informação que representa para o *POI engine* um ponto de interesse.

A classe *poi*, contém para além da estrutura acima representada, quatro construtores, sendo um deles vazio, outro recebe um objeto do mesmo tipo e torna-se uma cópia do mesmo, outro recebe todas as informações necessárias à criação de um *poi*, e atribui esses valores à própria estrutura de dados e, por fim, um outro construtor que armazena o conteúdo de um objeto *Feed* em formato de ponto de interesse.

Esta classe contém também um método de armazenamento de *pois* em base de dados.

6.1.2.2.2 Motor de feeds:

O motor de *feeds* é constituído por duas classes, *Feeds* e *FeedsSingleton*, sendo a classe *Feeds* o objeto que contém a estrutura de informação e propriedades que constituem um *Feed*.

Esta classe contém três construtores, sendo eles um construtor vazio, um duplicador de objetos *Feed*, e um outro que recebe a totalidade dos parâmetros acima representados e os atribui aos mesmos.

A classe *FeedsSingleton*, conforme o seu nome diz, é uma representação do padrão de design *singleton*. Este objeto é o motor de *feeds* propriamente dito, uma vez que tem a responsabilidade e o conhecimento necessário para recolher dados dos endereços URL do tipo RSS.

Poi engine

O processo de recolha de informação RSS, ocorre no método publico *getRssFeeds* da classe *FeedsSingleton*. Este método recolhe os endereços URL do tipo RSS guardados na base de dados, obtendo depois para cada um deles o seu conteúdo XML. Este processo é feito com a utilização da biblioteca *SyndicationFeed* e *XmlReader*. O objeto *XmlReader* contém o endereço URL onde se encontra a informação a recolher. O objeto *SyndicationFeed* irá requisitar o conteúdo XML do endereço guardado no objeto *XmlReader*, o excerto de Código 7 representa este processo de recolha de informação em XML.

```
public Feed[] getRssFeeds()
{
    DAL.enderecosUrlSingleton enderecos = DAL.enderecosUrlSingleton.Instance;
    DAL.poiDatabase.enderecosURLDataTable
    rows = enderecos.getUrlsByType(requestType);

    var tempRssData = new List<Feed>();
    foreach (DAL.poiDatabase.enderecosURLRow row in rows)
    {
        XmlReader reader = XmlReader.Create(row.url);
        SyndicationFeed feedsObject = SyndicationFeed.Load(reader);
        reader.Close();
        Feed feed = new Feed();
        Poi.Poi poi = new Poi.Poi();
        foreach (SyndicationItem item in feedsObject.Items)
        {
            (...)
        }
    }
    return tempRssData.ToArray<Feed>();
}
```

Código 7 – Método de recolha de conteúdos *Feed*

6.1.2.2.3 Web crawler

O módulo de web *crawler* tem a função de pesquisar a internet por páginas que contenham no seu conteúdo informações relevantes para a criação de pontos de interesse formativos da área de saúde. Esta pesquisa irá ser baseada em palavras chave ou frases pré-definidas. Inicialmente foi pensado utilizar os motores de pesquisa Google e Bing, no entanto, o motor de pesquisa Google devolve conteúdo *javascript*, o que obriga à implementação de um *parser* separado e especializado nesta linguagem. Assim sendo, e uma vez que, o motor de pesquisa Bing da Microsoft devolve o conteúdo das pesquisas em *HTML*, como pretendido, optou-se por utilizar neste protótipo apenas o Bing. Assim, o *parser* inicialmente projetado para o módulo de criação de pontos de interesse irá receber os dados necessários à criação de pontos de interesse na linguagem prevista.

Esta pesquisa irá obter as frases ou palavras chave a pesquisar da base de dados, para construir o endereço de pesquisa. Irão ser efetuados dois tipos de pesquisa: uma por páginas de internet HTML, e uma outra pesquisa por sites com *feeds*, por forma a alimentar o motor de *feeds*. Para fazer estas pesquisas, vai ser utilizada a biblioteca *HtmlAgilityPack*, da forma representada no excerto de Código 10.

A biblioteca *HtmlAgilityPack* permite carregar páginas de internet à semelhança de um navegador de internet. Isto é efetuado através de um objeto *HttpRequest*. Após o pedido, o conteúdo da página é extraído para um outro objeto do tipo *HttpResponse*, para se poder extrair os endereços *URL* e retirar informação para gerar pontos de interesse.

6.1.2.2.4 Web scrapper

A função deste módulo, é recolher a informação necessária à criação de pontos de interesse das páginas de internet cujos endereços *URL* estão armazenados na base de dados. Este mecanismo foi implementado em duas classes, cada uma com a sua função, sendo uma delas a classe *HtmlPoiSingleton* e outra a *HtmlRemoval*. Esta funcionalidade está em desenvolvimento neste momento.

Classe *HtmlRemoval*: esta classe implementa a lógica de comparação de conteúdos por expressões regulares. Os métodos desta classe, recebem como parâmetro o conteúdo de um elemento *HTML* e verificam se o mesmo contém algum texto que seja validado.

Por exemplo, para validar se um elemento *HTML* contém uma data, foi utilizada a expressão regular apresentada no Código 8.

```
public string validateDate (string str)
{
    Regex r = new Regex(@"\d{4}-\d{2}-\d{2}");
    Match match = r.Match(str);
    if (match.Success)
    {
        DateTime dt = DateTime.ParseExact(match.Value, "dd-MM-yyy",
        CultureInfo.InvariantCulture);
    }

    return dt.ToString();
}
```

Código 8 – Função de recolha de data de um elemento *HTML*

No caso de o elemento em análise ser do tipo âncora, temos que guardar o endereço *URL* normalmente presente no atributo *href* do elemento. Para isso foi implementada a função apresentada no Código 9.

```
public string getHref (HtmlNode node )
{
    string link = node.GetAttributeValue("href", "");

    return link
}
```

Código 9 – Função de extração do atributo *href* de elementos âncora

Classe *HtmlPoiSingleton*: A classe *HtmlPoiSingleton*, é uma função que implementa o padrão *singleton*, e tem um método de acesso onde é efetuado o processo de recolha de informação, sendo esse método chamado de *getHtmlContent*. Neste método, são requisitados à base de dados todos os endereços *URL* do tipo *HTML* através da classe *enderecosUrlSingleton*. Conforme demonstra o Código 10 a seguir apresentado.

```
public bool getHtmlContent()
{
    DAL.enderecosUrlSingleton enderecos =
DAL.enderecosUrlSingleton.Instance;
    DAL.poiDatabase.enderecosURLDataTable rows =
enderecos.getUrlsByType(requestType);
    var tempHtmlData = new List<HtmlPoi>();
    foreach (DAL.poiDatabase.enderecosURLRow row in rows)
    {
        HttpRequest request = (HttpRequest)WebRequest.Create(row.url);
        HttpResponse response = (HttpResponse)request.GetResponse();

        if (response.StatusCode == HttpStatusCode.OK)
        {
            Stream receiveStream = response.GetResponseStream();
            StreamReader readStream = null;

            if (response.CharacterSet == null)
            {
                readStream = new StreamReader(receiveStream);
            }
            else
            {
                readStream = new StreamReader(
receiveStream, Encoding.GetEncoding(response.CharacterSet)
);
            }

            string data = readStream.ReadToEnd();
            string body = ExtractHeaderContentFromHtml(data);
            string content = ExtractHtmlContentForPoi(body);
            response.Close();
            readStream.Close();
        }
    }
    return true;
}
```

Código 10 – Pedido de HTML através da biblioteca *HtmlAgilityPack*

O conteúdo de cada um dos endereços é recolhido, sendo-lhes retirado o cabeçalho. Assim retira-se ao *scraper* algum peso de desempenho uma vez que não será necessário inspecionar essa parte do código. O processo de remoção do cabeçalho, é feito com a utilização da biblioteca *AngleSharp*, o conteúdo *HTML* é varrido e todas as *tags* encontradas que existam numa lista definida anteriormente, são removidas. O Código 11 mostra a implementação da função *ExtractHeaderContentFromHTML* responsável pelo processo de remoção do cabeçalho.

```
private string ExtractHeaderContentFromHtml(string input)
{
    List<string> tagsToRemove = new List<string>
    {
        "doctype",
        "head",
        "script",
        "style"
    };
    HtmlParser hp = new HtmlParser();
    List<IElement> tags = new List<IElement>();
    List<string> nodeTypes = new List<string>();
    var hpResult = hp.Parse(input);
    try
    {
        foreach (var tagToRemove in tagsToRemove)
            tags.AddRange(hpResult.QuerySelectorAll(tagToRemove));

        foreach (var tag in tags)
            tag.Remove();
    }
    catch (Exception ex)
    {
        Console.WriteLine(string.Format("Error in cleaning html. {0}",
ex.Message));
    }
    var content = hpResult.QuerySelector("body");
    return (content).InnerHtml;
}
```

Código 11 – Remoção de código *HTML* não relevante

De seguida, é verificado se para o endereço URL em análise já existe uma sequência de pesquisa conhecida. Isso é feito utilizando o *id_enderecosUrl* e requisitando todos os registos da tabela *parserSequence* com o *fk_enderecosUrl* igual ao *id_enderecosUrl* do endereço URL em análise.

Esta tabela contém os atributos *id* dos elementos *HTML* recolhidos. Caso os mesmos existam, também guarda as classes *CSS* desses mesmos elementos, e guarda um histórico do identificador *id_parserSequence* no campo *parent*, por forma a saber-se qual a ordem de

Poi engine

pesquisa de elementos. Todas as sequências são atribuídas a um objeto do tipo *enderecosUrl*, por forma a identificar a quem pertence a sequência.

Caso não existam registos na tabela *parserSequence*, terá que se processar todo o mecanismo de pesquisa de informação referente a *POIs*, caso contrário, pode utilizar-se a sequência de elementos *HTML* conhecida, retirando o peso de processamento da pesquisa de informação.

Pesquisa de *POI's* sem sequência:

O resultado da função representada em Código 11, é enviado à função *ExtractHtmlContentForPoi* que recebe o conteúdo *HTML* como parâmetro. Esta função utiliza a biblioteca *AngleSharp* e procura por elementos *HTML* do tipo âncora (*a*), tentando desta forma identificar os atalhos para as páginas dos eventos. O excerto de Código 12 demonstra a obtenção de elementos *HTML* específicos.

```
HtmlParser hp = new HtmlParser();
    List<IElement> tags = new List<IElement>();
    List<string> nodeTypes = new List<string>();
    var hpResult = hp.Parse(input);

    List<string> textNodesValues = new List<string>();
    try
    {
        //foreach (string keyword in keywords)
        tags.AddRange(hpResult.QuerySelectorAll("a"));
    }
```

Código 12 – Recolha de objetos através do tipo de elemento *HTML*

Após encontrar todos os elementos *HTML* do tipo âncora, percorre-se cada um desses elementos, e procura-se o primeiro elemento pai com mais do que um filho. Assume-se que os vários elementos *HTML* com a informação necessária à criação de pontos de interesse se encontrem dentro de uma estrutura comum.

Após encontrar o elemento possivelmente contendo a informação necessária à criação de um *POI*, percorrer-se-ão os elementos filhos analisando o conteúdo de texto e o tipo de elemento *HTML*, na tentativa de, assim, se encontrarem os elementos necessários à criação de pontos de interesse. O tipo de elemento *HTML* ajuda a especificar o tipo de conteúdo que se pode encontrar, por exemplo nos elementos do tipo âncora pretende-se retirar o atributo *href*, nos elementos *div*, *span*, *strong* ou *p* apenas será necessário o conteúdo de texto.

Tendo em conta o tipo de elemento *HTML*, pode-se então invocar métodos de comparação separados, sendo que o formato de uma data é muito particular, pode-se verificar a existência

da mesma no conteúdo através de algumas expressões regulares. Para comparações de texto, ter-se-á em conta fatores como o tamanho do conteúdo de texto, sendo que o texto mais longo tem maior probabilidade de ser o atributo conteúdo do ponto de interesse e um texto mais curto, a descrição. Para o título compara-se com algumas palavras chave, uma vez que os eventos formativos contêm títulos com algumas características comuns (i.e., curso, congresso).

No caso de se conseguir criar um ponto de interesse com sucesso, será armazenada a sequência de elementos *HTML*, esta informação deverá estar armazenada na tabela *parserSequence* da base de dados os campos e regras desta tabela podem ser vistos na Tabela 13.

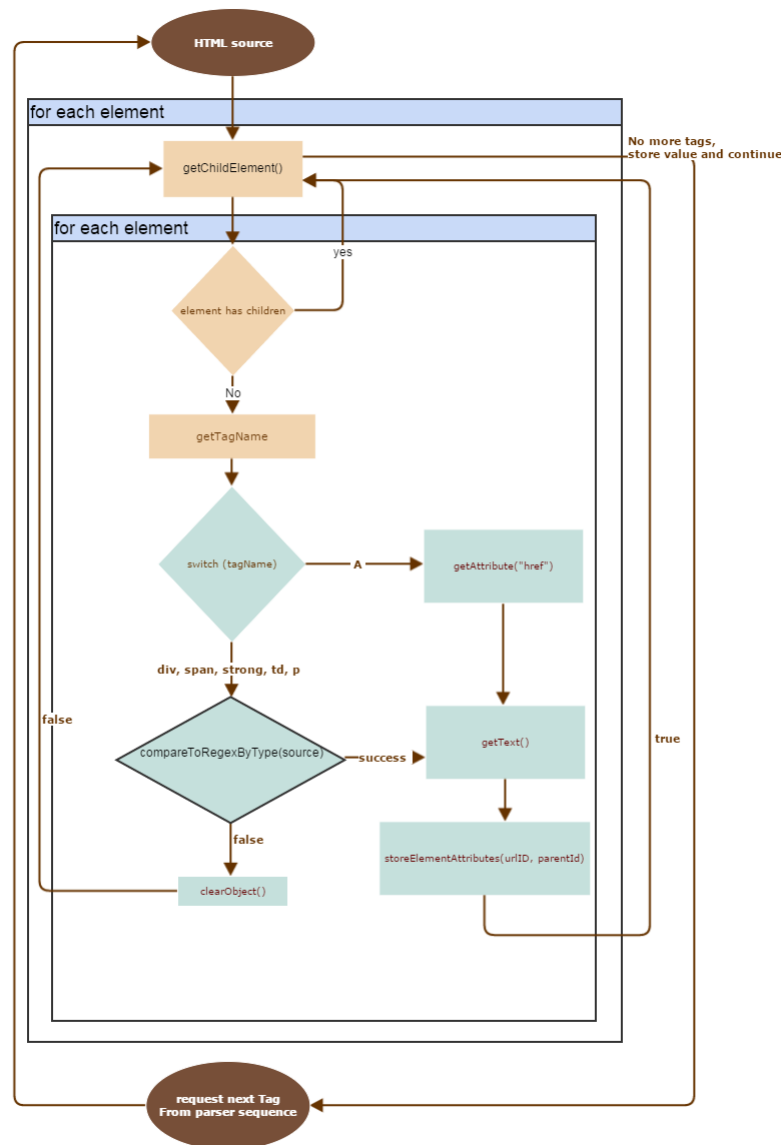


Figura 24 - Fluxograma de pesquisa sem sequência de elementos

Pesquisa de POI's com sequência:

Este tipo de pesquisa tem menos processamento, uma vez que se consegue retirar a informação quase diretamente. A Figura 25 mostra uma sequência de informação.

```
34 SELECT * FROM parserSequence where fk_enderecosUrl=41;
```

Grid view Form view

Total rows loaded: 4

	id_parserSequence	fk_enderecosUrl	parent	taqName	taqId	taqClass
1	25	41	0	tr	NULL	NULL
2	26	41	25	td	NULL	NULL
3	27	41	26	img	NULL	NULL
4	28	41	26	p	NULL	NULL

Figura 25 – Registos sequência elementos *HTML*

Para tornar o processo automático, o processamento desta sequência irá tomar partido de uma função recursiva. Sendo que irá tratar cada um dos registos independentemente.

Esta função recebe como parâmetro o registo, seguindo o exemplo na Figura 25, com *fk_enderecosUrl* 41 e como *parent* 0. Este registo irá obrigar a que seja o elemento pai comum, e a informação relevante ao ponto de interesse, encontrar-se-á nos elementos filhos.

Esse método irá verificar a existência de nós filhos, e caso os tenha, requisita à base de dados os nós com *fk_enderecosUrl* 41 e *parent* 25, e invocar-se-á a si mesmo, com este novo registo como parâmetro, e sucessivamente até o elemento em questão não ter elementos filhos. Quando for esse o caso, dependendo do tipo de elemento, a informação a recolher será diferente, como podemos ver na Figura 26, por exemplo para um elemento do tipo âncora será necessário recolher o valor a apresentar, e verificar se o mesmo é texto ou algum tipo de elemento *HTML*, e também o atributo *href* que contém o URL para uma especificação do evento mais detalhada, é também necessário verificar se o endereço é parcial ou completo. No caso de ser um elemento do tipo texto (i.e., *div*, *span*, *p*, *strong*), apenas deverá ser necessário recolher o texto. No final deverá ser guardado o texto, e para identificar que tipo de atributo é, deverá ter em conta fatores como tamanho do texto para o conteúdo, comparar a existência das palavras chave de pesquisa para identificar com aproximação o

titulo, pesquisar a data através de expressões regulares, e por sua vez converter para um formato por defeito. A categoria, pode ser comparada com as existentes. Embora este algoritmo não pareça muito eficiente, parece funcional e requer menos processamento que o algoritmo sem sequência.

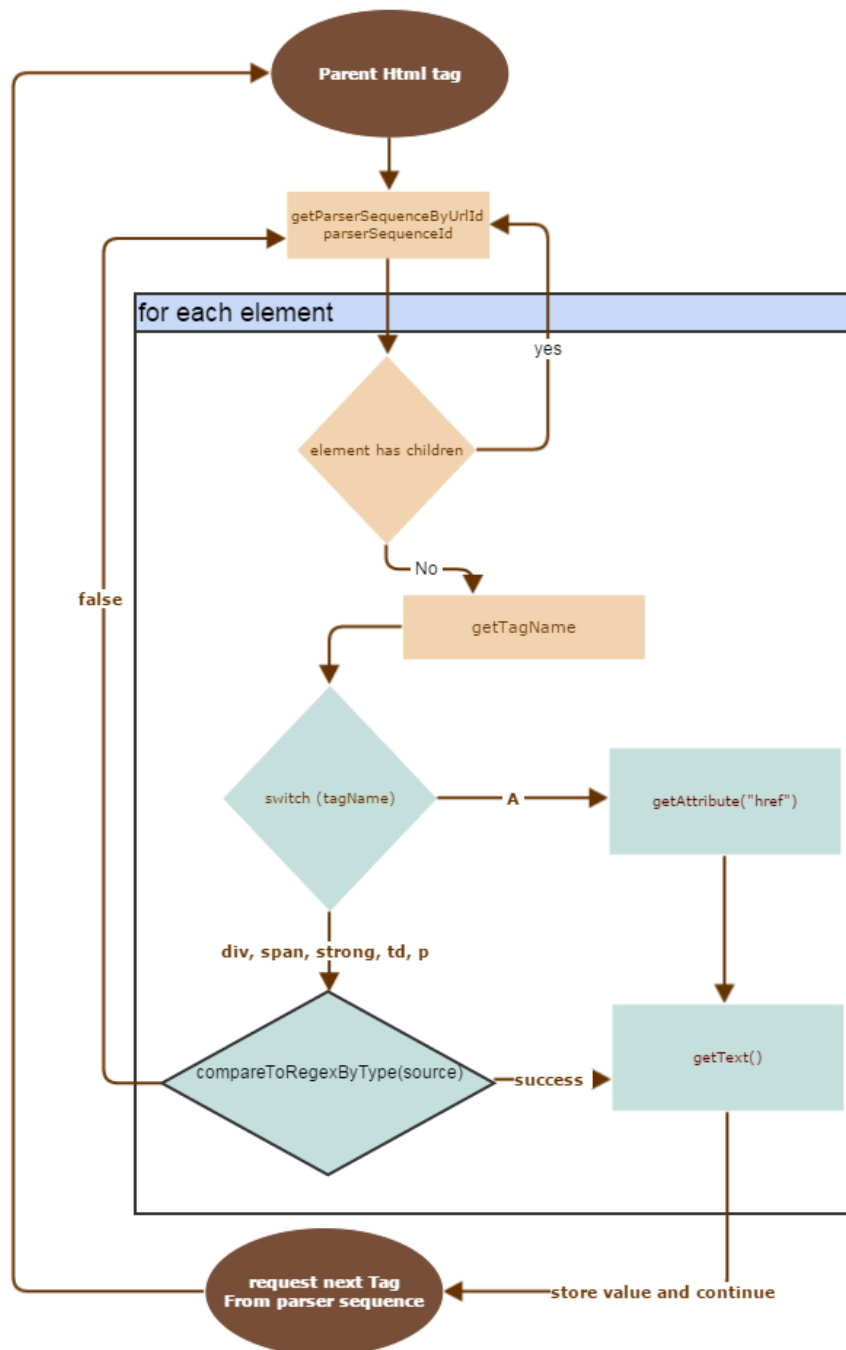


Figura 26 – Fluxograma de pesquisa com sequência de elementos

7 Avaliação de resultados

Neste capítulo irão ser avaliadas as diversas formas implementadas de recolha de informação para pontos de interesse. Será feito um estudo estatístico das mesmas e concluída qual a melhor abordagem.

7.1 Grandezas a avaliar

A grandeza a ser avaliada será a quantidade de pontos de interesse conseguida por cada uma das formas disponíveis para criar pontos de interesse, existindo até ao momento duas desenvolvidas, sendo elas por RSS e por envio de ficheiro CSV.

7.1 Hipóteses de teste e metodologias de avaliação

Para decidir qual o melhor modelo de classificação iremos comparar os algoritmos de criação de pontos de interesse com a taxa de sucesso e insucesso dos mesmos. Assim, teremos como hipótese nula a igualdade das medidas e como hipótese alternativa a solução proposta obtém melhores resultados.

7.2 Testes estatísticos

Para suportar as afirmações relativas à comparação de modelos serão usados testes estatísticos. No caso em estudo, serão utilizados os testes não paramétricos para amostras emparelhadas. Para testes não paramétricos será usado o teste de *Wilcoxon signed rank test for paired samples*.

Para realizar os testes aos dois algoritmos de criação de pontos de interesse, foram criados 12 casos de teste para cada um dos métodos. Cada linha da Tabela 14 representa um caso de teste e os valores encontrados nas células, o número de pontos de interesse criados no decorrer do teste.

Então em teoria, para este teste temos que ter uma hipótese nula H_0 e uma outra alternativa H_a , para o caso de teste em estudo foi definido:

- H_0 : é aceitável se não houver diferença entre os algoritmos ($med1 = med2$)
- H_a : existe uma diferença entre os algoritmos ($med1 < med2$)

O valor de α foi definido a 0.05.

Tabela 14 – Dados obtidos para análise estatística

Test	RSS	CSV	Diff	Abs Diff	Rank of Abs Diff	Positive Ranks	Negative Ranks
1	12	10	2	2	3	3	
2	83	20	63	63	12	12	
3	68	30	38	38	10	10	
4	38	24	14	14	7	7	
5	26	25	1	1	1,5	1,5	
6	23	30	-7	7	4		4
7	41	40	1	1	1,5	1,5	
8	59	50	9	9	5	5	
9	42	55	-13	13	6		6
10	51	25	26	26	8	8	
11	67	35	32	32	9	9	
12	83	30	53	53	11	11	
Total:						68	10

Após a inserção dos valores obtidos dos algoritmos, ocorreram uns cálculos apresentados na Tabela 14. Foi calculada a diferença entre os algoritmos em cada tentativa, o sinal do resultado indica qual dos algoritmos obteve mais pontos de interesse, a diferença absoluta indica o valor absoluto da coluna *difference*.

Após obter estes valores, é necessário calcular a classificação e a classificação sinalizada, entre os dois algoritmos.

De seguida são calculados os somatórios positivo e negativo da classificação sinalizada, sendo no caso em estudo 68 e 10 respetivamente, e é preenchida a Tabela 15, com cálculos e valores retirados da Tabela 14.

Tabela 15 – Valores obtidos para processar o teste

α	0,05
tails	2
n	12
T	10
T-crit	116
mean	39
variance	162,5
std dev	12,74755
z-score	2,274947
T-crit	13,96526
p-value	0,022909
sig	yes

Real Statistics
functions

SignRank	10
SRTEST	0,011455
2 tail test	0,022909
Effect size	
r	0,293694

Destes somatórios calcula-se o teste estatístico, que é o valor absoluto mínimo de ambos os somatórios, ou seja, o valor de T será 10, como mostra a Tabela 15.

Tabela 16 – Resultados obtidos teste *Wilcoxon signed rank test for paired samples*

Wilcoxon Signed-Rank Test for Paired Samples					
	RSS	CSV	one tail	two tail	
median	46,5	30	alpha	0,05	
			mean	39	
count	12		std dev	12,74755	
# unequal	12		z-score	2,274947	
T+	10		effect r	0,464372	
T-	68		T-crit	18,03215	14,01526
T	10		p-value	0,011455	0,022909
			sig (norm)	yes	yes
			T-crit	17	13
			sig (table)	yes	yes
			SRankPROB	0,011765	0,023529

Os resultados obtidos no teste para estes dados foram os encontrados na Tabela 16, e com estes valores, pode-se concluir que a hipótese H_0 é rejeitada, uma vez que o valor de $p\text{-value} < \alpha$, havendo assim provas suficientes que sugerem que existe uma diferença significativa entre os algoritmos.

8 Conclusões

Neste capítulo serão apresentadas as conclusões atingidas e o trabalho futuro para potencializar o valor do projeto.

8.1 Objetivos atingidos

Tal como apresentado neste documento, a solução pretendida tem foco em fornecer pontos de interesse formativos para a área da saúde, a sistemas de recomendação turísticos. Este documento relata o estudo e desenvolvimento de um protótipo de aplicação que pesquisa e recolhe a informação com vista à criação destes pontos de interesse.

Foram desenvolvidos neste projeto, alguns módulos de pesquisa da informação pretendida. Estes módulos utilizam serviços externos de pesquisa, poupando assim o esforço de processar algoritmos de pesquisa na aplicação.

Outro objetivo atingido, foi a implementação de um módulo de leitura de *feeds*, recolhendo assim de forma simples, informações necessárias à criação de pontos de interesse.

Foi também iniciado o desenvolvimento, de um *web scrapper*, que tinha o objetivo de retirar dados para criação de pontos de interesse do código fonte de páginas de internet, estando até ao momento a conseguir recolher alguns elementos *HTML* importantes. No entanto, os algoritmos de comparação não foram ainda desenvolvidos.

Apreciação final

Foi também desenvolvida uma *API* Rest, que disponibiliza os pontos de interesse. Estando ainda numa fase de desenvolvimento inicial, fornece apenas os pontos de interesse sem filtros de pesquisa.

Por fim, foi parcialmente desenvolvido um portal de internet com uma funcionalidade de criação de pontos de interesse por submissão de ficheiros *CSV*.

Dada a dimensão e complexidade do projeto e também as alterações de requisitos durante o processo de desenvolvimento, é de notar que existe ainda espaço para melhorias e trabalho futuro. Todavia, o projeto no seu estado atual, permite a recolha de pontos de interesse através da tecnologia *RSS*, com pesquisa automática de fontes, e permite também a criação de pontos de interesse manual através de um portal de internet.

8.2 Trabalho futuro

A próxima etapa deste projeto passa pela implementação de algoritmos de comparação de texto, por forma a permitir a criação de pontos de interesse com base em páginas de internet de forma automática. É também necessária a implementação dos componentes em falta no portal de internet, nomeadamente o módulo de *ACL*, por forma a controlar os acessos ao portal de internet e também à *API*. São também necessárias as implementações dos módulos de gestão de utilizadores e grupos, e também, a funcionalidade de edição e validação de pontos de interesse. Sem estas funcionalidades concluídas, não é possível tornar este protótipo numa aplicação, uma vez que a informação não é validada.

Será também necessário, desenvolver melhoramentos ao código por forma a potenciar a performance da aplicação. Deverão também ser desenvolvidos testes de carga e funcionais à aplicação.

8.3 Apreciação final

Este projeto foi desenvolvido desde o seu início, o que exigiu a aplicação de matérias lecionadas durante o Mestrado de Engenharia Informática, assim como o estudo de outras técnicas e vertentes necessárias à realização deste projeto.

Devido ao gosto de explorar novas técnicas e para tornar este projeto também numa forma de crescimento técnico, existiram inúmeras situações de aprendizagem transversais a todas as matérias envolventes a este projeto.

Com o período de tempo disponível foi conseguido um bom planeamento de uma estrutura capaz de albergar as funcionalidades implementadas. Foram também concluídas algumas das funcionalidades planeadas, que permitem a fornecer os pontos de interesse pretendidos. Este projeto mostrou-se uma ótima experiência a nível técnico pelos conhecimentos adquiridos. Estes fatores trazem uma grande satisfação profissional e pessoal.

Referências

- Adomavicius, G., Rokach, L. & Shapira, B., 2011. Recommender Systems Handbook. In *Media*. pp. 217–253. Available at: <http://www.springerlink.com/index/10.1007/978-0-387-85820-3> [Accessed February 7, 2016].
- Almeida, A., 2006. Technical Manual - User Modeling and Recommender System 's Data Access Layer. , pp.1–32.
- Almeida, A., 2009. ToursPlan. Available at: <https://www.gecad.isep.ipp.pt:9090/projects/toursplan/Default.aspx> [Accessed February 6, 2016].
- Amazon, L., Guo, X. & Xia, Z., 2012. The Architecture of Open Source Applications. *University of California Berkeley*. Available at: <http://software-carpentry.org/2011/05/06/> [Accessed February 14, 2016].
- Anacleto, R. et al., 2010. PSiS Mobile. *ICWN'10 - The 2010 International Conference on Wireless Networks. Abstract*, p.112. Available at: <http://recipp.ipp.pt/handle/10400.22/2720> [Accessed February 7, 2016].
- Barnes, C., Blake, H. & Pinder, D., 2009. *Creating and Delivering Your Value Proposition: Managing Customer Experience for Profit*, London and Philadelphia. Available at: http://books.google.com/books?id=8d73Cpt_khwC&pgis=1 [Accessed January 24, 2016].
- Berka, T., & Plößnig, M., 2004. *Designing Recommender Systems for Tourism*, Salzburg. Available at: http://www.salzburgresearch.at/wp-content/uploads/2010/10/enter_ploessnig.pdf.
- Bhattacharya, S. & Basu, J., 2007. Distress , Wellness and Organizational Role Stress among IT Professionals : Role of Life Events and Coping Resources. *Journal of the Indian Academy of Applied Psychology*, 33(2), pp.169–178. Available at: <http://medind.nic.in/jak/t07/i2/jakt07i2p169.pdf> [Accessed January 29, 2016].
- Bogers, T. & Van Den Bosch, A., 2009. *Collaborative and content-based filtering for item recommendation on social bookmarking websites*, Available at: <http://www.springerlink.com/index/10.1007/978-0-387-85820-3>.
- Bolin, M. et al., 2005. Automation and customization of rendered web pages. *Proceedings of the 18th annual ACM symposium on User interface software and technology - UIST '05*, p.163. Available at: <http://dl.acm.org/citation.cfm?id=1095034.1095062> [Accessed February 14, 2016].
- Brandstetter, T., 2011. 2 plan Work 2-gether Agile taskboard. Available at: <http://to-do-2-plan.com/free-task-board> [Accessed February 13, 2016].
- Carneiro, J.M.N., 2013. RealCrono –sistema de difusão de resultados em tempo real na web. Available at: <http://recipp.ipp.pt/handle/10400.22/5849> [Accessed February 14, 2016].
- Cheong Christopher, Cheong France, F.J., 2013. USING DESIGN SCIENCE RESEARCH TO INCORPORATE GAMIFICATION INTO LEARNING ACTIVITIES. In *Pacific Asia Conference on Information Systems*. Victoria, p. 14. Available at: <http://www.pacis-net.org/file/2013/PACIS2013-156.pdf> [Accessed December 29, 2015].
- Ciurana Simó, E.R., 2012. Development of a tourism recommender system. , p.1 recurs electrònic. Available at: <http://hdl.handle.net/2099.1/16444>.

- Coelho, B. da S., 2010. *User Modeling and Recommendation Strategies for Tourism*. Instituto Superior de Engenharia do Porto. Available at: <http://recipp.ipp.pt/handle/10400.22/1981> [Accessed October 12, 2016].
- Cording, P.H., 2011. *Algorithms for Web Scraping*. Technical University of Denmark. Available at: www.imm.dtu.dk [Accessed February 14, 2016].
- Costa, R., Ferreira, M. & Nicola, S., 2012. Utilização De Metodologias. , p.118. Available at: http://recipp.ipp.pt/bitstream/10400.22/2817/1/DM_RuiCosta_2012_MEEC.pdf [Accessed January 26, 2016].
- Coyle, L. & Cunningham, P., 2003. Exploiting Re-ranking Information in a Case-Based Personal Travel Assistant 1.
- Dudhe, A. & Sherekar, S., 2014. Performance Analysis of SOAP and RESTful Mobile Web Services in Cloud Environment. , pp.1–4. Available at: <http://research.ijcaonline.org/rtinfosec/number1/rtinfosec1401.pdf>.
- Edelson, H., 2011. Routing Topologies for Performance and Scalability with RabbitMQ. Available at: <http://blog.springsource.org/2011/04/01/routing-topologies-for-performance-and-scalability-with-rabbitmq/> [Accessed February 14, 2016].
- Erhun, F. & Keskinocak, P., 2003. *Game theory in business applications*, STANFORD. Available at: http://web.stanford.edu/~ferhun/paper/GT_Overview.pdf [Accessed January 26, 2016].
- Formacaosaude.com, 2016. Category Archive for “Formação” | FormacaoSaude.com. Available at: <http://www.formacaosaude.com/categoria/formacao/> [Accessed February 15, 2016].
- Gomes, a R. & Cruz, J.F., 2004. *A experiência de stress e “burnout” em psicólogos portugueses: um estudo sobre as diferenças de género*, CIPsi. Available at: <http://repositorium.sdum.uminho.pt/bitstream/1822/3944/4/2-Gomes&Cruz-SO-Revista-UM.pdf> [Accessed February 7, 2016].
- Google, 2016. Google Advanced Search. Available at: https://www.google.com/advanced_search.
- de Graaff, V., van Keulen, M. & de By, R.A., 2012. Towards geosocial recommender systems. In *Proceedings of the 4th International Workshop on Web Intelligence & Communities - WI&C '12*. New York, New York, USA: ACM Press, p. 1. Available at: <http://dl.acm.org/citation.cfm?doid=2189736.2189748> [Accessed October 15, 2016].
- Greco, S., Matarazzo, B. & Slowinski, R., 2001. Rough sets theory for multicriteria decision analysis. *European Journal of Operational Research*, 129(1), pp.1–47. Available at: http://sedok.narod.ru/s_files/poland/22.pdf [Accessed February 6, 2016].
- Harris, A. (Ana L.N. de C.H., 1999. *Revisão da Teoria dos Sistemas Nebulosos*. Escola Politécnica da Universidade de São Paulo. Available at: http://www.fec.unicamp.br/~luharris/tese/arquivos/Harris-ALNC_Cap3.pdf [Accessed January 28, 2016].
- Inc., R.H., 2012. ActiveMQ Tuning Guide. Available at: https://access.redhat.com/documentation/en-US/Fuse_ESB_Enterprise/7.1/html-single/ActiveMQ_Tuning_Guide/index.html [Accessed February 14, 2016].
- Levesque, T., 2009. Build an RSS reader in 5 minutes » Thomas Levesque’s .NET blog. Available at: <http://www.thomaslevesque.com/2009/02/13/build-an-rss-reader-in-5-minutes/> [Accessed February 15, 2016].

Apreciação final

- Luz, N., Anacleto, R. & Almeida, A., (1) Tourism Mobile and Recommendation Systems - A State of the Art | Nuno Luz - Academia.edu. Available at: https://www.academia.edu/1292794/Tourism_Mobile_and_Recommendation_Systems_-_A_State_of_the_Art [Accessed December 31, 2015].
- Luz, N., Anacleto, R. & Almeida, A., 2010. Tourism Mobile and Recommendation Systems - A State of the Art. In *EEE'10*. Porto, p. 3. Available at: http://recipp.ipp.pt/bitstream/10400.22/1467/1/COM_NunoLuz_2010_GECAD.pdf.
- McColl, R.C. et al., 2014. A performance evaluation of open source graph databases. *Proceedings of the first workshop on Parallel programming for analytics applications - PPAA '14*, pp.11–18. Available at: <http://dl.acm.org/citation.cfm?id=2567634.2567638> [Accessed February 16, 2016].
- Microsoft Developer Network (Microsoft), 2016. Advanced Operator Reference. Available at: <https://msdn.microsoft.com/en-us/library/ff795620.aspx> [Accessed October 20, 2016].
- Mun, W.K. et al., 2015. Medical Tourism Destination SWOT Analysis : A Case Study. In *SHS Web of Conferences12*. Kuala Lumpur: EDP Sciences, p. 8. Available at: <http://www.shs-conferences.org>.
- Nahon, J., 2011. A Comparative Analysis of REST and SOAP. , (August), p.64.
- Nicola, S., 2016. Análise de valor de negócio. , p.28. Available at: <https://moodle.isep.ipp.pt/course/view.php?id=3709>.
- Nicola, S., Ferreira, E.P. & Ferreira, J.J.P., 2012. A novel framework for modeling value for the customer, an essay on negotiation. *International Journal of Information Technology and Decision Making*, 11(3), pp.661–703. Available at: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84863473238&partnerID=40&md5=fadbee5462f209d6e32a60d695c55a0c> [Accessed January 31, 2016].
- Nicola, S., Ferreira, E.P. & Ferreira, J.J.P., 2014. A Quantitative Model for Decomposing & Assessing the Value for the Customer. *Journal of Innovation Management*, 1, pp.104–138. Available at: https://sigarra.up.pt/fbaup/pt/publs_pesquisa.FormView?P_ID=114690 [Accessed January 26, 2016].
- Nicola, S., Pinto, E. & Ferreira, J.J.P., 2013. J. J. Pinto Ferreira ASSETS MANAGEMENT - A CONCEPTUAL MODEL DECOMPOSING VALUE FOR THE CUSTOMER AND A QUANTITATIVE. *International Journal for Quality Research* 9(1) 89–106 ISSN 1800-6450, 9(1), pp.89–106. Available at: <http://www.ijqr.net/journal/v9-n1/7.pdf> [Accessed January 17, 2016].
- Nunes, H., Almeida, A. & Martins, C., 2016. Gathering data for professional tourism points of interest. In *Proceedings of the Ninth International C* Conference on Computer Science & Software Engineering - C3S2E '16*. New York, New York, USA: ACM Press, pp. 125–126. Available at: <http://dl.acm.org/citation.cfm?doid=2948992.2948996> [Accessed October 17, 2016].
- Osterwalder, A., 2004. The Business Model Ontology. , p.172. Available at: http://www.hec.unil.ch/aosterwa/PhD/Osterwalder_PhD_BM_Ontology.pdf [Accessed February 16, 2016].
- Pant, G., Srinivasan, P. & Menczer, F., 2004. *Crawling the Web*, Bloomington, Iowa. Available at: https://www.google.pt/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0ahUKEwjgmybhp_KAhWPhhoKHYVHDzwQFggqMAE&url=http://www.springer.com/?SGWID=4-102-45-113505-p28710520&usq=AFQjCNGNisZO3G6d4Z1Xo2ICHR13R25WjQ&sig2=90EOKplzUNUbZJBxqiVDhA&bvm=bv.
- Pawlak, Z., 1982. Rough sets. *International Journal of Computer & Information Sciences*, 11(5), pp.341–356. Available at: <http://sci2s.ugr.es/keel/dataset/includes/catlmbFiles/1982 - Pawlak.pdf>

[Accessed January 28, 2016].

- Pazzani, M. & Billsus, D., 2007. Content-based Recommendation Systems. *Content-based Recommendation Systems Michael*, p.18.
- Penman, R.B., Baldwin, T. & Martinez, D., 2009. Web Scraping Made Simple with SiteScraper. *Text*. Available at: file:///Users/heldernunes/Downloads/sitescraper.pdf [Accessed February 14, 2016].
- Pessoa, A. & Simões, J., 2003. Mineração de Dados Espaço-Temporal Aplicada a Previsão Climática Utilizando a Teoria dos Conjuntos Aproximativos. , p.6. Available at: http://mtc-m16c.sid.inpe.br/col/lac.inpe.br/worcap/2003/10.30.13.19/doc/worcap_versaofinal_alex2003.pdf [Accessed January 28, 2016].
- Potti, P.K., 2011. On the Design of Web Services: SOAP vs. REST. *Master Thesis*, p.106. Available at: http://digitalcommons.unf.edu/cgi/viewcontent.cgi?article=1139&context=etd.
- Ricci, F., Rokach, L. & Shapira, B., 2011. Introduction to Recommender Systems Handbook. In *Recommender Systems Handbook*. Boston, MA: Springer US, pp. 1–35. Available at: http://link.springer.com/10.1007/978-0-387-85820-3_1 [Accessed October 15, 2016].
- Rostański, M., Grochla, K. & Seman, A., 2014. Evaluation of highly available and fault-tolerant middleware clustered architectures using RabbitMQ. *2014 Federated Conference on Computer Science and Information Systems, FedCSIS 2014, 2*, pp.879–884. Available at: http://www.scopus.com/inward/record.url?eid=2-s2.0-84912058920&partnerID=tZotx3y1 [Accessed February 14, 2016].
- Silva, A. (Publico), 2014. Peso do turismo na economia do país está muito acima da média mundial. *Publico*. Available at: http://www.publico.pt/economia/noticia/turismo-pesa-mais-no-emprego-e-na-economia-em-portugal-do-que-no-resto-do-mundo-1629417 [Accessed December 6, 2015].
- Smith, H. (World H.O., Brown, H. (World H.O. & Khanna, J. (World H.O., 2009. *Continuing education meetings and workshops: effects on professional practice and health-care outcomes*, Geneva: Brown H; Khanna J; Smith H; Available at: http://apps.who.int/rhl/effective_practice_and_organizing_care/Cd003030_smithh_com/en/ [Accessed December 19, 2015].
- Snyder, B., Bosanac, D. & Davies, R., 2011. ActiveMQ in Action. *Online*, p.406. Available at: http://eds.a.ebscohost.com/eds/detail/detail?vid=1&sid=19393cc0-88df-4ae2-b259-f7c3830e5640%40sessionmgr4001&hid=4210&bdata=Jmxhbmc9cHQYnlmc2l0ZT1lZHMtbGl2ZSZZY29wZT1zaXRl#AN=edsgcl.268249116&db=edsgao [Accessed February 14, 2016].
- SQLite, 2016. SQLite Database Speed Comparison. Available at: https://www.sqlite.org/speed.html [Accessed February 16, 2016].
- Swarbrooke, J. & Horner, S., 2001. *Business Travel and Tourism*, Oxfordshire: Routledge. Available at: https://books.google.com/books?hl=pt-PT&lr=&id=6Dgek2vv8fEC&pgis=1 [Accessed January 29, 2016].
- Tanscheit, R., 2012. Sistemas Fuzzy. *LN Sistemas - DEE-PUC-Rio*, 113, pp.113–134. Available at: http://www2.ica.ele.puc-rio.br/Downloads/41/LN-Sistemas Fuzzy.pdf [Accessed January 28, 2016].
- Vaishnavi, V. & Kuechler, B., 2004. Design Science Research in Information Systems Overview of Design Science Research. *Ais*, p.45.
- Web, C. et al., 2013. Custom Web Scrapping For Content Aggregation From E-Commerce Websites *1,2. , 2(9). Available at: http://www.ijesrt.com/issues pdf file/Archives 2013/September-

Apreciação final

2013/56.pdf [Accessed February 14, 2016].

Yang, Y. et al., 2014. How Does Perceived Value Affect Travelers ' Satisfaction and Loyalty ? *Social Behavior and Personality*, 42(10), pp.1733–1744. Available at: <http://eds.b.ebscohost.com/eds/pdfviewer/pdfviewer?sid=fded06e2-a736-4a30-ace3-f6ddc5bcc601%40sessionmgr113&vid=2&hid=108> [Accessed January 24, 2016].

Zadeth, L., 1965. *Fuzzy Sets-Information and Control-1965*, Berkeley. Available at: <http://www.cs.berkeley.edu/~zadeh/papers/Fuzzy Sets-Information and Control-1965.pdf> [Accessed February 6, 2016].

Zukerman, I. et al., 2001. Predictive Statistical Models for UserModeling. *User Modeling and User-Adapted Interaction*, 11 (1-2)(1), pp.5–18.

Anexos

Algoritmos de Web crawling

Web *crawlers*, são programas que exploram a estrutura gráfica da web por forma a navegarem de página de internet em página de interna, historicamente os web *crawlers* foram também conhecidos como *wanderers*, *robots*, *spiders*, *fish*, e *worms* (Pant et al. 2004).

Uma das vantagens desta tecnologia, o custo de performance, sendo este bastante rápido, conseguindo em alguns casos, pesquisar milhares de páginas em alguns minutos, esta performance varia mediante a complexidade da pesquisa, e largura de banda. O objetivo dos web *crawlers*, é adquirir páginas de internet para repositórios locais, sendo que começa a recolher os dados através de um endereço inicial, e vai navegando, através dos atalhos encontrados, de página para página recolhendo também os seus conteúdos (Pant et al. 2004).

Segundo o estudo “*Crawling the Web*” realizado por Gautam Pant, Padmini Srinivasan e Menczer Filippo da universidade de Iowa, alguns dos mais interessantes algoritmos de web *crawling* são: *Naive Best-First Crawler*, *SharkSearch*, *Focused Crawler* e *Context Focused Crawler*, usando todos eles paradigmas e abordagens distintas.

- ***Naive Best-First Crawler***

A abordagem de pesquisa deste *crawler* é baseada num vetor de palavras chave por frequência de ocorrências. Os resultados são apresentados após um calculo estatístico entre a similaridade do cosseno entre os conteúdos da página e a descrição do fornecida pelo utilizador, sendo atribuída uma pontuação aos endereços não visitados nesta página através deste valor. Estes endereços são então armazenados num repositório por prioridade baseada nestas pontuações. Este repositório é então utilizado nas pesquisas seguintes, sendo estas feitas por ordem de priorização do repositório. O cálculo da pontuação das páginas é feito através da similaridade do cosseno entre a página p e a *query* q fornecida pelo utilizador:

$$\text{sim}(q, p) = \frac{V_q \cdot V_p}{\|V_q\| \cdot \|V_p\|}$$

onde V_q e V_p , são o termo de frequência (TF) dos vetores de ocorrências da *query* de pesquisa e das páginas. $V_q \cdot V_p$ é o produto dos dois vetores e $\|v\|$ é a norma de Euclides do vector V .

Houve ao longo dos anos melhoramentos neste algoritmo, nomeadamente a inserção de *multi threads* no *crawler* e com a possibilidade de definir a profundidade da pesquisa. Este algoritmo passou a ser conhecido como *best-N-first*, onde N é o número de *threads* disponíveis (Pant et al. 2004).

- ***SharkSearch***

Este algoritmo usa um método de similaridade idêntico ao do algoritmo *Naive Best-First Crawler*, na estimativa da relevância dos endereços das páginas de internet. Tem, no entanto, um sistema de pontuações mais refinado. O texto âncora, texto à volta de links e pontuações herdadas de páginas anteriores, pode influenciar a pontuação de links.

O *SharkSearch*, contém um título de profundidade, que permite ao *crawler* para a pesquisa se encontrar alguma página considerada não importante. Para o controlo de todas as informações de cada endereço URL definido, é associada uma profundidade e uma pontuação inicial a esses endereços. O título de profundidade é definido inicialmente pelo utilizador, enquanto a pontuação de um endereço não visitado é definida como:

$$score(url) = \gamma . inherited(url) + (1 - \gamma) . neighborhood(url)$$

Onde $\gamma < 1$ é um parâmetro, a pontuação *neighborhood* significa que foi encontrado um endereço url na página e o *inherited* é a pontuação calculada a partir da pontuação herdada dos endereços anteriores, e é calculada através da fórmula:

$$inherited(url) = \begin{cases} \delta . sim(q,p); & se\ sim(q,p) > 0 \\ \delta . inherited(p); & senão \end{cases}$$

Onde $\delta < 1$ é um parâmetro, q é a query e p é a página de onde foi retirado o endereço URL da página corrente.

A pontuação *neighborhood* utiliza o texto da âncora e o texto à volta numa tentativa de refinar a pontuação global do endereço URL, permitindo a diferenciação entre links encontrados na mesma página. Para isso este algoritmo atribui uma pontuação ao link e ao contexto a todos os endereços. A pontuação do link é a similaridade do texto do link com a query q, por exemplo, $sim(q, anchor_text)$. A pontuação do contexto aborda o contexto do link por forma a incluir algumas palavras à volta do link. O contexto aumentado conseguido, *aug_context* é usado para calcular a pontuação contexto, através da formula:

$$context(url) = \begin{cases} 1; & se\ anchor(url) > 0 \\ sim(q, aug_context); & senão \end{cases}$$

No final, obtém-se a pontuação *neighborhood* através das pontuações de link e contexto da seguinte forma:

$$neighborhood(url) = \beta . anchor(url) + (1 - \beta) . context(url)$$

Onde $\beta < 1$ e é também um parâmetro. (Pant et al. 2004)

- **Focused Crawler**

A ideia por detrás deste *crawler* é classificar as páginas visitadas em categorias numa taxonomia de tópicos. Este algoritmo necessita de uma taxonomia de tópicos como o Yahoo ou a *Open Directory Project (ODP)*, e os endereços URL iniciais são fornecidos pelo utilizador. Estes endereços fornecidos pelo utilizador são automaticamente classificados para várias categorias da taxinomia. Esta classificação pode ser corrigida pelo utilizador, adicionar novas categorias à taxinomia, e influenciar as pontuações das categorias marcando-as como “boas”. O *crawler* utiliza os endereços URL fornecidos para criar um classificador *bayesiano* que calcula a probabilidade ($Pr(c|p)$) que uma página visitada p pertence à categoria c na taxinomia. A pontuação de relevância associada a cada página visitada é calculada como:

$$R(p) = \sum_{c \in \text{"boas"}} PR(c|p)$$

Quando o *crawler* está num modo de focos leve, utiliza a pontuação de relevância da página visitada para pontuar os endereços URL de lá extraídos. As pontuações dos endereços são adicionadas à lista de endereços analisada. Depois de uma forma similar ao algoritmo *Naive Best-First Crawler*, o algoritmo escolhe qual o melhor endereço URL a visitar a seguir. No modo de focos alto, para um endereço visitado p , o classificador bayesiano procura um nó folha c^* na taxinomia, com a probabilidade máxima de incluir p . Se nenhum dos endereços pai na taxinomia de c^* estiver marcado como “boa” pelo utilizador, então os endereços URL da página p visitada são extraídos e adicionados à lista de endereços (Pant et al. 2004).

- **Context Focused Crawler**

Este tipo de algoritmos utiliza um classificador *bayesiano* por forma a guiar o *crawler*. No entanto, os classificadores estimam a distância do link entre uma página visitada e outras páginas relevantes. Este tipo de estimativa é importante para remover informação não útil em pesquisas na internet por exemplo.

Este algoritmo utiliza um gráfico de contexto (*context graph*) de L camadas correspondentes a cada página fornecida. As páginas correspondentes aos links contidos na página fornecida, são inseridos na camada 1. Os links contidos nas páginas da camada 1 são inseridos na camada 2 e assim sucessivamente. Pode-se obter os endereços URL contidos numa camada através de um mecanismo de pesquisa. A Figura 27 retrata um gráfico de contexto para o endereço definido <http://www.biz.uiowa.edu/programs/>. Quando o algoritmo termina a pesquisa através das páginas definidas e recolhe as informações para os gráficos de contexto (*context graph*), as páginas da mesma camada (i.e. camada 2), e de cada gráfico de contexto, são combinados numa única camada.

Isto proporciona um novo conjunto de camadas chamada de mistura de gráficos de contexto (*merged context graph*). De seguida são inseridos num documento a concatenação das páginas definidas, daí são identificados os termos para definir um vocabulário que irá ser utilizado na classificação.

São criados alguns classificadores *Bayesianos*, um por camada na mistura de gráficos de contexto (*merged context graph*). Todas as páginas numa camada são usadas para calcular $Pr(t|c_l)$, que é a probabilidade de ocorrência do termo t na classe c_l correspondente à camada l . Uma probabilidade mais alta, $Pr(c_l) = 1/L$, é atribuída a cada classe onde L é o numero de camadas. A probabilidade de uma página p pertencer a uma classe c_l é calculada como $Pr(c_l|p)$. Estas probabilidades são calculadas para cada classe. A classe com a maior probabilidade é tratada como a classe vencedora (camada). No entanto, se a probabilidade da classe vencedora, for menor que o imite definido, a página visitada é classificada para a classe “outras”. Esta classe “outras” representa páginas que não se assemelham bem a nenhuma das classes do gráfico de contexto (*context graph*). Caso a classe vencedora exceder o limite definido, a página é classificada na classe vencedora. A Figura 27 demonstra as várias camadas do gráfico de contexto.

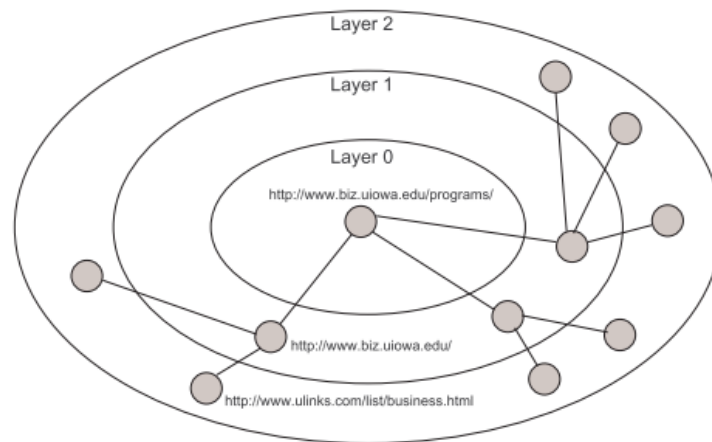


Figura 27 – Gráfico de contexto (*context graph*) (Pant et al. 2004)

O conjunto de classificadores correspondentes ao gráfico de contexto (*context graph*) providência um mecanismo para estimar a distância dos links de uma página visitada até outras páginas relevantes. O algoritmo mantém uma fila para cada classe, com as páginas visitadas e classificadas por classe. Cada fila é ordenada por pontuação de probabilidade $Pr(c_l|p)$. E quando o algoritmo precisar de um RUL para visitar, escolhe um URL no topo da página e coloca-o na fila com l mais pequeno. Assim tenderá a escolher primeiro páginas que pareçam

mais próximas das páginas relevantes. Os links externos dessas páginas serão explorados antes dos links externos de páginas que pareçam estar mais distanciadas (Pant et al. 2004).

Anexo 2

Filas de Mensagens (message queues)

Muitos sistemas tem a necessidade de comunicar entre si, no entanto muitas vezes, é necessário o envio de mensagens, da qual apenas necessitamos da certeza que as mesmas serão entregues. Neste tipo de situações não faz sentido a espera por uma resposta do outro sistema, onde muitas vezes o processo de resposta obriga ao tratamento da mensagem. Uma boa solução para este tipo de situações, é a implementação de filas de mensagens.

As filas de mensagens fornecem um protocolo de comunicação assíncrona permitindo ao produtor e ao consumidor de mensagens a comunicação sem interação entre sistemas, tornando-os assim independentes. A complexidade do sistema de filas de mensagens não se faz notar do lado do produtor e consumidor das mensagens, sendo que o produtor apenas coloca as mensagens, mediante critérios necessários, e o consumidor acede à fila de mensagens e processa-as tendo em conta os critérios definidos. (Carneiro 2013)

Neste subcapítulo iremos enumerar algumas tecnologias de filas de mensagens disponíveis no mercado. O estudo apresentado é baseado no conteúdo da tese de mestrado “RealCrono – sistema de difusão de resultados em tempo real na web”, do instituto Superior de Engenharia do Porto.

8.3.1.1 RabbitMQ

O *RabbitMQ* é um *message broker* e um servidor de filas (*queueing server*) que pode ser usado para diversas aplicações partilharem dados via protocolos comuns ou simplesmente colocar tarefas em filas para ser executadas por nós de cálculo de sistemas distribuídos. É uma aplicação de *middleware* que suporta vários protocolos de mensagens entre eles, os mais conhecidos *Streaming Text Oriented Messaging Protocol (STOMP)* e *Advanced Messaging Queuing Protocol (AMQP)* (Rostański et al. 2014).

Sendo que o *RabbitMQ*, implementa uma arquitetura de broker, significa que as mensagens são colocadas numa fila, num servidor de mensagens (*RabbitMQ*) para posteriormente serem entregues aos clientes.



Figura 28 – Esquema de comunicação RabbitMQ (Edelson 2011)

A Figura 28 demonstra o processo de comunicação de mensagens utilizando o *RabbitMQ*, sendo este o componente a laranja na figura. A utilização do *RabbitMQ* minimiza como roteamento, balanceamento de carga ou filas de mensagens persistentes (Carneiro 2013).

8.3.1.2 ZeroMQ

O *ZeroMQ* é um sistema de mensagens e não é uma aplicação, mas sim uma biblioteca de *sockets* similares aos TCP sockets, com a diferença que os *sokets* do *ZeroMQ* conseguem comunicar simultaneamente com vários *hosts* (Amazon et al. 2012). O seu funcionamento como demonstra a Figura 29, é com o envio de mensagens para *queues*, que serão posteriormente entregues aos recetores.

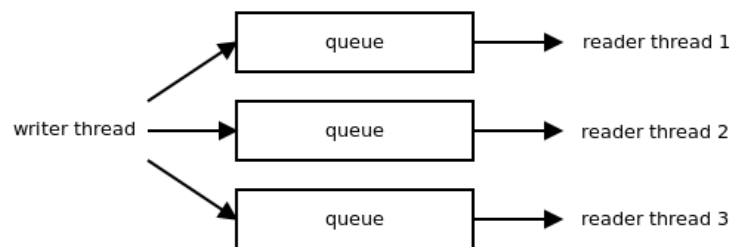


Figura 29 – Filas de comunicação ZeroMQ (Amazon et al. 2012)

O *ZeroMQ* é um sistema de mensagens leve, que implementa topologias de comunicação P2P, e é projetado especialmente para alto desempenho e cenários de baixa latência. A implementação de *zeroMQ* implica o conhecimento profundo de protocolos de comunicação, o que o torna de mais difícil utilização. (Carneiro 2013)

8.3.1.3 Apache ActiveMQ

Apache ActiveMQ é um broker de mensagens para comunicação remota entre sistemas através da especificação. Utiliza o JMS (*Java Message Service*). A tecnologia de implementação do *ActiveMQ* é Java, no entanto estão disponíveis API's em *C/C++*, *.NET*, *Perl*, *PHP*, *Python*, *Ruby*, entre outras (Snyder et al. 2011). Para além de disponibilizar um servidor broker de mensagens como o *RabbitMQ*, implementa também topologias do tipo P2P à semelhança do *ZeroMQ* (Carneiro 2013).

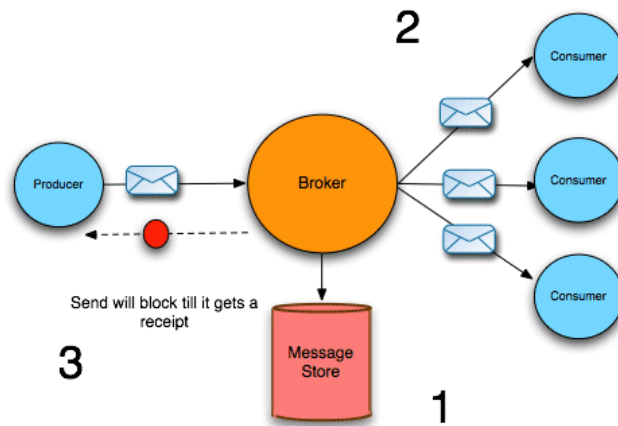


Figura 30 – Estrutura de comunicação do ActiveMQ (Inc. 2012)

O funcionamento do *ActiveMQ* está demonstrado na Figura 30, e similarmente com os casos anteriores, funciona como um *message broker* com o envio de mensagens para *queues*, que serão posteriormente entregues aos recetores.

No entanto, devido ao elevado número de funcionalidades que oferece a sua simplicidade na implementação acrescenta custos de desempenho (Carneiro 2013).

Anexo 3

Gathering data for professional tourism points of interest

Hélder Nunes
Institute of Engineering – Polytechnic
of Porto, Portugal
1030489@isep.ipp.pt

Ana Almeida
GECAD-Knowledge Engineering and
Decision Support Group
Institute of Engineering – Polytechnic
of Porto Porto, Portugal
amn@isep.ipp.pt

Constantino Martins
GECAD-Knowledge Engineering &
Decision Support Group Institute of
Engineering –
Polytechnic of Porto
Porto, Portugal
acm@isep.ipp.pt

Abstract - Being tourism a major economic and social factor in some countries, with a percentage of 2.9% of world GDP, it makes planning tourism routes and presentation of tourism points of interest to potential customers, very important. Many tourists plan their trips in advance using information gathered in Internet research. This action can be very stressful due to the excess of unnecessary information obtained in this kind of research.

There's also a fast growth of tourism interests in categories like sports, professional areas and medicinal tourism types, creating a business opportunity for tourism route planning tools. This opportunity also requires information for events in these specific tourism areas, in order to provide it to potential customers.

keywords - Recommendation system, professional tourism, points of interest.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.

Copyright is held by the owner/author(s).
C3S2E '16, July 20-22, 2016, Porto, Portugal
ACM 978-1-4503-4075-5/16/07.
<http://dx.doi.org/10.1145/2948992.2948996>

However this type of tourists is growing fast and there's a real business opportunity to provide this kind of information for tourist. This scientific study is based on this type of information gathering, creating the possibility to include in tourism recommendation systems, the capacity to provide professional events as point of interest.

INTRODUCTION

Going on vacation should be a battery recharging action for everyone. A tourism route well planned can foster an enriching and motivating experience (Ciurana Simó 2012). However, considering the time a tourist spend planning a tourism route for an unknown location, can be a very stressful action, de to the excessive information the Internet provides on researches. This makes this action difficult to perform since we retrieve a all lot of unnecessary and duplicate information (Ciurana Simó 2012). In order to ease this problem, IT companies created a new concept of research software called tourism recommendation systems. Those systems provide data as point of interest and with minimal effort. This kind of software helps a lot in tourism route plan, like professional tourism, there are some tourism areas that are being lacked off due to lower search in the subject (Adomavicius et al. 2011).

DEVELOPMENT

In order to provide a professional tourist with points of interest containing events that covers it's professional area, and interests, it will be needed to crawl the web gathering this type of data. In order to perform this action a spider algorithm was studied in order to verify the usability to crawl Internet pages for some pre-defined keywords that potentially have the needed data to generate professional tourism points of interest.

After the crawling the web for relevant information on Internet sites the system scrap the crawled sites using a Internet scrapper to scrap the needed info to generate the points of interest (POI).

There's also another data gathering process in study to complement this poi engine, using Rich Site Summary

April 5, 2016, Porto City, Portugal

(RSS) technology. There's a list of pre-defined Internet addresses on the database containing this type of data. This process will periodically gather the contents of these feeds and verify for new entries. If that's the case it will create a new entry on the POI engine database.

After gathering the information from the known list of Internet addresses, the POI engine system will then convert the gathered data into poi's and feed them to the specified recommendation system afterwards.

In order to guarantee a successful communication between the systems, a message queue will be implemented to transport the poi's from the POI engine to the recommendation system. After a study off several message queue systems, like RabbitMQ, ZeroMq and ActiveMQ (Carneiro 2013).

- **RabbitMQ:** this system is a message broker server that uses Advanced Message Queuing Protocol (AMQP). This server uses a broker architecture, which pushes the messages into a queue in a central node before delivering the message to the customer. This approach is a easy to implement solution and eases the routing, load balancing or persistent message queuing with few code lines (Carneiro 2013).
- **ZeroMQ:** It's a light message system, implemented to guarantee high performance communications and low latency. This system supports several complicated message communication scenarios, like P2P topology and other. This system implementation requires high communication protocol knowledge by the developer, and routing and load balancing configurations must be settle as well (Carneiro 2013).
- **ActiveMQ:** It's a much used message queue that uses a message broker server like RabbitMQ, and also provides P2P topology implementations similar to ZeroMq. This system is also known to provide high framework and support, but also low performance. (Carneiro 2013).

After identifying the requirements for this system, a list of known frameworks and development topologies were studied. The list of requirement is as follow:

- **Web optimized** – the system will fetch information on web pages, so web optimization will ease the process;
- **Scalability** – the system will need the capacity to grow on the future;
- **Easy to learn and implement** – due to the lack of implementation time;
- **Available frameworks** – this system should accept future migration to other systems;
- **Maintenance costs** - this system is supposed to work in its own;
- **License costs;**
- **Multi platform** – this system should be prepared to work on several platforms;

After study of several available technologies and frameworks and since the system is going to integrate a .net system in a near future, the same technology was chosen, for support reasons.

For database technologies, the same principle applies, since the system will be migrated to a existing one, we'll maintain a MySQL database. A study of relational database servers was made from two database engines:

- **SQLite;**
- **MySQL.**

To test the database engines on write capacity, a study was made based on a study performed by SQLite available on their site (SQLite 2016). The results and test cases are shown on Tabela 6

Table 17 – database tests results (in seconds)

	MySQL	SQLite
1000 INSERTs	0.114	0.223
25000 INSERTs in a transactions	2.184	0.757
25000 INSERTs into an indexed table	3.197	1.402
100 SELECTs without an index	2.760	2.526
100 SELECTs on a string comparison	4.640	3.372
Creating an index	0.318	0.659
5000 SELECTs with an index	1.270	1.162
1000 UPDATEs without an index	8.410	0.638
25000 UPDATEs with an index	8.134	3.104
25000 text UPDATEs with an index	6.982	1.725
INSERTs from a SELECT	1.537	1.599
DELETE without an index	0.975	0.560
DELETE with an index	2.262	0.752
A big INSERT after a big DELETE	1.815	1.485
A big DELETE followed by many small INSERTs	1.704	0.406
DROP TABLE	0.015	0.254

After studying this results, it was decided, based on database performance and development requirements, to go with SQLite server engine.

EVALUATION AND VALUE

This project will help professionals to gain new knowledge turning them into more habilitated workers, by providing information about professional events in a single place, they don't have to lose much time searching for them.

This project will be based on the capacity of the web crawler to find this data needed to generate the point of interest, and also in the feeds mechanism that is based on the information provided by the RSS web sites, if both these engines fail to gather information, the recommendation system won't provide the desired points of interest.

CONCLUSIONS

This project will complement an existent recommendation system and create a business opportunity to attract new customers providing this new type of point of interest to them. This will simplify the search and planning of tourism routes.

REFERENCES

- Adomavicius, G., Rokach, L. & Shapira, B., 2011. Recommender Systems Handbook. In *Media*. pp. 217–253. Available at: <http://www.springerlink.com/index/10.1007/978-0-387-85820-3> [Accessed February 7, 2016].
- Almeida, A., 2006. Technical Manual - User Modeling and Recommender System's Data Access Layer. , pp.1–32.
- Almeida, A., 2009. ToursPlan. Available at: <https://www.gecad.isep.ipp.pt:9090/projects/toursplan/Default.aspx> [Accessed February 6, 2016].
- Amazon, L., Guo, X. & Xia, Z., 2012. The Architecture of Open Source Applications. *University of California Berkeley*. Available at: <http://software-carpentry.org/2011/05/06/> [Accessed February 14, 2016].
- Anacleto, R. et al., 2010. PSIS Mobile. *ICWN'10 - The 2010 International Conference on Wireless Networks. Abstract*, p.112. Available at: <http://recipp.ipp.pt/handle/10400.22/2720> [Accessed February 7, 2016].
- Barnes, C., Blake, H. & Pinder, D., 2009. *Creating and Delivering Your Value Proposition: Managing Customer Experience for Profit*, London and Philadelphia. Available at: http://books.google.com/books?id=8d73CPT_khwC&pgis=1 [Accessed January 24, 2016].
- Berka, T., & Plößnig, M., 2004. *Designing Recommender Systems for Tourism*, Salzburg. Available at: http://www.salzburgresearch.at/wp-content/uploads/2010/10/enter_ploessnig.pdf.
- Bhattacharya, S. & Basu, J., 2007. Distress, Wellness and Organizational Role Stress among IT Professionals: Role of Life Events and Coping Resources. *Journal of the Indian Academy of Applied Psychology*, 33(2), pp.169–178. Available at: <http://medind.nic.in/jak/t07/i2/jakt07i2p169.pdf> [Accessed January 29, 2016].
- Bogers, T. & Van Den Bosch, A., 2009. *Collaborative and content-based filtering for item recommendation on social bookmarking websites*, Available at: <http://www.springerlink.com/index/10.1007/978-0-387-85820-3>.
- Bolin, M. et al., 2005. Automation and customization of rendered web pages. *Proceedings of the 18th annual ACM symposium on User interface software and technology - UIST '05*, p.163. Available at: <http://dl.acm.org/citation.cfm?id=1095034.1095062> [Accessed February 14, 2016].
- Brandstetter, T., 2011. 2 plan Work 2-gether Agile taskboard. Available at: <http://to-do.2-plan.com/free-task-board> [Accessed February 13, 2016].
- Carneiro, J.M.N., 2013. RealCrono –sistema de difusão de resultados em tempo real na web. Available at: <http://recipp.ipp.pt/handle/10400.22/5849> [Accessed February 14, 2016].
- Cheong Christopher, Cheong France, F.J., 2013. USING DESIGN SCIENCE RESEARCH TO INCORPORATE GAMIFICATION INTO LEARNING ACTIVITIES. In *Pacific Asia Conference on Information Systems*. Victoria, p. 14. Available at: <http://www.pacis-net.org/file/2013/PACIS2013-156.pdf> [Accessed December 29, 2015].
- Ciurana Simó, E.R., 2012. Development of a tourism recommender system. , p.1 recurs electrònic. Available at: <http://hdl.handle.net/2099.1/16444>.
- Coelho, B. da S., 2010. *User Modeling and Recommendation Strategies for Tourism*. Instituto Superior de Engenharia do Porto. Available at: <http://recipp.ipp.pt/handle/10400.22/1981> [Accessed October 12, 2016].
- Cording, P.H., 2011. *Algorithms for Web Scraping*. Technical University of Denmark. Available at: www.imm.dtu.dk [Accessed February 14, 2016].
- Costa, R., Ferreira, M. & Nicola, S., 2012. Utilização De Metodologias. , p.118. Available at: http://recipp.ipp.pt/bitstream/10400.22/2817/1/DM_RuiCosta_2012_MEEC.pdf [Accessed January 26, 2016].
- Coyle, L. & Cunningham, P., 2003. Exploiting Re-ranking Information in a Case-Based Personal Travel Assistant 1.
- Dudhe, A. & Sherekar, S., 2014. Performance Analysis of SOAP and RESTful Mobile Web Services in Cloud Environment. , pp.1–4. Available at: <http://research.ijcaonline.org/rtinfosec/number1/rtinfosec1401.pdf>.
- Edelson, H., 2011. Routing Topologies for Performance and Scalability with RabbitMQ. Available at: <http://www.rabbitmq.com/> [Accessed February 14, 2016].

April 5, 2016, Porto City, Portugal

- <http://blog.springsource.org/2011/04/01/routing-topologies-for-performance-and-scalability-with-rabbitmq/> [Accessed February 14, 2016].
- Erhun, F. & Keskinocak, P., 2003. *Game theory in business applications*, STANFORD. Available at: http://web.stanford.edu/~ferhun/paper/GT_Overview.pdf [Accessed January 26, 2016].
- FormacaoSaude.com, 2016. Category Archive for "Formação" | FormacaoSaude.com. Available at: <http://www.formacaoSaude.com/categoria/formacao/> [Accessed February 15, 2016].
- Gomes, a R. & Cruz, J.F., 2004. *A experiência de stress e "burnout" em psicólogos portugueses: um estudo sobre as diferenças de género*, CIPsi. Available at: <http://repositorium.sdum.uminho.pt/bitstream/1822/3944/4/2-Gomes&Cruz-SO-Revista-UM.pdf> [Accessed February 7, 2016].
- Google, 2016. Google Advanced Search. Available at: https://www.google.com/advanced_search.
- de Graaff, V., van Keulen, M. & de By, R.A., 2012. Towards geosocial recommender systems. In *Proceedings of the 4th International Workshop on Web Intelligence & Communities - WI&C '12*. New York, New York, USA: ACM Press, p. 1. Available at: <http://dl.acm.org/citation.cfm?doid=2189736.2189748> [Accessed October 15, 2016].
- Greco, S., Matarazzo, B. & Slowinski, R., 2001. Rough sets theory for multicriteria decision analysis. *European Journal of Operational Research*, 129(1), pp.1–47. Available at: http://sedok.narod.ru/s_files/poland/22.pdf [Accessed February 6, 2016].
- Harris, A. (Ana L.N. de C.H., 1999. *Revisão da Teoria dos Sistemas Nebulosos*. Escola Politécnica da Universidade de São Paulo. Available at: http://www.fec.unicamp.br/~luharris/tese/arquivos/Harris-ALNC_Cap3.pdf [Accessed January 28, 2016].
- Inc., R.H., 2012. ActiveMQ Tuning Guide. Available at: https://access.redhat.com/documentation/en-US/Fuse_ESB_Enterprise/7.1/html-single/ActiveMQ_Tuning_Guide/index.html [Accessed February 14, 2016].
- Levesque, T., 2009. Build an RSS reader in 5 minutes » Thomas Levesque's .NET blog. Available at: <http://www.thomaslevesque.com/2009/02/13/build-an-rss-reader-in-5-minutes/> [Accessed February 15, 2016].
- Luz, N., Anacleto, R. & Almeida, A., (1) Tourism Mobile and Recommendation Systems - A State of the Art | Nuno Luz - Academia.edu. Available at: https://www.academia.edu/1292794/Tourism_Mobile_and_Recommendation_Systems_-_A_State_of_the_Art [Accessed December 31, 2015].
- Luz, N., Anacleto, R. & Almeida, A., 2010. Tourism Mobile and Recommendation Systems - A State of the Art. In *EEE'10*. Porto, p. 3. Available at: http://recipp.ipp.pt/bitstream/10400.22/1467/1/COM_NunoLuz_2010_GECAD.pdf.
- McColl, R.C. et al., 2014. A performance evaluation of open source graph databases. *Proceedings of the first workshop on Parallel programming for analytics applications - PPAA '14*, pp.11–18. Available at: <http://dl.acm.org/citation.cfm?id=2567634.2567638> [Accessed February 16, 2016].
- Microsoft Developer Network (Microsoft), 2016. Advanced Operator Reference. Available at: <https://msdn.microsoft.com/en-us/library/ff795620.aspx> [Accessed October 20, 2016].
- Mun, W.K. et al., 2015. Medical Tourism Destination SWOT Analysis : A Case Study. In *SHS Web of Conferences12*. Kuala Lumpur: EDP Sciences, p. 8. Available at: <http://www.shs-conferences.org>.
- Nahon, J., 2011. A Comparative Analysis of REST and SOAP. , (August), p.64.
- Nicola, S., 2016. Análise de valor de negócio. , p.28. Available at: <https://moodle.isep.ipp.pt/course/view.php?id=3709>.
- Nicola, S., Ferreira, E.P. & Ferreira, J.J.P., 2012. A novel framework for modeling value for the customer, an essay on negotiation. *International Journal of Information Technology and Decision Making*, 11(3), pp.661–703. Available at: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84863473238&partnerID=40&md5=fadbee5462f209d6e32a60d695c55a0c> [Accessed January 31, 2016].
- Nicola, S., Ferreira, E.P. & Ferreira, J.J.P., 2014. A Quantitative Model for Decomposing & Assessing the Value for the Customer. *Journal of Innovation Management*, 1, pp.104–138. Available at: https://sigarra.up.pt/fbaup/pt/publs_pesquisa.FormView?P_ID=114690 [Accessed January 26, 2016].
- Nicola, S., Pinto, E. & Ferreira, J.J.P., 2013. J. J. Pinto Ferreira ASSETS MANAGEMENT - A CONCEPTUAL MODEL DECOMPOSING VALUE FOR THE CUSTOMER AND A QUANTITATIVE. *International Journal for Quality Research* 9(1) 89–106 ISSN 1800-6450, 9(1), pp.89–106. Available at: <http://www.ijqr.net/journal/v9-n1/7.pdf> [Accessed January 17, 2016].
- Nunes, H., Almeida, A. & Martins, C., 2016. Gathering data for professional tourism points of interest. In *Proceedings of the Ninth International C* Conference on Computer Science & Software Engineering - C3S2E '16*. New York, New York, USA: ACM Press, pp. 125–126. Available at: <http://dl.acm.org/citation.cfm?doid=2948992.2948996> [Accessed October 17, 2016].

- Osterwalder, A., 2004. The Business Model Ontology. , p.172. Available at: http://www.hec.unil.ch/aosterwa/PhD/Osterwalder_PhD_B_M_Ontology.pdf [Accessed February 16, 2016].
- Pant, G., Srinivasan, P. & Menczer, F., 2004. *Crawling the Web*, Bloomington, Iowa. Available at: https://www.google.pt/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0ahUKEWjgmcybhp_KAhWPhhoKHYVHDzwQFggqMAE&url=http://www.springer.com/?SGWID=4-102-45-113505-p28710520&usg=AFQjCNGNisZO3G6d4Z1Xo21CHR13R25WjQ&sig2=90EOKplzUNUbZJBxqiVDhA&bvm=bv
- Pawlak, Z., 1982. Rough sets. *International Journal of Computer & Information Sciences*, 11(5), pp.341–356. Available at: <http://sci2s.ugr.es/keel/dataset/includes/catImbFiles/1982-Pawlak.pdf> [Accessed January 28, 2016].
- Pazzani, M. & Billsus, D., 2007. Content-based Recommendation Systems. *Content-based Recommendation Systems Michael*, p.18.
- Penman, R.B., Baldwin, T. & Martinez, D., 2009. Web Scrapping Made Simple with SiteScraper. *Text*. Available at: <file:///Users/heldernunes/Downloads/sitescraper.pdf> [Accessed February 14, 2016].
- Pessoa, A. & Simões, J., 2003. Mineração de Dados Espaço-Temporal Aplicada a Previsão Climática Utilizando a Teoria dos Conjuntos Aproximativos. , p.6. Available at: http://mtc-m16c.sid.inpe.br/col/lac.inpe.br/worcap/2003/10.30.13.19/doc/worcap_versaofinal_alex2003.pdf [Accessed January 28, 2016].
- Potti, P.K., 2011. On the Design of Web Services: SOAP vs. REST. *Master Thesis*, p.106. Available at: <http://digitalcommons.unf.edu/cgi/viewcontent.cgi?article=1139&context=etd>.
- Ricci, F., Rokach, L. & Shapira, B., 2011. Introduction to Recommender Systems Handbook. In *Recommender Systems Handbook*. Boston, MA: Springer US, pp. 1–35. Available at: http://link.springer.com/10.1007/978-0-387-85820-3_1 [Accessed October 15, 2016].
- Rostański, M., Grochla, K. & Seman, A., 2014. Evaluation of highly available and fault-tolerant middleware clustered architectures using RabbitMQ. *2014 Federated Conference on Computer Science and Information Systems, FedCSIS 2014*, 2, pp.879–884. Available at: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84912058920&partnerID=tZ0tx3y1> [Accessed February 14, 2016].
- Silva, A. (Publico), 2014. Peso do turismo na economia do país está muito acima da média mundial. *Publico*. Available at: <http://www.publico.pt/economia/noticia/turismo-pesa-mais-no-emprego-e-na-economia-em-portugal-do-que-no-resto-do-mundo-1629417> [Accessed December 6, 2015].
- Smith, H. (World H.O., Brown, H. (World H.O. & Khanna, J. (World H.O., 2009. *Continuing education meetings and workshops: effects on professional practice and health-care outcomes*, Geneva: Brown H; Khanna J; Smith H; Available at: http://apps.who.int/rhl/effective_practice_and_organizing_care/Cd003030_smithh_com/en/ [Accessed December 19, 2015].
- Snyder, B., Bosanac, D. & Davies, R., 2011. ActiveMQ in Action. *Online*, p.406. Available at: <http://eds.a.ebscohost.com/eds/detail/detail?vid=1&sid=19393cc0-88df-4ae2-b259-f7c3830e5640%40sessionmgr4001&hid=4210&bdata=Jmxhbm9cHQtYnlmc2l0ZT1lZHMtbGl2ZS5y29wZT1zaXRl#AN=e dsgcl.268249116&db=edsgao> [Accessed February 14, 2016].
- SQLite, 2016. SQLite Database Speed Comparison. Available at: <https://www.sqlite.org/speed.html> [Accessed February 16, 2016].
- Swarbrooke, J. & Horner, S., 2001. *Business Travel and Tourism*, Oxfordshire: Routledge. Available at: <https://books.google.com/books?hl=pt-PT&lr=&id=6Dgek2vv8fEC&pgis=1> [Accessed January 29, 2016].
- Tanscheit, R., 2012. Sistemas Fuzzy. *LN Sistemas - DEE-PUC-Rio*, 113, pp.113–134. Available at: <http://www2.ica.ele.puc-rio.br/Downloads/41/LN-Sistemas Fuzzy.pdf> [Accessed January 28, 2016].
- Vaishnavi, V. & Kuechler, B., 2004. Design Science Research in Information Systems Overview of Design Science Research. *Ais*, p.45.
- Web, C. et al., 2013. Custom Web Scrapping For Content Aggregation From E-Commerce Websites *1,2. , 2(9). Available at: <http://www.ijesrt.com/issues pdf file/Archives 2013/September-2013/56.pdf> [Accessed February 14, 2016].
- Yang, Y. et al., 2014. How Does Perceived Value Affect Travelers ' Satisfaction and Loyalty ? *Social Behavior and Personality*, 42(10), pp.1733–1744. Available at: <http://eds.b.ebscohost.com/eds/pdfviewer/pdfviewer?sid=fded06e2-a736-4a30-ace3-f6ddc5bcc601%40sessionmgr113&vid=2&hid=108> [Accessed January 24, 2016].
- Zadeth, L., 1965. *Fuzzy Sets-Information and Control-1965*, Berkeley. Available at: <http://www.cs.berkeley.edu/~zadeth/papers/Fuzzy Sets-Information and Control-1965.pdf> [Accessed February 6, 2016].
- Zukerman, I. et al., 2001. Predictive Statistical Models for UserModeling. *User Modeling and User-Adapted Interaction*, 11 (1-2)(1), pp.5–18.

Anexo 4

Script base de dados

```
BEGIN TRANSACTION;
```

```
CREATE TABLE "users" ("id" integer not null primary key autoincrement, "name" varchar not null, "first_name" varchar not null default "", "last_name" varchar not null default "", "email" varchar not null, "password" varchar not null, "remember_token" varchar null, "created_at" datetime null, "updated_at" datetime null);
```

```
INSERT INTO `users`
```

```
(id,name,first_name,last_name,email,password,remember_token,created_at,updated_at)
```

```
VALUES
```

```
(1,'hnunes','','heu.nunes@gmail.com','$2y$10$cDEB7OcvCQ2s/XawJDpP9uoZXWblA2eUHas  
kRyTZYBeMyk157eMGi','LcoSylr9tzZaeVdzJJqFu1gQklbrfKkC1sIK2QtaiLwm7UHFhrj93QSHD40g  
';2016-09-13 09:07:10','2016-10-21 15:12:44');
```

```
CREATE TABLE siteType (
```

```
id_siteType INTEGER PRIMARY KEY AUTOINCREMENT
```

```
NOT NULL
```

```
UNIQUE,
```

```
type TEXT NOT NULL
```

```
CHECK (type IN ('rss',
```

```
'html',
```

```
'search_engine') )
```

```
DEFAULT html
```

```
);
```

```
INSERT INTO `siteType` (id_siteType,type) VALUES (1,'html'),
```

```
(2,'rss'),
```

```
(3,'search_engine');
```

```
CREATE TABLE searchKeywords (
```

```
id_searchKeywords INTEGER PRIMARY KEY AUTOINCREMENT
```

```

                NOT NULL

                UNIQUE,
searchQuery    TEXT    NOT NULL
);

INSERT INTO `searchKeywords` (id_searchKeywords,searchQuery) VALUES (1,'cursos
profissionais de medicina alternativa Porto');

CREATE TABLE poiEngine (
    id_poiEngine    INTEGER PRIMARY KEY AUTOINCREMENT
                NOT NULL
                UNIQUE,
    fk_endercosURL  INTEGER REFERENCES siteType,
    fk_poi          INTEGER REFERENCES poi (id_poi)
);

CREATE TABLE poiCategories (
    id_poiCategories    INTEGER PRIMARY KEY AUTOINCREMENT
                UNIQUE
                NOT NULL,
    fk_categories       INTEGER REFERENCES categories (id_categories),
    fk_poi              INTEGER REFERENCES poi (id_poi)
);

INSERT INTO `poiCategories` (id_poiCategories,fk_categories,fk_poi) VALUES (153,1,1),
(154,39,2055),
(155,40,1),
(156,43,1),
(157,44,1),
(158,45,1),
(159,46,1),

```

...

(221,99,2424),

(222,99,2426),

(223,45,2430);

CREATE TABLE poi (

id_poi INTEGER PRIMARY KEY AUTOINCREMENT

UNIQUE

NOT NULL,

title VARCHAR (500),

description VARCHAR (1500),

url VARCHAR (100),

date DATE

);

INSERT INTO `poi` (id_poi,title,description,url,date) VALUES (2055,'Curso de Auxiliar de Ação Médica','A I9PROJECT - Formação e eLearning encontra-se a promover o Curso de Auxiliar de Ação Médica no Porto. O curso que se destina a pessoas com escolaridade mínima obrigatória, interessadas em desenvolver a sua atividade profissional...','http://www.formacaosaude.com/2015/02/curso-de-auxiliar-de-acao-medica/','0001-01-01 00:00:00'),

(2056,'Cursos de Inglês (Nível)','A I9PROJECT promove, no Porto, vários cursos intensivos de inglês dispostos por nível. Os cursos ...','http://www.formacaosaude.com/2014/11/cursos-de-ingles-nivel/','0001-01-01 00:00:00'),

(2057,'Curso de Formação Inicial de Formadores','O Curso de Formação Inicial de Formadores destina-se a todos os interessados em trabalhar como formadores ...','http://www.formacaosaude.com/2014/09/curso-de-formacao-inicial-de-formadores/','0001-01-01 00:00:00'),

(2058,'Curso de Primeiros Socorros','O Curso de Formação em Primeiros Socorros com Suporte Básico de Vida é uma das mais ...','http://www.formacaosaude.com/2014/09/curso-de-primeiros-socorros/','0001-01-01 00:00:00'),

(2059,'Formação em Arte-Terapia: Práticas Artísticas','Arte-Terapia é um processo terapêutico que se serve do recurso expressivo a fim de conectar os mundos internos e

externos ...','http://www.formacaosaude.com/2014/08/formacao-em-arte-terapia-praticas-artisticas/','0001-01-01 00:00:00'),

(2060,'Cursos de Francês (Nível)','O ensino da língua francesa A I9PROJECT desenvolve cursos de Francês de nível e também cursos de Francês técnico ...','http://www.formacaosaude.com/2014/03/cursos-de-frances-nivel/','0001-01-01 00:00:00'),

(2061,'Curso de Socorrismo Pediátrico','DESTINATÁRIOS: Educadores e auxiliares de infância, professores, pais e outros profissionais cuja atividade ...','http://www.formacaosaude.com/2013/02/curso-de-socorrismo-pediatrico/','0001-01-01 00:00:00'),

(2062,'Curso de Francês para a Saúde','Programa do Curso de Francês para a Saúde: Entidade: I9PROJECT Duração: 40 horas ...','http://www.formacaosaude.com/2012/04/curso-de-frances-para-a-saude/','0001-01-01 00:00:00'),

...

```
CREATE TABLE "password_resets" ("email" varchar not null, "token" varchar not null, "created_at" datetime null);
```

```
CREATE TABLE parserSequence (
```

```
    id_parserSequence INTEGER    PRIMARY KEY AUTOINCREMENT
```

```
        UNIQUE
```

```
        NOT NULL,
```

```
    fk_enderecosUrl  INTEGER    REFERENCES enderecosURL (id_enderecosURL) ON DELETE NO ACTION
```

```
        ON UPDATE NO ACTION
```

```
        NOT NULL,
```

```
    parent          INTEGER,
```

```
    tagName         VARCHAR (20),
```

```
    tagId          VARCHAR (50),
```

```
    tagClass       VARCHAR (100)
```

```
);
```

```

INSERT INTO `parserSequence`
(id_parserSequence, fk_enderecosUrl, parent, tagName, tagId, tagClass) VALUES
(25,41,0,'tr',NULL,NULL),

(26,41,25,'td',NULL,NULL),

(27,41,26,'img',NULL,NULL),

(28,41,26,'p',NULL,NULL),

(29,10,0,'tr',NULL,NULL),

(30,10,29,'td',NULL,NULL),

(31,10,30,'img',NULL,NULL),

(32,10,30,'p',NULL,NULL),

(33,65,0,'tr',NULL,NULL),

(34,65,33,'td',NULL,NULL),

(35,65,34,'img',NULL,NULL),

(36,65,34,'p',NULL,NULL);

CREATE TABLE "migrations" ("migration" varchar not null, "batch" integer not null);

INSERT INTO `migrations` (migration, batch) VALUES
('2014_10_12_000000_create_users_table',1),

('2014_10_12_100000_create_password_resets_table',1);

CREATE TABLE enderecosURL (
    id_enderecosURL INTEGER PRIMARY KEY AUTOINCREMENT
        UNIQUE
        NOT NULL,
    url VARCHAR (500),
    fk_siteType INTEGER REFERENCES siteType (id_siteType) ON DELETE CASCADE
        NOT NULL
);

INSERT INTO `enderecosURL` (id_enderecosURL, url, fk_siteType) VALUES
(1, 'http://www.formacao.saude.com/categoria/formacao/feed/', 2),

```

(2,'http://www.workapt.com/feed/',2),
(3,'http://www.chporto.pt/ver.php?cod=0K0F',1),
(6,'http://doitbetter.pt/course-category/saude/',1),
(8,'http://www.imt.pt/',1),
(9,'http://www.portaleducacao.com.br/medicina-alternativa',1),
(10,'http://www.imt.pt/cursos.php',1),
(11,'https://www.umc.pt/index.php/cursos-e-formacoes',1),
(12,'http://www.esmot.com/',1),
(13,'http://www.escolashantala.com.br/cursos-mtc.html',1),
(14,'https://www.umc.pt/index.php/cursos-e-formacoes/saidas-profissionais',1),
(15,'http://www.topcursos.org/',1),
(16,'http://www.ipnaturaologia.com/cursos.html',1),
...
(70,'https://pt.wikipedia.org/wiki/Escola_de_Medicinas_Alternativas_e_Complementares',1),
(71,'https://www.umc.pt/index.php/cursos-e-formacoes/ensino-presencial/cursos/16-curso-pos-laboral',1),
(72,'http://www.ipnaturaologia.com/cursos.html',1),
(73,'http://www.ipnaturaologia.com/taxonomy/term/81/feed',2),
(74,'https://www.ualg.pt/pt/rss.xml',2),

(75,'http://www.sab.org.br/portal/index.php/medicinaeterapias?format=feed&type=rss',2),
(76,'http://institutohipocratesonline.com/index.php/medicinas-nao-convencionais/42-historia.feed?type=rss',2),

(77,'http://www.visitporto.travel/_layouts/GFI.PTur.Visitar/Feeds.aspx?content=1&lang=pt',2),
(78,'http://www.unesc.net/portal/rss/index/22',2),
(79,'http://www.guiamuriae.com.br/secao/noticias/educacao/feed/',2),

```
(80,'http://portal.medicinaoriental.com.br/site/modulos/noticias/noticias_rss.php',2),
```

```
(81,'http://www.blogcentagro.com/rss/ultimas/',2),
```

```
(82,'http://institutohipocratesonline.com/index.php/medicinas-nao-convencionais.feed?type=rss',2);
```

```
CREATE TABLE categories (
```

```
    id_categories INTEGER    PRIMARY KEY AUTOINCREMENT
```

```
                UNIQUE
```

```
                NOT NULL,
```

```
    description VARCHAR (300)
```

```
);
```

```
INSERT INTO `categories` (id_categories,description) VALUES (39,'Formação'),
```

```
(40,'Público em Geral'),
```

```
(41,'Auxiliar de Ação Médica'),
```

```
(42,'Auxiliar de saúde'),
```

```
(43,'Cardiopneumologia'),
```

```
(44,'Enfermagem'),
```

```
(45,'Farmácia'),
```

```
(46,'Fisioterapia'),
```

```
(47,'Medicina'),
```

```
(48,'Medicina Dentária'),
```

```
(49,'Medicina Veterinária'),
```

```
(50,'Nutrição'),
```

```
(51,'Para Profissionais'),
```

```
(52,'Psicologia'),
```

```
(53,'Radiologia'),
```

```
(54,'Serviço Social'),
```

```
(55,'Terapia da Fala'),
```

...

(90,'Alimentação'),

(91,'Ecologia'),

(92,'Naturopatia'),

(93,'Higienismo/Higiene de Vida'),

(94,'Acupunctura do Pé e da Mão'),

(95,'Deontologia'),

(96,'Corpo Teórico'),

(97,'Generic'),

(98,'curso'),

```
CREATE UNIQUE INDEX "users_email_unique" on "users" ("email");
```

```
CREATE INDEX "password_resets_token_index" on "password_resets" ("token");
```

```
CREATE INDEX "password_resets_email_index" on "password_resets" ("email");
```

```
COMMIT;
```

Anexo 5

Exemplo ficheiro Csv

Tabela 18 - Ficheiro Csv para inserção de pontos de interesse

title	description	url	date	categories
title_test_1	description_test1	http://www.url_test1.com/cursos.html	12/12/16	category_Test1
title_test_2	description_test2	http://www.url_test15.com/coiso.html	12/12/16	category_Test2
title_test_3	description_test15	http://www.url_test15.com/coiso.html	12/12/16	category_Test3
title_test_4	description_test1	http://www.url_test1.com/cursos.html	12/12/16	category_Test4
title_test_5	description_test2	http://www.url_test15.com/coiso.html	12/12/16	category_Test5
title_test_6	description_test15	http://www.url_test15.com/coiso.html	12/12/16	category_Test6
title_test_7	description_test1	http://www.url_test1.com/cursos.html	12/12/16	category_Test7
title_test_8	description_test2	http://www.url_test15.com/coiso.html	12/12/16	category_Test8
title_test_9	description_test15	http://www.url_test15.com/coiso.html	12/12/16	category_Test9
title_test_10	description_test1	http://www.url_test1.com/cursos.html	12/12/16	category_Test10
title_test_11	description_test2	http://www.url_test15.com/coiso.html	12/12/16	category_Test11
title_test_12	description_test15	http://www.url_test15.com/coiso.html	12/12/16	category_Test12
title_test_13	description_test1	http://www.url_test1.com/cursos.html	12/12/16	category_Test13
title_test_14	description_test2	http://www.url_test15.com/coiso.html	12/12/16	category_Test14
title_test_15	description_test15	http://www.url_test15.com/coiso.html	12/12/16	category_Test15
title_test_16	description_test1	http://www.url_test1.com/cursos.html	12/12/16	category_Test16
title_test_17	description_test2	http://www.url_test15.com/coiso.html	12/12/16	category_Test17
title_test_18	description_test15	http://www.url_test15.com/coiso.html	12/12/16	category_Test18
title_test_19	description_test1	http://www.url_test1.com/cursos.html	12/12/16	category_Test19
title_test_20	description_test2	http://www.url_test15.com/coiso.html	12/12/16	category_Test20
title_test_21	description_test15	http://www.url_test15.com/coiso.html	12/12/16	category_Test21
title_test_22	description_test1	http://www.url_test1.com/cursos.html	12/12/16	category_Test22
title_test_23	description_test2	http://www.url_test15.com/coiso.html	12/12/16	category_Test23
title_test_24	description_test15	http://www.url_test15.com/coiso.html	12/12/16	category_Test24
title_test_25	description_test1	http://www.url_test1.com/cursos.html	12/12/16	category_Test25
title_test_26	description_test2	http://www.url_test15.com/coiso.html	12/12/16	category_Test26
title_test_27	description_test15	http://www.url_test15.com/coiso.html	12/12/16	category_Test27
title_test_28	description_test1	http://www.url_test1.com/cursos.html	12/12/16	category_Test28
title_test_29	description_test2	http://www.url_test15.com/coiso.html	12/12/16	category_Test29
title_test_30	description_test15	http://www.url_test15.com/coiso.html	12/12/16	category_Test30