M.I. - U.C.E. em Engenharia de Linguagens
Arquivo Digital de Trabalhos Práticos

Igor Sá João Cadinha Mauro Castro

30 de Junho de 2008

Conteúdo

Prefácio

O presente documento serve simultaneamente de relatório e manual de utilização do repositório de trabalhos práticos desenvolvido no âmbito do projecto integrado da U.C.E. em Engenharia de Linguagens. Neste documento falamos da análise do sistema, da base de dados e da aplicação WEB que desenvolvemos. Abordamos as decisões de implementação que tomamos e as funcionalidades que implementamos.

Actualmente os repositórios encontram-se muito em voga e mostram-se de extrema utilidade. Como tal, o CCSDS (Consultative Committee for Space Data Systems) criou uma norma que sugere um conjunto de recomendações que devem ser seguidas no desenvolvimento de repositórios. Esta norma chama-se O.A.I.S. (Open Archive Information System) e tem como principal objectivo assegurar uma preservação permanente ou a longo prazo de informação digital.

No desenvolvimento deste projecto usamos várias tecnologias e linguagens de programação. Para compreender na totalidade este documento o leitor deverá ter conhecimentos básicos de MySQL, PHP, XML, javascript, HTML e Perl. Todas estas tecnologias e linguagens foram-nos extremamente úteis em alguns pontos chave deste projecto. O MySQL foi usado para criar a base de dados que serve de apoio à aplicação e o PHP e o XML foi usado para criar a aplicação em si, gerando código HTML através destes. Sempre que foi pertinente acrescentamos javascript às páginas para facilitar a sua utilização e aumentar a sua usabilidade. O perl foi usado na comunicação com outros repositórios e para criar um motor de pesquisa e recuperação de documentos. Durante a leitura deste documento encontrará também alguns diagramas UML usados para descrever o sistema desenvolvido e com o intuito de facilitar a compreensão do sistema desenvolvido.

O sistema por nós desenvolvido poderá ser encontrado em $http://epl.di.uminho.pt/el07\_g4/$. Algumas funcionalidades apenas podem ser utilizadas por utilizadores registados no sistema, no entanto este registo é instantâneo, como tal aconselha-se a devida experimentação aquando da leitura deste documento.

Capítulo 1

Introdução

1.1 Propósito e Contexto

Com este projecto integrado pretende-se sedimentar os conhecimentos introduzidos nas aulas teóricas dos 4 módulos complementares desta U.C.E.: Engenharia Gramatical, Processamento Estruturado de Documentos Análise e Transformação de Software e Scripting no Processamento de Linguagem Natural. Para atingir este objectivo, foi-nos proposto criarmos um arquivo digital de trabalhos práticos de alunos. Este arquivo será desenvolvido de acordo com o modelo de referência (norma internacional) para a implementação de repositórios digitais: O.A.I.S. (Open Archive Information System). Esta normal internacional pretende sugerir um conjunto de recomendações que deverão ser tomadas em conta aquando do desenvolvimento deste tipo de arquivos digitais. Esta norma pretende assegurar a preservação da informação a longo prazo, formalizando um série de princípios que deverão ser seguidos por todos aqueles que desenvolvem este tipo de arquivos e assegurados por todos aqueles que os mantêm.

1.2 Objectivos da Norma O.A.I.S.

Tal como foi dito na secção anterior, a norma O.A.I.S. pretende fornecer os princípios para a preservação da informação a longo prazo. Deste modo, os objectivos desta norma prendem-se com a preservação da informação e com a sua disseminação. Deste modo, podemos retirar da norma os seguintes objectivos principais:

- fornecer uma ferramenta de trabalho para a compreensão e contextualização dos conceitos de arquivos e sua necessidade de preservação da informação a longo prazo;
- fornecer uma ferramenta de trabalho, incluindo terminologias e conceitos, para descrever e comparar arquitecturas e operações de actuais e futuros arquivos;
- fornecer uma ferramenta de trabalho para descrever e comparar diferentes estratégias e técnicas de preservação da informação a longo prazo;
- fornecer uma base para comparação de modelos de dados da informação digital preservada por arquivos e para a discussão de como esses modelos de dados e respectiva informação podem variar ao longo do tempo;

- expandir o consenso sobre os elementos e processos de acesso e preservação da informação digital a longo prazo;
- guiar a identificação e produção de standards relacionados com a O.A.I.S..

Assim, esta norma define um conjunto mínimo de responsabilidades para um arquivo ser considerado um O.A.I.S..

1.3 Conceitos Básicos num Sistema O.A.I.S.

A O.A.I.S. utiliza alguns termos e nomenclaturas próprias que à primeira vista parecem confusas. Entre estas devemos salientar a definição de S.I.P. (Submission Information Package), a definição de A.I.P. (Archival Information Package) e a definição de D.I.P. (Dissemination Information Package). Um S.I.P. é um pacote de informação que é entregue pelo produtor (utilizador que submete um trabalho prático) ao arquivo digital, que será usado na construção de um ou mais A.I.P.. Um A.I.P. é um pacote de informação que consiste informação em si oriunda do S.I.P. e na informação de descrição da preservação, que será arquivada no arquivo digital. Um D.I.P é um pacote que resulta da derivação de um ou mais A.I.P., recebido pelo consumidor em resposta a um pedido ao arquivo digital. Associado a estes três conceitos básicos da arquitectura de um sistema O.A.I.S. temos a noção de produtor (producer), consumidor (consumer), e gestão (management). Os produtores são o papel desempenhado por todas aquelas pessoas ou sistemas clientes que fornecem a informação a ser preservada. O consumidor é o papel desempenhado por todas as pessoas, ou sistemas clientes, que interagem com o arquivo digital para procurar informação do seu interesse na informação preservada e para acederem a essa informação detalhadamente. A gestão é o papel desempenhado por todos aqueles que tratam da politica do arquivo digital e que asseguram o seu bom funcionamento. Estes são os conceitos básicos necessários para a correcta compreensão da arquitectura de um sistema O.A.I.S..

1.4 Objectivos do Arquivo Digital de Trabalhos Práticos

O nosso arquivo digital de trabalhos práticos tem como principais objectivos permitir a submissão, o arquivo e a disseminação de trabalhos práticos de alunos, respeitando a norma O.A.I.S.. Derivado da norma O.A.I.S. o nosso arquivo tem ainda como objectivo a preservação de informação digital a longo prazo. Para concretizar este arquivo foi necessário desenvolvermos as seguintes tarefas:

- Análise, caracterização e especificação da estrutura e composição de um OD;
- Derivação do modelo relacional que servirá para armazenar os ODs (depois de armazenados passarão a ser A.I.P.);
- Criação da base de dados;
- Criação de uma interface Web (Website) para incorporação de ODs;
- Criação de uma interface Web para pesquisar e aceder aos ODs armazenados;
- Análise do problema com vista a conceber a arquitectura do sistema ADTPs e a funcionalidade a incluir;

- Identificação dos standards a usar para o manifesto e os formatos de arquivo;
- Implementar um sistema de disseminação que permita ao utilizador descarregar cada trabalho prático num ZIP completo, descarregar um ZIP com o relatório, apresentação ou código, onde para facilitar a selecção do conteúdo que se pretende descarregar seja possível navegar interactivamente sobre o filesystem do Trabalho Prático, acedendo on-line à árvore de directorias, e visualização de ficheiros;
- Implementar um sistema de pesquisa textual sobre os ficheiros do relatório;
- Implementar uma API de pesquisa sobre o arquivo através de WebServices, de modo a que os vários arquivos possam consultar a base de trabalhos práticos dos outros arquivos.

Capítulo 2

Análise do Sistema

2.1 Arquitectura Geral do Sistema

Tal como foi dado a entender anteriormente, na arquitectura de um sistema O.A.I.S. temos produtores, gestores e consumidores. De uma maneira geral, um produtor submete informação sob a forma de um S.I.P., que é armazenada sob a forma de um A.I.P.. A informação é guardada, mantida e preservada pelo gestores. Em resposta a um pedido feito por um consumidor, o arquivo disponibiliza um ou mais S.I.P. em resposta. A este ultimo processo chama-se disseminação.

2.2 Gramática Representativa do Sistema

Tal como aprendemos no módulo de Engenharia Gramatical desta U.C.E., podemos usar gramáticas para definir o sistema e os seus requisitos. As gramáticas permitem-nos ter uma ideia global daquilo que pretendemos implementar e de quais são os requisitos do sistema. A seguir está apresentada a gramática que especifica o Arquivo Digital, desde a submissão dos trabalhos práticos, à gestão dos trabalhos internamente e à sua disseminação.

1. ADTPs : Submeter Gestao Disseminacao
2. Submeter : ObjectoDigital
3. ObjectoDigital : trabalhoPratico MetaInformacao
4. MetaInformacao : manifestoXML
5. | PedirMetaInformacao
6. PedirMetaInformacao : titulo subtitulo Autores Orientadores Contexto palavrasChave resumo
7. Autores : Autor
8. | Autores ; Autor
9. Autor : nome , email

10. Orientadores : Orientador
11. | Orientadores Orientador
12. Orientador : nome , email
13. Contexto : disciplina , curso
14. Gestao : InserirDadosBD CriarAIP
15. Disseminação : CriarDIP

2.3 Arquitectura Específica do Sistema

Para nos ajudar na visualização do funcionamento da arquitectura do sistema, passamos a mostrar um diagrama que demonstra da melhor forma o funcionamento de uma arquitectura de um arquivo O.A.I.S.. Basicamente, o sistema recebe um pacote de submissão (SIP) com vários ficheiros de

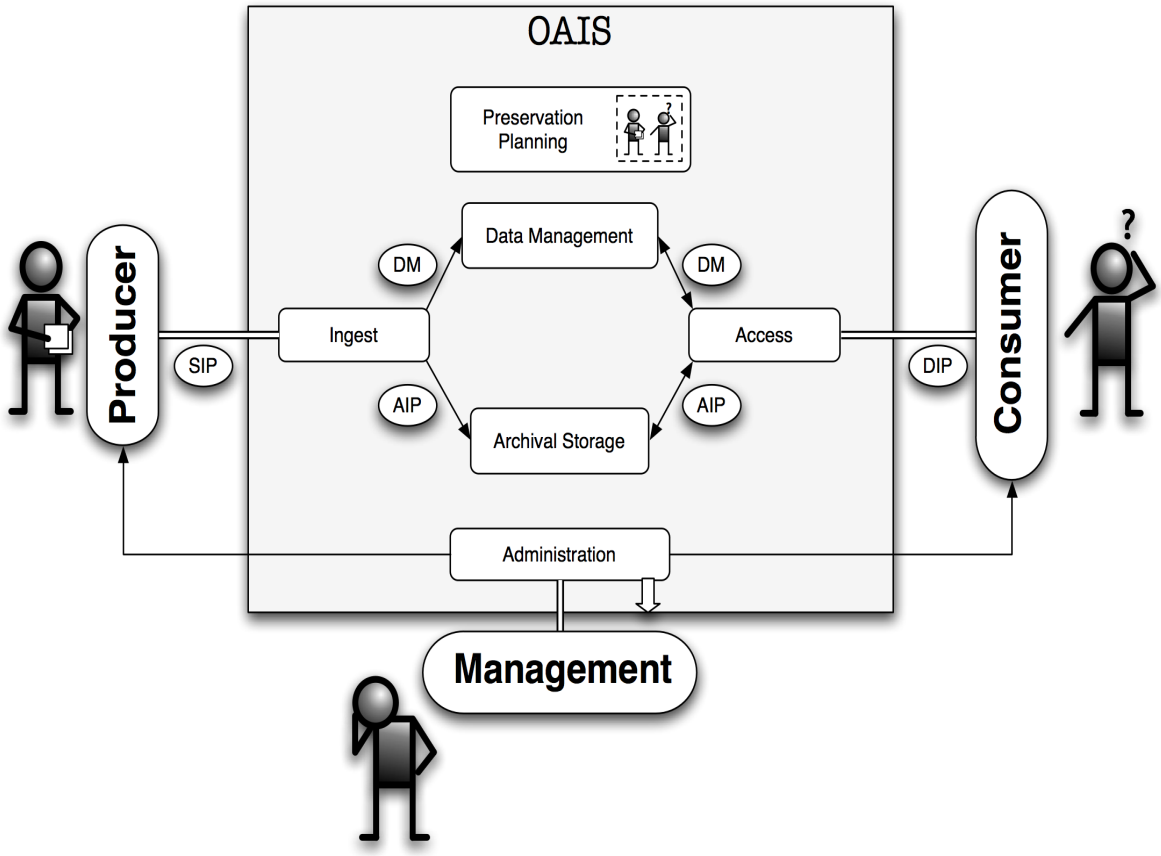

Figura 2.1: Diagrama Representativo de um Sistema O.A.I.S.

tipos diversos (por exemplo, TeX, BibTeX, imagens, PDFs, código-fonte, programas executáveis). O sistema procede à ingestão do SIP, convertendo o conjunto de ficheiros em formatos própios para armazenamento constituindo um novo pacote designado por pacote de arquivo (AIP). Esse AIP é depois transformado, segundo regras de difusão próprias do sistema, num conjunto de novos pacotes prontos para serem difundidos/distribuídos, designados por pacotes de disseminação (DIP).

O ADTPs realiza sobre o AIP um vasto conjunto de operações de gestão do arquivo, destinadas a pesquisar e a manter os ficheiros.

Resumindo o modelo O.A.I.S. comporta 3 actores (o produtor que é o autor ou a entidade ou pessoa que detém os direitos sobre o material a arquivar, o administrador do arquivo que terá o controlo das tarefas de transformação e gestão, e o consumidor que será o futuro utilizador do sistema e que irá realizar operações de pesquisa, consulta e download), 3 megaprocessos (ingestão que corresponde à entrada de materiais no arquivo, gestão/administração que corresponde às acções de transformação e gestão dos objectos digitais armazenados, e disseminação que corresponde à difusão/distribuição de objectos digitais), e 3 pacotes (SIP - "Submission Information Package", AIP - "Archival Information Package"e DIP - "Dissemination Information Package").

Capítulo 3

Desenvolvimento Geral da Aplicação WEB

Neste capítulo iremos percorrer de uma forma geral todos os conceitos, processos e tecnologias que usamos para criar a aplicação WEB base do nosso arquivo. Iremos ver como implementamos a base de dados do arquivo, o sistema de login da aplicação e as linguagens escolhidas para implementar a base de dados e a aplicação WEB.

3.1 Base de Dados

Nesta secção vamos explicar o funcionamento da base de dados de apoio a este arquivo digital. Esta base de dados não só serve para guardar as informações relativas aos ficheiros submetidos nos trabalhos práticos, mas também guarda as informações relativas aos utilizadores, autores e trabalhos práticos com os quais estes se relacionam. O sistema gestor de bases de dados escolhido para guardar a informação foi o MySQL, pois parece servir perfeitamente os nossos interesses, para além de ser open-source.

No que diz respeito a base de dados em si, passo de seguida a descrever as entidades desta, em que consistem e quais é que são os seus atributos. Em primeiro lugar temos a entidade Utilizadores. Esta entidade representa todos os utilizadores do arquivo digital e guarda as informações referentes a estes, como por exemplo, o nome do utilizador, a sua pass-word, o seu e-mail, etc. Esta entidade serve unicamente de apoio à autenticação dos utilizadores do arquivo digital, apesar de conter também algumas informações referentes a estes que são úteis noutras entidades. Depois, temos a entidade TrabalhosPraticos, que representa os trabalhos práticos submetidos. Esta entidade atribuí um número identificador ao trabalho prático e está relacionada com os utilizadores, com o contexto, com os autores e com os ficheiros. De seguida temos a entidade Autores, onde guardamos o nome dos autor e o seu e-mail. Esta entidade encontra-se também relacionada com a entidade Instituicoes, pois cada autor tem que pertencer a uma instituição. A entidade Instituicao guarda os nomes das instituições às quais pertencem os autores e as disciplinas. A entidade disciplinas guarda o nome e o ano lectivo das disciplinas. Temos ainda a entidade Contexto que guarda o contexto em que se insere cada trabalho prático. Para terminar, temos a entidade Ficheiros que guarda o nome, a categoria e a descrição dos ficheiros submetidos em cada trabalho prático. Neste momento, esta base de dados parece-nos ser a mais adequada para usarmos no nosso arquivo. O

programa que usamos para gerar o código SQL a partir do qual criamos a nossa base de dados foi o DBDesinger4. Este programa permite-nos criar facilmente a base de dados através de um interface gráfico.

De seguida apresentamos o diagrama E-R da base de dados criada.

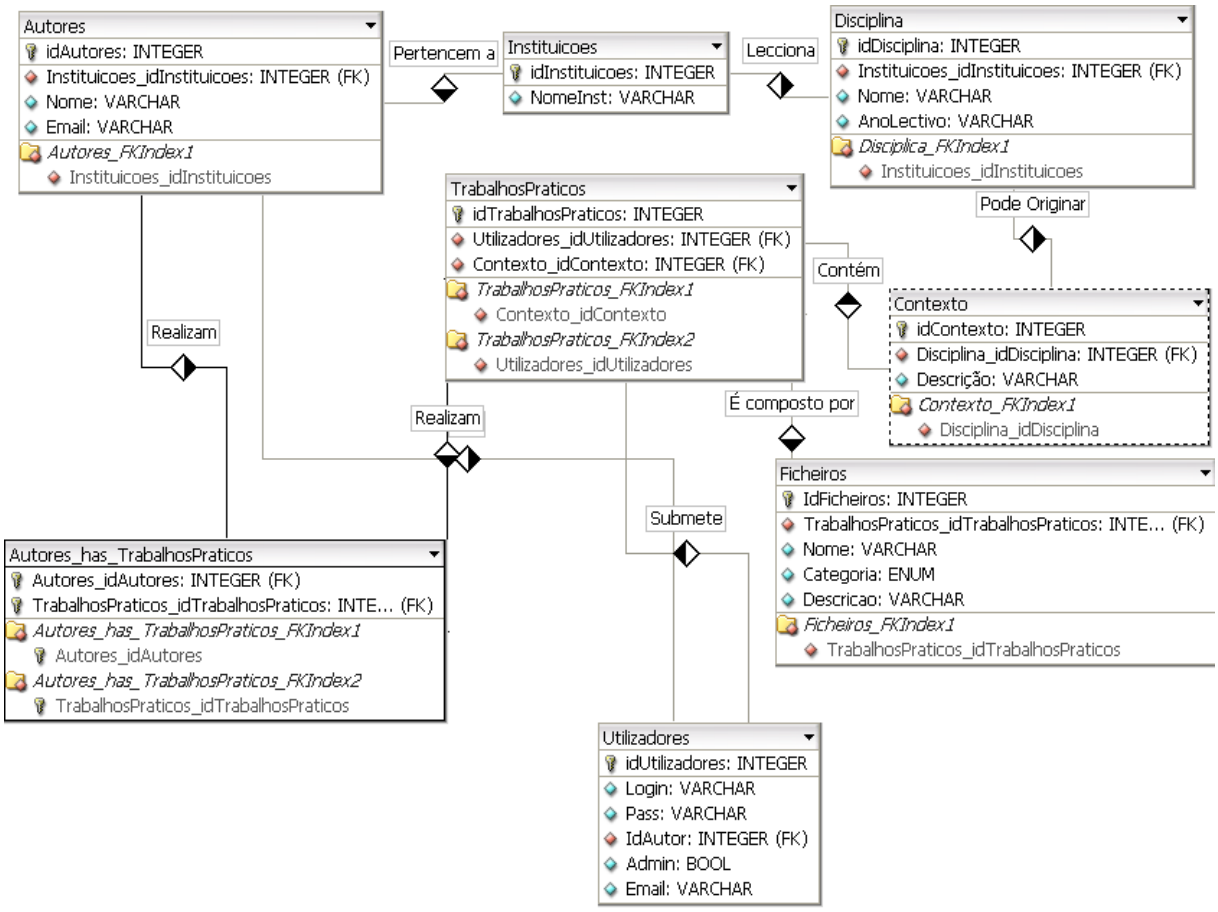

Figura 3.1: Diagrama E-R da Base de Dados do Arquivo Digital

No seguimento deste relatório, sempre que for pertinente, recorreremos a explicações mais detalhadas sobre o conteúdo da base de dados.

3.2 Sistema de Login e Autenticação

Sistema de login e autenticação usado é bastante simples, mas simultaneamente bastante eficiente. Quando o utilizador faz login no site, guardamos o seu username e a sua palavra pass numa sessão no servidor e, antes de mostrarmos cada página, verificamos se estes se encontram correctos. As variáveis de sessão são variáveis que são guardadas no servidor e são únicas para cada browser que fez login no site. Estas variáveis apenas expiram quando o browser deixa de contactar com o

servidor. Os utilizadores não têm acesso a estas variáveis, logo tornam-se bastante seguras.

Apesar de o registo ser fácil e instantâneo, decidimos disponibilizar os trabalhos práticos para consulta por parte de utilizadores não registados pois, muitas vezes, muitos dos utilizadores quando vêm que é necessário registo preferem aceder a outros repositórios. Porém, para submeter um trabalho é necessário registar-se. Após o registo, cada utilizador fica com os privilégios de aluno, ou seja, para além dos privilégios de um utilizador normal, pode ainda submeter trabalhos práticos.

3.3 Interface Gráfico

O sistema de login e autenticação faz parte da interface gráfica, como tal usamos a mesma linguagem para criar esta interface. Em relação à primeira etapa deste projecto alteramos o interface desenvolvido pois o anterior por vezes tornava-se confuso. No desenvolvimento de todo este interface, sempre que possível tentamos completar os inputs do utilizador com auto-completes feitos em javascript por uma questão de usabilidade. O utilizador foi sempre tido em conta no desenvolvimento da aplicação pois pretendíamos criar uma aplicação fácil de usar e bastante intuitiva. Por vezes este tipo de sistemas tornam-se aborrecidos para os utilizadores pois pede-se muitas informações sobre o conteúdo submetido para que a documentação sobre o trabalho seja a mais completa possível, a fim de tornar as pesquisas mais fáceis.

De seguida mostramos uma imagem da base do nosso interface gráfico. Este é o esquema base da interface usada para toda a aplicação web.

Figura 3.2: Interface do Arquivo Digital de Trabalhos Práticos

Capítulo 4

Sistema de Submissão

4.1 Estrutura dos Trabalhos Práticos

Antes de percebermos como funciona o sistema de submissão de trabalhos práticos é importante percebermos qual é a estrutura deste.

Um trabalho prático pode ser constituído por três componentes: relatório, apresentação, aplicação

- Relatório: é constituído por um ficheiro em PDF podendo ter ainda o ficheiro TeX ou XML que lhe deu origem.
- Apresentação: é constituída por um ficheiro em PDF podendo opcionalmente ter ainda o ficheiro PPT ou XML que lhe deu origem.
- Aplicação: é constituída por um conjunto de ficheiros zipados contendo o executável, o código fonte, makefile, um ficheiro README, etc.

Além destas três componentes, o trabalho prático tem ainda alguns campos de metainformação própria, que se enunciam a seguir:

- titulo, subtítulo
- autor(es), (nome, email)
- orientador (nome, email)
- contexto (disciplina, curso)
- palavras-chave
- resumo

4.2 Taxonomia

Um dos maiores problemas nos repositórios é a catalogação da informação. Assim, surgiram várias taxonomias para tentar colmatar estes problema. Neste projecto usamos como referência a taxonomia do ACM. Deste modo, sempre que um utilizador submete um trabalho prático é convidado a associar uma ou várias chaves taxonómicas ao seu trabalho para facilitar as pesquisas sobre os trabalhos que se encontram no repositório. Para guardar a árvore taxonómica do ACM usamos um ficheiro XML e para mostrar esta, quer para catalogação, quer para procura, criamos um ficheiro xslt que transforma a árvore conforme mais nos convém. Esta implementação da taxonomia fornece-nos várias vantagens, entre as quais podemos destacar o facto de se alguma vez actualizarmos a árvore taxonómica, não temos que mexer na estrutura do site, bastando para tal apenas actualizar o ficheiro XML.

A taxonomia do ACM pode ser vista como uma árvore ou como um grafo. Pode ser considerada um grafo uma vez que cada nodo da árvore pode ter como referência outro nodo da árvore, sem que este seja filho do anterior. Uma vez que permitimos que cada trabalho prático pertença a mais que um nodo da árvore, resolvemos ignorar estas referências, deixando a cargo do utilizador tratar destas. Esta decisão permitiu-nos criar um sistema de navegação tipo árvore sobre a taxonomia que é bastante apelativo e fácil de usar para o utilizador, quer nas pesquisas, quer na catalogação dos seus trabalhos.

4.3 Submissão de Trabalhos Práticos

Podemos dividir o processo de submissão de trabalhos práticos em 5 etapas: a submissão do ficheiro ZIP, a análise do conteúdo do ficheiro submetido, a associação de uma taxonomia ao trabalho, a associação de uma disciplina ao trabalho e a inserção dos dados do trabalho. Resolvemos falar de cada uma destas etapas independentemente, focalizando os seus pontos fortes e aquilo que deve ser tido em conta.

4.3.1 Submissão do Ficheiro ZIP

Uma vez que cada trabalho prático é constituído por vários ficheiros, para facilitar a submissão deste resolvemos apenas aceitar que o utilizador submetesse um ficheiro ZIP com o todos os ficheiros desse mesmo projecto. Assim, torna-se necessário processarmos todo o conteúdo do ficheiro para pedirmos as informações adicionais sobre estes ao utilizador. Antes de mais, guardamos o ficheiro no servidor mas alteramos o seu nome para o identificador do trabalho prático que foi atribuído a essa submissão. Depois descompactamos o ficheiro para uma pasta que também vai ter como nome esse identificador. Na raiz dessa pasta vão ser criados 3 ficheiros. O ficheiro folder.txt guarda o nome de todas as pastas que se encontram nessa directoria. O ficheiro file.txt guarda o nome de todos os ficheiros que se encontram nessa directoria. Finalmente, o ficheiro content.txt guarda o caminho e o nome de todos os ficheiros que se encontram no ficheiro ZIP. Este ultimo ficheiro apenas se encontra na raiz da pasta, ao passo que os outros dois encontram-se em todas as pastas que se encontravam dentro do ZIP. Os ficheiros file e content são obtidos através do comando ls e são executados em todas as pastas do conteúdo do ZIP através de invocações recursivas.

Esta solução foi encontrada como alternativa à opção de o utilizador ter que submeter todos os ficheiros individualmente. Optamos por guardar o ficheiro comprimido e o seu conteúdo descomprimido para permitirmos ao utilizador ter acesso a ambos sem ser necessário estar a descompactar

os ficheiros ou a compactá-los na altura, poupando assim processamento no servidor e tornando a aplicação mais rápida. Com esta solução poupamos ainda processamento do servidor pois a análise ao conteúdo do ficheiro ZIP apenas é feita uma vez.

4.3.2 Análise do Conteúdo do Ficheiro ZIP

Depois de analisados os ficheiros que compõem o ZIP, é permitido ao utilizador associar informação a cada ficheiro. A cada ficheiro pode ser associado uma breve descrição do que é, a sua categoria (Apresentação, Relatório ou Aplicação), o seu tipo e ainda o sistema operativo no qual foi criado. Todas estas informações são opcionais, não sendo necessário a sua introdução para continuar o processo de submissão.

4.3.3 Associação Taxonómica

Depois de concluído o passo anterior, é permitido ao utilizador associar ao seu trabalho elementos da taxonomia do ACM. O utilizador pode associar mais do que um contexto ao seu trabalho. A lista taxonómica do ACM é obtida através de uma transformação num ficheiro XML, transformação esta que gera código HTML e javascript. Assim, podemos actualizar a árvore taxonómica sempre que quisermos, bastando para tal substituir o ficheiro XML onde se encontra guardada a taxonomia. Caso o utilizador não pretenda associar nenhum elemento taxonómico ao seu trabalho prático basta apenas clicar em submeter que não se guarda nenhuma informação a este respeito.

4.3.4 Associação de uma Disciplina

Tal como foi especificado, cada trabalho prático tem a ele associado uma disciplina. Como tal, depois de pedidas as informações a respeito da taxonomia, pede-se ao autor que escolha um disciplina de entre as disciplinas disponíveis e pede-se também que escolha o contexto ao qual aquele trabalho prático pertence. Para este efeito usamos um interface semelhante ao interface da taxonomia (em árvore), mas o utilizador apenas pode escolher uma disciplina. Depois de concluído este passo, o utilizador é encaminhado para a inserção dos dados do trabalho.

4.3.5 Inserção dos Dados do Trabalho Prático

Neste último passo para a submissão de um trabalho prático, pede-se ao utilizador que introduza alguns dados sobre o trabalho que acabou de submeter. Desta forma, pedimos ao utilizador que insira o título, o subtítulo, os autores e os respectivos e-mails, um resumo e algumas palavras chave do trabalho prático. Destes dados, o único que é obrigatório para o utilizador é os autores e os seus e-mails. Para facilitar a tarefa ao utilizador, usamos o javascript para criar um auto-complete para estes campos. Para além disso, se o autor que se precisa de introduzir não estiver na lista do auto-complete, o autor adicionado é acrescentado automaticamente à base de dados e aparecerá disponível logo na próxima submissão.

Ainda neste passo, permitimos que o utilizador indique se o finalizou a sua submissão. Caso ele indique que finalizou o trabalho, este deixa de estar disponível para edição, mas passa a estar disponível para consulta por parte de todos os utilizadores. Falta apenas acrescentar que o processo de submissão pode ser abandonado em qualquer altura, podendo depois ser continuado quando o utilizador assim o entender.

4.4 Criação do A.I.P.

Depois de passarmos por todas estas fases na submissão de um trabalho prático, temos toda a informação que precisamos reunida para podermos criar um A.I.P.. Durante o processo de submissão, as informações vão sendo guardadas por forma a que se o utilizador abandonar a submissão, os dados introduzidos não sejam perdidos. O diagrama seguinte tenta compilar todos os passos da submissão para se criar um A.I.P..

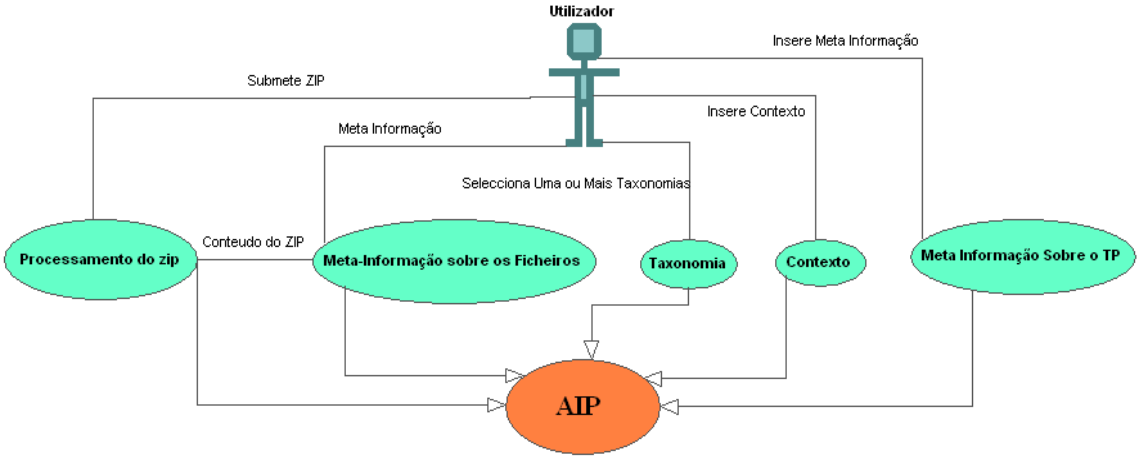

Figura 4.1: Diagrama UML do Sistema de Submissão

Capítulo 5

Arquivo

Neste capítulo descrevemos a forma como guardamos a informação referente a um A.I.P., gerado após o processo de submissão de um trabalho prático. Sobre o arquivo pouco à a dizer. Os ficheiros são guardados no file system, mantendo a sua estrutura original, apenas alteramos o nome do ficheiro ZIP para o numero identificador do trabalho prático. A metainformação referente ao trabalho prático é guardada na base de dados, a fim de estar disponível para as pesquisas feitas por consumidores. Finalmente, o conteúdo do relatório e da apresentação é analisado, sendo o resultado da análise guardado numa DB_File. Uma DB_File é uma base de dados que fica guardada em ficheiro, ou seja, é uma hash map em ficheiro. Desta forma, sempre que o utilizador dá como terminado a submissão do seu trabalho prático, é executado sobre a sua submissão um programa em perl criado por nós que guarda as informações referentes ao conteúdo do seu relatório e da sua apresentação. Estas informações serão muito úteis para pesquisa sobre os trabalhos práticos guardados. O diagrama seguinte explica-nos em que consiste o sistema de arquivo.

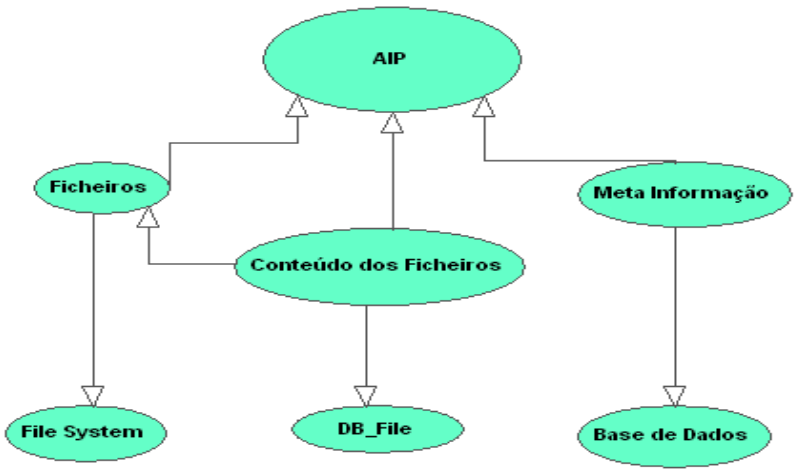

Figura 5.1: Diagrama UML do Sistema de Arquivo

Capítulo 6

Disseminação

A disseminação é uma partes mais importantes de um sistema OAIS (Open Archive Information System), pois é na disseminação que se realizam todas as operações que visam disponibilizar para os consumidores a informação que estamos a guardar. Neste capítulo falamos um pouco da disseminação e explicamos alguns passos que seguimos para atingirmos o resultado final.

6.1 Análise dos Requisitos da Disseminação

Quando se refere a Disseminação no contexto do nosso Arquivo Digital de Trabalhos Práticos, estamos a falar das várias formas possíveis que um consumidor tem de aceder à informação guardada pelo sistema. No nosso sistema, esta informação é guardada sob a forma de AIPs mas é fornecida aos consumidores sob a forma de DIPs. Um AIP (Archival Information Package) é um pacote de informação que consiste num trabalho prático submetido e na descrição da informação preservada. Um DIP (Dissemination Information Package) é um pacote de informação que deriva de um ou mais AIPs, que é recebido pelo consumidor em resposta a um pedido feito por este ao nosso arquivo digital. Sendo assim, um DIP contém a totalidade ou parte de um objecto digital, meta-informação do pacote, uma lista do seu conteúdo (manifesto), informação sobre os direitos de autor do pacote.

Durante a análise dos requisitos da disseminação, a primeira questão que se colocou foi saber quando é que se deveria gerar um DIP. Podemos gerar um DIP sempre que um consumidor o pede ou podemos gera-lo quando algum utilizador submete um AIP e actualiza-lo sempre que este é alterado. Cada uma destas abordagens tem a sua vantagem. A primeira permite poupar bastante espaço no servidor, mas corre o risco de tornar o repositório muito lento pois sempre que cada utilizador requisita um DIP estaremos a criá-lo no momento. A segunda abordagem tem como vantagem o facto de tornar o repositório mais rápido, pois exige menos processamento (Teoricamente são submetidos menos trabalhos do que são consultados pelos utilizadores).

Tal como falamos nos objectivos, interessa-nos disseminar os trabalhos práticos de três formas:

1. O trabalho prático completo;
2. Apenas uma das partes do trabalho:
    - Relatório;

- Código;
- Apresentação;

3. Apenas um ficheiro do trabalho prático

No decurso da análise de requisitos detectamos que também seria interessante permitir ao utilizador, durante a navegação na árvore de ficheiros, seleccionar alguns ficheiros e criar um DIP apenas com os ficheiros seleccionados. Estes DIP têm que ser gerados obrigatoriamente quando o utilizador o requisita pois, caso contrário, ao guardarmos todas as combinações possíveis de componentes de um trabalho prático correríamos o risco de ficar sem espaço em disco no servidor, pois mais um ficheiro num trabalho prático aumentaria o seu tamanho exponencialmente.

Um DIP enviado para um utilizador tem que conter as seguintes informações:

1. a árvore de directorias referente ao trabalho prático;
2. uma nota com os direitos de autor;
3. meta-informação referente ao pacote (autores, título, data, resumo...);
4. um manifesto com o conteúdo do pacote.

No que diz respeito aos direitos de autor, existem três alternativas possíveis. Uma dessas alternativas é existir apenas uma política de direitos de autor para todos os trabalhos práticos e, quando um utilizador submete um, ele está a concordar que abdica de todos os seus direitos. Outra alternativa é cada utilizador submeter os seus direitos de autor num ficheiro junto do trabalho prático. A ultima é o utilizador escolher um de vários direitos de cópia quando submete o seu trabalho.

A meta-informação referente ao DIP consiste no título do trabalho prático, autores, resumo, data de submissão, entre outras informações referentes ao trabalho prático. O manifesto do DIP é lista de ficheiros que estão no DIP. Uma forma de enviar-se esta informação para o utilizador é juntar a meta-informação e o manifesto no mesmo ficheiro.

No que diz respeito à navegação sobre os trabalhos práticos concluímos haver duas formas de navegar sobre estes. A primeira é através de uma árvore de ficheiros. A segunda é permitir a navegação sobre as pastas e ficheiros e depois mostrando a meta-informação referente a cada um destes.

6.2 Decisões de Implementação

No decurso da implementação dos objectivos pretendidos, tomamos algumas decisões resultantes da análise dos requisitos. Vamos de seguida documentar estas.

6.2.1 DIPs

Para nós, existem cinco formas de DIPs:

1. A totalidade do objecto digital (trabalho prático);
2. Apenas a apresentação;

3. Apenas o código;

4. Apenas o Relatório;

5. Partes do trabalho prático seleccionadas pelo utilizador;

Todas estas formas de DIPs estão disponíveis para o utilizador, respeitando a definição de DIP mencionada na análise de requisitos, ou seja, para além dos ficheiros, adicionamos ainda a meta-informação, o manifesto e os direitos de autor ao DIP.

Criação do DIP

Os pacotes de disseminação são gerados durante a submissão do trabalho prático, com a excepção do pacote em que o conteúdo é escolhido pelo consumidor. Aos pacotes é adicionada ainda informação sobre o seu conteúdo e sobre o trabalho prático. Esta decisão decorre da análise de requisitos, uma vez que gerar os DIPs quando são pedidos pelo utilizador pode sobrecarregar o servidor, ao passo que criar os DIPs personalizados durante a submissão levaria a falta de espaço no disco do servidor. Em cada alteração efectuada num AIP, nós decidimos que essa alteração é feita sobre os DIPs gerados anteriormente, aquando da primeira submissão.

Meta-informação e Lista do Conteúdo (manifesto)

A meta-informação do trabalho prático e a lista do conteúdo do DIP estão presentes num único ficheiro xml de nome README. Este ficheiro possui já arvore de directorias do DIP, estando então organizado da seguinte forma:

- Meta-informação
  - Titulo, subtítulo
  - Autores
  - Orientador
  - Disciplina
  - Data de submissão
  - Resumo
- Lista do conteúdo
  - possui a árvore de directorias, possuindo também para cada ficheiro a respectiva meta-informação

O dialecto XML em que está escrito este ficheiro README foi definido por nós, mas inspirado em outros que consultamos. Este dialecto parece ser perfeito para guardar toda a meta-informação e o manifesto. O facto de estes dois estarem definidos num único ficheiro favorece a simplicidade do DIP, tornando-o mais amigável para o utilizador.

Direitos de Cópia

Para que seja possível a submissão de um trabalho, é necessário concordar com os termos do arquivo.Esta noticia de COPYRIGHT é única, isto é, todos os trabalhos presentes neste arquivo terão os mesmos direitos cópia. Os direitos de cópia são adicionados ao DIP em ficheiro pois o consumidor de DIP pode não ser obrigatoriamente um utilizador e não ter conhecimento deste direitos, como tal este ficheiro serve para relembrar os consumidores que conhecem estes direitos e para informar aqueles que não os conhecem.

6.2.2 Navegação Sobre os Trabalhos Práticos

É possível ao utilizador navegar sobre o trabalho prático, podendo aceder ao conteúdo das pastas e ver os ficheiros nelas contidos. Para cada ficheiro pode aceder-se à respectiva meta-informação, e também fazer-se o download individual do ficheiro. Pode ainda criar-se um DIP à medida do utilizador, isto é, ao navegar-se sobre o trabalho prático é possível escolher quais os ficheiros que se quer adicionar ao DIP. Depois de escolhidos os ficheiros é criado o DIP incluindo também toda a informação adicional (direitos de cópia,meta-informação e lista do conteúdo). A visualização da meta-informação referente a cada ficheiro é feita em simultâneo com a navegação por questões de usabilidade. Mostrar a meta-informação numa janela separada da navegação pode fazer com que o utilizador se perca, levando a que este receba um DIP que não era aquilo que ele desejava.

6.3 Conclusão

Deste modo, a partir do A.I.P. arquivado no nosso sistema criamos um D.I.P. por forma a distribuirmos a informação pedida pelo consumidor. O diagrama seguinte explica de forma sucinta a forma como funciona o sistema de disseminação da informação.

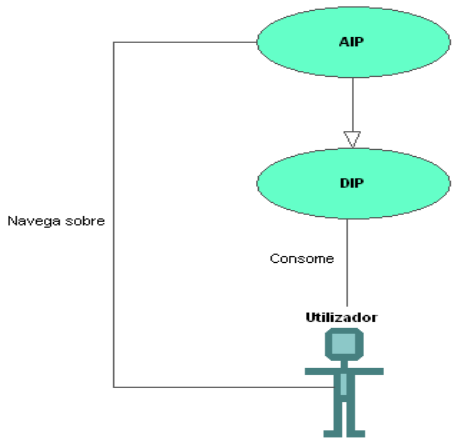

Figura 6.1: Diagrama UML do Sistema de Disseminação

Capítulo 7

Pesquisa

Existem várias maneiras de pesquisar, ou seja, de aceder à informação guardada no nosso repositório. De seguida passamos a descrever estas maneiras.

7.1 Consulta dos Trabalhos Criados pelo Próprio Utilizador

Para consultar os trabalhos práticos submetidos pelo utilizador, este tem à sua disposição a opção Meus Trabalhos Práticos no menu do lado esquerdo. Ao seleccionar esta opção são listados todos os trabalhos submetidos pelo utilizador que se encontra autenticado no sistema. Como um professor também pode submeter trabalhos práticos, este também tem disponível esta opção. Os trabalhos listados mostram algumas informações sobre estes, das quais podemos destacar o título, o resumo, a disciplina... Nos trabalhos práticos que não se encontram finalizados está disponível uma opção para editá-los. Os trabalhos práticos finalizados, aparecem na lista mas apenas se encontram disponíveis para consulta, pois uma vez que já se encontram finalizados já não podem ser editados.

7.2 Pesquisa por Keywords

Tal como era pretendido no enunciado deste projecto integrado, implementamos um sistema de pesquisa de trabalhos práticos por keywords. Esta opção encontra-se disponível seleccionando a opção Procura no menu. Para além da pesquisa por keywords adicionamos a funcionalidade de pesquisar simultaneamente por autor, título e subtítulo. Assim, nesta opção o utilizador apenas tem que indicar o que é que quer procurar, podendo escolher mais que uma opção de cada vez. A pesquisa por keywords procura na base de dados por trabalhos que tenham keywords semelhantes às introduzidas. Estas keywords que são procuradas são as keywords que são pedidas ao utilizador para introduzir. Caso este não tenha introduzido nenhuma na aquando da submissão do trabalho, o seu trabalho não aparecerá em nenhuma procura deste tipo.

7.3 Pesquisa por Taxonomia

Para além das pesquisas por keywords à qual adicionamos algumas funcionalidades, neste projecto integrado também se pretendia que fosse implementado um sistema de pesquisa por taxonomia. Assim, servimo-nos mais uma vez do XML para resolver este problema. Assim, sempre que o utilizador selecciona a opção Procurar por Assunto tem acesso à árvore taxonómica do ACM. O utilizador pode navegar nesta e escolher qualquer elemento desta, quer seja folha ou nó, e de seguida é encaminhado para a lista de elementos que estão associados a este e que estão associados a filhos deste. Para este sistema tivemos como referência o eprints e acrescentamos-lhe a funcionalidade de consulta dos elementos filhos dos nós. Tal como foi dito anteriormente, esta procura tem por base o ficheiro xml onde se encontra guardada a estrutura da árvore e tem um ficheiro xslt que aplica transformações nessa árvore para gerar o código HTML e javascript que compõem a página onde se pode navegar sobre esta árvore.

7.4 Pesquisa no Conteúdo dos Ficheiros

Esta foi uma funcionalidade adicionada numa segunda fase da realização deste projecto. Aquando da criação do A.I.P. é processado o conteúdo do relatório e da apresentação. Do seu conteúdo guardamos os radicais das palavras que encontramos, usando o JSpell para obtermos esses radicais. Depois, implementamos um sistema de pesquisa sobre essas palavras guardadas na DB_File. Nesta pesquisa permitimos o uso de operadores lógicos AND, OR e NOT de forma a que o utilizador consiga obter os resultados mais próximos daquilo que procura. Mais uma vez usamos o JSpell para obtermos os radicais das palavras pesquisadas pelo utilizador. O resultado final apresentado apresenta-se ordenado por relevância em relação às palavras pesquisadas. Esta pesquisa é muito eficiente e permite obtermos resultados mais próximos daquilo que procuramos, pois ao analisarmos o conteúdo do relatório e da apresentação estamos a disponibilizar uma pesquisa sobre o conteúdo do trabalho prático e não sobre o título ou subtítulo que muitas vezes é demasiado genérico, podendo por vezes não corresponder aquilo que o utilizador procura.

Capítulo 8

WebServices

Neste capítulo iremos falar da API que implementamos que permite que o nosso repositório comunique com outros repositórios por forma a completar as pesquisas feitas sobre os segundos e também permitir que outros repositórios pesquisassem sobre os trabalhos práticos que nós guardamos. Desta forma, de uma forma muito simplificada, abrimos o nosso repositório a outros repositórios, ou seja, formamos uma rede de repositórios que comunicam entre si por forma a disponibilizarem os seus conteúdos ao utilizador, evitando que este tenha que repetir a sua pesquisa nos segundos. Podemos considerar os webservices como uma das ferramentas mais importantes da disseminação se considerarmos que, a partir deste momento, o nosso repositório é um consumidor noutros repositórios e que entre os nossos consumidores estão outros repositórios. De seguida passamos a descrever os métodos, tecnologias e decisões que tomamos e usamos nesta implementação, justificando sempre que possível a sua utilização.

8.1 Objectivos

Nesta pretendia-se implementar uma API que permitisse a partilha de arquivos e consulta automática de trabalhos práticos de outros repositórios. Desta forma, era necessário implementar uma API de pesquisa sobre o arquivo através de webservices, de modo a que os vários arquivos possam consultar a base de trabalhos práticos dos outros arquivos. Durante as aulas dedicadas a este projecto integrado ficou decidido que deveria ser implementada a seguinte API:

- TextSearch: Query —> Result*
  - Query: String do género word AND word;
  - Result: Um resultado é um bloco HTML com título, link para o objecto digital e autores;
- TaxonomySearch: Category+ —> Result*
  - Category: Código ACM de uma categoria da taxonomia;
  - Result: Um resultado é um bloco HTML com título, link para o objecto digital e autores;
- KeywordSearch: Keyword+ —> Result*
  - Keyword: uma palavra chave ou um par "campo:palavra-chave". Os campos suportados: título, autor, orientador e resumo;

– Result: Um resultado é um bloco HTML com título, link para o objecto digital, e autores.

8.2 Webservices

Um webservice é um sistema de software desenhado para suportar a interoperabilidade entre máquinas e a interacção sobre uma rede. No nosso caso, o nosso webservice é apenas uma API Web que pode ser acedida pela internet e que executa no nosso servidor os pedidos do cliente. Muitos sistemas usam um dialecto XML para comunicar entre os clientes e o servidor chamado SOAP. Na próxima secção explicamos mais detalhadamente o que é o SOAP.

Deste modo, um webservice é um conjunto de ferramentas que pode ser usado de várias maneiras. Existem três estilos principais de utilização:

- Remote Procedure Calls(RPC): Os webservices RPC apresentam várias funções ou métodos, a que normalmente se dá o nome de interface, que são conhecidos pela maioria dos programadores. Neste caso, o servidor apresenta o resultado do método chamado pelo cliente.
- Service-oriented Architecture(SOA): Neste tipo de webservices, ao contrário dos RPC, a unidade básica de comunicação é uma mensagem, em vez de uma operação.
- Representational State Transfer(RST): Neste tipo de webservices tenta-se emular o protocolo HTTP ou semelhantes, restringindo o interface a um conjunto bem conhecido de operações. Nos RST foca-se principalmente a troca de recursos, em vez de mensagens ou operações.

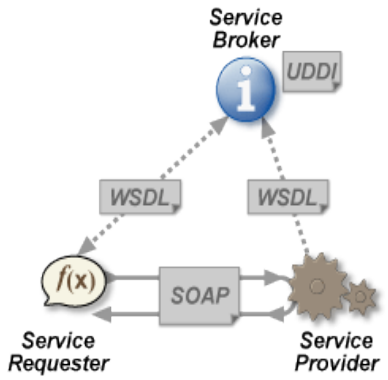

Figura 8.1: Arquitectura de um Webservice

8.3 SOAP

O SOAP é um protocolo para a troca de mensagens através de redes, normalmente HTTP ou HTTPS, baseado em XML. O SOAP utiliza o protocolo das aplicações da internet como protocolo de transporte. Tanto o SMTP como o HTML são protocolos da camada de aplicações válidos e usados para o transporte para o SOAP, mas o HTTP ganhou maior aceitação pois funciona

melhor com as infrastucturas de internet actuais (especialmente porque o HTTP funciona bem com as firewalls). O XML foi escolhido como standard para a formato das mensagens devido à grande utilização por grandes empresas e por programadores open-source. Adicionalmente, existem ainda uma grande variedade de ferramentas grátis que tornam bastante fáceis as migrações para implementações baseadas em SOAP. O tamanho da sintaxe do XML pode ser um beneficio mas, simultaneamente, um ponto contra. Enquanto promove a fidelidade para os humanos, pode também retardar a velocidade de processamento. Deste modo, o SOAP apresenta as seguintes vantagens:

- Usar o SOAP sobre HTTP permite uma comunicação mais fácil através de proxies e firewalls do que as tecnologias de execução remotas anteriores;
- O SOAP é versátil o suficiente para permitir o uso de outros protocolos de comunicação;
- O SOAP é independente de plataformas;
- O SOAP é independente de linguagens;
- O SOAP é simples e extensível.

No entanto, o SOAP apresenta algumas desvantagens:

- Devido ao XML, o SOAP pode ser consideravelmente mais lento do que outras tecnologias semelhantes.
- Ao depender do HTTP para o transporte, os papeis na interacção cliente-servidor são fixos, ou seja, apenas o cliente pode fazer pedidos ao servidor e o servidor não pode notificar o cliente.

Apesar de tudo, no consenso geral, o SOAP continua a ser uma optima maneira de implementar webservices, sendo usado na maioria destas aplicações.

8.4 Decisões de Implementação

Nesta secção falamos um pouco das decisões que fomos tomando durante a execução desta etapa do projecto integrado, salientando a nossa escolha do tipo de webservice, utilização do SOAP e estratégia de desenvolvimento, entre outras coisas.

8.4.1 Tipo do Webservice

O tipo de webservice que decidimos usar é o RST, pois com o uso do SOAP, permite de forma rápida e eficaz implementar este. Uma vez que as operações permitidas pelo servidor foram previamente discutidas nas aulas, todos aqueles que desenvolveram aplicações cliente para o nosso webservice conhecem profundamente as operações permitidas, logo seria lógico usarmos este tipo de servidor. Para além disso, era pretendido que o servidor respondesse a cada cliente um determinado objecto bem definido, título, link para o objecto digital e autores, ou seja, o servidor responde com informação e recursos, pelo que não permite executar operações.

8.4.2 Uso do SOAP

Apesar das suas desvantagens, as vantagens do SOAP e a sua larga utilização em webservices demonstram as suas capacidades excelentes. Neste tipo de repositórios, e uma vez que pretendemos essencialmente utilizar webservices para completar pesquisas sobre os repositórios, não se justifica que o servidor tenha a capacidade de notificar os clientes sempre que é submetido um novo trabalho prático. Deste modo, o SOAP apresenta-se como uma excelente solução para esta implementação.

8.4.3 Estratégia de Desenvolvimento

Depois de decidido, de uma maneira geral, o que iríamos implementar e como iríamos implementar, faltava decidirmos por onde começar, se pelo lado do cliente, se pelo lado do servidor. Começar pelo lado do cliente tinha como vantagem permitir que os outros grupos pudessem testar os seus servidores, assim que estes estivessem implementados. Por outro lado, enquanto estes não estivessem implementados não teríamos maneira de testar o nosso cliente. Começar pelo lado do servidor permitia que os outros grupos pudessem experimentar os seus clientes assim que o nosso servidor estivesse pronto. Deste modo, decidimos começar a implementação dos webservices pelo lado do servidor, pois se todos os grupos acabassem ao mesmo tempo, depois poderíamos todos testar simultaneamente os nossos clientes e os nossos servidores. Esta decisão foi tomada por todos os grupos, pelo que todos tivemos o trabalho facilitado nos testes.

8.4.4 Perl para Cliente e Servidor

Depois de tomadas todas as decisões anteriores o projecto estava pronto para ser iniciado. Mas antes de o começarmos a implementar era necessário escolher a linguagem que deveríamos utilizar. Uma destas alternativas possíveis seria utilizar C#.net para desenvolver o servidor. Esta solução não seria muito viável pois a nossa base de dados é em MySQL e este não comunica bem com esta. Outra hipótese seria utilizar java. Esta hipótese foi abandonada pois seria moroso. Outra hipótese possível seria ainda usar o PHP para implementar os webservices. Porém, o PHP não tem a qualidade nem a fiabilidade do Módulo DBI para o perl para ligação à base de dados. Deste modo decidimos usar o Perl para codificar tanto o lado do cliente como o lado do servidor. Para além disso o Perl apresenta excelentes APIs para o processamento de expressões regulares o que nos permite receber uma string ou um array com os pedidos dos clientes e fazer uma conversão para SQL. Como única desvantagem temos o facto de termos que criar uma CGI para podermos disponibilizar o serviço.

8.5 Conclusões

Após o planeamento e realização deste serviço pudemos concluir que a criação de um webservice dinamizou o nosso arquivo digital de trabalhos práticos pois, ao permitir que outros arquivos comunicassem com o nosso e como o nosso comunica com outros, estamos a melhorar o serviço fornecido a todos os utilizadores. O webservice foi de fácil implementação e a partir dele disponibilizamos um serviço útil e fiável. O SOAP é permite-nos a troca de mensagens de forma útil e eficiente.

Capítulo 9

Considerações Finais

9.1 Propostas de Melhoria

Após e durante a realização deste trabalho prático foram notórias alguma melhorias que podiam ser feitas a este. Em parte por falta de tempo, poderíamos ter acrescentado algumas funcionalidades a este repositório que o tornariam ainda mais interessante. Uma dessas melhorias seria a implementação do sistema de avaliação dos trabalhos submetidos por alunos. Visto que muitos destes trabalhos serão submetidos por alunos teria toda a lógica disponibilizar a quem consulta os trabalhos arquivados a nota obtida após a avaliação destes. Seria também interessante permitir que fosse anexado a cada trabalho um relatório da avaliação destes, para que fosse mais claro para quem os consulta os melhores aspectos e os pontos mais fortes que cada trabalho contém. Esta funcionalidade poderia acrescentar rigor ao nosso repositório e sem duvida que iria aumentar a credibilidade dos trabalhos por eles guardados, uma vez que quem os consulta teria a certeza que cada trabalho era revisto antes de ser disponibilizado.

Outra funcionalidade que poderia ter sido implementada era o uso de um servidor de mail para informar os utilizadores sempre que ocorresse alguma avaliação dos seus trabalhos. Para além disso, poderíamos utilizar o mesmo servidor para validar os registos de novos utilizadores no servidor. Assim, conseguiríamos manter os utilizadores informados do status dos seus trabalhos submetidos e poderíamos acrescentar a funcionalidade de sistema de submissão e avaliação de trabalhos práticos a este repositório.

Outra funcionalidade que poderia ter sido implementada era a análise da existência de virus nos ficheiros ZIP submetidos. Pensamos que não há muito a acrescentar sobre esta funcionalidade.

9.2 Conclusão

Após a realização deste projecto integrado pudemos concluir que todos os objectivos foram concluídos com sucesso, porém ficamos com um gosto amargo por sabermos que mais funcionalidades poderiam ser acrescentadas a este projecto. Pensamos que todas as funcionalidades referidas nas propostas de melhoria iriam valorizar este repositório. A nossa aplicação privilegia a funcionalidade e a usabiliadade, tentando facilitar ao máximo as tarefas pedidas ao utilizador. Sempre que possível usamos o javascript para auxiliar o utilizador nas tarefas necessárias.

No decorrer deste projecto, apercebemos-nos das vantagens do uso do XML como linguagem de anotação e como ferramenta de auxilio para a criação de páginas WEB. Neste projecto o seu uso centrou-se principalmente no que toca a taxonomia mas esta utilização abriu-nos portas para outras aplicações possíveis deste. Outra linguagem que também nos foi muito útil foi o perl. A grande quantidade de módulos disponíveis permitem que seja possível desenvolver aplicações poderosas e eficientes em pouco tempo e de forma bastante simples.

Com a realização deste projecto vimos ainda a importância de uma boa análise de pré-requisitos e modelação do sistema (através do uso de gramáticas), quer para facilitar o processo de desenvolvimento da aplicação, quer também para nos certificarmos que o sistema responde ao requisitos. Devemos ter muito cuidado ao construir a base de dados, para que esta permita ser adaptada a novas situações sem trazer problemas ao que já foi desenvolvido.

Pensamos que realizamos um bom trabalho e teríamos todo o gosto em acrescentar novas funcionalidades a este num futuro próximo.

Capítulo 10

Bibliografia

1. Allen,J., Hornberger,C., ”Mastering PHP 4.1”, Sybex
2. http://www.w3schools.com
3. http://eprints.di.uminho.pt