Integrador Automático de Notícias

Daniel Silva, Pedro Abreu, Pedro Mendes, e Vasco Vinhas

Faculdade de Engenharia da Universidade do Porto Rua Dr. Roberto Frias, s/n
4200-465 Porto, Portugal

Resumo A proliferação dos serviços noticiosos na Internet resulta da crescente importância deste meio de comunicação no que concerne à publicação de conteúdos, tendo as entidades competentes definido um conjunto de normas por via a consolidar o processo de difusão. Contudo, constatam-se elevados níveis de dispersão e não unificação de fontes de dados. Neste contexto identificam-se oportunidades de aplicação de conceitos de informação semi-estruturada, armazenamento e pesquisa orientada ao documento. É proposta uma solução capaz de fornecer serviços de integração e pesquisa no contexto da problemática exposta, recorrendo a XML e tecnologias associadas: XSD, XSL, XPath, XQuery, XUpdate e eXist. Utilizando as tecnologias mencionadas alcançou-se o acesso automatizado às fontes de dados, conversão da informação semi-estruturada para formato normalizado, actualização automática dos conteúdos locais, e interfaces de administração para gestão de fontes e utilizadores, assim como pesquisa de informação, gestão de subscrições e organização em árvore multinível das fontes de dados.

1 Introdução

Os serviços noticiosos disponíveis na *web* consistem numa das formas mais eficientes para qualquer indivíduo com pretensões de se manter informado com elevada regularidade em determinadas áreas de conhecimento. Com o progresso da tecnologia e a evolução das normas de representação, as notícias têm vindo a perder o seu impacto como forma única de manter um dado sítio *web*. Actualmente, os utilizadores procuram formas mais directas de agregar a informação noticiosa, potencialmente proveniente de diversas fontes, livres de eventuais conteúdos publicitários e com menores necessidades a nível de exigências de rede, assegurando que os dados provenientes da Internet contêm na sua globalidade informação noticiosa.

Foi neste contexto que surgiu, por volta do ano 2000, a norma RSS (*Really Simple Syndication* na versão 2.0)[1], que consiste num dialecto XML[2] normalizado tendo em vista o armazenamento simplificado de informação e meta informação noticiosa. As entidades produtoras de conteúdos, se assim o entenderem, têm neste formato um novo meio para os publicar, sendo assegurado que do lado dos utilizadores novas formas são oferecidas para os integrar em aplicações de âmbito mais alargado ou especificamente orientadas a este domínio. Nesta gama de ferramentas, existem já muitas que oferecem ao utilizador formas mais ou menos eficazes de garantir uma actualização constante dos conteúdos.

Dada a relevância da norma no domínio da publicação de conteúdos, as ferramentas existentes para consulta têm na sua maioria uma componente comercial associada e surgem das mais diversas formas. Existem ferramentas *online*, *desktop*, tanto multi-plataforma como orientadas a determinado sistema operativo e, sob a forma de extensões para outras aplicações. Serão por ventura estas últimas as mais utilizadas, na medida em que estão associadas a ferramentas de uso diário dos utilizadores comuns, como clientes de correio electrónico e navegadores *web*, levando a uma coexistência de funcionalidades em tudo útil na eficiente gestão do tempo que o utilizador tem disponível para a consulta de conteúdos[3].

De entre estas a destacar o *Mozilla Thunderbird*[4], que sendo um cliente de correio electrónico, integra nas suas principais funcionalidades um agregador de RSS. Esta ferramenta permite uma gestão personalizada das *feeds* subscritas mesmo que, protegidas por camadas de autenticação e apresenta diálogos de busca avançados permitindo na situação mais complexa especificação de intervalos de tempo. O acesso distribuído é contudo limitado pois implica conhecimentos por parte do utilizador de criação de configurações distribuídas de aplicações. Para o cliente de correio electrónico da *Microsoft*, existem extensões desenvolvidas por *third-parties*, de entre as quais se destacam o *Attensa*[5] e o *intraVnews*[6] contudo face à aplicação da *Mozilla*, apresentam algumas limitações. O *Attensa* apenas funciona com a versão inglesa do *Microsoft Oulook*, e apresenta erroneamente caracteres nas notícias usados em outros idiomas. O *intraVnews* não apresenta estas deficiências, suportando ainda a norma *Atom*[7] todavia, a sua instalação só é possível em *Microsoft Outlook XP* ou *2003*.

Os navegadores *web* em geral também apresentam algumas funcionalidades de forma a facilitar a consulta deste tipo de conteúdos. Maior destaque poderá ser dado ao *Mozilla Firefox*[8] que nas primeiras versões (1.0) tinha possibilidade através de extensões evidenciar as notícias desta forma. Na versão actual (2.0), estas funcionalidades encontram-se presentes no pacote base da aplicação. A nova versão do navegador da *Microsoft* (*Internet Explorer 7.0*[9]), apresenta igualmente funcionalidades neste âmbito. A presença deste tipo de funcionalidades é hoje factor decisivo no que concerne à avaliação de um navegador *web*.

Todavia, a relevância dos serviços noticiosos não se remete apenas aos conteúdos publicados numa perspectiva actual ou local. A forma como surgem e os canais onde são publicados são aspectos extremamente importantes e mesmo críticos em diversas situações e contextos, pois destes factores muitas vezes se inferem questões como relevância sócio-cultural ou mesmo político-económica. É precisamente neste aspecto que a grande maioria das ferramentas disponíveis apresenta algumas falhas, não sendo as funções e metodologias de pesquisa facilitadas nem alargadas aos múltiplos campos característicos de uma notícia. Ainda neste aspecto, as notícias, quando descarregadas da Internet são armazenadas em sistemas de ficheiros, quase sempre sem qualquer tipo de indexação, quer sobre conteúdos textuais, quer sobre metadados associados. Não existe igualmente uma preocupação evidente em manter um repositório de notícias, o que impossibilita a caracterização de um dado fenómeno se à sua ocorrência estiver associado um intervalo de tempo considerável. A título de exemplo deste tipo de utilização,

podemos invocar a recuperação de notícias relacionadas com a problemática do défice orçamental de Portugal, tema recorrente nos últimos anos.

O projecto desenvolvido pretende dar resposta a alguns destes problemas, nomeadamente a construção e manutenção de um arquivo dinâmico de conteúdos noticiosos, oferecendo ainda a flexibilidade necessária para que os dados a manipular sejam apresentados de uma forma personalizada a cada utilizador.

Ao longo do presente documento é realizada, no capítulo 2, uma análise às diversas tecnologias associadas ao conceito de informação semi-estruturada. No capítulo 3 são apresentados e descritos os componentes genéricos do projecto, enquanto o quarto encerra a divulgação dos resultados. O 5º capítulo é dedicado à identificação das conclusões e melhoramentos futuros.

2 Informação Semi-Estruturada

A quantidade de informação hoje disponível para o público em geral teve um incremento sem precedentes. O acesso aos conteúdos, que têm verdadeiramente relevância para quem os pesquisa, é actualmente mais difícil, dado o elevado número de fontes disponíveis e a grande diversidade de meios para estruturar a informação.

Formatos completamente estruturados exigem definição de normas e desenvolvimento de ferramentas sofisticadas que permitam a sua eficiente manipulação, muito embora tais características restrinjam a disponibilidade da informação, levando a formatos fechados e dificilmente descodificáveis.

A alternativa não estruturada remete os métodos de pesquisa para técnicas de processamento que procuram simular a inteligência humana no que concerne aos processos interpretativos, o que pelos padrões actuais é ainda algo que está longe de ser atingido, nomeadamente ao nível da correlações de temas e respectiva localização temporal.

A solução de compromisso passa pela adopção de métodos semi-estruturados, de armazenamento de informação, que tornam possível a associação entre informação e meta-informação relevantes na captação da significância dos dados. Neste âmbito, a meta-linguagem XML, definida pelo W3C é o padrão corrente.

2.1 Processamento

Os documentos XML, devido à sua estrutura de anotações, podem ser representados esquematicamente como árvores em memória. Este facto foi determinante na concepção das duas principais APIs que definem as formas de processar e manipular os documentos.

DOM (*Document Object Model*)[10] especifica o carregamento completo da árvore para memória, fornecendo, posteriormente, um conjunto de métodos por via a permitir acesso e alteração dinâmica dos nós constituintes da mesma. SAX (*Simple* API *for* XML)[11] oferece uma abordagem guiada por eventos, emitidos pelos componentes validadores, permitindo o registo dos *handlers* correspondentes e consequente execução de tarefas específicas. Esta abordagem é mais

eficiente no que respeita aos recursos necessários ao processamento dos documentos, sendo contudo menos flexível do que DOM aquando da existência de necessidade de manipulação do documento a nível da sua estrutura.

Uma outra forma de processar documentos XML passa pela especificação de folhas de estilo XSL[12]. Os documentos XSL, também eles XML, definem um conjunto de primitivas que possibilitam a especificação de acções a despoletar aquando do processamento por um motor adequado, de forma semelhante à abordagem com SAX. Genericamente, as acções focam-se na criação de um outro documento, tipicamente também ele XML, sendo esta uma das formas mais compactas e concisas de especificar conversões entre ficheiros obedecendo a esquemas de dados distintos.

2.2 Interrogação

Na medida em que os ficheiros XML apresentam a versatilidade necessária para possibilitar a representação de informação oriunda de diversos domínios, a necessidade de criar uma linguagem padrão para interrogação sistemática tornou-se emergente.

XQuery[13] surge neste contexto. A linguagem retira alguns conceitos de outras conhecidas de interrogação como SQL, e OQL e poder-se-á mesmo considerar como uma extensão da segunda versão de XPath[14], na medida em que os resultados devolvidos por perguntas semelhantes nas duas são equivalentes. XQuery necessita, contudo, de blocos de informação representados em XPath 1.0 para satisfazer necessidades de localização unívoca de nós pretendidos. Pretende ser sobretudo um padrão de fácil compreensão, um pouco à imagem do SQL, para as bases de dados relacionais, adaptado a documentos semi-estruturados.

Apesar de ser completo a nível de capacidades interrogativas, não inclui ainda mecanismos que permitam alterar, dinamicamente, quer o conteúdo, quer a estrutura de documentos. Para atingir tais fins, existem outras normas que definem formas para actualizar os documentos em causa. Entre elas, é de destacar o XUpdate[15] da XML:DB[16] que define uma estrutura para documentos XML cujo conteúdo exprime operações de interrogação e actualização.

2.3 Armazenamento

O armazenamento de informação formatada segundo XML é atingível de diversas formas, tendo cada uma delas um conjunto de valências e aspectos mais prejudiciais neste âmbito[17].

A forma mais imediatista de atingir este objectivo passa pelo simples armazenamento num sistema de ficheiros. Desta forma os tempos de acesso são completamente determinados pela arquitectura física que o suporta e pelas características do sistema operativo que o mantém.

Outra forma de manter a informação persistente consiste em armazenar como BLOBS (*Binary Large Objects*) ou CLOBS (*Character Large Objects*) em bases de dados puramente relacionais. Seguindo esta abordagem, há obviamente perda

de informação no que concerne aos metadados associados, muito embora alguns fornecedores deste tipo de produtos tenham vindo a incluir tipos de dados e estruturas orientadas a documentos XML como é o caso da *Oracle*[18] e *PostgresSQL*[19].

A terceira alternativa consiste em ter bases de dados orientadas especificamente a XML, usando esta estrutura para assim armazenar nativamente os documentos em causa. Este tipo de método de armazenamento é mais adequado quando a informação a salvaguardar consiste primordialmente em documentos com grandes porções contínuas de texto.

Para o projecto, optou-se pela última abordagem dado que a informação noticiosa integra em si os conceitos que mais partido tiram da mesma.

3 Aplicação

O Integrador Automático de Notícias foi desenhado segundo a arquitectura de três camadas para aplicações baseadas no paradigma cliente/servidor. Este desenho proporciona, quando comparado com arquitecturas de camada dupla, um incremento dos níveis de desempenho, flexibilidade, manutenção, reusabilidade, e escalabilidade[20]. Com o intuito de promover a inteligibilidade do funciona-

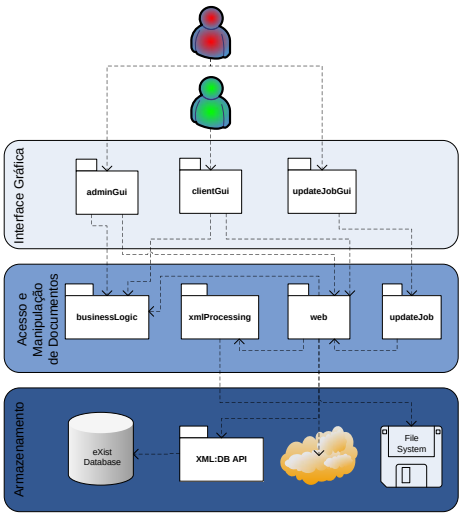

Figura 1. Diagrama da Arquitectura do Integrador Automático de Notícias

mento multinível do sistema, optou-se por apresentar em secções individuais cada um dos seus componentes agregadores, ilustrados através da Figura 1.

3.1 Armazenamento e Esquema de Dados

Todos os documentos relacionados com a aplicação são armazenados em formato XML na base dados nativa XML eXist[21]. Os documentos encontram-se divididos de acordo com o seu âmbito de utilização no sistema geral. Deste modo, é realizada uma primeira divisão dicotómica de conteúdos entre documentos referentes a repositórios de notícias e documentos relacionados com a administração e configuração da aplicação.

Seguindo este último ramo, encontram-se as colecções respeitantes à informação relativa às fontes com o respectivo período de actualização de dados e descrição textual, encontrando-se organizadas em categorias, de forma a facilitar a sua gestão. Ainda no directório de administração é armazenado o documento com a informação relativa aos utilizadores do sistema, de modo a possibilitar os processos de autenticação. Para além da informação pessoal de cada utilizador e das suas credenciais de autenticação, é armazenada a sua organização pessoal das fontes de dados numa estrutura em árvore multinível, em tudo semelhante à tradicional organização de sistemas de ficheiros[22]. Esta opção possibilita que cada utilizador personalize a organização das fontes de dados do modo que mais lhe aprouver, potenciando o grau de usabilidade da aplicação. De referir, por último, que o esquema de dados permite que uma mesma fonte de dados se encontre replicada em diversas (sub)categorias.

O ramo do repositório de notícias encontra-se estruturado de modo aplanado, procedendo-se à constituição de um documento por cada fonte de dados. A actualização das notícias por documento é responsabilidade do módulo descrito na secção 3.3. Tendo em consideração o facto da norma RSS não contemplar um identificador unívoco para cada entrada noticiosa, nem obrigar a que o campo *data de publicação* seja consistente com sua a posição relativa no documento, o modelo de actualização do Integrador tem por pressuposto que as entidades responsáveis pela publicação dos conteúdos noticiosos procedem à modificação dos documentos RSS segundo o modelo FIFO[22], procedendo à identificação de cada entrada pelo seu título. Esta assumpção, através dos diversos testes realizados a múltiplas fontes de dados revelou-se perfeitamente válida e, mesmo que venha a ser quebrada, o único impacto na aplicação traduz-se na repetição ou exclusão de algumas notícias. Esta decisão foi assumida após o estudo experimental de diversas fontes de dados de referência, quer de âmbito nacional, quer internacional, concluindo-se que a única análise consistente verificada foi a da opção descrita, não se verificando outras tidas como válidas *a priori*, tais como a análise posicional por data de publicação e comparação de diversos campos.

3.2 Acesso e Manipulação de Documentos

Este conjunto de pacotes, em conjunto com o componente responsável pela actualização periódica de informação, descrito no ponto seguinte, enquadra-se na camada de lógica de negócio do modelo seguido. Resulta da necessidade de encapsular um vasto conjunto de funcionalidades directamente relacionadas com o âmago da aplicação, para posterior utilização por parte das interfaces com o

utilizador. De entre este conjunto vasto de funcionalidades são de destacar as áreas identificadas na Figura 1.

Analisando em particular cada um dos referidos pacotes, refere-se, de imediato, o *xmlProcessing*, responsável pela manipulação e transformação da totalidade dos conteúdos noticiosos guardados, numa primeira fase, sem tratamento, no sistema de ficheiros local. A estes dados não tratados, resultado do acesso aos documentos publicados na *web*, é-lhes aplicada uma transformação XSL com o intuito de seleccionar apenas as notícias ainda não armazenadas, descartando simultaneamente diversos elementos, promovendo a conversão para um formato universal a todas as fontes tratadas pelo Integrador.

O pacote *businessLogic* encerra a representação das entidades mais significativas para a aplicação sob a forma de classes. De grosso modo, assume a responsabilidade das funções de mapeamento entre as realidades salvaguardadas nos documentos em base de dados e entidades manipuláveis pelas demais classes do Integrador. Para melhor compreensão, e a título de exemplo não exaustivo, é possível manipular entidades representadoras de clientes (entendidos como os utilizadores finais da aplicação responsáveis pela execução de consultas), notícias e fontes de dados.

Por último, e revestindo-se de uma importância superior, encontra-se o pacote *web* que tem um conjunto relativamente vasto de classes que suportam todas as actividades relacionadas com o acesso a dados, encontrando-se estes, quer em base de dados nativa XML, quer nos locais referenciados como pontos de publicação das entidades noticiosas monitorizadas. No caso dos documentos se encontrarem em base de dados, foi desenvolvido um conjunto de operações que, para além do simples acesso, permitem a inserção, edição e remoção de conteúdo, utilizando para tal efeito as tecnologias apresentadas na secção 2.2.

3.3 Actualização Periódica de Informação

De modo a garantir uma actualização constante dos documentos contendo informação noticiosa, para as fontes declaradas segundo o modelo exposto na secção 3.1, foi desenvolvido um serviço tendo em vista a execução periódica de tarefas, segundo intervalos de tempo definidos. O serviço de actualização está estruturado de forma a garantir a execução paralela das várias tarefas de actualização, seguindo uma arquitectura caracterizada por múltiplos fios de execução. Apesar deste aspecto ser, indubitavelmente, aquele que se reveste de maior relevância no contexto do projecto, e igualmente no componente em análise, duas outras funcionalidades foram desenvolvidas, sem as quais, este não poderia ser considerado completo. Estas duas funcionalidades estão relacionadas com os seguintes pontos:

- Construção das estruturas de dados que materializam as tarefas a executar. Envolve naturalmente o recurso às facilidades oferecidas pelo componente previamente descrito na secção 3.2.
- Cancelamento das tarefas activas. Sempre que o utilizador directamente desencadeie este conjunto de actividades e nas situações em que ocorram erros

de execução, o módulo desactiva todas as tarefas activas, esperando que as que estejam eventualmente a correr terminem a sua execução.

3.4 Interface Gráfica

A interface gráfica com o utilizador foi desenvolvida tendo por base a plataforma de desenvolvimento *Java Swing*[23]. Esta opção justifica-se com a necessidade da utilização de um meio de eficácia comprovada de interacção com humanos, sendo simultaneamente importante manter a independência relativamente ao sistema operativo dos sistemas dos utilizadores finais. Tal como é perceptível através da estruturação do presente capítulo, da consulta da Figura 1, e da leitura da secção 3.1, em especial o parágrafo dedicado à explicitação da organização da estrutura de dados, foram desenvolvidas interfaces tendo em vista a sua utilização por utilizadores de dois níveis distintos: o administrador da aplicação, de cariz mais técnico, e o agente pesquisador de notícias, sem necessidade de competências informáticas.

Os pontos seguintes esclarecem a natureza destes dois tipos de utilizador ao mesmo tempo que apresentam, de modo sumário, as principais funcionalidades das interfaces.

Gestão de Actualização Graficamente, o serviço de gestão da actualização periódica de informação, descrito na secção 3.3, materializa-se num *tray icon* localizado na *system tray*. Esta opção é justificada pelo número reduzido de operações permitidas ao utilizador. Por outro lado, trata-se também de uma forma de manter o controlo do serviço, espacialmente próximo da localização habitual dos mecanismos de controlo dos servidores que suportam a camada persistente de dados, nomeadamente o *Apache Tomcat* da *Apache Software Foundation*[24]. O menu associado ao ícone está igualmente disposto de forma semelhante face ao seu congénere do Tomcat. As opções acima descritas fundamentam-se em dois princípios fundamentais de usabilidade, especificamente a normalização de sequências de acções[25] e o princípio da localização espacial[26].

Gestão de Entidades Pretendeu-se, com a definição desta interface, fornecer o suporte a todas as actividades relativas ao domínio de dados da administração do sistema. Este domínio foi já alvo de referência na secção 3.1, consistindo, basilarmente, em duas realidades: utilizadores do sistema e fontes de dados. Desta forma, é fornecido um mecanismo visual para as operações básicas de inserção, eliminação e edição das referidas entidadades, estando cada um dos tipos armazenados em documentos distintos. De referir ainda o facto de as fontes de dados estarem categorizadas segundo uma estrutura em árvore de nível único, conduzindo à permissão da migração de fontes de dados entre as diversas categorias.

Consulta e Pesquisa Personalizada A interface gráfica de apoio à consulta e pesquisa de notícias encerra mais funcionalidades do que as referidas nos pontos

anteriores e, ao invés destas, foi desenhada para utilização por parte do agente pesquisador de notícias. São identificáveis três grandes conjuntos de actividades disponibilizadas:

- Gestão de Subscrições Permite-se ao utilizador que este realize a operação de subscrição de fontes de dados de entre as previamente disponibilizadas através do processo descrito no ponto anterior. Naturalmente, a funcionalidade simétrica encontra-se, igualmente, contemplada.
- Gestão da Organização de Fontes Este conjunto de funcionalidades possibilita que o utilizador proceda à livre organização das diferentes fontes de dados subscritas, segundo uma estrutura em árvore multinível que admite replicação de entidades.
- Pesquisa e Análise de Informação Trata-se do âmago da aplicação, prevendo a apresentação de um formulário onde é possível a especificação das palavras chave a pesquisar, quer no título, quer no corpo da notícia. É igualmente disponibilizado um filtro por data de publicação através da definição de um intervalo (aberto ou fechado) de datas. Ainda que por omissão a pesquisa seja efectuada em todas as fontes de dados subscritas pelo utilizador, é possível a escolha do conjunto de fontes a interrogar.

De modo a potenciar o grau de compreensão das entidades descritas na presente secção, dedicada às interfaces gráficas com o utilizador, aconselha-se a leitura do capítulo 4, onde, a propósito da exposição dos resultados do projecto, se apresentam ilustrações de algumas interfaces do projecto.

4 Resultados

As metas inicialmente definidas foram totalmente alcançadas, sendo que grande parte das funcionalidades são suportadas por tecnologias relacionadas com XML: eXist, XPath, XSL, XQuery e XUpdate. Deste modo, os principais requisitos funcionais implementados, agregadores dos vários conceitos do projecto, encontram-se listados de seguida:

- Acesso automatizado às fontes de dados.
- Conversão da informação semi-estruturada para um formato normalizado.
- Actualização automática dos conteúdos locais.
- Interface de administração de gestão de fontes e utilizadores totalmente funcional (ver Figura 2-A).
- Interface totalmente funcional de pesquisa de informação, gestão de subscrições e organização em árvore multinível das fontes de dados (ver Figura 2-B).

No último passo da Figura 2-B, é gerada uma interrogação em XQuery sempre que é efectuada uma pesquisa de notícias. A título de exemplo, e traduzindo a interrogação requerida pelo utilizador, apresenta-se o código correspondente através da Figura 3.

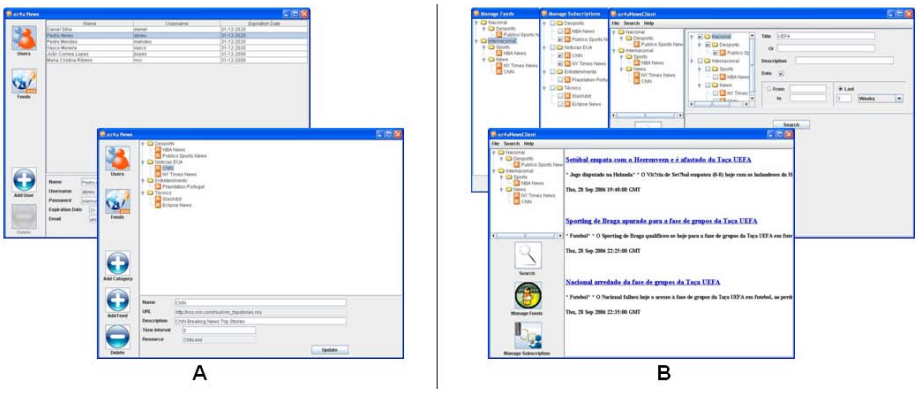

Figura 2. A-GUI de Gestão de Entidades. B-GUI de Consulta e Pesquisa Personalizada

```
xquery version "1.0";
for   $x in doc("publicodesporto.xml")/news/item
let       $listaMeses := ("Jan","01", "Feb","02","Mar","03","Apr","04","May","05","Jun","06","Jul","07","Aug","08","Sep","09","Oct","10","Nov","11","Dec","12"),
          $listaTimezones := ("Z", "GMT", "Z", "EDT", "-04:00", "EST", "-05:00", "CST", "-06:00", "CDT", "-05:00", "PST", "-08:00", "PDT", "-07:00" ),
          $data := $x/pubDate, $diasemana := substring($data, 1, 3), $dia := substring($data, 6, 2), $mesLetra := substring($data, 9, 3),
          $ano := substring($data, 13, 4), $hora := substring($data, 18, 2), $minuto := substring($data, 21, 2), $segundo := substring($data, 24, 2),
          $timezone := substring($data, 27, 3),$mes := subsequence($listaMeses , index-of($listaMeses, $mesLetra) + 1, 1),
          $timezoneValue := subsequence($listaTimezones , index-of($listaTimezones, $timezone ) + 1, 1),
          $totaldata := xs:dateTime(concat(concat(concat($ano, "-"),concat(concat($mes, "-"),concat(concat($dia, "T"),concat(concat($hora, ":"),concat(concat($minuto, ":"),$segundo))))), $timezoneValue))
where contains($x/title, "UEFA") and contains ($x/description, "") and $totaldata ge xs:dateTime("2006-09-26T00:00:01Z") and $totaldata le xs:dateTime("2006-10-17T00:00:01Z")
order by $totaldata descending
return <res><news>{$x}</news><dateZ>{$totaldata}</dateZ></res>
```

Figura 3. Exemplo de uma Interrogação de Pesquisa de Notícias

5 Conclusão e Futuros Desenvolvimentos

Tendo em conta a apresentação dos resultados efectuada no capítulo anterior, encontram-se identificados diversos temas que resultariam em desenvolvimentos futuros do projecto. A análise seguinte apresenta-se estruturada de acordo com o grau crescente de dificuldade dos componentes, e consequentemente, da previsão da data da sua implementação.

Identificada como funcionalidade de dificuldade reduzida, encontra-se classificada a análise de texto mais avançada com a introdução de pesquisa *fuzzy* e *case insensitive*. Esta funcionalidade poderia ser implementada através da utilização das respectivas funções disponibilizadas pela norma XQuery. Ainda nesta classe de desenvolvimentos, é de considerar a introdução de operações lógicas avançadas na pesquisa e múltiplos intervalos de datas, bem como a salvaguarda de esqueletos de interrogações típicas, acessíveis por todo os utilizadores. No capítulo da segurança, prevê-se o refinamento das operações de autenticação, promovendo a utilização de toda a informação respeitante aos utilizadores. Por último, neste domínio, identifica-se como futuro desenvolvimento o suporte a múltiplos idiomas e fusos horários.

Considerados como desenvolvimentos de complexidade superior, encontram-se os relacionados com operações sobre o conteúdo noticioso. De entre estes, há a destacar a análise do contexto em que os termos são empregues[27], a tradução automática de conteúdos para um conjunto pré-definido de idiomas

e a incorporação na análise do conteúdo apontado pelos hiperligação presentes na fonte de dados. Numa vertente mais relacionada com as pesquisas encetadas pelos utilizadores, importa referir a possibilidade de salvaguarda das mesmas por utilizador e a manutenção do histórico de interrogações efectuadas.

6 Agradecimentos

Neste capítulo final, os autores gostariam de agradecer ao professor Eugénio de Oliveira pelas repetidas lições respeitantes à importância do estudo e redacção de artigos científicos. Uma nota especial também para os professores João Correia Lopes e Maria Cristina Ribeiro pela atenção prestada ao trabalho dos autores aquando da sua condição de alunos da disciplina de Linguagem, Anotação e Processamento de Documentos.

Referências

1. RSS Advisory Board - http://www.rssboard.org/
2. Bradley, N.: The XML Companion Third Edition Addison Wesley (2001)
3. Most used RSS aggregators - http://www.newsonfeeds.com/faq/aggregators/
4. Mozilla Thunderbird - http://www.mozilla.com/en-US/thunderbird/
5. Attensa - http://www.attensa.com/
6. intraVnews - http://www.intravnews.com/
7. Atom - http://tools.ietf.org/html/rfc4287/
8. Mozilla Firefox - http://www.mozilla.com/en-US/firefox/
9. Internet Explorer 7 - http://www.microsoft.com/windows/ie/default.mspx
10. W3C DOM Interest Group - Document Object Model - http://www.w3.org/DOM/
11. SAX Project Page - http://www.saxproject.org/
12. XSL 1.0 Technical Report - http://www.w3.org/TR/xsl/
13. XQuery 1.0 Technical Report - http://www.w3.org/TR/xquery
14. XPath 1.0 Technical Report - http://www.w3.org/TR/xpath
15. XUpdate Table of Contents
    http://xmldb-org.sourceforge.net/xupdate/xupdate-req.html
16. XML:DB Initiative for XML Databases
    http://xmldb-org.sourceforge.net/
17. Bourret, Ronald: Consulting, writing, and research in XML and databases - http://www.rpbourret.com/xml/
18. Oracle. XML to the Power of SQL
    http://www.oracle.com/technology/tech/xml/xdkhome.html
19. PostgreSQL Project Page - http://www.postgresql.org/about/advantages
20. Eckerson, Wayne W. "Three Tier Client/Server Architecture: Achieving Scalability, Performance, and Efficiency in Client Server Applications." Open Information Systems 10, 1 (Janeiro 1995). Capítulo 3, Página 20.
21. eXist Project Page - http://exist.sourceforge.net/
22. Tanenbaum, Andrew S. "Modern Operating Systems", 2nd ed, Prentice-Hall, 2001.
23. JAVA Swing Project Page - http://java.sun.com/products/jfc/
24. Apache Tomcat Project Page- http://tomcat.apache.org/
25. U.S. Department of Health and Human Services. 2006. "Research-Based Web Design & Usability Guidelines". Capítulo 2, página 11.

26. Constantine, L. L., and Lockwood, L. A. D. Software for Use: A Practical Guide to the Essential Models and Methods of Usage-Centered Design. Reading, MA: Addison-Wesley, 1999. Capítulo 8.
27. Sarmento, Luís - Agrupamento de contexto de palavras polisémicas, Relatório Técnico 2006.