

Arquivo Digital de Trabalhos Práticos

Carlos Eduardo da Silva Pereira *N.º* 11070
Nuno Ernesto Salgado Oliveira *N.º* 10994

29 de Junho de 2008

Resumo

Este projecto surge no âmbito da UCE-30 de Engenharia de Linguagens do primeiro ano do Mestrado em Informática.

Os arquivos digitais surgiram com o forte aumento da necessidade de preservar informação que, de algum modo, pode ser preservada digitalmente. Esta informação é guardada sob a forma de Objectos Digitais, que na sua essência são a unidade básica para repositórios electrónicos.

Com o rápido aumento de repositórios a serem criados, foi necessário implementar medidas para normalizar a construção deste tipo de sistemas de informação, nascendo assim o método OAIS (Open Archival Information System). Este método comporta, muito sumariamente, três fases importantes para a construção e divulgação dos objectos digitais, o que se traduz na criação de pacotes de informação denominados por os SIPs, AIPs e DIPs.

Com este relatório de desenvolvimento pretende-se explicar e documentar todas as fases relevantes à implementação de um arquivo digital, centrando esforços num caso específico de um arquivo digital de trabalhos práticos, seguindo o método OAIS.

Palavras Chave: Arquivo Digital, Objecto Digital, Repositório Electrónico, OAIS, SIP, AIP, DIP.

Conteúdo

1	Introdução	5
2	Análise	7
2.1	Introdução ao OAIS	7
2.2	Análise do Problema	9
2.3	Análise Temporal de Desenvolvimento	10
3	Captura de Requisitos	12
3.1	Actores	12
3.2	Funcionalidades	13
4	Concepção do Arquivo	15
4.1	Estrutura Estática	15
4.2	Restrições à Estrutura e à Funcionalidade	17
5	Implementação	18
5.1	Tecnologias Utilizadas	18
5.2	Base de Dados e Entidades	19
5.3	Criação da Interface Base	21
5.3.1	Decisões de Implementação	21
5.3.2	Resultados Obtidos	22
5.4	Submissão e Arquivo de Trabalhos Práticos	22
5.4.1	Discussão de Implementação e Decisões	25
5.4.2	Resultados Obtidos	27
5.5	Pesquisa - Base	27
5.5.1	Discussão de Implementação e Decisões	30
5.5.2	Resultados Obtidos	31
5.6	Disseminação	33
5.6.1	Árvore de Directorias	33
5.6.2	Copyright	36

5.6.3	Manifesto e Meta-Informação	39
5.6.4	Disseminação	41
5.7	Pesquisa - Information Retrieval	44
5.7.1	Considerações Iniciais	45
5.7.2	Discussão de Ideias	47
5.7.3	Decisões Tomadas	48
5.7.4	Pesquisa	50
5.7.5	Resultados Obtidos	53
5.8	Web Services	54
5.8.1	Discussão e Decisões de Implementação	55
5.8.2	Os Serviços	56
5.8.3	Resultados Obtidos	57
6	Conclusão	59
6.1	Trabalho Futuro	60
A	Apêndice - Descrição dos Ue Cases	62
A.1	Registo	62
A.2	Pesquisar	62
A.2.1	Pesquisar Por Palavra Chave	62
A.2.2	Pesquisa por Taxonomia	63
A.3	Alterar Dados Pessoais	63
A.4	Submeter TP	63
A.5	Consultar Trabalho Prático	65
A.6	Alterar TP	65
A.7	Inserir Extensões	65
A.8	Inserir Propriedades	66
A.9	Confirmar/Cancelar Registo Professor	66
B	Apêndice - Diagrama ER	68
C	Exemplo de Manifesto Gerado Automaticamente	70

Lista de Figuras

2.1	Modelo de referência OAIS	8
2.2	Diagrama de distribuição temporal de tarefas	11
3.1	Diagrama de Use Cases para o sistema ADTPis	14
4.1	Diagrama de Classes para o sistema ADTPis	15
4.2	Gramática que define a estrutura do sistema ADTPis	16
5.1	Diagrama Entidade-Relacionamento para a base de dados de suporta ao sistema ADTPis	20
5.2	Página principal do sistema ADTPis	23
5.3	Página de área pessoal de um utilizador do sistema ADTPis	23
5.4	Página para alteração da informação de um trabalho	24
5.5	Página principal da fase de submissão e armazenamento dos dados do OD	28
5.6	Página de submissão de ficheiros para compor o trabalho prático	29
5.7	Página de pesquisa de trabalhos práticos por caminho da ta- xonomia ACM	31
5.8	Página que apresenta a informação de um trabalho prático resultante de uma pesquisa	32
5.9	Página renovada para o upload de ficheiros e criação de novas directorias	35
5.10	Página de apresentação de escolha do Copyright associado ao TP	38
5.11	Página de Consulta de um TP, com a disponibilização de DIPs	43
5.12	Página de criação de Pacotes de informação personalizados	44
5.13	Aspecto do DER focado nas tabelas referentes ao suporte de informação para IR	46
5.14	Gráfico que descreve o decréscimo de eficiência em função do número de palavras indexadas	50

5.15	Diagrama do fluxo de execução da tarefa de Indexação de Palavras	51
5.16	Diagrama do fluxo de execução da tarefa de Pesquisa por IR	52
5.17	Interface de Pesquisa no interior de documentos.	53
5.18	Interface de Apresentação dos Resultados da Pesquisa no interior de documentos.	54
5.19	WSDL com folha de estilo associado para o serviço de pesquisa textual.	58
5.20	Resultado da utilização do web service definido	58
B.1	Diagrama ER para o sistema ADTPis	69

Capítulo 1

Introdução

Nos dias de hoje, há informação sobre tudo e em todo o lado. A Internet é a principal aposta para difusão dessa informação. Muitos utilizam simples páginas de internet com a informação guardada na própria página, outros usam sítios de internet com suporte de bases de dados para divulgar a informação.

Contudo, há ainda muitos casos em que a informação não está digitalizada, ou se está, tem vários defeitos. Para combater esses defeitos e normalizar os pacotes de informação, são criados repositórios electrónicos

Um repositório, é como o próprio nome indica, um sistema de informação documentada que guarda a meta-informação e a localização associada a vários objectos digitais, assegurando a preservação por longos tempos desses objectos [TUT], apesar da verdadeira longevidade de um objecto digital ser de aproximadamente 5 anos [FCR⁺07].

Estes objectos digitais podem ser qualquer coisa passível de ser digitalizada, como por exemplo artigos, livros, documentos de empresas, documentos históricos, documentos políticos, etc.

Contudo, a palavra digitalizar pode muitas vezes ser mal interpretada. A digitalização de um documento não é sempre, converter esse documento para uma imagem digital através de um *scanner*, podendo, posteriormente, ser destruído o objecto físico. Pelo contrário, a digitalização de um documento pode ser a passagem da sua informação para documentos em formato digital, como é o caso do XML e a utilização de normas como METS¹, que depois poderá ser guardada em bases de dados apropriadas para acolher tal informação. No entanto essa informação não vale de muito caso o documento em si seja destruído, logo é intuitivo que a digitalização de documentos incorpore o armazenamento da localização física do objecto.

¹METS é uma norma que fomenta a normalização e estruturação da meta-informação de objectos digitais, possibilitando a troca destes objectos entre vários repositórios que sigam as mesmas normas

Um repositório electrónico é ainda visto como meio de difusão de informação. Assim um repositório num sentido menos abstracto ganha a forma de um Arquivo Digital o qual disponibiliza uma *interface web* para que qualquer utilizador da internet possa aceder à informação que disponibilizam sob a forma de objectos digitais.

Um objecto digital, como já foi dito, é algo que pode ser digitalizado. No contexto deste documento, um trabalho prático, será um objecto digital.

Um trabalho prático realizado numa universidade ou numa outra qualquer instituição de ensino, por um grupo de alunos, pode ser armazenado num repositório de trabalhos práticos, o que leva a uma maior facilidade no desenrolar de todo um processo de entrega de trabalhos e correcção dos mesmos pelo professor. Contudo, no contexto de um arquivo digital, o que interessa é que esses trabalhos práticos fiquem acessíveis ao público com a devida informação associada, e catalogados de forma coerente usando-se ontologias ou taxonomias apropriadas.

A estrutura deste relatório de desenvolvimento será a seguinte: No Capítulo 2 vai ser introduzido teoricamente o tema dos arquivos digitais e a norma OAIP, e será, com base nesta introdução, feita uma análise concisa do problema a resolver, o qual concerne na criação de um arquivo digital de trabalhos práticos. Por fim serão analisados todos os passos a dar no sentido de concluir a tarefa proposta, documentando-os com um diagrama de Gantt.

No Capítulo 3 será apresentado um diagrama de Use Cases com as funcionalidades mais importantes que o arquivo digital de trabalhos práticos deverá fornecer aos seus diversos utilizadores.

No Capítulo 4 será mostrada a estrutura estática do arquivo digital, assim como apresentadas algumas restrições a essa mesma estrutura.

No Capítulo 5 vão ser descritos todos os passos importantes que foram levados a cabo para a construção do arquivo digital. Começa-se por desenhar a base de dados de suporte, explicando-se cada uma das entidades principais. De seguida passa-se à apresentação sucinta das tecnologias utilizadas, e por fim, são apresentados os vários passos de implementação, sendo para cada um mostrada a análise prévia, as discussões e decisões de implementação, e por fim os resultados obtidos, documentados com algumas imagens.

Por fim, no Capítulo 6 serão retiradas as conclusões sobre aquilo que foi conseguido, mostrando-se ainda o trabalho que se poderá vir a fazer no futuro.

Capítulo 2

Análise

Neste capítulo vai ser introduzido, numa perspectiva a muito alto nível, o modelo OAIS o qual será a base para a construção do arquivo digital de trabalhos práticos. Será ainda feita uma análise, descrição e contextualização do problema. Tendo em conta esta análise feita, serão extrapoladas tarefas de desenvolvimento e documentação da peça de software a construir, as quais serão mantidas num diagrama de Gantt, para organização pessoal, temporal e de recursos.

2.1 Introdução ao OAIS

O OAIS – *Open Archive Information System* é visto como um modelo de referência para a construção de arquivos digitais. Ou seja, é a framework de desenvolvimento para este fim.

Com esta framework, o que se pretende é que todos os arquivos a serem construídos sejam de certo modo iguais, no sentido de terem um funcionamento idêntico e seguirem exactamente as mesmas regras de ingestão e arquivo dos Objectos digitais que lhes são fornecidos. Assim, com todas as regras que propõe, mas não obriga (daí ser *aberto – open*), este modelo de referência é uma norma ISO.

O seu ambiente é simples, e o diagrama da Figura 2.1 demonstra na sua essência todo o processo e fases de vida dos objectos digitais num arquivo que segue esta norma. Neste diagrama podem então ver-se todos os actores participantes no processo, e as três estruturas de informação que são geradas para manutenção do bom funcionamento do arquivo.

- O Produtor é o conjunto de pessoas que fornecem material ao arquivo para este o preservar;
- O Administrador é o conjunto de recursos humanos que se dedicam à manutenção do arquivo e gerem as suas políticas.

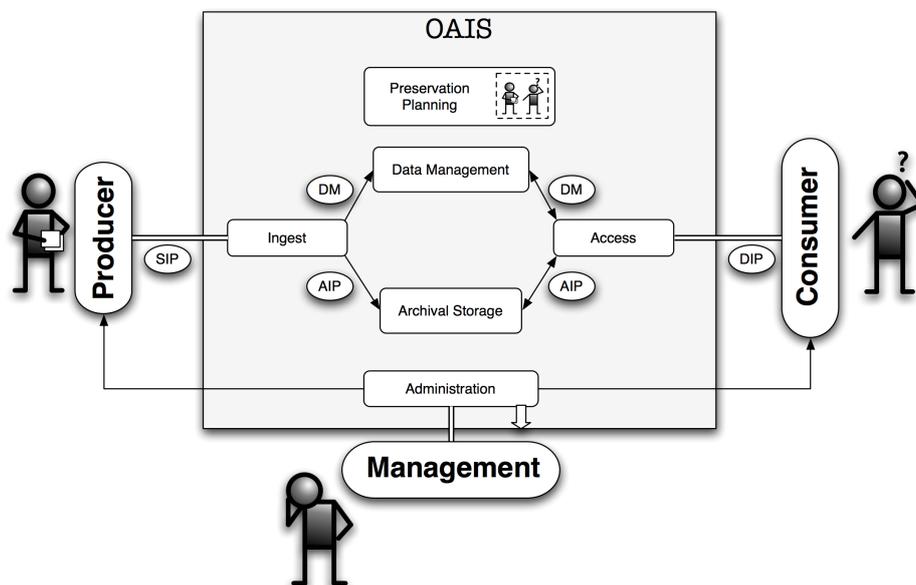


Figura 2.1: Modelo de referência OAIS

- O Consumidor é o conjunto de pessoas que navega pelo arquivo, efectuando pesquisas e descarregando informação sob a forma de pacotes.

Como já foi referido, nas três grandes fases do processo de um arquivo digital OAIS são geradas estruturas de informação, mais correctamente denominados por pacotes de informação. Estes pacotes de informação são vistos como uma capsula que engloba dois outros tipos de estruturas [CCS01]: uma é o conteúdo do pacote, a outra é a preservação de dados, normalmente é manifesto em XML que segue normas pré-definidas. Porém esta é como que uma classe abstracta, pois as suas sub-classes concretas são aquelas que realmente são usadas e criadas em cada fase.

A seguir define-se mais em concreto, mas de forma breve, o que é cada um destes pacotes.

Submission Information Package (SIP) é um pacote de informação fornecido pelo produtor ao arquivo digital. O seu conteúdo é, nada mais nada menos, que o conteúdo da informação que o utilizador pretende que o repositório preserve, acompanhado por um ficheiro (ou conjunto de ficheiros) que descrevem e preservam a meta-informação que está associada ao conteúdo, físico do pacote em si. Este pacote é submetido pelo produtor, como já foi dito, e ingerido pelo arquivo OAIS, gerando um novo pacote de informação: o AIP.

Archival Information Packages (AIP) é um ou mais pacotes de informação que são guardados para preservação segundo regras ditadas pela framework OAIS. Estes pacotes podem ser vistos como meta-informação do conteúdo do SIP e guardados em bases de dados relacionais, juntamente com uma referência para a localização exacta do conteúdo físico do SIP.

Dissemination Information Packages (DIP) é um ou mais pacotes de informação que derivam directamente dos AIP. Estes pacotes contêm toda ou apenas uma parte do pacote AIP, isto é, do conteúdo físico da informação de um AIP, juntamente com outros documentos necessário como é o caso do manifesto que mostra toda a meta-informação (de forma bem estruturada) do pacote DIP que o consumidor vai descarregar.

De um modo geral, a framework OAIS foi introduzida nesta pequena secção de introdução à mesma.

De seguida proceder-se-á à análise do problema tendo como base esta introdução, e ao escalonamento de tarefas, horários e metodologias de desenvolvimento.

2.2 Análise do Problema

O objectivo principal do projecto prático, como já tem vindo a ser anunciado em secções anteriores deste relatório, é o desenvolvimento de um arquivo digital para trabalhos práticos de alunos.

Pretende-se guardar não só trabalhos práticos, isto é, todos os documentos do trabalho na sua forma física, mas também a informação associada ao trabalho prático e ainda aquela que está associada a cada um dos seus documentos constituintes. Para isto será seguido o modelo internacional para a criação de repositórios digitais OAIS, que como se viu na secção anterior, este modelo comporta três grandes fases de funcionamento: a primeira refere-se à fase de submissão de um dado pacote de informação denominado SIP, a segunda é a fase em que o sistema de arquivo extrai a informação submetida e a incorpora numa base de dados, ou seja é criado um novo pacote de informação denominado por AIP, e a terceira e última grande fase, é a disseminação da informação submetida, que é denominada por DIP

Deste modo, o sistema a desenvolver deve seguir os três grandes passos referentes às fases descritas acima. No primeiro passo, um utilizador deve poder submeter toda a informação física e lógica de um trabalho prático, isto é, ficheiros e informação associada referentes ao relatório, à apresentação ou ao código produzido. Junto aos ficheiros do trabalho prático, o utilizador

deve submeter um manifesto em formato XML seguindo um modelo instaurado como norma, como é o caso do METS, e que será usado pelo sistema, na segunda fase, para catalogar toda a informação do trabalho prático.

Na última fase, por requisição do utilizador (não necessariamente aquele que submeteu o trabalho prático) o sistema terá que criar pocotes de difusão da informação contida em base de dados, sendo portanto possível mostrar toda a informação contida no arquivo digital.

Um arquivo digital não é um simples programa que corre numa única máquina, pois assim não teria utilidade nenhuma. É sim uma aplicação cliente-servidor para a web sendo portanto necessário criar ainda uma interface que suporte a submissão da informação assim como a disseminação da mesma.

De uma forma geral, o problema a resolver foi descrito. Não se entrou em detalhes sobre a arquitectura do sistema, nem sobre as tecnologias a utilizar, pois não é, de todo, o mais importante, nesta fase, para o leitor perceber aquilo que se pretende ter no final.

2.3 Análise Temporal de Desenvolvimento

Dada a descrição do problema acima documentada, é agora possível traçar um mapa ou uma rota para organização pessoal e temporal sobre como e quando resolver os vários passos que a implementação desta peça de software requer.

Dado o contexto em que o projecto se insere, é bastante proveitoso seguir uma metodologia ágil, a SCRUM [Agi], como metodologia de desenvolvimento do sistema de arquivo de trabalhos práticos. Muitas das fases de desenvolvimento deverão ser apresentadas em intervalos pequenos de tempo, centrando-se esforços em pequenas partes de todo o projecto para cada uma dessas apresentações. Decidiu-se no entanto que para cada uma destas pequenas fases se iria seguir uma metodologia de desenvolvimento muito aproximada àquilo que foi proposta pela IBM no seu famoso processo de desenvolvimento de software RUP (Rational Unified Process) [JBR99]

Assim, pensou-se na distribuição temporal mostrada em 2.2 das várias tarefas de desenvolvimento que foram extrapoladas partindo da análise previamente feita.

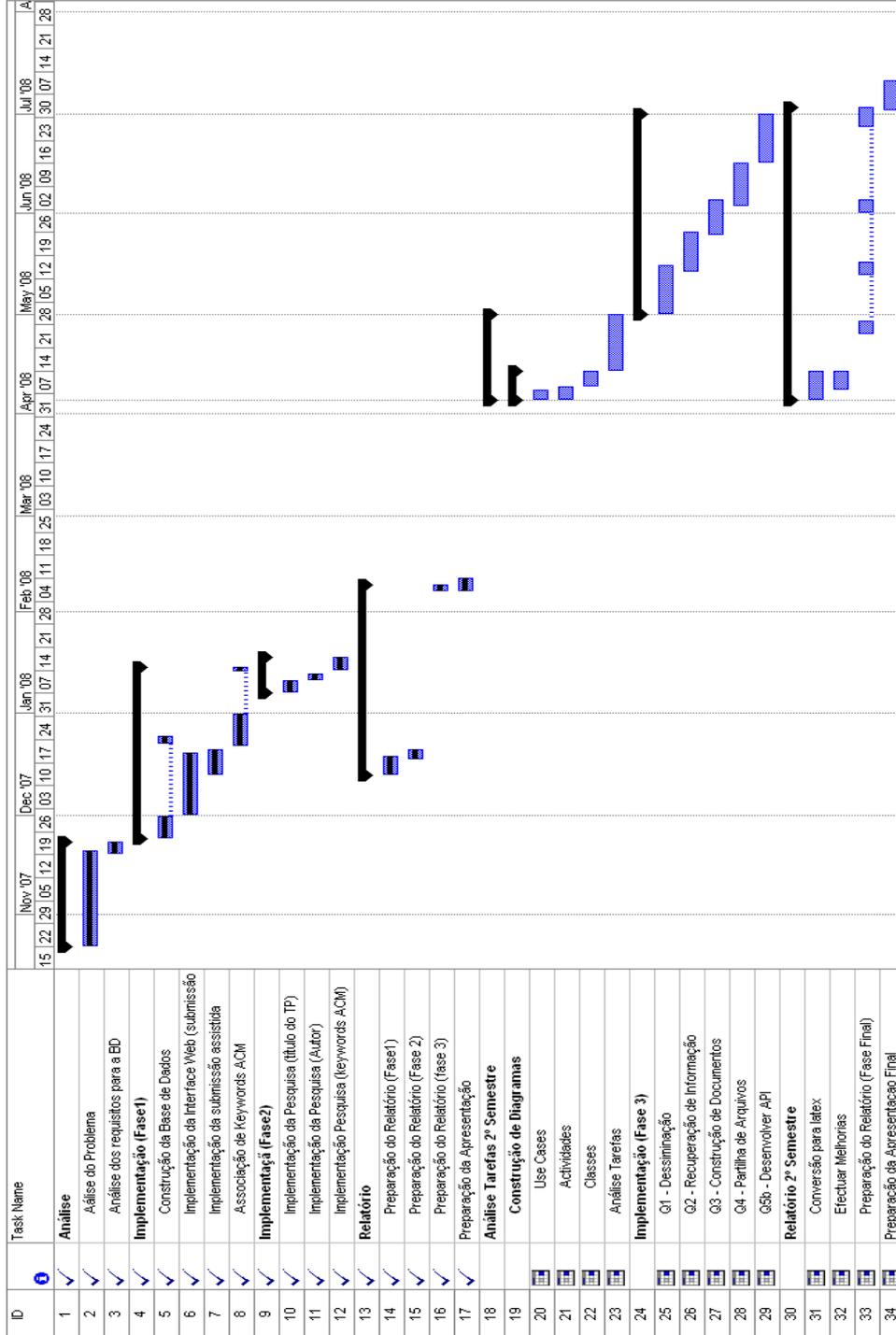


Figura 2.2: Diagrama de distribuição temporal de tarefas

Capítulo 3

Captura de Requisitos

Tendo-se descrito no Capítulo 2 o problema a resolver, no presente serão delineados os requisitos e as principais funcionalidades do produto final que se quer construir.

Como já foi dito, vai ser seguido o modelo de referência OAIS para a criação d repositório digital. Ora isto implica uma certa interacção com os utilizadores do sistema, tendo sido definida a necessidade de se ter uma interface web que funcione como camada de acesso ao repositório. Esta camada de acesso deve rever em si todas as necessidades que cada utilizador terá.

Assim, antes de se proceder a uma análise mais aprofundada das funcionalidades do arquivo digital, vai-se identificar e descrever todos os actores (conjuntos de utilizadores) que interagem com o sistema.

3.1 Actores

Identificaram-se quatro actores principais que irão actuar sobre o sistema: o Aluno, o Professor, o Administrador e o Visitante. Cada um deles tem necessidade que o sistema lhe forneça algumas funcionalidades. De seguida indentificar-se-ão todos estes actores, explicando o papel que desempenham.

O Aluno

O actor Aluno desempenha um papel importantíssimo no sistema. Fazendo o paralelo com o modelo apresentado na Figura 2.1, o actor identificado como Aluno pode ser mapeado no actor Produtor do modelo em questão. Portanto é o Aluno quem faz a submissão dos trabalhos práticos, com toda a informação associada aos pacote num manifesto. Para além de os submeter, o Aluno pode alterar as informações referentes aos seus trabalhos práticos (que são dele apenas se ele for o seu autor, ou co-autor).

Apesar de poder submeter trabalhos práticos, um Aluno não poderá, de maneira alguma, remover este tipo de objectos digitais.

O Professor

O actor Professor não tem um papel muito importante na interacção com o sistema. Simplesmente pode aceder aos seus trabalhos práticos (que são dele apenas se for dado como orientador) e alterar os seus conteúdos, tal como se de um aluno de tratasse.

O Administrador

O actor Administrador tem como papel principal a administração do sistema. É ele o único capaz de aceder aos conteúdos a base de dados directamente, adicionando novas propriedades e novas informações pré-fixas à base de dados. Outra acção que ele pode desempenhar é a aprovação ou rejeição de registos quando o registo é um professor.

O Visitante

Este actor, Visitante, é tido como o grupo de pessoas que apenas acede ao sistema para poder elaborar pesquisas ou fazer descargas de pacotes de trabalhos práticos. Este actor, tal como o Aluno, pode ser mapeado num dos papéis apresentados no modelo OAIS (Figura 2.1).

3.2 Funcionalidades

Da descrição feita acima, é possível observar muitos dos requisitos funcionais e mais importantes para sistema. Segue-se então uma listagem com alguns destes requisitos. Assim o sistema deve:

- Possibilitar ao Aluno a submissão de trabalhos práticos;
- Possibilitar ao Aluno a alteração de informação dos seu trabalhos práticos;
- Possibilitar ao Professor a alteração de informação dos trabalhos práticos que orientou;
- Permitir que o Administrador possa aprovar ou rejeitar registos de professores;
- Permitir que o Administrador possa fazer manutenção de dados da base de dados;

- Permitir que se façam pesquisas sobre o arquivo por intermédio de palavras chave, como título do trabalho, ou então por palavras da taxonomia ACM;
- Permitir que se possam fazer registos de utilizadores para as categorias de professores ou alunos.
- Fornecer uma área pessoal para visualização de informação sua e dos seus trabalhos;
- Fornecer ao utilizador meios para alterar as suas definições pessoais.

Estas funcionalidades são mostradas na Figura 3.1 onde é mostrado o diagrama de Use Cases referente ao sistema ADTPis (Arquivo Digital de Trabalhos Práticos Information System), que a partir daqui será a denominação para o arquivo digital que se pretende construir.

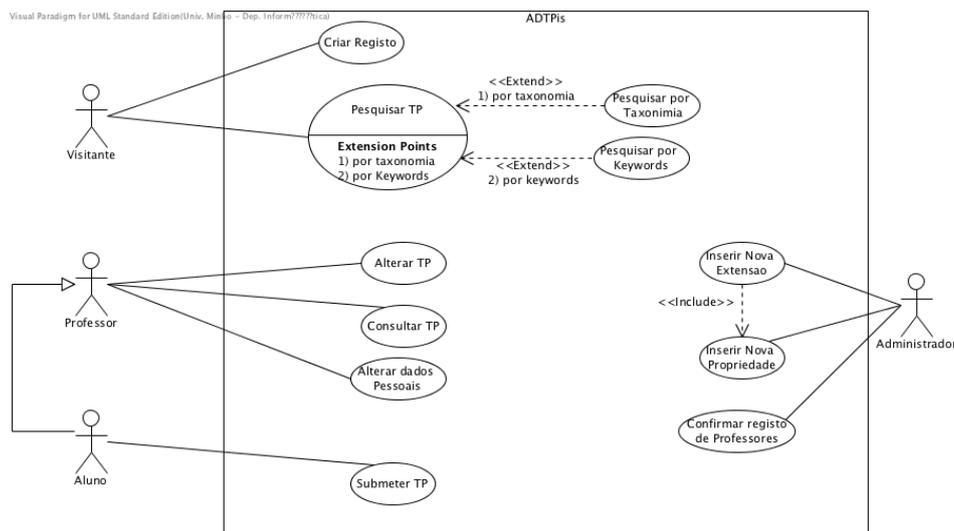


Figura 3.1: Diagrama de Use Cases para o sistema ADTPis

Estes diagramas de Use Cases, por si só, não dizem tudo referente às funcionalidades do sistema. Para isso foram descritas todos os use cases deste sistema, e estão presentes no Apêndice A.

Obtidos os requisitos, tem-se uma ideia ainda mais clara daquilo que se quer desenvolver, podendo assim poder-se passar à concepção do sistema.

Capítulo 4

Concepção do Arquivo

Neste capítulo vai ser proposta a estrutura para o sistema ADTPis.

A proposta da estrutura do sistema baseia-se essencialmente num diagrama de classes, pois demonstra de forma quase unívoca a representação e disposição da informação no sistema. Para além do diagrama de classes, vai ser apresentada também uma gramática que tenta, de uma forma não mais além que o diagrama de classes, descrever essa estrutura.

4.1 Estrutura Estática

Antes de se seguir para a implementação de uma forma concisa da representação estruturada da informação no sistema, vai-se apresentar, como se pode ver na Figura 4.1, um diagrama de classes que descreve o relacionamento entre classes de informação do sistema.

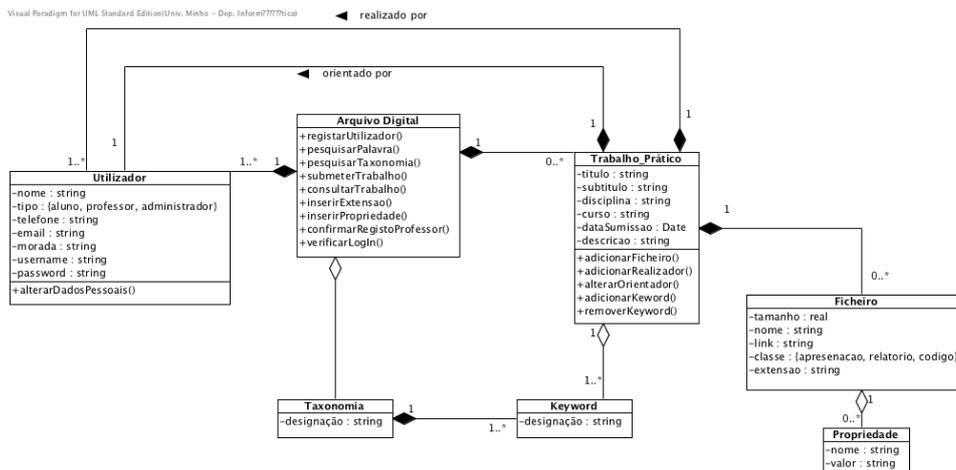


Figura 4.1: Diagrama de Classes para o sistema ADTPis

Dado o contexto do projecto, faz todo o sentido apresentar também esta mesma estrutura por meios de uma gramática. A Figura 4.2 mostra essa gramática. À primeira vista pode parecer haver algumas incoerências entre a gramática e o diagrama, mas não existem. São apenas duas maneiras distintas para apresentar a mesma estrutura. Numa (Figura 4.1) é possível ver melhor como são interligadas as várias entidades, noutra (Figura 4.2) é mais perceptível a constituição e a definição de cada uma das entidades gerais do sistema.

```

Adtp          : Od+
Od            : MetaInformacao Tp
MetaInformacao : titulo subtitulo? Autor+ Entregador Orientador
               Contexto palavraTaxonomia+ resumo dataSubmissao
Entregador    : Autor
Autor         : id nome e-mail
Orientador    : id nome e-mail
Contexto      : disciplina curso
Tp            : Ficheiro+
Ficheiro      : Classe Tipo Propriedade+ link+
Classe        : relatorio
               | apresentacao
               | codigo
Propiedade    : nome descricao Tipo+
Tipo          : extensao

```

Figura 4.2: Gramática que define a estrutura do sistema ADTPis

Repare-se que as entidades são iniciadas por letras maiúsculas, e os atributos das entidades são iniciados por letras minúsculas. Note-se ainda que os atributos das entidades podem não estar representados na sua totalidade, pois não são de todo importantes para a percepção da estrutura do sistema a este nível.

A tradução da gramática para linguagem natural é simples, ficando explicada da seguinte forma:

O sistema Adtp é constituído por um ou mais Od (que denota Objecto Digital). Um Od é constituído por Meta-Informação e um Tp (que denota trabalho prático físico). A Meta-Informação de um Od tem alguns atributos, os quais não interferem na estrutura, e é constituído por um Autor ou mais, um Entregador (que corresponde ao Autor que faz a submissão do Od), um Orientador e por um Contexto. Um Autor e um Orientador são praticamente iguais, pois podem ter a mesma informação associada, mas o Autor é um aluno, e um Orientador é um professor.

Um Contexto tem informação sobre a área e datas onde o Od é realizado. Um Tp é um conjunto não vazio de Ficheiros. Um Ficheiro tem uma Classe, um Tipo, um conjunto de Propriedades associados e uma localização no repositório. A Classe de um ficheiro pode ser uma de três possíveis: relatório, apresentação, código. Um tipo é basicamente uma extensão, e uma Propriedade é constituída por um nome, uma descrição e uma lista de Tipos aos quais podem estar associados.

4.2 Restrições à Estrutura e à Funcionalidade

Olhando para a gramática que representa a estrutura do Sistema, detectam-se facilmente algumas restrições importantes a efectuar. Estas restrições podem não só dizer respeito à estrutura física em si, mas também à estruturação lógica da informação:

- **pertence(MetaInformação.entregador, MetaInformação.autores)**
- Esta restrição significa que quem submete o trabalho deve ser um dos autores do trabalho, não podendo ser uma pessoa externa a fazer a entrega.
- **compatíveis(Ficheiro.tipo, Propriedade.tipos)** - Esta restrição significa que as propriedades do ficheiro devem estar coerentes com o tipo do ficheiro. Não podendo haver, portanto, nenhum ficheiro com um tipo x, e com propriedades referentes a um tipo y.
- Um trabalho prático deve ter sempre pelo menos um ficheiro associado.

Em relação às restrições funcionais, não se encontram muitas que ainda não foram ditas, pelo que apenas se destaca o facto de apenas um utilizador registado no sistema como sendo aluno poder submeter trabalhos.

Um outro aspecto importante a destacar sobre a restrição da funcionalidade é que nenhum trabalho prático¹ submetido no arquivo digital poderá ser removido, nem mesmo pelo administrador.

¹Leia-se trabalho pratico como o conceito geral apresentado como sendo um objecto digital, e não o conceito que foi apresentado na definição da gramática

Capítulo 5

Implementação

Neste grande capítulo serão apresentados todos os passos que foram dados para a implementação das diversas fases emergentes no Capítulo 2 e apresentadas no diagrama da figura 2.2.

As primeiras secções relacionam-se com escolhas efectuadas em relação a tecnologias utilizadas para o desenvolvimento e com o desenho e construção da base de dados. Nas restantes secções deste capítulo são realmente apresentados os passos de implementação, sendo cada um estruturado da seguinte forma: primeiro faz-se uma breve análise àquilo que está inerente à tarefa, depois são apresentadas algumas discussões sobre a implementação assim como as decisões que daí surgem. No final mostram-se resultados daquilo que foi feito, documentando-se, para além do texto, com imagens.

Os resultados finais apresentados por todo este Capítulo podem ser visualizados e utilizados em http://epl.di.uminho.pt/el07_g2.

5.1 Tecnologias Utilizadas

Para construção do sistema ADTPis que será explicada um pouco por cada parte do presente capítulo, foram utilizadas tecnologias variadas. Desde a construção da base de dados relacional até à mais ínfima função do site, a escolha e utilização dessas tecnologias têm o seu fundamento.

Foi utilizado o SGBD (Sistema Gestor de Bases de Dados) MySQL, como tecnologia para a construção da base de dados. A escolha baseia-se no facto de o MySQL ser gratuito e as ferramentas de trabalho serem bastante mais leves quando comparados com SQLServer ou Oracle.

No que diz respeito ao desenvolvimento do site em si, escolheu-se como tecnologia base o HTML por razões óbvias, aliado às folhas de estilo CSS, com as quais se pode dar, facilmente, um ar mais profissional ao site. Por outro lado, para se poder dar alguma funcionalidade ao site, foram ainda

utilizadas as tecnologias PHP e JavaScript.

Escolheu-se o PHP como tecnologia *server-side* porque é simples de utilizar, existe muita informação sobre a mesma, não correndo o risco de se ficar parado sem se saber como fazer qualquer coisa. Quanto à escolha do JavaScript como tecnologia *client-side*, deve-se ao facto de em certas ocasiões, ser necessário ter um conteúdo dinâmico no site, do lado do cliente, que não se poderia efectuar apenas com a utilização de PHP. A sua comum utilização faz com que haja inúmeras fontes de informação, espalhadas por todo o lado, em livros, internet, logo foi também um ponto importante para a sua escolha.

Utilizou-se também XML para se poder guardar fora da Base de Dados, a informação da árvore da taxonomia ACM. Sendo utilizado XML DOM (incorporado em javascript) para se proceder à transformação do ficheiro XML em informação funcional e capaz de ser utilizada na aplicação em desenvolvimento.

5.2 Base de Dados e Entidades

O desenho da base de dados relacional foi completamente baseada na ideia original do diagrama de classes assim como na gramática apresentada no Capítulo 4, Figuras 4.1 e 4.2, respectivamente.

Deste modo definiram-se as seguintes entidades que foram mapeadas em tabelas na base de dados:

TP Esta entidade é a mais importante entre todas as entidades, pois é o sistema é construído à volta dela. Representa as principais características do OD em meta-informação.

Pessoa Representa todas as pessoas que tenham qualquer interacção com o sistema. Tem como atributos as características mais comuns e inerentes a uma pessoa humana, isto é, os seus dados pessoais.

Users Contém a informação sobre os utilizadores. Complementa a informação que existe na entidade Pessoa. A informação desta entidade diz respeito à informação de interacção com o sistema.

TipoUser Contém os tipos de utilizador que são possíveis existir na interacção com o sistema.

Ficheiro É a entidade responsável por todos os ficheiros pertencentes a um trabalho prático. Contém toda a informação que um ficheiro possa ter, nomeadamente a sua meta-informação bem como a sua localização no sistema de ficheiros.

5.3 Criação da Interface Base

Em capítulos anteriores tem-se vindo a descrever e a analisar o sistema que se conceberá no presente capítulo. Essa análise feita foi fundamental para se terem organizadas as ideias daquilo que se quer implementar.

Nesta altura é tudo colocado nos sítios certos, e a poeira levantada na análise começará agora a assentar.

Foi definido que o arquivo digital teria uma interface web como meio de comunicação entre o utilizador e o repositório, portanto o primeiro passo a dar na construção do arquivo e suas funcionalidades específicas de arquivo digital que segue o modelo OAIS é a implementação de uma interface web.

No Capítulo 3 foi definido que o sistema deveria fornecer ao utilizador as seguintes funcionalidades:

- registo de novo utilizador
- área pessoal;
- alteração dos dados pessoais;
- submissão de trabalhos práticos;
- alteração de trabalhos práticos;
- pesquisa de trabalhos práticos;
- etc...

5.3.1 Decisões de Implementação

Dados estes requisitos, a base de dados já construída e uma escolha pré-definida de tecnologias de implementação de interfaces web, começou-se então a pensar em como interligar as coisas e a criar-se um web site seguindo-se modelos generalizadamente utilizados pela comunidade de desenvolvimento para a web.

Assim a estrutura a seguir foi a que se descreve de seguida.

A página inicial, que mostra a *cara* de todo o sistema, deve mostrar desde logo um ar sério e simples, para cativar todos os utilizadores. Para além do aspecto deve mostrar todas as funcionalidades possíveis de se fazerem no momento de entrada. Essas funcionalidades, seguindo-se modelos de referência de desenho web, devem estar presentes do lado direito numa coluna. Para além das funcionalidades, deve-se ter em conta alguma navegação sobre informação relativa ao site que se está a construir.

Tendo-se idealizada a interface base, é necessário dar semântica àquilo que é apresentado ao utilizador. Para isso as atenções serão centradas nas

funcionalidades dadas ao utilizador para cada um dos menus em que se encontra, segundo-se a implementação de uma navegação contextual.

Na página inicial um utilizador terá, como funcionalidades principais a de efectuar um registo, a de efectuar login ou a de efectuar pesquisas sobre o repositório.

A funcionalidade de criação de registos e de login não necessitam de grande debruço, pelo que se seguirão novamente modelos bastante referenciados para implementação destas funcionalidades. Assim pode-se avançar para a área pessoal, a qual é acedida depois de se efectuar o login. Aqui, as funcionalidades a apresentar ao utilizador já devem ser diferentes, pois o contexto inicial foi alterado.

A área pessoal deverá depender de uma premissa, isto é, do tipo de utilizador. No caso de ser aluno ou professor deve ser apresentada para além da informação, uma listagem de trabalhos práticos submetidos ao ADTPis e dos quais sejam autores ou orientadores, respectivamente. Por defeito deverá ser possível alterar os dados do utilizador e apresentar esta informação. Mas no caso de ser aluno deve ainda ser oferecida a funcionalidade de submeter trabalhos práticos, etc.

Decidiu-se que estes contextos seriam identificados usando-se variáveis de sessão, e os casos mais gerais o contexto de navegação, dependeria de um *workflow* de navegação pré-definido.

5.3.2 Resultados Obtidos

Para apresentar os resultados, nada melhor do que um pequeno conjunto de imagens. Assim, na Figura 5.2 mostra-se a página inicial do repositório com o desenho decidido na secção anterior. Na Figura Fareapessoal é mostrado o ambiente na área pessoal de um utilizador que comporta desde logo a funcionalidade de submeter trabalhos práticos, a qual será desenvolvida na secção 5.4. Apesar de apenas ser apresentada a aí a submissão de trabalhos práticos, fica desde já mostrada a forma como se pode alterar a informação de um trabalho prático.

Essa funcionalidade é acedida a partir de um click no nome do trabalho prático. Ao utilizador é então apresentada essa funcionalidade como se pode ver na Figura 5.4.

5.4 Submissão e Arquivo de Trabalhos Práticos

Na secção anterior foi desenhada a base de todo o arquivo digital, tendo assim neste momento tudo aquilo que é necessário para avançar com a primeira grande fase do ciclo de vida de um arquivo digital quando seguindo o modelo de referência da OAIS, devidamente documentado no Capítulo- 2.

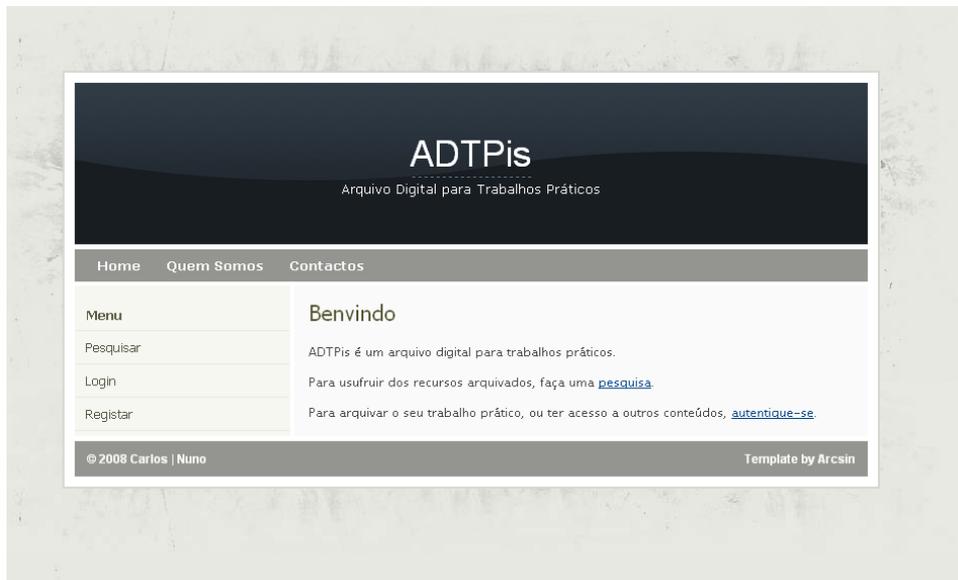


Figura 5.2: Página principal do sistema ADTPis



Figura 5.3: Pagina de área pessoal de um utilizador do sistema ADTPis

Menu

Sair

Trabalhos Práticos

Submeter

Dados Pessoais

Alterar

Trabalho prático

Título:

Subtítulo:

Descrição:

Realizado Por

- tuna2 - Nuno
- pnpo - Nuno
- byak - Carlos Pereira

Novo Autor:

Orientado Por

- prh - Pedro Henriques

Novo Orientador:

Taxonomia
Brevemente

Ficheiros de Apresentação Ficheiros de Relatório Ficheiros de Código

- [index \(html\)](#)

Figura 5.4: Página para alteração da informação de um trabalho

A primeira fase de vida de um arquivo digital é a submissão de um objecto digital no seu repositório. Como foi analisado em capítulos anteriores, a submissão de um objecto digital, neste caso um trabalho prático, deve ser feita num único passo: o utilizador submete um SIP ao arquivo. Onde um SIP é um ficheiro compactado constituído por duas partes essenciais – uma árvore de directorias que representa a organização física do trabalho, e um manifesto em XML seguindo a norma METS de documentação de meta-informação para objectos digitais de repositórios electrónicos, o qual para além da meta-informação acerca do OD, contém também a reflexão da estrutura de directorias do mesmo ficheiro compactado.

Quanto à submissão de ODs em arquivos digitais, não há muito mais a relatar, porque o modelo OAIS é bastante claro para esta primeira fase.

A segunda fase de vida de um arquivo digital é o armazenamento de informação relativa ao pacote submetido. Esta informação diz respeito às duas estruturas comentadas à pouco, ou seja, à meta-informação no ficheiro XML e à representação física do OD.

Assim, depois de ser submetido um pacote, deve ser usado o ficheiro de meta-informação, denominado de *manifesto*, para se proceder ao armazenamento dos dados do OD em base de dados compatível. O processamento do ficheiro XML é a base de toda esta fase, que conclui no final com o pacote de representação intermédia do OD e já introduzido também: o AIP. Para além de se guardar esta meta-informação na base de dados é necessário manter a localização física do OD, para que o repositório não fique incoerente com aquilo que está em base de dados, e essencialmente, para que o arquivo possa ter uma terceira fase de vida.

Contudo, repare-se que da análise de requisitos e de funcionamento patente no Capítulo 3 e consequentemente no Apêndice A, alguns passos desta fase do OAIS foram bastante alterados. Ficando a primeira e segunda fases do OAIS desfiguradas e misturadas.

5.4.1 Discussão de Implementação e Decisões

Analisadas que estão a primeira e segunda fase de vida de um arquivo digital, passa-se agora à apresentação de alguns problemas que foram surgindo à medida que se foi pensando numa solução.

Dado que o seguimento do modelo OAIS não foi continuado para estas duas fases como deveria ter sido, então foi necessário pensar numa nova framework para estes dois passos. A discussão levou a uma simples maneira de fazer as coisas: em vez de se ter uma submissão única de um ficheiro, passaria a ter-se uma submissão assistida de todo o OD: meta-informação do Trabalho Prático e submissão dos seus ficheiros um a um.

Para esta solução foi necessário pensar em outras duas coisas bastante importantes: como ir guardando a meta-informação do trabalho prático

para garantir consistência na base de dados, e como e onde armazenar o trabalho em si.

Para a primeira tarefa chegou-se a algumas soluções:

Utilização de objectos em variáveis de sessão Esta solução comporta uma limpeza completa na arquitectura da solução, pois muito do código que se iria escrever ficaria modulado e acessível a partir de um único ponto, tendo-se a certeza que não se iria ter código específico de cada classe *hard-coded* na interface da solução. A criação destes objectos levaria à criação de outro tipo de estruturas de processamento que poderiam facilitar os acessos à base de dados para a inserção de um trabalho prático, podendo até fazer-se tal inserção em apenas uma transacção. A utilização de variáveis de sessão comportam um *design pattern* comum no desenvolvimento para a web, contudo, apesar do seu uso generalizado, ter muita informação a passar entre o cliente e o servidor, não é benéfico para a solução.

Utilização de ficheiros Uma outra solução proposta foi a utilização de ficheiros de texto SQL para se ir guardando as instruções necessárias para que no fim fosse calculada a inserção total na base de dados. Mas também com esta solução se iria ter algum desgaste na informação que ia transitar de cliente para servidor, e mesmo assim, não ajudava na integridade das referências que iam posteriormente ficar na base de dados.

Guardar directamente na Base de Dados Esta solução é simples, eficaz e bastante intuitiva dada a organização da estrutura do OD trabalho prático. Por cada fase de inserção de meta-informação acerca do trabalho, seria feito o seu armazenamento em base de dados. Tendo-se a necessidade de apenas passar uma ou duas variáveis de sessão com o identificador do trabalho prático. Com esta solução apenas surge um problema: a qualquer altura o utilizador pode cancelar a submissão do trabalho prático acabando por ficar incoerente a informação na base de dados. Porém aliando isto à funcionalidade de alterar trabalhos práticos, então a solução óbvia é deixar que o trabalho fique invisível para o exterior do arquivo até que o utilizador complete a submissão do mesmo.

Dadas estas propostas, não foi difícil escolher aquela que iria vigorar. A terceira hipótese parece ser a mais sensata, apesar de ao contrário da primeira comportar uma menor limpeza no código da solução. Assim o problema do armazenamento da meta-informação ficou resolvido, faltando apenas resolver a questão do armazenamento físico do OD.

Apenas uma proposta foi pensada, ficando desde logo assente: os ficheiros e estrutura do trabalho prático seria guardada no sistema de ficheiros do

servidor, mantidos numa directoria única por trabalho. Para se ter uma directoria única, utilizar-se-ia a ideia que as base de dados têm para manter, por exemplo, um trabalho prático único, isto é, utilizar um identificador único por trabalho. Ora dado que a necessidade de armazenar coisas no sistema de ficheiros apenas surge na fase de submissão de ficheiros, então a pasta pode ser criada logo após à inserção da meta-informação na base de dados, utilizando-se o id do trabalho prático para o id da pasta a criar.

5.4.2 Resultados Obtidos

Após implementação das ideias discutidas anteriormente, chega-se ao fim destas fases do ciclo de vida do arquivo com uma solução bastante intuitiva, seguindo-se um *workflow* que é apresentado ao utilizador no topo dos formulários da submissão, como se pode ver na Figura 5.5, que representa a página de submissão da meta-informação relativa ao trabalho prático em geral.

Todas as outras fases de submissão, nomeadamente a inserção dos autores do trabalho prático, do orientador, das palavras da taxonomia, da submissão de ficheiros e suas propriedades seguem a mesma filosofia de desenvolvimento da interface, sendo mais ou menos complexas, utilizando mais ou menos tecnologias, como é o caso da página da escolha das referências taxonómicas, que utiliza javascript e XML DOM para processar a árvore da taxonomia ACM patente no servidor num ficheiro XML.

Note-se porém que a submissão de ficheiros, nesta fase, e como se pode ver na Figura 5.6 são feitos sem qualquer informação dada ao utilizador sobre para onde vão ficar tais ficheiros, guardando-se todos os ficheiros na raiz da pasta criada. Na Secção 5.6 será dada uma maior ênfase a esta parte da manutenção de paths relativas e de estruturas de directorias para se ter uma maior organização do conteúdo físico de um trabalho prático.

5.5 Pesquisa - Base

A pesquisa de informação pode ser vista como uma interface para a terceira e última fase do ciclo de vida de um arquivo digital, a disseminação, cujo tópico de discussão e implementação será desenvolvido na Secção ??.

Conforme foi analisado na capítulo referente à captura de requisitos, deve ser possível oferecer ao utilizador (visitante incluído) uma zona de pesquisa. A pesquisa deverá poder ser efectuada mediante palavras chave como o título de um trabalho, o autor, ou mesmo o orientador, ou então mediante a construção de um caminho na árvore de taxonomia ACM.

O resultado da pesquisa deverá surgir sob a forma de listagens de trabalhos práticos, pois são a unidade básica do arquivo, e sobre os quais existe

ADTPis

Arquivo Digital para Trabalhos Práticos

[Home](#) [Quem Somos](#) [Contactos](#)

TP
Dados -> Autores -> Orientador -> Ficheiro Upload -> Dados -> Propriedades

Dados

Responsável:

Título:

Subtítulo:

Disciplina:

Curso:

Descrição:

Data Submissão: - -

Keywords ACM: Brevemente

© 2008 Carlos | Nuno Template by Arcsin

Figura 5.5: Página principal da fase de submissão e armazenamento dos dados do OD

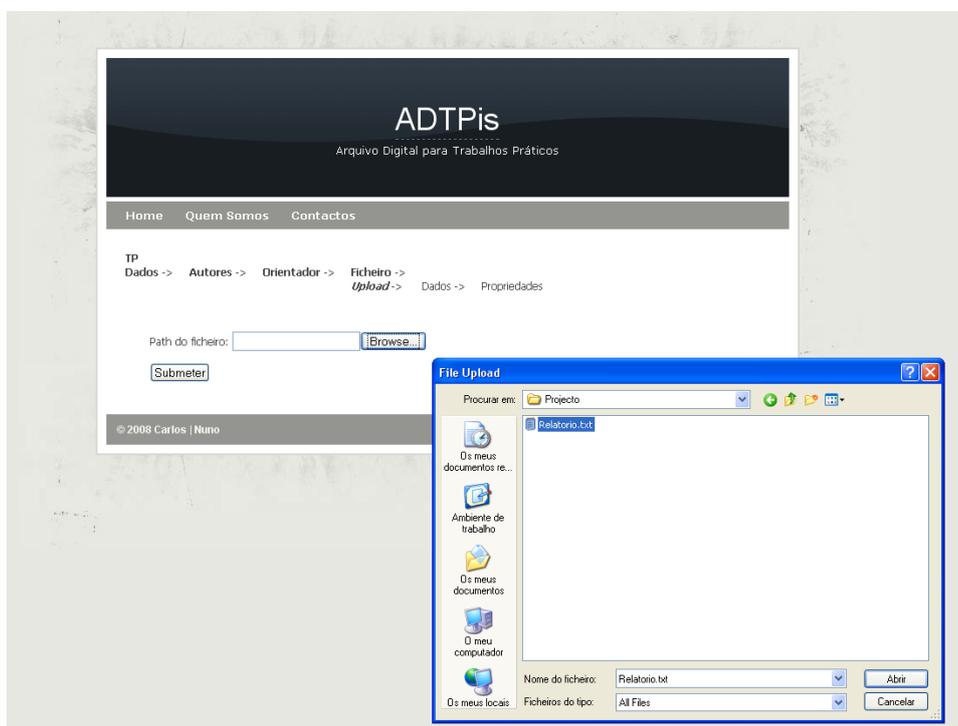


Figura 5.6: Página de submissão de ficheiros para compor o trabalho prático

toda a informação que um utilizador pretende aceder. É ainda a partir da pesquisa que o utilizador terá acesso aos pacotes de difusão do trabalho, conforme será explicado mais à frente no presente capítulo.

5.5.1 Discussão de Implementação e Decisões

Sempre que se aborda o tema pesquisa no contexto do desenvolvimento para a web, um *design pattern* extremamente generalizado surge à ideia de quem procura maneira de resolver tal problema: o *Google*. Apesar de ter uma aparência extremamente simples no que diz respeito à interface, é extremamente poderoso no motor que utiliza por trás. Obviamente que para este projecto não se quer um poderio tão exagerado para a pesquisa pelo que nem de perto se pensou em utilizar motores de busca como o do Google. Porém, a interface de pesquisa, sendo bastante simples e já bastante difundida, será um ponto de partida para a realização da interface de pesquisa do arquivo digital, assim como a interface onde se apresentam os resultados.

Nesta primeira fase de construção da base da pesquisa, não se vai querer fazer busca ao conteúdo dos documentos que compõem um trabalho prático, tópico que apenas será abordado na segunda iteração sobre a pesquisa no arquivo digital (Secção 5.7). Assim o que se pretende é apenas efectuar pesquisa na meta-informação de um trabalho.

Não havendo muitas opções para se discutir, propor e decidir, optou-se por seguir em frente com as pesquisas simples na base de dados, onde está, precisamente, a meta-informação para o trabalho prático.

Os motores de bases de dados, mais propriamente as linguagens de bases de dados — SQL — fornecem aos seus utilizadores facilidades de pesquisa nos seus conteúdos, que porém não se deve abusar deles. Essas facilidades são as possibilidade de se escreverem expressões regulares por meio das palavras reservadas “LIKE” ou ‘%’ juntamente com as expressões usuais de *query*.

Assim a implementação do motor de pesquisa não foi mais além que uma ou várias *queries* à base de dados, utilizando-se mecanismos simples de conjunção de argumentos, quando passados pelo utilizador. Imaginando que o utilizador faz uma pesquisa por título do trabalho, então a expressão SQL básica a ser passada ao servidor de base de dados será a seguinte, imaginando ainda que em *\$título* fica guardado o valores passado pelo utilizador:

```
SELECT      *
           FROM TP
           WHERE TP.Titulo LIKE '%$título%'
```

A pesquisa por árvore taxonómica é igualmente simples em termos de queries à base de dados. A única diferença, e daí o facto de se ter decidido

fazer um tipo de pesquisa independente da outra, é que é necessário processar o ficheiro XML com a árvore taxonómica para se poder dar ao utilizador a possibilidade de ele criar um caminho nesta árvore.

5.5.2 Resultados Obtidos

Como já foi abordado na secção anterior, viu-se a necessidade de criar duas interfaces de pesquisa que surtem no mesmo resultado. Na Figura 5.7 é mostrada a interface para a pesquisa por taxonomia, onde o utilizador deve sugerir um caminho na árvore da taxonomia ACM para obter resultados de pesquisa.

The screenshot shows the ADTPis website interface. At the top, the logo 'ADTPis' is displayed with the subtitle 'Arquivo Digital para Trabalhos Práticos'. Below the logo is a navigation bar with links for 'Home', 'Quem Somos', and 'Contactos'. On the left side, there is a 'Menu' section with links for 'Pesquisar', 'Login', and 'Registrar'. The main content area is titled 'Pesquisa Por Taxonomia' and features a search form. The form includes a label 'Escolha o caminho para o termo:' followed by four dropdown menus. The selected options in the dropdowns are 'Software', 'SOFTWARE ENGINEERING', 'Requirements/Specifications', and 'Elicitation methods (e.g., rapid prototyping, interviews, JAD)'. A 'Submeter' button is positioned below the dropdowns. At the bottom of the page, the footer contains the text '© 2008 Carlos I Nuno' on the left and 'Template by Arcsin' on the right.

Figura 5.7: Página de pesquisa de trabalhos práticos por caminho da taxonomia ACM

Esses resultados seguem a mesma ideia que os resultados referentes às pesquisas no google, isto é, é apresentada ao utilizador uma listagem de trabalhos práticos que fazem *match* com os argumentos de pesquisa. Para cada uma destas entradas é disponibilizado uma referência para um trabalho prático, e que resulta numa folha bastante simples mas que apresenta toda a informação do trabalho e a partir de onde se podem realizar outras pesquisas implícitas no texto de apresentação do trabalho. Por exemplo, ao clicar no nome de um autor, é mostrada uma nova lista com todos os trabalhos desse autor, como se se tivesse feito pesquisa por aquele autor. Na Figura 5.8 mostra-se uma dessas páginas.

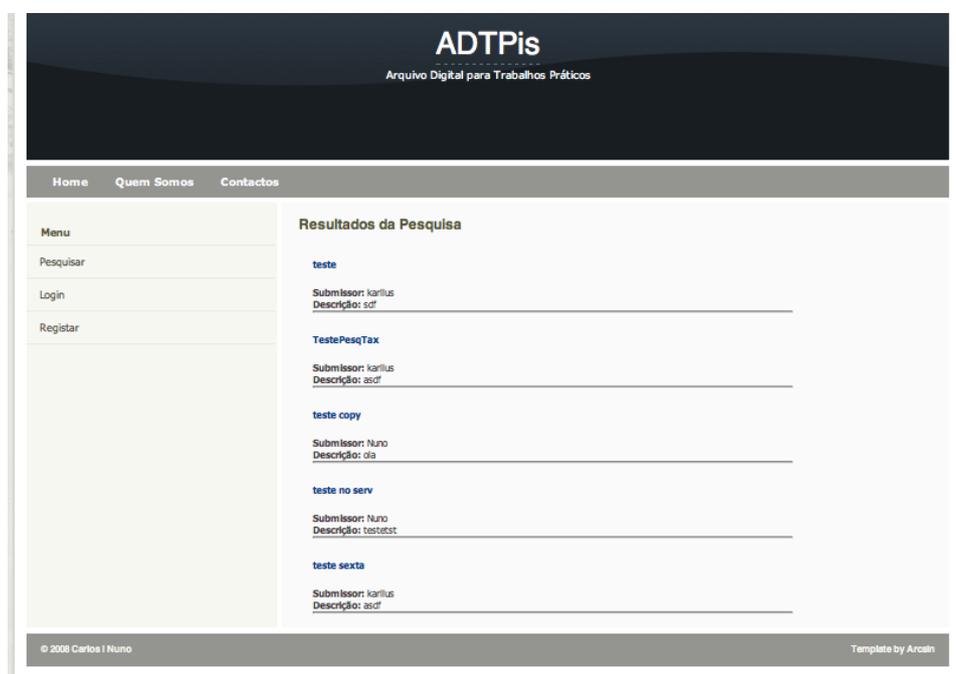


Figura 5.8: Página que apresenta a informação de um trabalho prático resultante de uma pesquisa

5.6 Disseminação

Em capítulos anteriores apresentaram-se pontos de vista, discussões e decisões essenciais para o desenvolvimento de tarefas relativas às duas primeiras fases do ciclo de vida de um Objecto Digital. Nesta secção está-se então preparado para se concluir este ciclo de vida, adicionando-se à análise feita anteriormente a análise relativa à disseminação de um OD, que no presente contexto se materializa num trabalho prático (TP).

A disseminação de um TP, muito sumariamente, corresponde à possibilidade de facultar conteúdos de informação referentes ao mesmo, dado por um DIP. Por um lado esta última fase pode corresponder a um único *click* para fazer o *download* de um ficheiro comprimido; porém, por outro lado, há que se proceder à construção deste mesmo ficheiro comprimido. Esta tarefa é particularmente complexa dado que envolve inúmeras sub-tarefas. Estes pequenos passos que podem ser vistos como filhos da disseminação serão então amplamente analisados e discutidos até se chegar a uma decisão final de como se irão conceber. Contudo, antes de se avançar na discussão é importante identificar quais são exactamente estes passos.

Um DIP de um TP deve conter no seu interior quatro estruturas de informação essenciais, as quais se listam a seguir, como sendo aquelas que vão merecer maior foco de análise:

- A estrutura da árvore de directorias;
- Uma declaração dos direitos de cópia;
- Metainformação;
- Manifesto com informação sobre o pacote;

Juntamente com estes quatro elementos, vão-se analisar algumas partes de integração desta última fase com as outras fases anteriores, por forma a garantir a possibilidade de se ter uma abordagem robusta, e possibilitar a disseminação final dos pacotes.

5.6.1 Árvore de Directorias

Até esta fase do projecto, o armazenamento dos ficheiros relativos a um TP foi pensado de forma muito linear. Decidiu-se, inicialmente, que para cada TP, os seus ficheiros estariam ao mesmo nível da árvore de directorias. A materialização desta ideia foi realizada sem qualquer tipo de dificuldade, pois não houve preocupação em guardar qualquer tipo de estruturação: um ficheiro pertencia a um trabalho prático e nada mais se dizia.

Contudo, para um TP ser consistente deve haver uma estruturação do sistema de ficheiros de modo a permitir o endereçamento, em vários níveis, de ficheiros que compõem o TP.

Existindo uma estrutura nivelada – no sentido de ser constituída por níveis de directorias – de ficheiros, o utilizador já não tem a necessidade de se preocupar se dois ficheiros vão ter o mesmo nome, desde que fiquem em directorias diferentes. Pode parecer um pequeno problema, mas endereça na perfeição um grave problema de consistência entre aquilo que o utilizador pretende, e aquilo que o sistema fornece.

Deste modo pensou-se nas seguintes maneiras de resolver o problema:

- **Criação de directorias on line:** Com esta opção, o utilizador seria responsável por construir a sua própria árvore de directorias, antes de submeter os seus ficheiros, ou ao mesmo tempo que está a proceder a essa tarefa.
- **Análise das URLs:** Com esta opção todo o trabalho ficaria do lado do servidor. A análise de caminhos e verificação de partes comuns nestes caminhos seria um meio para verificar que ficheiros ficariam no mesmo nível, e que directorias seria necessário criar.

Ambas as opções acima referidas são válidas. A par delas poder-se-iam considerar outras possíveis soluções, mas estas pareceram ser as que mais consenço trouxeram para o desenvolvimento futuro.

Prós e Contras

Em relação à primeira abordagem há vários *contra* essencialmente por ser o utilizador a controlar a criação de directorias. Este problema não é grave pois ainda que ele gere um milhão de directorias, o espaço ocupado é sempre relativo ao espaço dos ficheiros. Contudo este problema endereça outro que diz respeito à criação de manifestos dos pacotes que serão abordados mais à frente na secção 5.6.3, e que tem a ver com o processamento de directorias e seus conteúdos.

Quanto a argumentos favoráveis, pode-se dizer que é uma óptima solução pois não necessita de grande esforço de processamento do servidor para encontrar qual a directoria em que dado ficheiro vai ser *encaixado*, ao contrário da segunda abordagem!

A segunda abordagem tem um mal grande que é o ter que *adivinhar* exactamente qual a directoria em que dado ficheiro vai ficar. A análise de caminhos e partes comuns nos mesmos pode ser vantajosa se os ficheiros a serem adicionados estiverem numa mesma directoria antes de serem carregados para o sistema e forem para a uma mesma directoria após *upload*. No caso de não estarem todos na mesma directoria antes de serem carregados, aparentemente não existe forma simples de verificar se o utilizador pretende colocar todos na mesma directoria ou em directorias distintas.

Decisão

Dada a breve análise efectuada na subsecção anterior, optou-se por escolher a primeira opção. Como foi dito, não há qualquer problema em se ter milhões de directorias vazias, dado que o espaço em disco apenas concerne ao espaço utilizado pelos ficheiros do TP.

Resultados Obtidos

Na Figura 5.9 é mostrado então o resultado obtido após implementação. Como se pode ver, e comparar com a Figura 5.6, nesta nova versão o utilizador tem acesso a uma árvore de directorias permanentemente aberta, mostrando desde logo a disposição dos ficheiros que já foram submetidos.



Figura 5.9: Página renovada para o upload de ficheiros e criação de novas directorias

Nesta mesma zona o utilizador tem a facilidade de seleccionar qual a directoria onde quer inserir o seu ficheiro, ou então onde quer criar uma nova directoria. O caminho escolhido é sempre apresentado ao utilizador. Na Figura 5.9 a directoria seleccionada tem o nome de “Apresentacao” e está na raiz do trabalho prático.

Deste modo é possível criar uma estrutura mais organizada de ficheiros que compõem o trabalho prático.

5.6.2 Copyright

Como já foi relatado na introdução inicial deste capítulo, um DIP não é muito mais que um pacote com informação relativa a um TP. Essa informação pode ser um simples relatório de desenvolvimento que não exigiu muito dos autores, como pode também ser um programa completo com grande valor para os mesmos. Ora se esse pacote poder ser distribuído, então qualquer utilizador vai poder utilizar livremente essa informação se nada for dito em contrário.

Segundo [Cop], o copyright é uma forma de proteger os autores de trabalhos originais. No contexto do arquivo digital de trabalhos práticos, o trabalho pode não ser tão original quanto aquilo que o uso de copyright poderia estar à espera; dado que no mesmo arquivo digital podem estar trabalhos com o mesmo assunto e as mesmas ideias aplicadas. Neste caso o copyright é uma forma de proteger o trabalho feito por alguém e não propriamente proteger as ideias inerentes e o seu autor.

Deste modo é imperioso que cada trabalho submetido no arquivo digital esteja ao abrigo de um documento que dite uma dada lei de utilização da informação contida no TP. Com este mecanismo o utilizador que acede a esta informação tem obrigatoriamente que concordar com os termos indicados pelo documento a que se chamará daqui em diante de declaração de *copyright*.

Como soluções para este requisito importante, foram pensadas as seguintes:

- **Uma única declaração para todo o arquivo:** A utilização desta opção poderia não ser aceite por todos os utilizadores do arquivo. Por exemplo, no caso de ser pouco proibitiva poderiam existir produtores de TPs que não gostassem dos termos de protecção oferecidos pelo arquivo. Contudo seria bastante simples de implementar.
- **Uma declaração por TP com possibilidade de escolha fixa:** Esta solução permite mais flexibilidade que a anterior, podendo o produtor do TP escolher uma de entre um certo número de declarações fixadas pelo sistema. Assim poderia escolher aquela que mais se adequasse à sua ideia de protecção requerida para o seu trabalho.
- **Uma declaração por TP idealizada pelo utilizador:** Apesar de ser parecida com a solução anterior, apresenta algumas diferenças e lacunas. O facto de ser o utilizador a idealizar e escrever a sua própria declaração de copyright vai provocar incoerências entre as várias declarações existentes para cada trabalho. O texto escrito deixará de ser cuidado sob o ponto de vista da lei jurídica, o que poderá fazer a credibilidade do arquivo perder-se no seu todo. Outro problema anexado

a esta solução é o facto de o produtor não inserir qualquer declaração de copyright, a qual se pretende obrigatória.

Decisão

Feita a análise às várias soluções apresentadas, decidiu-se optar não por uma, mas sim por uma mistura entre duas delas.

A primeira solução é interessante pois coloca o arquivo digital num patamar guiado por regras, o que o faz tornar bem mais robusto. Contudo pelas razões indicadas em cima, a solução por si só não é flexível para todos os utilizadores.

A segunda solução apresentada, é, sem margem para dúvidas, aquela que rege melhor os interesses dos produtores, nesta caso, os alunos. O facto de se ter regras fixadas e a obrigatoriedade em escolher uma delas faz com o arquivo esteja sempre coerente e que haja sempre alguma variedade entre as várias possibilidades de utilização da informação.

Assim, juntando a primeira e a segunda solução adquire-se um estatuto regrado consistente e coerente.

Decidiu-se, com base em [Cop] que as opções fixas a dar ao produtor do TP seriam as seguintes:

- Os autores autorizam a utilização livre de todo o conteúdo do pacote, sem qualquer tipo de restrições.
- Os autores autorizam a utilização do conteúdo do pacote como meio de apoio para completar e/ou reutilizar noutros trabalhos de carácter público ou pessoal. Requerem no entanto que sejam preservados os direitos de autor.
- Os autores autorizam a utilização de todo o conteúdo do pacote apenas para uso pessoal mas sem vista a utilização pública para re-distribuição ou ainda alteração dos conteúdos do mesmo.
- Os autores autorizam a utilização de parte do conteúdo apenas para uso pessoal com fins de consulta. Apenas a utilização das ideias expressas em relatórios e outro ficheiros de conteúdos relacionados com a documentação é permitida. Não são permitidas re-distribuições do pacote e utilizações públicas de código de maneira alguma.

Como regra geral do arquivo, optou-se por ter um simples texto que pretende avisar o produtor que o TP inserido é propriedade dos seus autores, podendo este ser alterado de qualquer forma, mas que ao mesmo tempo passa a ser propriedade do arquivo, no sentido em que este não vai ser de modo algum removido pelos seus autores.

Foi ainda decidido que esta escolha da declaração de copyright seria apresentada ao produtor logo no momento inicial da submissão do seu trabalho prático. No caso de ele não concordar com os termos, é-lhe de imediato impossibilitada a submissão do trabalho.

Resultados Obtidos

Dada a discussão e decisões acima, foi bastante simples implementar esta tarefa. Cumprindo-se assim um requisito que até então não tinha sido analisado. Apesar de ser sabido que no conteúdo de um DIP devesse estar presente uma declaração de copyright.

Para conclusão desta tarefa, foi adicionada uma nova tabela à base de dados, a qual já se encontra na Figura 5.1, e que suporta os textos disponíveis para a declaração de copyright. Estes pequenos textos são utilizados para serem apresentados aos utilizadores aquando do início da submissão do trabalho prático.

A página final é apresentada na Figura 5.10, e como se pode ver, apresenta os textos que foram decididos e transcritos em cima. Sempre que for necessário adicionar mais textos de declarações na base de dados, esta página está preparada para os poder suportar, e manter o seu normal funcionamento.

Copyright

Caro utilizador
Ao confirmar uma das escolhas listadas imediatamente abaixo, está a concordar com as regras que regem o *ADTPis*.
Os direitos de autor de todos os conteúdos dos trabalhos práticos submetidos ao *ADTPis* são mantidos, podendo os seus autores alterar esses conteúdos sempre que quiserem e de todas as formas possibilitadas pelo arquivo. Porém após submissão, todos os trabalhos práticos passam ser também propriedade deste arquivo, no sentido em que nenhum poderá ser removido do repositório.
Os autores são os únicos responsáveis pela legalidade ou ilegalidade dos seus trabalhos. Em caso algum o arquivo se responsabilizará por fraudes, plágios e outros crimes semelhantes.

Os autores autorizam a utilização livre de todo o conteúdo do pacote, sem qualquer tipo de restrições.

Os autores autorizam a utilização do conteúdo do pacote como meio de apoio para completar e/ou reutilizar noutros trabalhos de carácter público ou pessoal. Requerem no entanto que sejam preservados os direitos de autor.

Os autores autorizam a utilização de todo o conteúdo do pacote apenas para uso pessoal e sem vista a utilização pública para re-distribuição ou ainda alteração dos conteúdos do mesmo.

Os autores autorizam a utilização de parte do conteúdo do pacote apenas para uso pessoal e com fins de consulta. Apenas a utilização das ideias expressas em relatórios e outros ficheiros de documentação é permitida. Não são, ainda, permitidas re-distribuições do pacote e utilizações públicas de código de maneira alguma.

Figura 5.10: Página de apresentação de escolha do Copyright associado ao TP

Repare-se que esta declaração de copyright será relacionada com o trabalho prático que anda não foi submetido, portanto este indicador não é logo inserido na base de dados, sendo passado para o passo seguinte no workflow da submissão do TP através de uma variável de sessão.

5.6.3 Manifesto e Meta-Informação

Para cada TP no arquivo digital, existe meta-informação associada que pode ser do interesse de todos os utilizadores aquando do acesso aos DIPs. Assim esta informação deve estar presente em cada DIP para se poder divulgar algumas características do trabalho que foi pesquisado, mas off-line.

Ainda em relação à informação do DIP, deve existir um mecanismo que ajude o utilizador a saber se o pacote está correcto e de acordo com aquilo que era suposto fazer *download*.

Para ambos os requisitos abordados acima, tentou-se propor algumas soluções que serão identificadas mais à frente. Estas possíveis soluções não se prendem a questões como “Qual a ideia base?”, mas sim a questões de cariz tecnológico. Isto deve-se ao facto de se ter desde logo decidido que esta informação seria disponibilizada ao utilizador por meios de ficheiros de texto. Os conflitos surgem quando se quer decidir em que tipo de ficheiros de texto disponibilizar esta informação.

Assim as hipóteses idealizadas foram as seguintes:

- **Utilizar ficheiros de Texto Pleno:** Esta solução é simples e bastante eficaz para o utilizador final que vai aceder a esta informação, devido ao facto de não haver qualquer tipo de estruturação mais complexa do texto com informação.
- **Utilizar ficheiros XML:** Com esta solução consegue-se ter, por exemplo, uma representação exacta da estrutura do sistema de ficheiros de um pacote, e ainda adicionar meta-informação a cada uma das partes envolventes no pacote.
- **Utilizar ficheiros JSON e YAML e outras linguagens de anotação:** Apesar de XML também ser uma linguagem de anotação como o JSON e o YAML, o XML é bastante mais popular que as anteriores. A utilização destas linguagens poderia dificultar a capacidade de leitura dos utilizadores menos acostumados com estas linguagens.

Decisões

Olhando a que toda a ideia base já foi delineada, como se referiu acima, então escolher uma tecnologia é muito mais simples e rápido. De todas as

tecnologias apresentadas para servir a ideia base, as que mais se adequam ao requisito final são as duas primeiras.

Contudo, optou-se pela utilização de ficheiros XML para difundir a informação relativa à meta-informação do trabalho e ainda para a composição estruturada do manifesto que retratará a estrutura de todo o pacote. Esta decisão, apesar de ser contraditória com o que foi dito acima relativamente à meta-informação, parece ser a mais eficaz para distribuir tal informação. As dificuldades de leitura dos ficheiros pode ser um ponto fraco, no entanto, hoje em dia, os *browsers* de internet abrem este tipo de ficheiros de maneira muito elegante e que facilita a sua leitura.

Ambos os ficheiros seguirão a estrutura dada por simples DTDs. A seguir são mostrados os DTDs para cada um deste tipo de ficheiros.

O seguinte DTD refere-se ao refere-se ao DTD para a meta-informação.

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!ELEMENT metainfo (trabalho)>
<!ELEMENT trabalho (titulo, subtitulo, curso, disciplina,
                    orientador, autores, descricao, data-submissao)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT subtitulo (#PCDATA)>
<!ELEMENT curso (#PCDATA)>
<!ELEMENT disciplina (#PCDATA)>
<!ELEMENT orientador (nome, email)>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT autores (autor)+>
<!ELEMENT autor (nome, email)>
<!ELEMENT descricao (#PCDATA)>
<!ELEMENT data-submissao (#PCDATA)>
```

O seguinte DTD refere-se ao DTD para o manifesto.

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!ELEMENT manifesto (nomeTrabalho, autores, dir+)>
<!ELEMENT nomeTrabalho (#PCDATA)>
<!ELEMENT autores (autor)+>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT dir (nome, (ficheiro | dir)*)>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT ficheiro (metainfo)>
<!ELEMENT metainfo (nome, extensao, tamanho, propriedade*)>
```

```
<!ELEMENT extensao (#PCDATA)>
<!ELEMENT tamanho (#PCDATA)>
<!ELEMENT propriedade (designacao, valor)>
<!ELEMENT designacao (#PCDATA)>
<!ELEMENT valor (#PCDATA)>
```

Com estes ficheiros em XML estruturados pelo DTD, tem-se acesso não só a toda a disposição dos ficheiros na árvore de directorias, mas também à meta-informação que a eles está inerente.

Resultados Obtidos

Dadas as decisões analisadas em cima, a implementação de processos de criação de ficheiros quer em XML para o manifesto e para a meta informação, quer em texto simples, para a declaração de copyright, foi um processo bastante simples.

Envolveu na sua essência a manipulação de ficheiros e queries simples à base de dados, para se poder popular os campos necessários ao preenchimento deste ficheiros.

Contudo surgiram dois problemas, um de implementação, que envolveu um maior esforço no desenvolvimento de um algoritmo eficaz e capaz de processar paths relativas para criação do manifesto; e outro, ainda não bem resolvido, de ordem tecnológica que está directamente ligado com a codificação textual em UTF-8 e ISO-8859-1.

Assim os resultados obtidos após implementação desta tarefa são a criação de três ficheiros, dois em formato XML para o manifesto e para a meta-informação, e outro em texto simples para a declaração de copyright do pacote que se pretende disponibilizar.

No Apêndice C apresenta-se o resultado de um manifesto gerado automaticamente.

5.6.4 Disseminação

A disseminação de pacotes tem vindo a ser analisada por todo o presente capítulo num sentido lato e anterior à verdadeira distribuição de pacotes, isto é, tem-se debruçado sobre os pequenos passos e não à publicação do pacote para distribuição do mesmo.

Nesta fase, tendo-se já tomadas decisões em relação aos conteúdos do pacote, é necessário juntar todas as partes e decidir em que fases da submissão é realmente criado o DIP e como o distribuir.

Criação de DIPs

A criação de um DIP não necessita de grandes análises em relação à fase de criação, pois este só pode mesmo ser criado depois de se ter toda a informação do TP guardada no sistema. Assim decidiu-se que os DIPs serão criados no momento final à submissão do trabalho prático, estando criado permanentemente estes ficheiros de distribuição para evitar ocupação do servidor sempre que um utilizador requer estes pacotes. A criação *on-the-fly* destes ficheiros comprimidos pode ocupar algum tempo de servidor dado que seria necessário de todas as vezes que um pacote era solicitado proceder à execução de um grande número de passos anteriores à compressão de todo o pacote num único ficheiro.

Para além da criação de DIPs no final da submissão de um TP é necessário ter em conta que estes TPs podem posteriormente ser alterados, havendo a necessidade de se criar novo(s) DIP(s). Assim, outra fase de criação de DIPs será aquando da finalização de uma alteração a um TP.

Tem-se vindo a falar em DIP no plural, mas sem se ter explicado o porquê até então. Isto deve-se ao facto de um DIP ser uma parte, ou se se quiser, um *slice* de todo um TP. No caso em concreto deste sistema, um DIP poderá ser uma um ficheiro comprimido que contenha todo o TP, apenas a parte referente ao relatório, ao código, à apresentação, ou ainda a algo escolhido pelo utilizador.

Distribuição de DIPs

A distribuição de DIPs está directamente ligada com a pesquisa de TPs no arquivo. Não faz qualquer sentido colocar um DIP fora do contexto de um TP, pelo que aquando do resultado da pesquisa faz todo o sentido deixar acessível ao utilizador final estes DIPs que são criados na altura da criação ou alteração do TP.

Assim, tendo-se uma página web com o resultado e informação referente a um TP, a disponibilização de links para estes documentos de forma agradável e intuitiva será a solução que se vai adoptar para a concepção da ideia aqui formada.

Para além da disponibilização de links directos, decidiu-se ainda que o utilizador irá poder navegar na árvore de directorias de todo um TP tendo a possibilidade de escolher os ficheiros que lhe são mais úteis e criar um DIP *on-the-fly* baseado nas suas escolhas. A opção de criar DIPs enquanto se navega no arquivo vai contra aquilo que foi definido anteriormente, no entanto é um mal necessário para se poder dar maior flexibilidade às escolhas do utilizador final.

Resultados Obtidos

A criação de DIPs foi um processo um tanto ao quanto moroso, pois envolveu não só a articulação entre todas as sub-tarefas analisadas e apresentadas anteriormente nesta secção, mas também a criação e utilização de partes da tecnologia PHP até então ainda não utilizadas.

À semelhança da criação de manifestos a partir de paths relativas dos ficheiros do TP, foi necessário criar um mecanismo idêntico, mas desta vez para recriar um sistema de ficheiros que serviria depois para a criação de ficheiros zip.

O conteúdo desses ficheiros zip é a organização de directorias que envolve os ficheiros com informação do trabalho e ainda os três ficheiros amplamente discutidos na sub-secção 5.6.3: Manifesto, Meta-Informação e Copyright.

No final desta implementação, adicionou-se uma nova área à página de consulta de um TP, que acarreta com todos os DIPs disponíveis por defeito, isto é:

- um pacote zip com o conteúdo de todo o trabalho prático em pesquisa;
- um pacote zip com o todos os ficheiros que dizem respeito ao Relatório do TP;
- um pacote zip com todos os ficheiros que dizem respeito ao Código do TP;
- um pacote zip com todos os ficheiros que dizem respeito à Apresentação do TP.

A disponibilização destes pacotes para download, são apresentados na Figura 5.11.



Figura 5.11: Página de Consulta de um TP, com a disponibilização de DIPs

Atente-se numa outra funcionalidade que está patente na Figura 5.11 ao fundo da página. Esta funcionalidade redirecciona o utilizador para uma página de navegação em toda a estrutura do TP (Figura 5.12), onde ele pode escolher ficheiros da árvore apresentada para fazer parte do seu pacote personalizado de informação daquele trabalho.

À medida que ele passa o rato por cima de um ficheiro, é apresentada meta-informação sobre esse ficheiro acrescida de uma pequena pré-visualização do conteúdo do ficheiro.

Ao seleccionar na árvore um ficheiro, esse ficheiro é adicionado ao pacote virtual podendo ser removido a qualquer altura.



Figura 5.12: Página de criação de Pacotes de informação personalizados

5.7 Pesquisa - Information Retrieval

A recolha de informação, ou o seu termo mais reconhecido – Information Retrieval (IR) – pode ser vista como uma ciência de procura de informação no interior de documentos [Wika].

Pode pensar-se que está apenas conectada à área da informática, mas é errado pensar assim. A recolha de informação é multi-disciplinar, podendo aparecer na área das ciências da computação, da psicologia cognitiva ou mesmo na área da justiça.

Todo o processo de busca de informação é iniciado quando um utilizador de um sistema que implemente pesquisa por IR efectua uma consulta introduzindo as palavras a pesquisar. Desse ponto o sistema apenas faz as operações relevantes para trazer ao utilizador a informação – que pode vir sob a forma de texto, de documentos, etc. – que contenha as palavras ou frases que foram usadas para a questão.

As operações relevantes levemente abordadas anteriormente, podem ser mais simples ou mais complexas, dependendo do modelo que o sistema utiliza. Há inúmeros modelos para a representação intermédia da informação e possível busca da mesma. Um dos mais utilizados, e pioneiros na área, baseia-se na teoria de conjuntos e tem como nome *Standard Boolean Model*[Wikb]. Neste caso as operações que o sistema efectua são simplesmente intersecções, reuniões ou outras operações aplicadas a vários conjuntos. Estes conjuntos contêm referências aos documentos que incorporam as palavras ou frases de pesquisa.

Para além das operações baseadas na teoria de conjuntos, existem ainda as baseadas em teorias algébricas ou probabilísticas, das quais se distinguem o *Vector Space Model* e o *Probabilistic Relevance Model (BM25)*.

Mais à frente, baseando-se nesta breve introdução à recolha de informação, será explicado aquilo que se pretende efectuar, e de que modo proceder para obter resultados palpáveis e interessantes.

5.7.1 Considerações Iniciais

Da introdução feita acima, surge a noção do modelo booleano para a recolha de informação. A sua implementação não é complicada, pois baseia-se numa pequena e simples parte da teoria de conjuntos matemática. Porém, antes de se poder efectuar tais pesquisas booleanas, há que preparar uma estrutura de suporte auxiliar, a partir da qual o sistema iniciará a construção dos variados conjuntos de referências para documentos.

Não havendo muito por onde escolher, e como todo o arquivo digital tem como base de suporte à informação uma base de dados, então decidiu-se que também o suporte relativo à recolha de informação seria acrescentado à base dados actual.

Assim foi necessário pensar que tipo de informação guardar para que o sistema construa, de forma eficaz, os conjuntos já mencionados. Para além daquilo que já se tem na base de dados, foi preciso adicionar uma nova entidade e um novo relacionamento.

A nova entidade a adicionar diz respeito às palavras que existem no conteúdo de cada documento que se guarda no arquivo digital. Como a mesma palavra pode estar em vários documentos, então o relacionamento a adicionar entre esta nova entidade e a entidade que representa os ficheiros

do arquivo tem cardinalidade de muitos para muitos. Ora daqui nasce uma tabela que faz o relacionamento entre as palavras e os ficheiros. Nesta última tabela decidiu-se que seria importante guardar o total de ocorrências de cada palavra que se encontra em cada um dos documentos.

A Figura 5.13 mostra o aspecto do Diagrama ER da base de dados focado nas novas tabelas adicionadas.

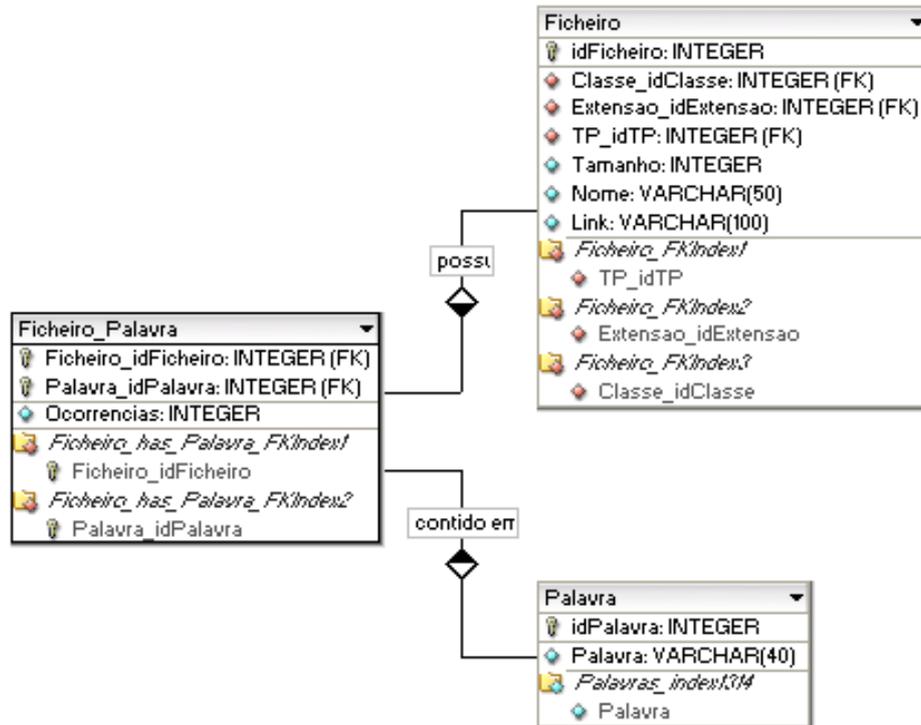


Figura 5.13: Aspecto do DER focado nas tabelas referentes ao suporte de informação para IR

Mas guardar toda e qualquer palavra que possa surgir num documento de texto pode ser uma operação muito pesada e que acaba por não ter qualquer utilidade, pois palavras comuns como “a”, “um”, “ela”, “para”, “como”, “de”, . . . não são úteis para efeitos de pesquisa de informação devido a serem palavras comuns e auxiliares no dialecto português.

Deste modo, um filtro baseado numa listagem com estas palavras mais comuns², poderia dificultar o processo de criação do índice de palavras, mas

²Em <http://acdc.linguateca.pt/acesso/tokens/lemas.cetempublico> encontra-se uma lista com palavras mais utilizadas na língua portuguesa. Esta lista será utilizada para proceder ao filtro anunciado acima.

havia a certeza de que nenhuma destas palavras ficaria na base de dados a ocupar espaço desnecessário.

As decisões tomadas nesta parte da presente secção, são tomadas como dados adquiridos, não sendo portanto feita qualquer discussão acerca de alternativas de concepção ou de implementação.

De seguida são apresentadas ideias de concepção para outras fases da tarefa de recolha de informação, que, ao contrário das anteriores, necessitam de discussão e análise para a sua final escolha.

5.7.2 Discussão de Ideias

Nesta parte da presente secção, vai tentar-se explicar quais as alternativas para os vários problemas que se encontraram no decorrer da análise e da concepção da pesquisa textual, quer para a altura da indexação de palavras, quer para a altura da pesquisa propriamente dita.

Indexação de Palavras

Como já foi referido, a indexação de palavras será feita com suporte de Base de Dados, para se poder ter um desempenho mais simples e não se estar a pensar naquilo que já foi muito pensado. Contudo, a interacção entre o processamento das palavras e a Base de Dados é algo que não é imediato no fluxo da tarefa de indexação. Até lá existe um grande caminho a percorrer.

Para se proceder ao processamento das palavras, é necessário, antes de mais, conseguir ler o conteúdo de um ficheiro. Esta tarefa é simples, mas tem as suas questões.

Imagine-se que se quer indexar documentos anotados em XML. A leitura simples, é algo não pretendido, pois sendo assim as *tags* dos documentos seriam também indexadas. Portanto há que fazer um pré-processamento a este tipo de ficheiros. Porém não são apenas os ficheiros XML que padecem deste mal. Ficheiros de HTML entre outros necessitam de ser pré-processados, pois o seu texto não é puro.

As alternativas para se poder ultrapassar este problema são apresentadas de seguida:

- **Processar todo e qualquer documento.** Como já foi dito em cima, a maior parte de tipos de documentos existentes não são de texto puro. O que leva a um pré-processamento inicial para se poder retirar as anotações ou outro tipo de codificação indesejável. Fazer isso para todos os tipos de documentos de texto é bastante trabalhoso pois todos eles apresentam diferentes formas de codificação do texto. Porém, para enriquecimento do arquivo a desenvolver, esta seria a solução óptima.

- **Escolher um pequeno conjunto de documentos.** Esta solução apresenta o mesmo problema que a anterior devido ao facto de se ter que fazer processamento do texto antes de se efectuar qualquer outra operação. Contudo difere, pois não é necessário conhecer a estrutura de todos os tipos de documentos existentes, mas sim apenas de alguns; aqueles que se fixam como indexáveis.
- **Indexar apenas documentos com texto puro.** Esta é a solução mais simples, pois não requer qualquer tipo de processamento prévio aos documentos. Mas tem o lado reverso da medalha, isto é, são extremamente poucos os documentos em texto puro, e raramente são utilizados para escrita de grandes textos. A utilização de apenas este tipo de documentos faria com que o arquivo perdesse alguma importância no que diz respeito ao tópico da Information Retrieval.

Mais à frente, será fundamentada a escolha relativa às três opções apresentadas.

Depois de se ter uma listagem de palavras vindas de um documento, e se ter passado por outras tarefas intermédias é necessário proceder à inclusão destas palavras no índice de palavras, ou seja, inserção das mesmas na Base de Dados. Porém o processo não é simples e imediato.

Confrontar uma nova palavra com as palavras que já existem na Base de Dados, não é um processo simples. Para este problema foram pensados dois processos, os quais podem não ser ainda os mais eficazes. De seguida listam-se as alternativas com uma breve descrição.

- **Criar lista com informação.** Esta opção faz com que se faça uma selecção de todas as palavras que estão em base de dados, e se coloquem numa lista. Essa lista será depois mantida em memória e confrontada com a palavra a verificar se é ou não inserida na base de dados.
- **Fazer pedido à base de dados.** Esta opção evita a manutenção de uma lista em memória. O que se pretende com ela é que por cada palavra que vem do texto se interrogue a base de dados para verificar se a palavra existe ou não. Dependendo da resposta, a palavra é inserida e o índice para essa palavra é criado, ou então não é inserida, mas o índice é igualmente criado.

5.7.3 Decisões Tomadas

Com base nas alternativa discutidas acima, nesta subsecção vai-se fundamentar uma escolha relativa a cada uma das sub-partes do problema da Recolha de Informação.

Indexação de Palavras

A Indexação de Palavras inicia-se com o processamento de ficheiros para se poder retirar as palavras que interessam para serem indexadas.

Como foi analisado, encontraram-se três opções para serem escolhidos os ficheiros que fariam parte da indexação de palavras, enunciando-se prós e contras para cada um deles. Relacionando-se todas, a melhor opção, e a escolhida para esta tarefa, foi a segunda, ou seja, será escolhido um pequeno conjunto de ficheiros que poderão ser processados para indexação.

O conjunto de ficheiros a poder ser processados pelo arquivo digital, será constituído pelos seguintes formatos:

- TXT. Texto puro, como não poderia deixar de ser;
- XML. Texto anotado. Para este tipo de ficheiro é necessário efectuar pré-processamento para serem eliminadas as tags;
- HTML. Estes documentos necessitam do mesmo tratamento que os documentos anotados em XML.
- \LaTeX . Necessita de um tratamento especializado semelhante àquele que será efectuado para os tipos de ficheiro de *markup*, contudo será um pouco diferente pois a estrutura de *codificação* textual é bastante diferente.

A decisão deve-se ao facto de esta ser a opção mais equilibrada quando comparada com as outras duas. Com esta consegue-se obter palavras de ficheiros que não apenas texto puro, e ainda palavras de ficheiros com estruturas mais complexas e que são os mais utilizados para a escrita de documentos, nos dias de hoje.

Após se ter a lista de palavras, é necessário, como já foi explicado em cima, proceder ao seu armazenamento. Esse armazenamento e criação de índice para cada palavra é condicionado pelo facto de a mesma já existir ou não.

Dadas as opções relatadas na área da discussão de ideias, é possível fundamentar agora a escolha.

A primeira escolha é uma opção excelente, mas apenas enquanto a base de dados tem muito poucas palavras indexadas. O gráfico da Figura 5.14 mostra exactamente onde e quando se terá problemas de eficiência.

Isto é, apesar de haver funções que já tratam da pesquisa em listas, estas operações, por mais optimizadas que sejam – normalmente podem-se efectuar em tempo $O(N)$, ou seja, linear no número de elementos da lista – levam o seu tempo a executar, e quanto maior é a lista, maior será o N , e portanto maior será o tempo de processamento e ocupação do servidor.

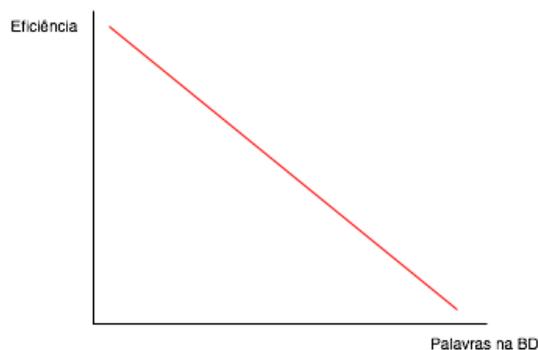


Figura 5.14: Gráfico que descreve o decréscimo de eficiência em função do número de palavras indexadas

Outro problema que surge é que para além de se inserir a informação em Base de dados, quando a palavra não existe, é necessário ainda manter a lista que está em memória actualizada, acabando por se estar a efectuar duas manutenções que podem ser evitadas.

No que diz respeito à segunda opção, apesar de poder não ser muito eficiente, comparada com outras técnicas que possam existir, quando comparada com a anterior é bastante mais eficiente. Tem-se que fazer milhares de pedidos à base de dados, mas são imensamente simples, isto é, materializam-se em queries do tipo:

```
1 | SELECT * FROM Tabela WHERE palavra="x"
```

onde x é a palavra a procurar na tabela de listagem de Palavras.

Ao contrário da opção anterior, não é necessário manter uma lista em paralelo com a manutenção da base de dados, o que simplifica bastante o processo de indexação.

Tendo em conta as fundamentações acima descritas, opção que será levada a cabo é a segunda, pois tem-se mais segurança, maior eficiência, e menos burocracias no processo de indexação.

Assim, e com base em tudo o que foi explicado anteriormente para esta tarefa de recolha de informação, apresenta-se na Figura 5.15 um diagrama que explica os vários passos para se proceder à indexação das palavras de um ficheiro.

5.7.4 Pesquisa

A pesquisa de palavras no interior de documentos, como já foi dito, não é um processo simples que apenas requer busca de informação a uma base de

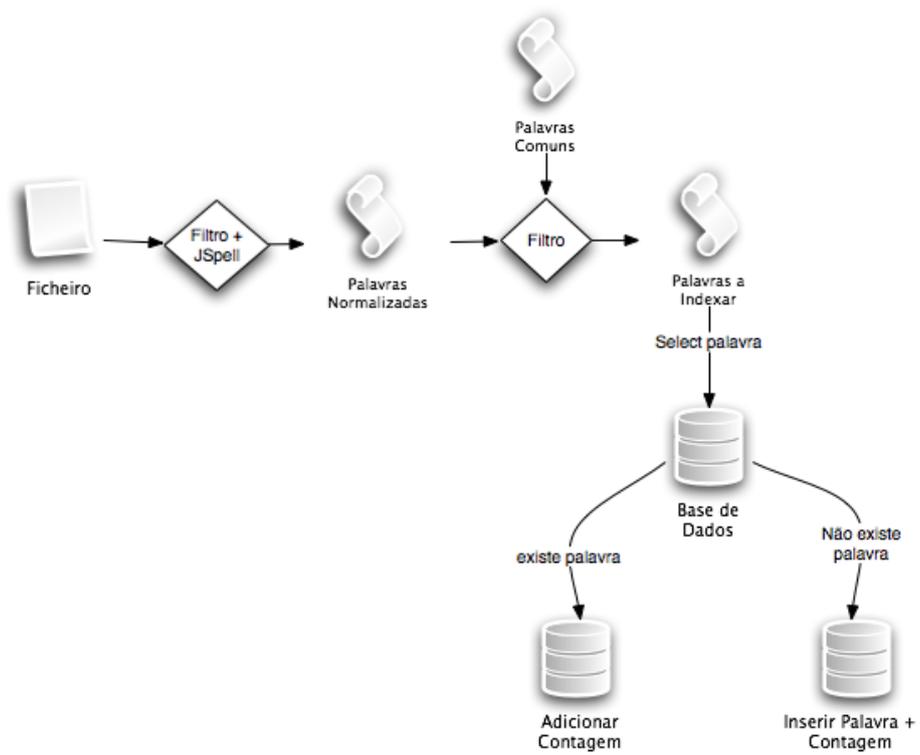


Figura 5.15: Diagrama do fluxo de execução da tarefa de Indexação de Palavras

dados. Pelo contrário. Antes de se poder efectuar tais pesquisas, é necessário ter por trás um mecanismo de extracção de conhecimento dos ficheiros para a base de dados. Este tópico já foi abordado na subsecção anterior, pelo que agora se está nas perfeitas condições de se enveredar pelo caminho das pesquisas.

Tendo em conta aquilo que foi introduzido na presente secção sobre os variados métodos de recolha de informação, vai-se então aplicar a teoria à prática.

Como foi dito, vai ser utilizado o método *Standard Boolean Model*, sobre o qual serão apenas utilizadas as conectivas AND e OR para se efectuar a união das várias listagens de identificadores de ficheiros que contenham as palavras introduzidas pelos utilizadores.

A Figura 5.16 mostra o fluxo de execução ao qual as palavras inseridas pelo utilizador irão ser submetidas.

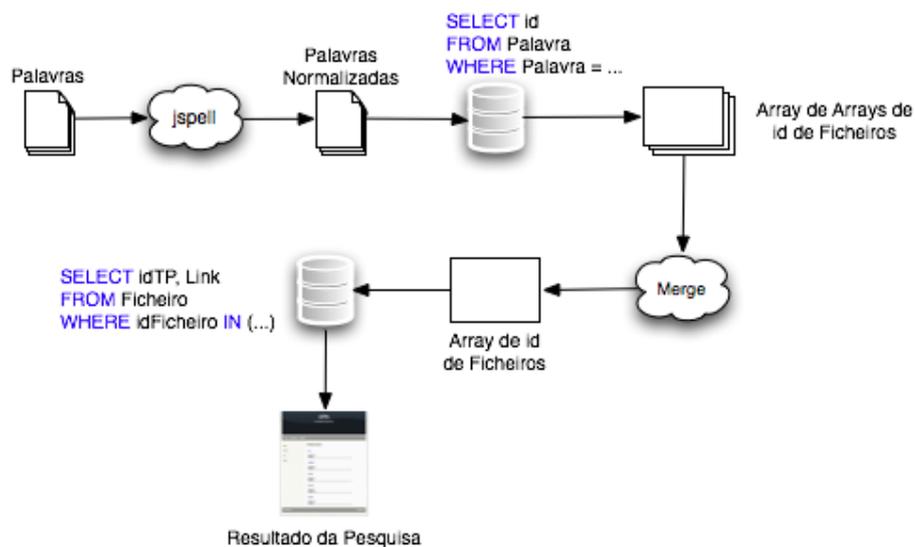


Figura 5.16: Diagrama do fluxo de execução da tarefa de Pesquisa por IR

O primeiro passo é a passagem das palavras ao *jspell* para se obter o seu radical, já que, como visto anteriormente, as palavras inseridas na Base de Dados na altura da sua indexação, estão já normalizadas.

O segundo passo é o acesso à base de dados para popular em memória uma estrutura que guarde os identificadores dos ficheiros que contêm as variadas palavras a pesquisar.

O terceiro passo, corresponde à execução da união por conectivas lógicas da estrutura que foi criada no passo anterior. As conectivas utilizadas, como já foi referido antes, são o E e o OU lógicos. Com o algoritmo para a

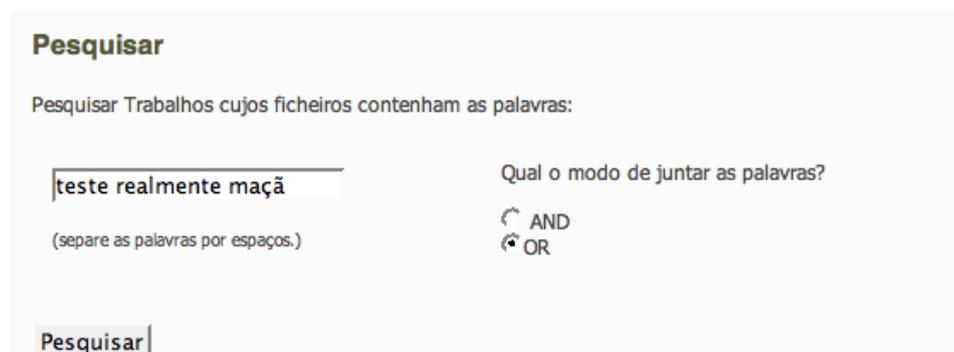
conectiva OU, pretende-se dar ao utilizador todos os ficheiros que tenham pelo menos uma palavra inserida, se existir correspondência entre as palavras e os ficheiros. Com a conectiva E, pretende-se fornecer ao utilizador, apenas os ficheiros que contenham em simultâneo as palavras inseridas. Da execução destes algoritmos é criada uma nova estrutura com os identificadores dos ficheiros que realmente contêm as várias palavras inseridas.

O quarto passo é o segundo acesso à base de dados, onde se vai buscar os identificadores dos trabalhos que contêm os ficheiros resultantes do merge do terceiro passo. Para além do identificador único do trabalho selecciona-se também o link para o ficheiro. Estes dois recursos que se recolhem da base de dados são suficientes para mostrar os resultados da pesquisa quer em termos de trabalho prático, quer em termos de links directos para os ficheiros.

Na próxima subsecção serão mostrados os resultados obtidos para a recolha de informação no interior de documentos.

5.7.5 Resultados Obtidos

A pesquisa por palavras no interior de documentos não necessita de uma interface muito elaborada, em relação àquela que foi concebida para a pesquisa por palavras chave e documentada na Secção 5.5. A Figura 5.17 mostra essa interface.



Pesquisar

Pesquisar Trabalhos cujos ficheiros contenham as palavras:

(separe as palavras por espaços.)

Qual o modo de juntar as palavras?

AND

OR

Pesquisar

Figura 5.17: Interface de Pesquisa no interior de documentos.

Neste exemplo vai fazer-se uma pesquisa pelas palavras “teste”, “realmente” e “maçã”. A conectiva escolhida para ligação dos ficheiros, neste caso é OR (OU).

É uma interface simples e muito intuitiva.

A Figura 5.18 mostra os resultados da pesquisa iniciada na figura anterior.

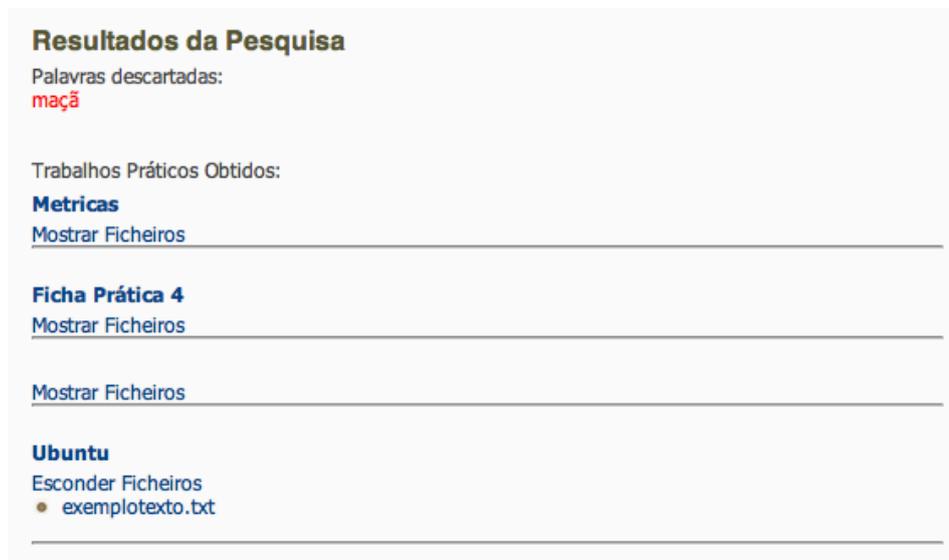


Figura 5.18: Interface de Apresentação dos Resultados da Pesquisa no interior de documentos.

Como se pode ver, a apresentação dos resultados segue um padrão semelhante àquele que foi utilizado para a pesquisa normal. Conseguindo-se aqui bastante reutilização.

No entanto apresenta algumas *nuances* que convém salientar. No início é apresentada uma lista de palavras que são excluídas à partida; a sua exclusão deve-se ao facto de a mesma conter caracteres como assentos, hífen ou sinais de pontuação, os quais não são incluídos na pesquisa por motivos de ordem tecnológica.

Quanto à listagem de resultados obtidos, é apresentado o título do trabalho e uma sub-lista de ficheiros desse trabalho que contém as palavras inseridas na pesquisa. Essa sub-lista aparece à partida escondida para não se ocupar muito espaço na página da apresentação de resultados, mas é possível fazer a sua expansão e consequentemente aceder ao conteúdo dos ficheiros.

O nome do trabalho apresenta desde logo um link que redirecciona o utilizador para a página de apresentação do trabalho prático, tal como mostra a Figura 5.8.

5.8 Web Services

As aplicações para a web, normalmente, dão aos seus utilizadores funcionalidades muito interessantes. Mas para quê tanto interesse se essas funci-

onalidades apenas podem ser acedidas por um único ponto ou apenas por clientes humanos?

Os web services são a solução para combater estas deficiências, promovendo a comunicação entre diferentes aplicações. Com eles uma qualquer aplicação para web deixa de ter apenas clientes humanos, e passa a ter clientes que são outras aplicações.

Os web services promovem a comunicação e integração entre aplicações, e fazem com que novas aplicações possam usufruir de funcionalidades já antigas.

É muito comum ter-se aplicações desenvolvidas em plataformas e em linguagens de programação diferentes. A comunicação entre estas torna-se muito complexa se não existir um meio simples e que abstraia o processo de transferência de dados da comunicação. Os web services utilizam o formato standard XML para encapsular os serviços fornecidos, e facilitar a transferência de pedidos web.

Essencialmente, os web services são baseados em três tecnologias bastante difundidas por toda a internet: o XML, o protocolo de comunicação de internet, HTTP, e o protocolo SOAP.

De todas as tecnologias mencionadas, aquela que está completamente direccionada à utilização de web services é o protocolo SOAP (Simple Object Access Protocol), que na sua essência é um documento XML que foi idealizado para invocação de procedimentos remotos, via RPC (Remote Procedure Calls) ou mesmo para trocas de mensagens entre servidores distribuídos [ws1].

Associado aos web services existe uma linguagem de especificação dos serviços WSDL (Web Services Description Language), que serve, entre outras coisas, para definir o tipo de dados que devem ser enviados e quais aqueles que vão ser retornados, ou seja, define e explica um serviço disponibilizado por um servidor. Baseia-se em XML para definir a estrutura do serviço.

5.8.1 Discussão e Decisões de Implementação

A implementação de web services, nos dias de hoje, é bastante simples, pois quase todas as linguagens de programação têm já bibliotecas que auxiliam o processo de desenvolvimento de web services.

Assim o único problema de implementação foi escolher em que linguagem de programação definir o servidor de serviços. Foram tido em conta as duas opções seguintes:

- **Perl.** Usando Perl como linguagem para definir os web services, tanto a parte de servidor, como a parte de cliente, tem-se facilidade em

desenvolver o código da script, no entanto para a definição dos procedimentos a disponibilizar não é a melhor, devido ao facto de se estar a usar php, e base de dados mysql como base de trabalho. Obviamente que com perl se pode aceder à base dados, mas neste caso causaria alguma quebra de rendimento devido ao facto de não se estar habituado à utilização de módulos perl como o DBI, por exemplo.

- **PHP.** Usando PHP para implementar ambas as partes do web service, ou seja, a parte do servidor e do cliente, tem-se igualmente facilidade, pois existem classes definidas e bem testadas e existe imensa documentação que ajuda na tarefa de desenvolvimento. Como a base do Arquivo Digital está implementada em php, então não haverá entorpia em estudar outro tipo de abordagens.

A decisão foi bastante simples, dada a breve discussão acima. Os argumentos para a utilização de php foram mais fortes. O facto de não causar qualquer tipo de quebra de rendimento, foi sem dúvida e mais importante nesta fase do desenvolvimento do arquivo digital de trabalhos práticos.

Assim, para se poder definir os serviços disponibilizados pela aplicação ADTPis, foi utilizada a classe **nusoap**³, e foram definidas as assinaturas para os mesmos.

5.8.2 Os Serviços

Definiu-se que os serviços a disponibilizar pelo arquivo digital seriam apenas sobre pesquisas à base de dados, resultando em listas de informação sobre trabalhos práticos.

Os modos de pesquisa sobre a base de dados para os serviços, foram exactamente aqueles que já eram funcionalidades específicas do arquivo digital. Contudo algumas diferenças separam os dois modos de pesquisa, pois a pesquisa para os web services é bastante mais expandida do que a que se revela como funcionalidade nativa.

Deste modo definiu-se as seguintes assinaturas para os procedimentos de pesquisa:

Procedimento de Pesquisa no interior dos ficheiros do relatório de cada trabalho prático.

1 | `TextSearch :: Query -> Result`

Onde,

`Query ::` É uma string formatada do tipo `palavraX (AND palavraY)*`.

³Esta classe escrita em e para PHP, pode ser encontrada no seguinte endereço: <http://dietrich.ganx4.com/nusoap/>

Result :: É uma string com um div html, com os vários resultados, formatados do seguinte modo: Nome do trabalho, link para o trabalho e autores.

Procedimento de Pesquisa sobre caminhos da taxonomia.

```
1 | TaxonomySearch :: Category -> Result
```

Onde,

Category :: É uma string formatada do tipo “palavraRAIZ(; palavraCORPO)* ; palavraFOLHA”

Result :: É uma string com um div html, com os vários resultados, formatados do seguinte modo: Nome do trabalho, link para o trabalho e autores.

Procedimento de Pesquisa sobre palavras chave.

```
1 | KeywordSearch :: Keyword -> Result
```

Onde,

Keyword :: É uma string formatada do tipo (Campo)?palavra (AND (Campo)?palavra).

Campo :: É um identificador dos seguintes: autor, orientador, titulo, resumo.

Result :: É uma string com um div html, com os vários resultados, formatados do seguinte modo: Nome do trabalho, link para o trabalho e autores.

5.8.3 Resultados Obtidos

Estes serviços foram definidos e associados ao servidor de web services do ADTPis, utilizando-se a classe acima enunciada, nusoap. Com esta classe tem-se gratuitamente a especificação de um documento WSDL para descrever cada um dos serviços disponibilizados, como se pode ver na Figura 5.19. Portanto estes serviços podem ser utilizados com recurso a WSDL, ou sem recurso ao mesmo.

A API da Figura 5.19 mostra aquilo que é necessário para se poder usar os serviços. As partes mais relevantes são o valor associado ao endpoint, que será o url para se poder aceder ao serviço; os valor associados ao input, que definem o tipo de dados para o serviço ingerir; os valores associados ao output, que definem o tipo de dados que o serviço retorna; finalmente a parte da documentação, que dá informação extra sobre a formatação dos conteúdos dos tipos de dados.

Em relação ao endpoint, para se poder usar o serviço com WSDL, é necessário adicionar “?wsdl” ao valor a ele associado.

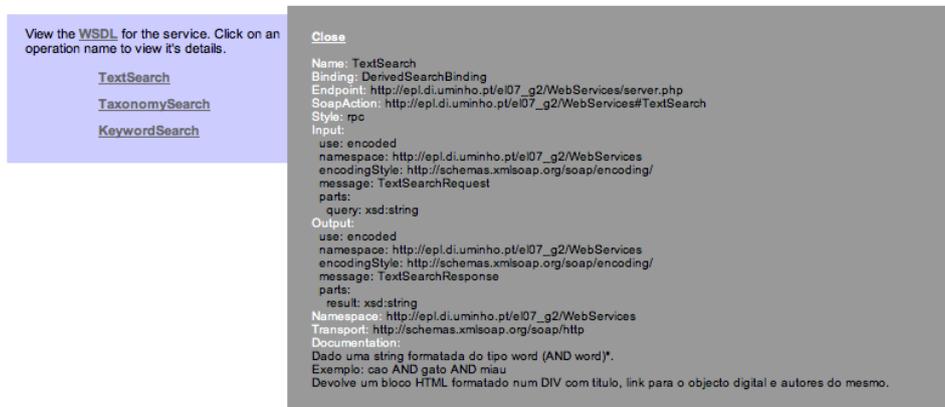


Figura 5.19: WSDL com folha de estilo associado para o serviço de pesquisa textual.

A Figura 5.20 mostra o resultado obtido de uma pesquisa por palavra chave. Neste caso os campos utilizados para a pesquisa foram o autor e o título do trabalho prático.



Figura 5.20: Resultado da utilização do web service definido

O resultado é um elemento div de HTML, cujo conteúdo é o título do trabalho, o link para a página de apresentação desse trabalho, e ainda uma listagem dos autores do mesmo trabalho, tal e qual como foi definido na API para os serviços descritos em cima.

Repare-se ainda na frase que se deu como input ao serviço. Apesar de esta estar formatada seguindo as normas definidas na API, o utilizador que faz a procura não necessita de saber qual a formatação da mesma. Apenas deve inserir as palavras separadas por espaços, como está habitualmente a fazer noutros motores de busca.

Capítulo 6

Conclusão

Neste documento tentou-se descrever com alguma síntese mas de forma bem documentada o processo de análise e desenvolvimento das diversas fases de vida dos elementos de um arquivo digital. E ainda de outras funcionalidades inerentes a estes arquivos.

A análise do problema e o levantamento de requisitos do sistema são duas fases muito importantes no processo de desenvolvimento de Software para que no futuro se consiga perceber sem grande esforço o que se pretende da peça do software a desenvolver.

Embora a análise e descrição do problema efectuadas tenham sido bastante simples, pareceu aos autores que ficou bem descrito o propósito do mesmo. Os requisitos especificados estão concisos e é simples de perceber as funcionalidades mais importantes do sistema ADTPis.

Em relação à descrição da estrutura do sistema, o facto de se utilizar uma gramática, não retira em nada a legibilidade e a compreensão da estrutura, pois ficam coerentes os relacionamentos entre as entidades que compõem o sistema, que, basicamente, é o que se pretende aquando da representação da estrutura de um sistema.

O digrama Entidade-Relacionamento desenhado e explicado descreve o modo como a informação ficará guardada no sistema.

O desenvolvimento da interface do arquivo, apesar de ser bastante trabalhosa, foi proveitoso no que diz respeito à aprendizagem de novas linguagens de programação web, assim como no cimentar dos conhecimentos em linguagens que já eram conhecidas.

Neste relatório, não foram explicadas as linhas de código desenvolvido, pois os autores não acharam de todo importante essas explicações, no contexto do relatório de desenvolvimento.

6.1 Trabalho Futuro

Em relação àquilo que foi feito, poderia-se melhorar pelo menos dois aspectos:

- Correção de erros de codificação de texto, aquando da criação de ficheiros
- Permitir a eliminação de pastas e alteração da localização de ficheiros aquando do upload e criação da estrutura organizada do trabalho prático.
- Ordenar os resultados das pesquisas textuais por número de ocorrências.

Bibliografia

- [Agi] Site com introdução às metodologias ágeis de desenvolvimento de software. <http://www.wrytradesman.com/articles/IntroToAgileMethods.pdf>.
- [CCS01] C. CCSDS. Reference model for an open archival information system, 2001.
- [Cop] Site americano com algumas definições sobre leis de copyright. <http://www.copyright.gov/circs/circ1.html#wci>.
- [FCR⁺07] Miguel Ferreira, Rui Castro, José Carlos Ramalho, Francisco Barbedo, Luis Corujo, and Luis Faria. Roda : Repositório de objectos digitais autênticos. In *9º CONGRESSO NACIONAL DE BIBLIOTECÁRIOS, ARQUIVISTAS E DOCUMENTALISTAS*, Ponta Delgada, Açores, Portugal, 03.29 2007.
- [JBR99] Ivar Jacobson, Grady Booch, and James Rumbaugh. *The unified software development process*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.
- [TUT] Site com definição de terminologias acerca de arquivos digitais. <http://www.lib.utexas.edu/dams/terminology.html>.
- [Wika] Site com informação sobre information retrieval. http://en.wikipedia.org/wiki/Information_retrieval.
- [Wikb] Site com informação sobre modelo de booleano para recolha de informação. http://en.wikipedia.org/wiki/Standard_Boolean_model.
- [ws1] Site brasileiro com algumas explicações sobre web services. http://devedge-temp.mozilla.org/viewsource/2002/soap-overview/index_pt_br.html.

Apêndice A

Apêndice - Descrição dos Use Cases

A.1 Registo

Utilizador: Visitante

Pré-Condição: Sistema online

Descrição:

0 utilizador introduz os seus dados no sistema

0 sistema valida a inserção

2.a) 0 sistema indica erro na introdução de um campo.

2.a.1) Voltar ao passo 1.

3) 0 sistema indica que o registo ocorreu com sucesso.

Pós-Condição: Utilizador Registado

A.2 Pesquisar

Utilizador: Visitante

Pré-Condição: Sistema Online

Descrição:

Extendido por ?Pesquisar por palavra-chave? ou ?Pesquisar por taxonomia?

Pós-Condição: Pesquisa efectuada com sucesso.

A.2.1 Pesquisar Por Palavra Chave

Utilizador: Visitante

Pré-Condição: Opção de Pesquisa seleccionada e Sistema Online

Descrição:

- 1) O sistema pede ao utilizador que introduza uma ou mais palavras para a procura.
O utilizador introduz a(s) palavra(s)-chave pretendida(s)
 - 2.a) O sistema indica que nenhuma palavra chave foi introduzida.
 - 2.a.1) Voltar ao passo 1.O sistema devolve a lista de trabalhos práticos encontrados
- Pós-Condição: Pesquisa por palavras chave efectuada com sucesso.

A.2.2 Pesquisa por Taxonomia

Utilizador: Visitante

Pré-Condição: Opção de Pesquisa seleccionada e Sistema Online

Descrição:

- 1) O sistema pede ao utilizador que seleccione uma entrada da presente na taxonomia.
 - 2) O utilizador selecciona uma das entradas na taxonomia.
 - 3) O sistema devolve a lista de trabalhos práticos encontrados
- Pós-Condição: Pesquisa por taxonomia efectuada com sucesso.

A.3 Alterar Dados Pessoais

Utilizador: Aluno, Professor

Pré-Condição: Login efectuado e Sistema Online

Descrição:

- 1) O sistema mostra a informação relativa ao utilizador em causa.
 - 2) O utilizador modifica os campos que pretende alterar.
 - 3) O sistema valida as alterações
 - 3.a) O sistema invalida as alterações
 - 3.a.1) Voltar ao passo 2.
 - 4) O sistema confirma as alterações.
- Pós-Condição: Alterações efectuadas com sucesso.

A.4 Submeter TP

Utilizador: Aluno

Pré-Condição: Login Efectuado e Sistema Online

Descrição:

- 1) Utilizador escolhe a sua opção de copyright.
 - 1.a) Utilizador não escolhe copyright.
 - 1.a.1) Finalizar operação.
- 2) O sistema guarda o copyright escolhido.
- 3) O utilizador introduz os dados relativos ao trabalho pratico.
- 4) O sistema valida os dados.
 - 4.a) O sistema invalida os dados
 - 4.a.1) Voltar ao passo 3.
- 4) O utilizador insere os autores do trabalho prático.
- 5) O sistema valida os dados.
 - 6.a) O sistema invalida os dados
 - 6.a.1) Voltar ao passo 5.
- 7) O utilizador selecciona o orientador.
- 8) O sistema valida os dados.
 - 8.a) O sistema invalida os dados
 - 8.a.1) Voltar ao passo 5.
- 9) O utilizador selecciona uma chave da Taxonomia ACM.
 - 9.a) O utilizador não selecciona chave da Taxonomia
 - 9.a.1) Ir para o passo 9
- 10) O sistema associa a chave ao trabalho prático.
- 11) O utilizador escolhe um ficheiro do seu Sistema de Ficheiros.
- 12) O utilizador submete o ficheiro
- 13) O sistema faz o upload do ficheiro.
 - 13.a) O upload do ficheiro foi mal sucedido.
 - 13.a.1) Voltar ao passo 11.
- 14) O utilizador confirma os dados do ficheiro.
- 15) O Sistema associa o ficheiro ao trabalho prático
- 16) O Sistema analisa a extensão do ficheiro
 - 16.a) O ficheiro tem propriedades necessárias a preencher
 - 16.a.1) Utilizador introduz as propriedades relativas ao ficheiro
 - 16.a.2) O Sistema valida as propriedades.
 - 16.a.2.1) O Sistema associa as propriedades ao ficheiro submetido
 - 16.a.2.a) O sistema não valida as propriedades
 - 16.a.2.a.1) Voltar ao passo 16.a.1)
 - 17) O utilizador confirma o fecho do trabalho prático.
 - 17.a) O utilizador pretende introduzir um novo ficheiro.
 - 17.a.1) Voltar ao passo 11.
 - 18) O sistema torna o trabalho visível a todos os utilizadores.

Pós-Condição: Submissão efectuada com sucesso.

A.5 Consultar Trabalho Prático

Consultar Trabalho Prático

Utilizador: Professor ou Aluno

Pré-Condição: Login Efectuado e Sistema Online

1) Utilizador selecciona um trabalho prático da sua lista pessoal

2) O sistema mostra o trabalho prático ao utilizador

Pós-Condição: O trabalho foi mostrado a utilizador.

A.6 Alterar TP

Utilizador: Professor e Aluno

Pré Condição: O trabalho prático foi submetido, Login foi efectuado e o Sistema está online.

1) O utilizador selecciona um trabalho prático

2) O sistema mostra o trabalho prático em modo de edição

3) O utilizador insere as alterações

4) O sistema valida as alterações

4.a) O sistema não valida as alterações

4.a.1) O sistema mostra ao utilizador o trabalho inalterado

4.a.2) Voltar ao passo 3

5) Sistema verifica completude do trabalho

5.a) Trabalho está completo

5.a.1) Sistema coloca o trabalho visível a todos os utilizadores

5.b) Trabalho está incompleto

5.b.1) Sistema coloca o trabalho invisível para os utilizadores

Pós-Condição: O trabalho foi alterado

A.7 Inserir Extensões

Utilizador: Administrador

Pré-Condição: Sistema Online e Login Efectuado

Descrição:

1) O utilizador introduz a extensão que pretende adicionar.

2) O sistema valida a nova extensão.

2.a) O sistema não valida a extensão.

2.a.1) O sistema indica que a extensão já existe.

- 2.a.2) Voltar ao passo 1.
 - 3) O sistema indica ao utilizador que podem ser introduzidas propriedades sobre a extensão.
 - 3.a) A extensão possui propriedades a introduzir.
 - 3.a) O utilizador introduz propriedades sobre a nova extensão.
 - 3.a.1) O sistema confirma as novas propriedades associando-as à extensão introduzida.
 - 3.a.1.a) O sistema valida as novas propriedades.
 - 3.a.1.b.1) O sistema invalida as novas propriedades.
 - 3.a.1.b.2) Voltar ao passo 3.a.
 - 3.b) A extensão não possui propriedades.
 - 4) Extensão introduzida.
- Pós-Condição: Extensão introduzida com sucesso no sistema.

A.8 Inserir Propriedades

Utilizador: Administrador

Pré-Condição: Login Efectuado e Sistema Online

Descrição:

- 1) O utilizador introduz a extensão à qual pretende introduzir propriedades.
 - 2) A extensão existe.
 - 2.a) A extensão não existe.
 - 2.a.1) Voltar ao passo 1.
 - 3) O utilizador introduz as propriedades relativas à extensão pré-seleccionada.
 - 4) O sistema valida as propriedades.
 - 4.a) O sistema invalida as propriedades.
 - 4.a.1) Voltar ao passo 3.
 - 5) Propriedades inseridas.
- Pós-Condição: Propriedades introduzidas com sucesso.

A.9 Confirmar/Cancelar Registo Professor

Utilizador: Administrador

Pré-Condição: Sistema Online e Login Efectuado.

Descrição:

- 1) O utilizador verifica a existência de um registo não confirmado de um professor.

- 1.a) O utilizador confirma o registo do professor.
 - 1.a.1) O sistema confirma a verificação.
 - 1.b) O utilizador cancela o registo do professor.
 - 1.b.1) O sistema confirma o cancelamento.
 - 1.b.2) O sistema remove o registo.
 - 2) Registo confirmado ou removido.
- Pós-Condição: Registo confirmado ou cancelado pelo administrador.

Apêndice B

Apêndice - Diagrama ER

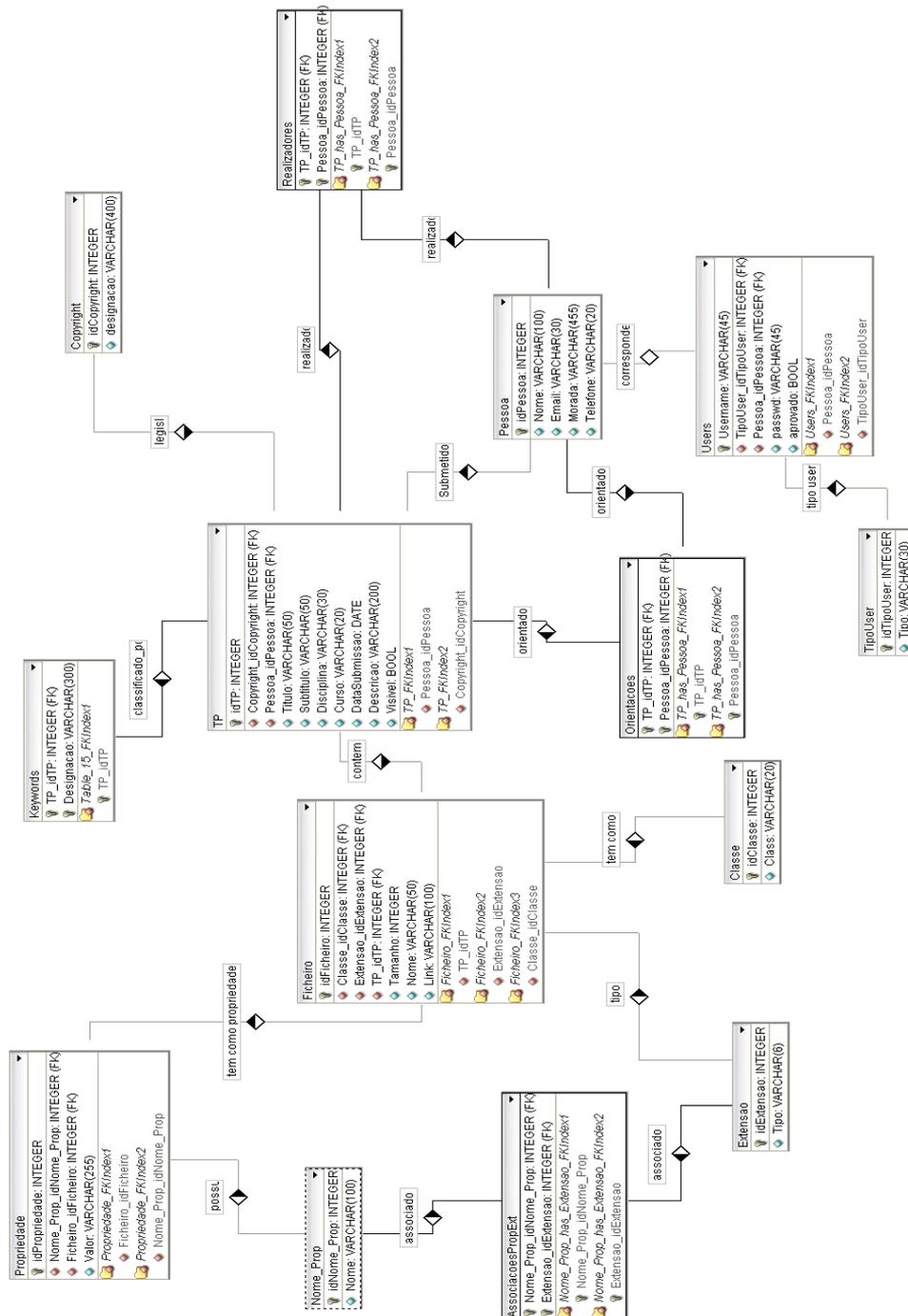


Figura B.1: Diagrama ER para o sistema ADTPis

Apêndice C

Exemplo de Manifesto Gerado Automaticamente

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<manifesto>  
<nomeTrabalho>  
Imagens  
</nomeTrabalho>  
<autores>  
<autor>  
Nuno  
</autor>  
</autores>  
<dir>  
<nome>  
  uploads  
</nome>  
<dir>  
<nome>  
  TP_77  
</nome>  
<dir>  
<nome>  
  Trabalho  
</nome>  
<dir>  
<nome>  
  Apresentacao  
</nome>  
<ficheiro>
```

```
<metainfo>
<nome>
platinum_32
</nome>
<extensao>
jpg
</extensao>
<tamanho>
653118
</tamanho>
<propriedade>
<designacao>
largura
</designacao>
<valor>
1024
</valor>
</propriedade><propriedade>
<designacao>
altura
</designacao>
<valor>
1024
</valor>
</propriedade></metainfo>
</ficheiro>
</dir>
<ficheiro>
<metainfo>
<nome>
water_fire_01
</nome>
<extensao>
jpg
</extensao>
<tamanho>
159431
</tamanho>
<propriedade>
<designacao>
largura
</designacao>
<valor>
1024
```

```
</valor>
</propriedade><propriedade>
<designacao>
altura
</designacao>
<valor>
1024
</valor>
</propriedade></metainfo>
</ficheiro>
<dir>
<nome>
Codigo
</nome>
<ficheiro>
<metainfo>
<nome>
YinYang
</nome>
<extensao>
png
</extensao>
<tamanho>
672650
</tamanho>
<propriedade>
<designacao>
largura
</designacao>
<valor>
1024
</valor>
</propriedade><propriedade>
<designacao>
altura
</designacao>
<valor>
760
</valor>
</propriedade></metainfo>
</ficheiro>
</dir>
</dir>
</dir>
```

```
</dir>  
</manifesto>
```