



**Fernando Miguel
Rocha de Oliveira**

**Procura de Informação em Redes de
Telecomunicações**



**Fernando Miguel
Rocha de Oliveira**

**Procura de Informação em Redes de
Telecomunicações**

dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Dr. A. Manuel de Oliveira Duarte, Professor Catedrático do Departamento de Engenharia Electrónica e Telecomunicações da Universidade de Aveiro

o júri

presidente

Prof. Dr. José Rodrigues Ferreira da Rocha
Professor Catedrático da Universidade de Aveiro

Prof. Dr. Aníbal Manuel de Oliveira Duarte
Professor Catedrático da Universidade de Aveiro

Prof. Dra. Ana Maria Ramalho Correia
Professora Associada com Agregação do Instituto Superior de Estatística e Gestão da Informação da Universidade Nova de Lisboa

Prof. Dr. José Carlos Gouveia Teixeira
Professor Associado do Departamento de Matemática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

agradecimentos

É meu desejo aproveitar este espaço para expressar algumas palavras de gratidão para com aqueles que, das mais diversas formas, me apoiaram e ajudaram a levar a cabo este projecto.

Antes de mais, a minha profunda estima e um sincero obrigado e ao meu orientador, Professor Doutor A.M. Oliveira Duarte, não apenas pelas palavras que me encorajaram a iniciar este projecto, mas também por todo o apoio científico e pedagógico prestado, com inúmeras e proficuas sugestões apresentadas ao longo desta dissertação.

Ao Professor Jorge Ferraz de Abreu, pelo apoio e apreciações que, numa fase ainda embrionária, fez a este projecto. O meu profundo apreço e agradecimento ao Professor Doutor Amaro S. de Sousa, pela colaboração prestada na preparação de publicações preliminares que serviram como ponto de partida para esta dissertação. O meu obrigado ao Professor Doutor Rui L. A. Aguiar, pelas sugestões e críticas que foi apresentando no decorrer do projecto. Ao Professor Doutor José Luís Oliveira e ao Professor Doutor Joaquim Sousa Pinto, igualmente pelas sugestões que foram fazendo ao meu trabalho. Gostaria, ainda, de agradecer a alguém que, de certa forma, tem influenciado o meu percurso profissional, o Engenheiro Manuel Lemos. Para ele, a minha gratidão, pela disponibilidade que sempre demonstrou ao longo de todos estes anos para dialogar comigo e discutir as minhas ideias.

Aos colaboradores e amigos da Universidade de Aveiro, em especial ao Grupo de Sistemas de Banda Larga, com os quais tive oportunidade de colaborar em projectos relacionados com esta dissertação, que, por sua vez, partiram de ideias derivadas da nossa convivência.

Um sincero obrigado aos colaboradores do projecto *Universal Usability Guide*, pelas longas horas de diálogo e debate sobre mais variadas ideias, que de forma tão significativa contribuíram para a elaboração desta dissertação, em particular, às Professoras Doutoradas Joanna McGrenere e Kori Inkpen, e aos Professores Doutores Brad Mehlenbacher e Ben Shneiderman.

À PT-Inovação, pela disponibilidade e colaboração prestadas neste projecto, em especial, aos Engenheiros Fausto Carvalho, Vasco Lagarto, José Neto e Alcino Lavrador, bem como à Doutora Gabriela Moura.

À EuroGTD, pela disponibilidade e apoio prestados à conclusão desta dissertação, em especial, ao Engenheiro José Santos e ao Doutor Eduardo Tapadas.

Um agradecimento especial à minha família, pais e irmã, que, pelo incentivo que sempre me deram, contribuíram de forma decisiva para que esta dissertação se tornasse uma realidade.

Por fim, desejo dedicar esta dissertação à memória de Armando de Oliveira, Avô e Amigo, cujas vivências e ensinamentos constituíram o primeiro impulso para a concretização da minha carreira profissional e continuam a marcar de forma indelével a minha vida pessoal.

resumo

O propósito desta dissertação é o de documentar o desenvolvimento de uma plataforma de *software* que permite, a utilizadores pouco versados na utilização de ferramentas de edição e publicação de conteúdos multimédia, um serviço de publicação de conteúdos e recursos informativos num servidor *WWW*, disponibilizá-los numa interface *Web* sob classificação temática, e apoiar nas buscas relativas aos documentos e outros recursos informativos residentes numa base de dados. Pretende-se propor novas técnicas de distribuição de informação e de aquisição de informação sobre os perfis dos utentes do serviço.

O propósito central desta dissertação foi o de proporcionar um ambiente simples e interactivo de publicação, indexação e classificação de conteúdos, através de uma plataforma de utilização em ambiente *WWW*.

O resultado final do projecto foi o desenvolvimento de uma plataforma de *software* que facilita a utilizadores com poucos conhecimentos técnicos a composição e publicação de documentos em ambientes *WWW* bem como a execução de outras tarefas não triviais. Além disso, foi dado especial relevo a tarefas automáticas de gestão e manutenção do serviço como, policiamento de links, remoção de recursos informativos desactualizados e notificação por *e-mail* de autores e administradores, em determinadas acções (alteração do estatuto do utilizador, notícias, etc.).

Este serviço, teve início através de um projecto designado por AMIS. O projecto AMIS (Agentes Móveis e Inteligentes de *Software*) decorreu entre Maio de 1998 e Novembro de 1999 no Instituto de Telecomunicações (pólo de Aveiro) com o apoio financeiro da Portugal Telecom através do Centro de Estudos de Telecomunicações.

Este projecto teve como validação um serviço disponibilizado aos vários parceiros e colaboradores do projecto de modo a contribuírem para um aperfeiçoamento do mesmo, tanto nas tecnologias de agentes como também na sua funcionalidade e ergonomia.

Também num ambiente de validação de ideias e propósitos a atingir no projecto AMIS, foi criado um Repositório de Informação – Desenvolvimento Científico Tecnológico e Regional (RI-DCTR).

O RI-DCTR apresenta as mesmas funcionalidades do projecto AMIS acrescido e adaptado de outros requisitos no âmbito do projecto DCTR. Este repositório de informação apoia as actividades de um projecto PRAXIS com o nome “Desenvolvimento Científico Tecnológico e Regional”.

Através da experiência adquirida nos projectos AMIS e RI-DCTR, o serviço foi adaptado a uma plataforma universal de publicação de recursos e conteúdos, que mais tarde serviu outros interesses como, por exemplo, o UUGuide (*Universal Usability Guide*), um projecto internacional da ACM (*Association for Computing Machinery*), abordado mais à frente nesta dissertação.

abstract

The purpose of this thesis is to document the development of a software platform that makes easy, for users with few technical knowledge, the composition and publication in a *Web* server, makes documents available in a *Web* interface under theme classification, supports searches related to documents and other info resources in the database. The work is also targeted towards the identification of new techniques for information distribution and for the acquisition of personalized user profiles.

The main purpose of this thesis was to afford a simple and interactive contents publication, indexation and grouping environment through a user *Web* platform interface.

The project final result was the development of a software platform that enables users with little technical knowledge the composition and publication of documents in *Web* environments as well as the execution of other non-trivial tasks. Besides, was given special importance to automatic service management and maintenance tasks like link policing, outdated content resources removal and e-mail notification from authors and administrators in certain actions (user status changes, news, etc.).

This service started with the AMIS project. The AMIS (Mobile and Intelligent Software Agents) occurred between May 1998 and November 1999 in the Institute of Telecommunications, Aveiro branch with the financial support of Portugal Telecom through CET (at present PT Innovation).

This project had, as validation a service, available for the partners and collaborators in a way to contribute for the improvement of the same, not only in the agent technologies, but also in functionality and ergonomics.

Also, in an idea and purposes validation goal for the AMIS project, was created the Information Repository – Scientific, Technological and Regional Development (RI-DCTR).

The RI-DCTR presents the same functionalities of the AMIS, added and adapted with other requirements for the DCTR project purpose. This information repository supports the activities of the PRAXIS project named DCTR (Scientific, Technological and Regional Development).

Through the accumulated experience in AMIS and RI-DCTR projects, the service was adapted to a universal platform to publish contents and resources that later served other projects such as the UUGuide (Universal Usability Guide), an ACM (Association form Computing Machinery) international project.

Índice

1	ENQUADRAMENTO E PROBLEMÁTICA: A EXPLOSÃO INFORMACIONAL NA INTERNET	21
1.1	NOTA INTRODUTÓRIA.....	21
1.2	A EXPLOSÃO INFORMACIONAL NA INTERNET	22
1.2.1	<i>O panorama global</i>	22
1.2.1.1	Número de hosts.....	22
1.2.1.2	Número de utilizadores.....	24
1.2.2	<i>Indicadores da Internet na união europeia</i>	25
1.2.2.1	A iniciativa eEurope 2002.....	27
1.2.3	<i>Realidade da Internet em Portugal</i>	28
1.2.3.1	Número de domínios .pt.....	29
1.2.3.2	Utilizadores da Internet em Portugal	31
1.2.4	<i>Estratégias em curso para aumento da penetração da Internet em Portugal</i>	34
1.2.4.1	E-Democracia.....	35
1.2.5	<i>Internet nacional e escolas</i>	36
1.3	PROBLEMÁTICA: A DISPONIBILIZAÇÃO E A LOCALIZAÇÃO DE INFORMAÇÃO NA INTERNET	37
1.3.1	<i>Os conteúdos</i>	38
1.3.2	<i>O problema de encontrar informação</i>	41
1.3.3	<i>A necessidade de credibilizar a informação</i>	41
1.3.4	<i>Conclusões</i>	42
2	UMA PROPOSTA DE ABORDAGEM AOS PROBLEMAS DA PUBLICAÇÃO DE DOCUMENTOS NA INTERNET	45
2.1	SERVIÇO DE CATALOGAÇÃO DE MATERIAL DIDÁCTICO.....	45
2.2	SERVIÇO DE INTERLIGAÇÃO AUTOMÁTICA ENTRE PÁGINAS.....	45
3	ESPECIFICAÇÃO	47
3.1	UTILIZAÇÃO DO SISTEMA.....	47
3.1.1	<i>Documentação da forma de utilização do sistema</i>	47
3.1.2	<i>Base de dados do sistema</i>	47
3.1.3	<i>Actores envolvidos no Serviço AMIS</i>	48

3.1.3.1	Utilizadores anónimos	48
3.1.3.2	Autores	49
3.1.3.3	Administradores	49
3.1.4	<i>Criação e publicação de documentos</i>	50
3.1.5	<i>Classificação de documentos publicados</i>	51
3.1.6	<i>Interligação de documentos publicados</i>	51
3.1.7	<i>Pesquisa de documentos</i>	52
3.2	ADMINISTRAÇÃO DO SISTEMA.....	52
3.2.1	<i>Gestão de contas de acesso</i>	53
3.2.2	<i>Gestão de documentos</i>	53
3.3	ESTRUTURA DO REPOSITÓRIO DE INFORMAÇÃO DO SERVIÇO AMIS	53
3.3.1	<i>Documentos</i>	54
3.3.2	<i>Ligações</i>	55
3.3.3	<i>Eventos</i>	55
3.3.4	<i>Classes</i>	56
3.3.5	<i>Portal AMIS</i>	57
3.3.5.1	<i>Lista de classes</i>	57
3.3.5.2	<i>Placard informativo</i>	57
3.3.5.3	<i>Menu de opções</i>	57
3.3.6	<i>Pesquisas</i>	59
4	IMPLEMENTAÇÃO	61
4.1	REQUISITOS DE HARDWARE DO SISTEMA	61
4.1.1	<i>Uma máquina destinada à função de servidor de informação ligada à Internet.</i>	61
4.2	REQUISITOS DE SOFTWARE DO SISTEMA	61
4.2.1	<i>Servidor de Web Apache</i>	61
4.2.2	<i>· Servidor de base de dados MySQL</i>	61
4.2.3	<i>· Linguagem de programação de páginas dinâmicas PHP</i>	61
4.2.4	<i>· Software de gestão de versões de software CVS</i>	62
4.2.5	<i>· Software de análise e processamento de documentos de HTML - SGMLS</i>	62
4.2.6	<i>· Software de gestão de projectos - Keystone</i>	63
4.3	CAPACIDADES IMPLEMENTADAS	63
4.3.1	<i>Base de dados do sistema</i>	63

4.3.2	<i>Páginas de acesso ao sistema</i>	64
4.3.2.1	Utilização de cookies para utilizadores registados.....	65
4.3.3	<i>Registo do administrador do sistema</i>	66
4.3.4	<i>Registo de novos autores</i>	66
4.3.5	<i>Aprovação da inscrição de um novo autor</i>	67
4.3.6	<i>Publicação de conteúdos</i>	68
4.3.6.1	Programas de publicação de conteúdos.....	68
4.3.6.2	O Protocolo HTTP	69
4.3.6.2.1	HTTP 1.0.....	70
4.3.6.2.1.1	Primitiva GET.....	70
4.3.6.2.1.2	Primitiva HEAD	71
4.3.6.2.1.3	Primitiva POST.....	71
4.3.6.2.2	HTTP 1.1	71
4.3.6.2.2.1	Primitiva OPTIONS	71
4.3.6.2.2.2	Primitiva TRACE.....	71
4.3.6.2.2.3	Primitiva PUT.....	72
4.3.6.2.2.4	Primitiva DELETE	72
4.3.6.3	Ferramentas de composição e publicação de documentos em servidores Web	72
4.3.6.4	Criação de pasta.....	73
4.3.6.5	Publicação de conteúdos.....	74
4.3.6.5.1	Publicação através do Netscape Composer.....	75
4.3.6.6	Classificação de conteúdos	76
4.3.6.6.1	Classificação de conteúdo por expressões chave.....	76
4.3.6.6.2	Classificação temática de conteúdos.....	76
4.3.7	<i>Criação de ligações</i>	78
4.3.8	<i>Criação de eventos</i>	78
4.3.9	<i>Notificação da aprovação de um autor</i>	79
4.3.10	<i>Interligação de um documento com outros documentos publicados</i>	79
4.3.10.1	Interligar documentos.....	79
4.3.10.2	Adicionar ligações aos documentos.....	79
4.3.11	<i>Alteração dos dados de um autor</i>	80
4.3.12	<i>Alteração das expressões chave de um documento</i>	80

4.3.13	<i>Classificação temática de ligações e eventos</i>	80
4.3.14	<i>Pesquisas</i>	80
4.3.15	<i>Apagar documentos</i>	81
4.3.16	<i>Alterar e apagar ligações</i>	81
4.3.17	<i>Alterar e apagar eventos</i>	81
4.3.18	<i>Alterar e apagar classes temáticas</i>	81
4.3.19	<i>Placard informativo</i>	81
4.4	CAPACIDADES A IMPLEMENTAR.....	82
4.4.1	<i>Controlo do acesso aos documentos</i>	82
4.5	CAPACIDADES EM ESTUDO E DESENVOLVIMENTO.....	82
4.5.1	CSS	82
4.5.1.1	Introdução.....	82
4.5.1.2	Exemplos do uso de CSSs	85
4.5.1.3	Editor de css	87
4.5.2	<i>HtDig – Motor de indexação e pesquisa de conteúdo estático</i>	87
4.5.2.1	Observações de motores de indexação.....	87
4.5.2.2	Descrição	88
4.5.2.3	Características gerais	88
4.5.2.4	Configuração.....	89
4.5.2.5	Conclusões	91
5	VALIDAÇÃO DO PROJECTO AMIS.....	93
5.1	REPOSITÓRIO DE INFORMAÇÃO DO DCTR.....	93
5.1.1	<i>Enquadramento do sistema de informação no projecto DCTR</i>	93
5.1.2	<i>Objectivos do RI-DCTR</i>	95
5.2	UNIVERSAL USABILITY GUIDE.....	95
5.2.1	<i>Enquadramento do sistema de informação no projecto UUGuide</i>	95
5.2.2	<i>Participação no projecto UUGuide</i>	96
6	POSSÍVEIS FUTURAS IMPLEMENTAÇÕES DO PROJECTO AMIS	99
6.1	NOTA INTRODUTÓRIA.....	99
6.2	CARACTERIZAÇÃO DOS AGENTES DE SOFTWARE.....	99
6.2.1	<i>Definição e especificação de agentes</i>	99
6.2.1.1	Definição de agente	99

6.2.1.2	Agência	100
6.2.1.3	Inteligência	101
6.2.1.4	Agentes de Software	101
6.2.1.4.1	Agentes Inteligentes para Interfaces com utilizador	101
6.2.1.4.2	Agentes Distribuídos.....	102
6.2.1.4.3	Agentes Móveis.....	103
6.2.2	<i>Aplicação de Agentes Inteligentes para Internet.....</i>	<i>103</i>
6.2.2.1	Agentes de gestão de redes	103
6.2.2.2	Agentes de filtragem de informação.....	104
6.2.2.3	Agentes para acesso à informação	105
6.2.2.3.1	Exemplos de Agentes de Filtragem de Informação	107
6.2.2.3.1.1	NetAttaché Pro	107
6.2.2.3.1.2	GoAgent!.....	107
6.2.2.3.2	Agentes para assistência à realização tarefas conhecidas	108
6.2.2.3.2.1	Microsoft Bob	108
6.2.2.3.2.2	Open Sesame!.....	109
6.2.2.3.3	Outros exemplos de agentes	109
6.2.2.3.3.1	Newbot.....	109
6.2.2.3.3.2	Jango	110
6.2.2.3.3.3	Firefly.....	110
6.2.2.3.3.4	WBI (Web Browser Intelligence)	110
6.2.2.3.3.5	Roboshopper	110
6.2.2.3.3.6	BargainFinder	111
6.2.2.3.3.7	BottomDollar	111
6.2.3	<i>Uso de agentes em Classificação, Indexação, Rastreo e Interligação de Diversas Fontes de Informação.</i>	<i>111</i>
6.2.3.1	Panorama actual	111
6.2.3.2	Problemas relacionados com as pesquisas de informação (quem a procura)	112
6.2.3.3	Motores de Busca e Agentes	113
6.3	PERSONALIZAÇÃO DO SERVIÇO PARA UTILIZADORES	114
6.3.1	<i>Registo de gostos, experiências e ações comuns dos utilizadores do serviço AMIS.....</i>	<i>115</i>
6.3.2	<i>Cookies como ferramenta de reconhecimento de utilizadores do serviço AMIS.</i>	<i>117</i>

6.4	XML – O GRANDE ÍCONE DA INFORMAÇÃO NA INTERNET.....	117
6.4.1	<i>Resumo</i>	117
6.4.1.1	Linguagens de marcas e linguagens de programação.....	118
6.4.2	<i>Evolução do XML</i>	118
6.4.2.1	História e Objectivos do XML.....	119
6.4.2.2	SGML: A mãe de todas as linguagens de marcas.....	120
6.4.2.3	XML versus SGML.....	120
6.4.3	<i>Descrição</i>	121
6.4.3.1	Vantagens e funcionalidades.....	121
6.4.3.1.1	A importância da extensibilidade.....	121
6.4.3.1.2	Minimização de marcas.....	122
6.4.3.1.3	Dois formatos de especificação de documentos.....	123
6.4.3.1.4	Conversão HTML/XML.....	124
6.4.3.1.5	Extensões em XML: Estilos e Links.....	125
6.4.3.1.5.1	Estilos em XML.....	125
6.4.3.1.5.2	Links em XML.....	125
6.4.3.2	Manipulação e gestão de documentos e conteúdo.....	126
6.4.3.2.1	Um exemplo simples de manutenção e gestão de documentos em XML 126	
6.4.3.2.2	A estrutura física.....	129
6.4.3.3	Exemplos de consolidação de ideias.....	131
6.4.3.3.1	MoDL.....	131
6.4.3.3.2	Generic XML Parser Class para PHP3.....	133
6.4.4	<i>Aplicações</i>	137
6.4.4.1	XML-RPC.....	137
6.4.4.1.1	Especificação.....	137
6.4.4.1.2	Vantagens e Objectivos.....	139
6.4.4.2	ICE.....	139
6.4.4.2.1	Introdução.....	139
6.4.4.2.2	O Problema.....	140
6.4.4.2.3	Sindicalização em ICE.....	140
6.4.4.2.4	Especificação (briefing).....	141
6.4.4.2.5	Conclusão.....	142

6.4.4.3	RDF.....	143
6.4.5	<i>Investigação e Desenvolvimento</i>	144
6.4.5.1	Federações de Agentes.....	144
6.4.5.2	XML/EDI.....	144
6.4.6	<i>Conclusões</i>	145
6.5	FEDERAÇÃO DE REPOSITÓRIOS E SISTEMA DE PESQUISA DISTRIBUÍDA.....	146
6.5.1	<i>Utilização de XML-RPC para pesquisas numa federação de repositórios</i>	147
6.6	METABASE PHP PACKAGE – INSTALAÇÃO, ACESSO E GESTÃO INDEPENDENTE DE SGBDS.....	148
6.6.1	<i>Introdução</i>	148
6.6.1.1	O que é o METABASE?	148
6.6.1.2	Objectivos	148
6.6.1.3	Estrutura do pacote METABASE.....	149
6.6.1.3.1	Interface de programação independente dos SGBDs.....	149
6.6.1.3.2	Drivers específicos de SGBDs	149
6.6.1.3.3	Analisador (parser) de descrições de esquemas.....	149
6.6.1.3.4	Gestor de bases de dados	149
6.6.2	<i>Descrição</i>	150
6.6.2.1	Desenvolvimento de aplicações baseadas no modelo tree-tier	150
6.6.2.1.1	Sistemas de Gestão de Bases de Dados.....	150
6.6.2.2	Criação de uma base de dados com PHP/METABASE.....	151
6.6.2.2.1	Esquema da base de dados.....	151
6.6.2.2.2	Instalação da base de dados com o METABASE	155
6.6.2.3	Manipulação de dados	156
6.6.2.3.1	Configuração do acesso à base de dados	156
6.6.2.3.2	Execução de queries à base de dados	158
6.6.2.3.2.1	<i>Queries</i> directos.....	158
6.6.2.3.2.2	<i>Queries</i> preparados	158
6.6.2.3.3	Retorno de resultados dos queries	159
6.6.2.4	Updates à base de dados	160
6.6.2.5	Portabilidade de dados (Database Dumping)	160
6.6.2.6	METABASE versus ODBC.....	160
6.6.3	<i>Conclusão – Um exemplo: Sistemas de informação comerciais</i>	161

6.7	ADAPTAÇÃO DO PROJECTO AMIS PARA A PUBLICAÇÃO DE CONTEÚDOS NA UNIVERSIDADE DE AVEIRO	161
7	CONCLUSÕES	163
	APÊNDICES.....	165
	A. ESQUEMA UML DA BASE DE DADOS DO SISTEMA.....	165
	B. PARTICIPAÇÃO NA III LUSOCOM NO ÂMBITO DO PROJECTO AMIS	167
	B.1 Abstracto da comunicação apresentada no âmbito do projecto AMIS - A problemática da divulgação da cultura científica usando os recursos proporcionados pela Internet e pelas redes telemáticas.....	167
	ACRÓNIMOS	169
	LISTA DE FIGURAS.....	171
	REFERÊNCIAS	173

1 Enquadramento e Problemática: a explosão informacional na Internet

1.1 Nota introdutória

A Internet surge através de um projecto criado em 1969 pelo departamento de defesa dos Estados Unidos da América. Esse projecto consistia no desenvolvimento duma rede segura para transmissão de informação entre instituições norte-americanas que trabalhavam no desenvolvimento e fornecimento de recursos militares, rede que abrangia também algumas universidades [1].

Na altura em que foi criada, designada por ARPANET¹, era constituída por 3 computadores, mas, com a adesão de grande parte das universidades norte-americanas, a rede cresceu rapidamente e de tal modo que teve de ser dividida em duas redes menores, separando os computadores das instituições militares e os computadores das instituições não militares.

A partir de 1983, após um crescimento contínuo da Internet, surge a necessidade de se reestruturar todos os mecanismos de coordenação, através do desenvolvimento na parte de engenharia da rede. Nesta fase, os serviços disponíveis eram escassos, tais como o *e-mail* e *newsgroups* [2]. À medida que os anos avançam, a tecnologia evolui, os custos de *hardware* e *software* baixam e cada vez mais países aderem à rede das redes, inicialmente em instituições de carácter académico.

A rede das redes sofre outro crescimento considerável quando em 1991 o IRC² ganha alguma fama nos acontecimentos históricos da Guerra do Golfo Pérsico. Através do IRC, muitas pessoas puderam saber detalhes do decorrer dos acontecimentos que eram relatados por pessoas que estavam a segui-los de muito perto. Os relatos chegavam no momento em que os acontecimentos estavam a suceder e, frequentemente, mais depressa do que através dos serviços noticiosos tradicionais [1].

Com o aparecimento da *World Wide Web* em 1992, inicia-se a grande difusão e divulgação da Internet. Através de um *browser*, um utilizador poderia consultar páginas de informação e

¹ *Advanced Research Projects Administration Network*

² *Internet Relay Chat*. Este serviço permite a conversação, em directo, de duas ou mais pessoas.

navegar através das mesmas graças a hiperligações nos documentos, que permitiam consultar outras páginas de uma forma dinâmica. Surge a necessidade de se criarem infra-estruturas e métodos de promoção da mesma e Portugal não é excepção. Na secção seguinte, é possível observar alguns gráficos que provam esse crescimento a nível global, europeu e nacional.

1.2 A explosão informacional na Internet

1.2.1 O panorama global

Nas secções seguintes, são apresentados alguns resultados de censos e estatísticas relacionadas com a Internet, desde máquinas registadas ao número de utilizadores. A ideia é a de tentar apresentar, com alguma exactidão, a expansão global que a Internet está a ter.

1.2.1.1 Número de hosts

Através de alguns dados específicos, foi realizado um estudo relativo ao crescimento “incontrolável” de conteúdos na Internet. Como é possível verificar pelos últimos resultados da *Networks Wizards*³, de Julho de 1999 a Julho de 2000, o número de *hosts*⁴ aumentou em mais de 80%, de 56 milhões para 93 milhões [3] (Fig. 1). De um modo geral, este aumento reflecte, não só, maior consciencialização e preocupação para com as novas tecnologias, mas também tecnologia a preços mais baixos e, um dos factores mais importantes, um grande aumento de conteúdos.

Date	Survey Host Count	Adjusted Host Count	Replied To Ping*	
Jul 2000	93,047,785		-	
Jan 2000	72,398,092		-	
Jul 1999	56,218,000		-	
Jan 1999	43,230,000		8,426,000	
Jul 1998	36,739,000		6,529,000	
Jan 1998	29,670,000		5,331,640	[first NEW Survey]
Jul 1997	19,540,000	26,053,000	4,314,410	[last OLD Survey]
Jan 1997	16,146,000	21,819,000	3,392,000	
Jul 1996	12,881,000	16,729,000	2,569,000	
Jan 1996	9,472,000	14,352,000	1,682,000	
Jul 1995	6,642,000	8,200,000	1,149,000	
Jan 1995	4,852,000	5,846,000	970,000	

³ Ver em <http://www.nw.com/>.

⁴ No caso presente, podemos definir um *Host*, como um computador registado num DNS (servidor de nomes de domínios).

Jul 1994		3,212,000		707,000
Jan 1994		2,217,000		576,000
Jul 1993		1,776,000		464,000
Jan 1993		1,313,000		

Fig. 1 - Fonte *Network Wizard* – número de *hosts* registados em DNS.

O gráfico seguinte, demonstra o crescimento exponencial de *hosts* de Janeiro de 1991 a Julho de 2000 (Fig. 2). Cada *host*, implica uma máquina ligada à Internet, registada num servidor DNS⁵. A grande maioria destas máquinas são *sites* na *Web* que alojam serviços e, por sua vez, disponibilizam conteúdos.

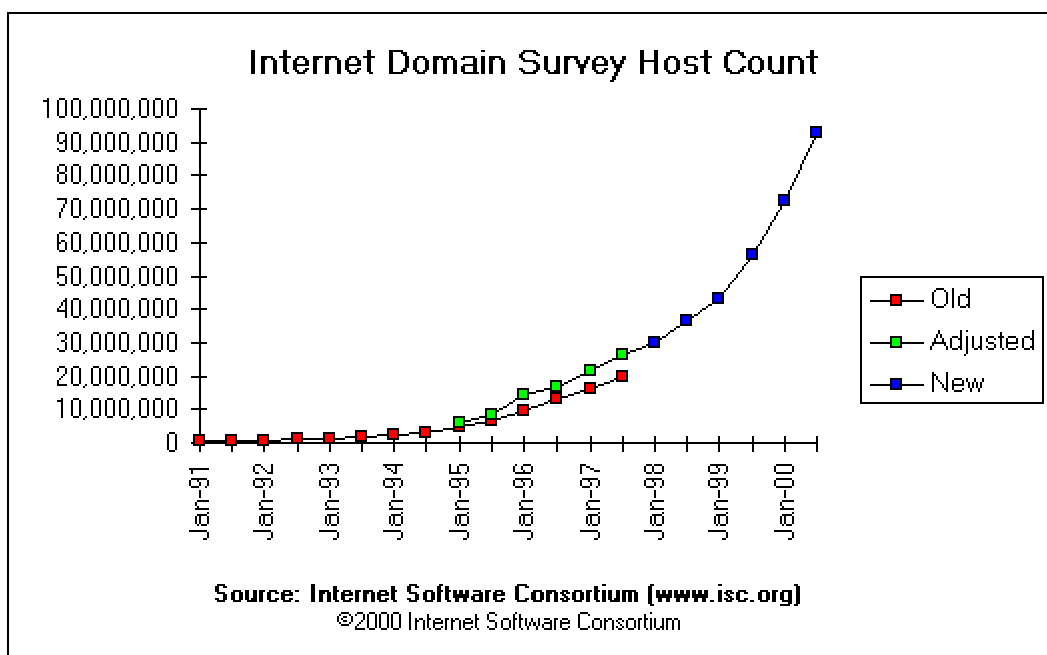


Fig. 2 - Fonte *Network Wizard* – contador de *hosts* registados de Janeiro de 1991 a Julho de 2000.

Como já foi referido, os *hosts* que disponibilizam maiores quantidades de conteúdos, são aqueles que disponibilizam os seus serviços na *Web*. Através de uma estatística da *netcraft*⁶, verificamos que dos 93 milhões de *hosts*, cerca de 27.6 milhões de *hosts* (29,6%) são utilizados para uso de servidores *Web*, ou seja, estas máquinas disponibilizam serviços e conteúdos em ambiente *Web*, por sua vez, mais requisitados e usados pelos utilizadores [4] (Fig. 3).

Server	December 2000	Percent	January 2001	Percent	Change
--------	---------------	---------	--------------	---------	--------

⁵ Domain Name Server. Servidor que reconhece o nome de uma máquina (*minha.maquina.pt*) em função do IP (192.168.0.99) ou vice-versa.

⁶ Ver em <http://www.netcraft.com/>.

Apache	15414726	60.04	16207982	58.75	-1.29
Microsoft-IIS	5025017	19.57	5901507	21.39	1.82
Netscape-Enterprise	1682737	6.55	1733097	6.28	-0.27
WebLogic	890791	3.47	1004571	3.64	0.17
Zeus	676526	2.63	693684	2.51	-0.12
Rapidsite	365807	1.42	371441	1.35	-0.07
Thttpd	321944	1.25	343172	1.24	-0.01
tigershark	139300	0.54	150937	0.55	0.01
AOLserver	125513	0.49	127980	0.46	-0.03
WebSitePro	110681	0.43	113480	0.41	-0.02

Fig. 3 - Fonte Netcraft - Estatística de servidores Web de Dezembro de 2000 a Janeiro de 2001.

Somando a totalidade de servidores *Web*, independentemente das preferências tecnológicas de cada um, que entre Dezembro de 2000 a Janeiro de 2001 deu-se um aumento na ordem dos 3 milhões de novos servidores Web, implicando 3 milhões de *websites*.

1.2.1.2 Número de utilizadores

Segundo o último censo da *CommerceNet*⁷, o número de utilizadores da Internet ascende aos 242 milhões (Fig. 4).

January 2000	
Africa	2.1 Million
Asia/Pacific	40 Million
Europe	70 Million
Middle East	1.9 Million
Canada & USA	120 Million
South America	8 Million
World Total	242 Million

Fig. 4 - População Global na Internet pelo centro de pesquisa da *CommerceNet*.

O que não deixa de ser, de certa forma, preocupante são os índices de população na Internet em áreas onde se agrupam os países de terceiro mundo (África e América do Sul) onde, comparativamente com a Europa, os índices populacionais são superiores em larga escala mas,

⁷ Ver em <http://www.commerce.net/>.

no entanto, a população europeia na Internet, ultrapassa “avassaladoramente” a população sul americana e africana. Penso que, a disparidade de utilizadores, se prende com as carências sócio-económicas que estas áreas do globo demonstram ter. De algum modo, os países industrializados têm conseguido transmitir a importância que as novas tecnologias, nomeadamente a Internet, apresentam.

Segundo dados oficiais da *CommerceNet* [5], prevêem-se 349 milhões de utilizadores em meados do ano 2000, 490 milhões no fim do ano de 2002 e 765 milhões no fim de 2005. Prevê-se um crescimento de 27% para Estados Unidos e Canadá, 20% para a Europa e 422% para a Ásia e Pacífico. Não são avançadas estimativas para África, Médio Oriente e América do Sul.

1.2.2 Indicadores da Internet na união europeia

Através de um estudo realizado pela *Fischer & Lorenz*⁸ para a união europeia, o crescimento do uso da Internet continuará a aumentar, de uma maneira estável, embora, esse crescimento seja menor nos países da Europa Sul. Alguns investigadores apontam estas diferenças como resultado de diferenças culturais, linguísticas e infra-estruturas de telecomunicações, sem dar prioridade a alguma das diferenças mencionadas. À medida que o número de utilizadores aumenta, crescerá também a penetração de infra-estruturas, com vantagens de carácter social, político e económico e, as fronteiras nacionais terão cada vez menor importância devido ao efeito “fim-de-distância” provocado pela Internet [6].

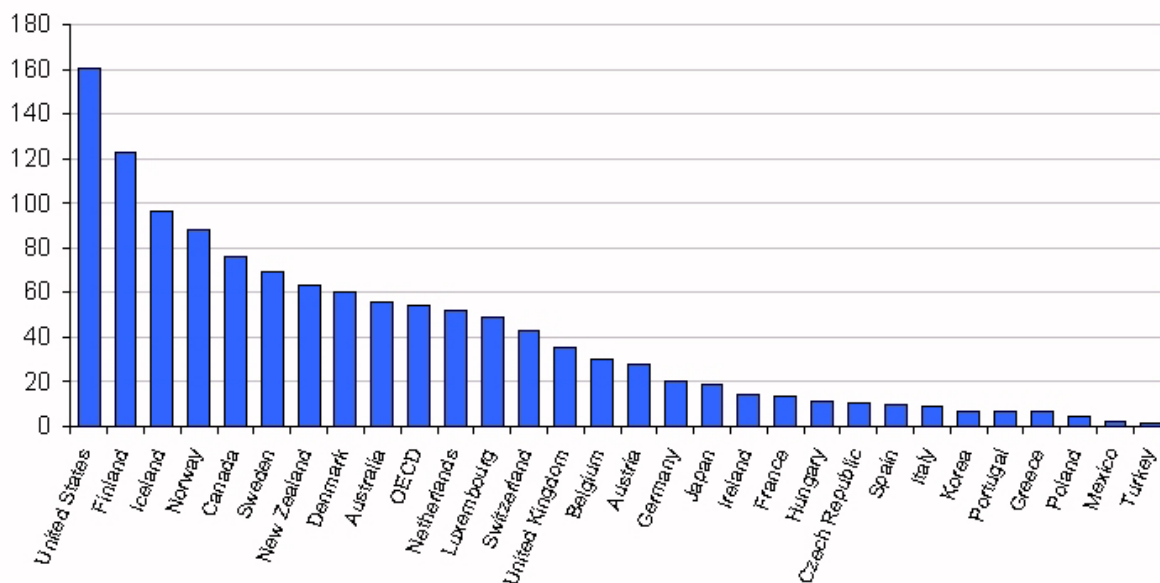
O mesmo estudo adianta que as novas tecnologias endereçarão exigências de capacidade, flexibilidade e mobilidade. Os serviços básicos de acesso à Internet como *e-mail* navegação na *web* tornar-se-ão uma comodidade. O mercado irá forçar a queda dos preços de ligação aos ISP, surgindo novos modelos e tarifários.

Comparativamente aos países da união europeia e restantes membros da OECD⁹, Portugal apresenta-se no fim da tabela o que, de certa forma, não deixa de ser um sinal preocupante quando comparado com outros países populacionalmente pequenos como, por exemplo, a Holanda, Luxemburgo e Bélgica (Fig. 5) [7].

⁸ Disponível em <http://www.fl.dk/>.

⁹ *Organization for Economic Co-Operation and Development*, conhecida em português por OCDE. Ver em <http://www.oecd.org/>.

Chart 2. Internet hosts per 1000 inhabitants, in the OECD area, Sept 1999

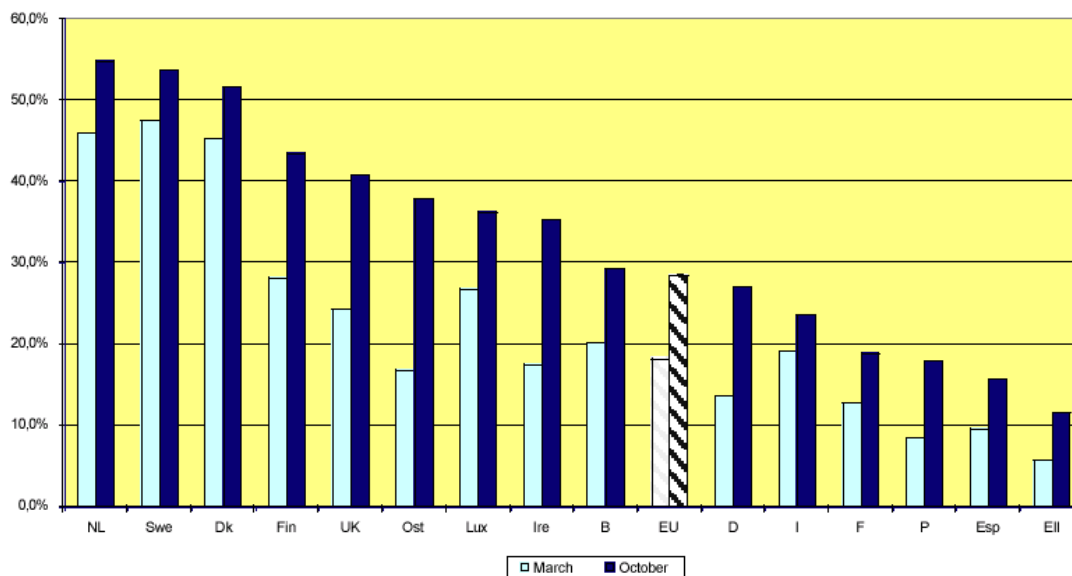


Source: OECD based on Telcordia Technologies (www.netsizer.com)

Fig. 5 - Número de *hosts* por 1000 habitantes na zona OCDE.

Relativamente à penetração da Internet nos lares da União Europeia, Portugal é um dos cinco estados membros, juntamente com a Áustria, Irlanda, Grécia e Alemanha, que demonstraram ter duplicado os índices de penetração da Internet (Fig. 6).

Internet Penetration in EU homes
March- October 2000



Fonte: Comissão Europeia, *Eurobarometer* Março – Outubro 2000

http://europa.eu.int/comm/information_society/europe/documentation/update/surv102000/

Fig. 6 - Penetração da Internet em lares da união Europeia.

A média europeia estima-se por 28.4% dos lares da União Europeia ligados à Internet. A estimativa de Março de 2000 cifrava-se em 18.3% [8].

1.2.2.1 A iniciativa eEurope 2002

Em Dezembro de 1999, a Comissão Europeia lançou a iniciativa *eEurope*¹⁰, com os seguintes objectivos essenciais:

- Colocar todos os europeus na era digital e em linha;
- Criar uma Europa digitalmente instruída, apoiada por uma cultura empresarial;
- Assegurar que o processo seja socialmente abrangente e ganhe a confiança dos consumidores.

Em Junho de 2000, foi adoptado o Plano de Acção *eEurope 2002* pelo Conselho Europeu da Feira. Especificava as medidas necessárias para alcançar esses objectivos até 2002. O Conselho Europeu solicitou também a apresentação de um relatório de progresso ao Conselho Europeu de Nice. O presente relatório contém a actualização da *eEurope* em resposta a essa solicitação. O relatório abrange os domínios que são da responsabilidade da Comissão Europeia e de outros protagonistas a nível europeu (por exemplo, o Conselho, o Parlamento Europeu ou o BEI) [9].

Como principais medidas para promover a internet na união europeia, a *eEurope* definiu medidas específicas:

- Cartões inteligentes, domínio em que as lacunas em termos de normas e aplicações comuns impedem o pleno desenvolvimento de soluções. A indústria assumiu os desafios definidos na *Cimeira dos cartões inteligentes*, realizada em Lisboa em Abril de 2000. Em 18 e 19 de Setembro de 2000, realizou-se em Atenas uma reunião subsequente em que se estabeleceram doze grupos de trabalho, liderados pela indústria, responsáveis pelos diversos objectivos neste domínio e uma *task force* para coordenar os trabalhos.
- *eContent*, relativamente ao qual a Comissão propôs um novo programa de 150 milhões de euros destinado a estimular o desenvolvimento e a utilização de conteúdos digitais europeus na Internet e a promover a diversidade linguística nos sítios *Web* europeus. O seu objectivo é ultrapassar as barreiras que impedem os produtores europeus de

¹⁰ Ver em http://europa.eu.int/comm/information_society/eeurope/.

conteúdos de explorarem plenamente as suas possibilidades na Internet a nível mundial. O programa deverá ser adoptado pelo Conselho ‘Telecomunicações’, em Dezembro de 2000.

- Educação, domínio em que a iniciativa *eLearning* e o reforço das acções do programa IST contribuirão para adaptar o sistema educativo à nova economia. Os recursos nacionais terão o apoio dos instrumentos comunitários adequados e das parcerias criadas entre as autoridades públicas e a indústria. A Comissão apresentará uma proposta pormenorizada de medidas de execução da iniciativa *eLearning* ao Conselho ‘Educação’, em 15 de Novembro de 2000.
- Redes de investigação: Com o lançamento do projecto *Géant*, foram dados os primeiros passos de um plano geral destinado a fortalecer a Europa como parceiro a nível mundial no que diz respeito a conectividade. O projecto garantirá a modernização das interligações entre as redes de investigação da Europa, que passarão a funcionar a 10 Gigabit/s no final de 2001, fornecendo assim aos investigadores europeus as ferramentas necessárias para trabalharem em conjunto interactivamente numa vasta gama de domínios de investigação.
- Fundos regionais, domínio em que a sociedade da informação tem sido integrada como uma prioridade por todos os Estados-Membros no novo período de programação 2000-2006, reflectindo assim as orientações da Comissão. Além disso, eEurope Regio será um dos três temas da nova geração de acções inovadoras (que implicam um montante total de 400 milhões de euros entre 2000 e 2006) no âmbito do FEDER.
- Lançamento do .eu: A organização ICANN¹¹ aceitou, em princípio, delegar a gestão do código .eu na União Europeia. A Comissão apresentará, antes do final de 2000, uma proposta formal para a criação do domínio de topo .eu. O Registo .eu será criado logo que se chegue a acordo com o Parlamento Europeu e o Conselho [10].

1.2.3 Realidade da Internet em Portugal

Seguindo o exemplo global e europeu, Portugal, de ano para ano, tem-se vindo a afirmar, não só em relação ao número de utilizadores, mas também em relação ao número de domínios registados .pt que, por sua vez, implicam um crescimento relativo à produção de conteúdos.

¹¹ Sociedade Internet para a Atribuição de Nomes e Números (*Internet Corporation for Assigned Names and Numbers*) - responsável, a nível mundial, pela gestão de nomes de domínio.

1.2.3.1 Número de domínios .pt

A FCCN¹², entidade reguladora dos domínios .pt, apresentou um crescimento de mais de 50% de registos entre Janeiro de 2000 a Janeiro de 2001 (Fig. 7) [11].

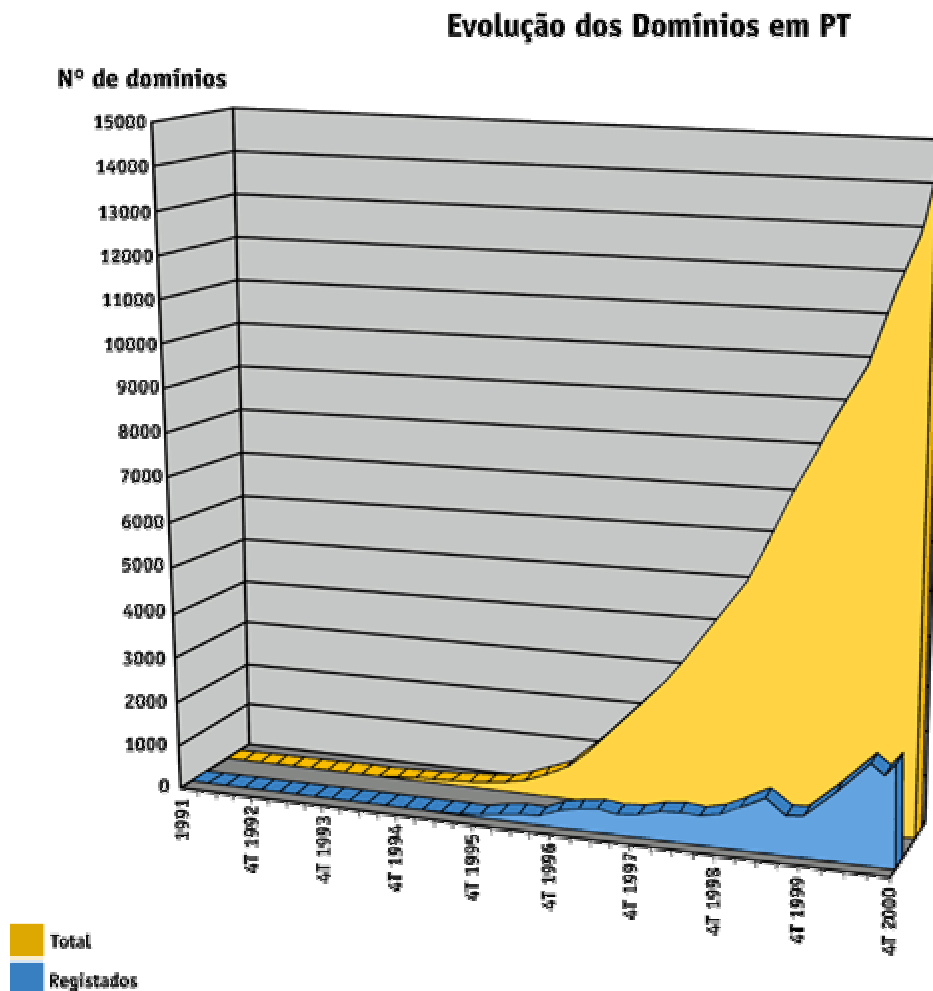


Fig. 7 - Evolução dos domínios em PT.

De salientar que, de maneira a captar e motivar mais entidades a registarem domínios .pt, entrou em vigor a 5 de Fevereiro de 2001, novos modelos de domínios (Fig. 8) [12]:

.net.pt	Poderão requerer subdomínios em .net.pt os prestadores de serviços de telecomunicações registados no Instituto das Comunicações de Portugal (ICP).
----------------	--

¹² Fundação para a Computação Científica Nacional. Ver em <http://www.fccn.pt/>.

.gov.pt	Poderão requerer subdomínios sob .gov.pt as entidades que integrem a estrutura do Governo da República Portuguesa.
.org.pt	Poderão requerer subdomínios em .org.pt todas as organizações sem fins lucrativos.
.edu.pt	Poderão requerer subdomínios sob .edu.pt os estabelecimentos de ensino públicos e os titulares de estabelecimentos de ensino privados ou cooperativos.
.int.pt	Poderão requerer subdomínios sob .int.pt todas as organizações internacionais e todas as representações diplomáticas devidamente registadas no Registo Nacional de Pessoas Colectivas (RNPC).
.publ.pt	Poderão requerer subdomínios sob .publ.pt os titulares de publicações periódicas registadas no Instituto da Comunicação Social (ICS).
.nome.pt	Poderão requerer subdomínios sob .nome.pt quaisquer pessoas singulares portadoras de Bilhete de Identidade emitido por Arquivo de Identificação português, bem como residentes em Portugal portadores de título de residência válido.
.com.pt	Qualquer pessoa/entidade poderá requerer subdomínios sob .com.pt.

Fig. 8 - Hierarquias PT.

Desde 5 de Fevereiro até 1 Março, 472 novos domínios, foram registados nas novas hierarquias (Fig. 9) [13]:

ANO DE 2001	
Hierarquias	Total de Activos
.com.pt	468
.edu.pt	1
.net.pt	1

.nome.pt	2
.pt	15353
Total de Domínios	15825

Fig. 9 - Total de domínios activos.

1.2.3.2 Utilizadores da Internet em Portugal

Segundo as previsões apresentadas pela FCCN para um crescimento do número de utilizadores de 23% para 35%, é comprovado com fonte na última sondagem efectuada pela Marktest¹³ em Portugal, a um universo de 7528382 indivíduos residentes em Portugal com 15 e mais anos.

No trimestre Setembro/Dezembro 2000, cerca de 32.3% dos inquiridos tinham acesso à Internet, 17.3% com acesso próprio em casa, 11.7% no local de trabalho, 8.9% na Escola/Universidade e em outros locais 5.1% [14].

No mesmo estudo, as intenções de ter acesso à Internet, 4.3% pensam vir a ter nos próximos 3 meses e 3.3% no espaço de 4 a 12 meses. De salientar, que 30.3% pensam vir a ter acesso, mas não prevêem data e 41.5% não pretendem ter acesso. Os inquiridos que não pretendem ter acesso à Internet alegam falta de interesse (8.1%), é cara ou não têm possibilidades (6.5%), não têm idade para isso (5.5%), não têm necessidade (5.4%), não sabem trabalhar (3.6%), material informático obsoleto (2.9%) e variados factores (4.7%).

Observando os valores dos inquiridos que não pretendem ter acesso à Internet, as maiores percentagens prendem-se com falta de conhecimentos e material informático e ISPs a preços inatingíveis. De alguma maneira, a promoção das novas tecnologias não depende só das intenções de cada um dos inquiridos, mas sim dos custos e conhecimentos que acarretam.

Em termos de tempo médio de utilização, 7.4% dos inquiridos dedica entre meia e uma hora. Entre uma e hora e meia 3.4% e mais de duas horas, 2.6%. Como serviço principal é referenciado o *e-mail* (14.5%), FTP (6.8%) e IRC ou *Chats* 6.5%. Não feita nenhuma referência a percentagens relativas à *Web*.

¹³ Ver em <http://www.marktest.pt/>.

A finalidade com que utilizam a Internet, 12% dos inquiridos refere para divertimento, 10.1% para fins profissionais, 10.0% para formação pessoal, 9.8% para fins académicos e 9.2% para obter notícias. Podemos concluir que, no que respeita à finalidade de utilização, o acesso é feito, em grande parte, a conteúdos informativos, académicos, profissionais e de carácter lúdico. Seguindo a mesma linha de raciocínio, 9.6% dos inquiridos procura notícias, 9.0% música, 7.8% cultura, 6.6% desporto, 6.0% cinema, 5.9% ciência, 5.7% informática, 5.3% viagens e lazer e 4.7% economia/finanças/negócios.

Por último, num inquérito à utilização das tecnologias de informação e comunicação pela população portuguesa, é apresentado o perfil dos utilizadores de computador em Portugal (Fig. 10) [15].

A – Perfil dos Utilizadores de Computador

A1- Utilizadores de computador por Nível de Escolaridade (%)

Nível de Escolaridade					
	Não sabe ler e escrever	1º Ciclo do ensino básico	2º Ciclo do ensino básico	3º ciclo e ensino secundário	Curso médio/superior
Utilizadores de computador		8	32	79	92

Percentagens em coluna

A2- Utilizadores de computador por Escalão Etário (%)

Escalão Etário						
	15-19 anos	20-24 anos	25-29 anos	30-39 anos	40-49 anos	> 50 anos
Utilizadores de computador	81	68	55	37	27	11

Percentagens em coluna

A3- Utilizadores de computador por Grandes Grupos de Profissões (%)

Grandes Grupos de Profissões									
	A	B	C	D	E	F	G	H	I
Utilizadores de computador	43	90	77	75	41	6	16	16	11

Percentagens em coluna

- A - quadros superiores da administração pública, quadros superiores e dirigentes de empresas
- B - especialistas das profissões intelectuais e científicas
- C - técnicos e profissionais de nível intermédio
- D - pessoal administrativo e similares
- E - pessoal dos serviços e vendedores
- F - agricultores e trabalhadores qualificados da agricultura e pescas
- G - operários, artífices, e trabalhadores similares
- H - operadores de instalações e máquinas e trabalhadores de montagem
- I - trabalhadores não qualificados

Fig. 10 - Perfil dos utilizadores de computador.

Como nota de conclusão, este estudo não refere a percentagem de utentes que utiliza a *World Wide Web* e, em relação à modalidade de acesso, refere apenas o modem (14.6%) e a RDIS (0.8%). Penso que seria importante a obtenção de valores relativos a outras modalidades de acesso, tais como, a televisão por cabo¹⁴ e ADSL¹⁵.

¹⁴ CATV, *Cable TV*. Permite acessos à Internet, através do mesmo cabo de televisão, na ordem dos 55Mbps em *downstream*, numa rede partilhada, e 3Mbps em *upstream*. Ver em <http://www.cable-modems.org/tutorial/03.htm>.

Com base nos valores apresentados, podemos concluir que a grande maioria dos *cibernautas* portugueses acede à Internet, principalmente, à procura de conteúdos de carácter informativo, académico ou profissional. À medida que o número de *sites*, portais e páginas da *Web* aumenta, torna-se necessário caracterizar a informação, o que implica a sua catalogação, classificação e indexação, de uma maneira uniforme, de modo a proporcionar aos potenciais utilizadores, e aos já existentes, serviços menos falíveis de procura, rastreio e classificação de informação.

1.2.4 Estratégias em curso para aumento da penetração da Internet em Portugal

Com vista a aumentar o número utilizadores da Internet em Portugal, o governo pretende dar curso a um conjunto de acções de incentivo económico e social.

A taxa de penetração da Internet em Portugal em 2003 deverá atingir 50%. Para tal, o governo vai promover o uso da Internet pelas famílias:

- Reduzir significativamente os custos de tráfego e oferecer sistemas de plafonamento desses custos. Dispor em 2003 de um sistema de acesso mínimo à Internet inteiramente gratuito (Internet de cidadania);
- Acelerar a introdução, competição e difusão de sistemas de banda larga (ADSL, Cablemodem, UMTS, etc.) em regime de *flat-rate*;
- Estimular a oferta competitiva de serviços destinados ao grande público;
- Desenvolver sistemas de interacção com os serviços públicos (*gnichet* digital em casa, serviços públicos *on-line*);
- Estender a Internet a todas as escolas do 1º ciclo (até final de 2001);
- Generalizar o recurso à Internet nos processos escolares de aprendizagem;
- Lançar diploma de competências básicas em TI (fase experimental no verão de 2000);
- Implantar sistema de certificação de competências em TI para fins profissionais;
- Generalizar formação e estímulo à escrita de páginas WWW por jovens;
- Estimular a criação de páginas pessoais com CV para efeitos de emprego. [16]

Além destas acções de incentivo directas, o governo pretende promover a Internet e novas tecnologias através outras medidas, não menos importantes:

¹⁵ *Asynchronous Digital Subscriber Line*. Permite acessos à Internet, através da rede telefónica, na ordem dos 8Mbps em *downstream* e 960Kbps em *upstream*. Ver em <http://www.icus.com/adsl/>.

- Aquisição de computadores pelas famílias;
- Uso da Internet em locais públicos;
- Uso da Internet pelas empresas;
- Uso da Internet pela administração pública e pelos cidadãos nas suas relações com o estado.

1.2.4.1 E-Democracia

O interesse no uso das novas tecnologias da informação, de forma a proporcionar um maior *input* e participação dos cidadãos na vida política tem aumentado com o crescimento do número de utilizadores da Internet. Hoje, 27 de Abril de 2001, numa pesquisa ao portal *Altavista* por *electronic democracy*, foram-me retornados 74.708 resultados de pesquisa, o que já é considerável.

A “democracia electrónica”, “teledemocracia” ou “democracia digital”, são termos usados regularmente como sinónimos para referir o uso das tecnologias de informação que permitem a interface entre políticos e cidadãos através de informação, votantes, votações e opinião [17].

Embora em Portugal este termo ainda não esteja divulgado de uma forma clara e consistente, são visíveis as várias iniciativas do Governo Português nesta matéria, mesmo antes do termo referido ser conhecido e divulgado. Um dos exemplos foi o lançamento da iniciativa “Livro Verde para a Sociedade da Informação em Portugal”¹⁶ em 1997. Este Livro Verde pretende dar cumprimento ao disposto na Resolução do Conselho de Ministros n.º 16/ 96 de 21/ 3/ 96, que cria a Missão para a Sociedade da Informação e estabelece um mandato para ‘promover um amplo debate nacional sobre o tema sociedade da informação, tendo em vista a elaboração de um Livro Verde que, nomeadamente, contenha propostas de medidas a curto, médio e longo prazos, a ser presente à Assembleia da República’ [18].

Em Portugal, é de assinalar, tal como já foi referido anteriormente, o interesse do poder político em informatizar os organismos e instituições administradas pelos mesmos. Veja-se o exemplo dos vários ministérios e institutos que se encontram disponíveis com serviços na Internet. Entre alguns exemplos, não menos importante e do interesse da sociedade civil, é a possibilidade da declaração dos impostos através da Internet na Direcção Geral de

¹⁶ Disponível em <http://www.mct.pt/sociedadedainformacao/>.

Contribuições e Impostos¹⁷. Através do portal da Administração Pública, Infocid¹⁸, que permite o acesso e esclarecimento quanto à declaração de impostos *online*, registo civil, registo comercial, certificados de admissibilidade e declarações para a segurança social, com cerca de 250 mil acessos de visualização mensais [19], podemos ter uma visão clara das vantagens que este serviço tem. Ainda podemos referir as colocações de professores no ensino estatal e os acessos ao ensino superior no *site* do Ministério da Educação¹⁹.

Daqui podemos concluir que as iniciativas que têm surgido do Governo Português têm tido grande aceitação da parte dos cidadãos e, pela adesão e acessos que têm tido, demonstram não só o interesse mas também a versatilidade em que estão disponíveis, poupando tempo e outros recursos.

1.2.5 Internet nacional e escolas

Com a introdução do acesso à Internet nas escolas, e em particular com a criação da Rede de Ciência, Tecnologia e Sociedade (RCTS²⁰) no âmbito da Iniciativa para a Sociedade de Informação em Portugal²¹, muitos alunos e professores passaram a ter a possibilidade de disponibilizar conteúdos de informação produzidos por si, de modo a poderem ser partilhados com outros utilizadores com acesso à Internet. A preocupação da integração de um novo ambiente educativo, social e processual na Sociedade da Informação tornou-se clara (ainda que não radicalmente prioritária). Exemplos como o Projecto Internet nas Escolas e o Programa Nónio XXI²².

A quantidade de materiais produzidos pelas escolas ou para as escolas, que tem vindo a ser disponibilizada através da Internet, é já considerável. A tendência é para que venha a aumentar cada vez mais. De salientar o empenho louvável prestado pelo Ministério da Ciência e Tecnologia²³, no que respeita a esta matéria. De acordo um estudo realizado pela Comissão

¹⁷ Disponível em <http://www.dgci.min-financas.pt/SiteDGCLnsf>.

¹⁸ Disponível em <http://www.infocid.pt/>.

¹⁹ Disponível em <http://www.min-edu.pt/>.

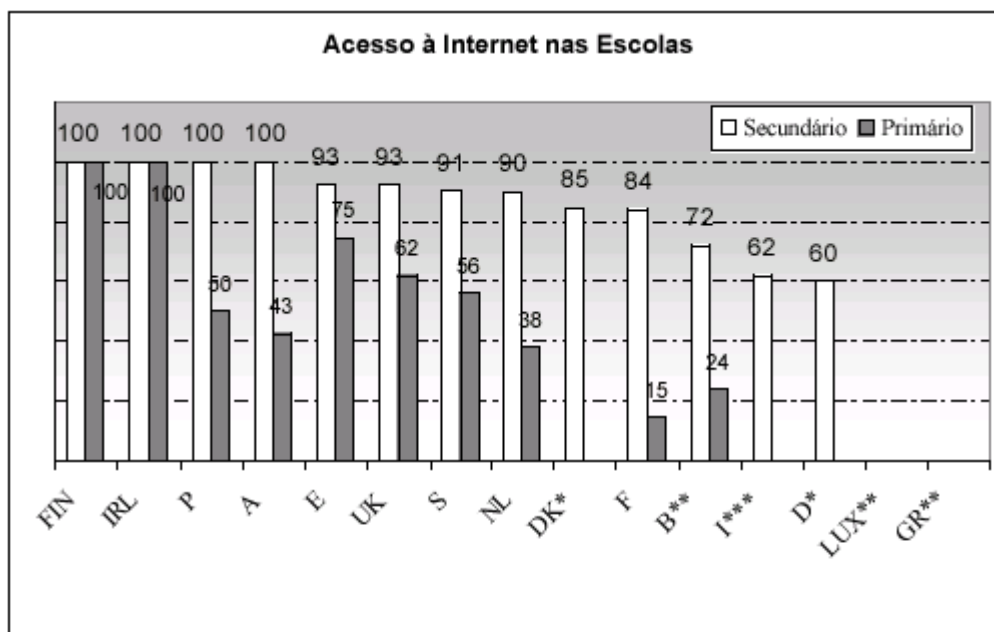
²⁰ Disponível em <http://www.fcen.pt/RCTS/index.shtml>.

²¹ Disponível em <http://www.mct.pt/PtSocInfo/indice.htm>.

²² Disponível em <http://www.dapp.min-edu.pt/nonio/nonio.htm>.

²³ Ver em <http://www.mct.pt/>.

Europeia, Portugal detém cerca de 50% das instituições primárias ligadas e 100% das instituições secundárias: (Fig. 11) [20]:



- * - Os dados apresentados foram agregados diversamente dos restantes países.
- ** - Os dados não incluem a comunidade francófona (Bélgica); não foram apresentados números ou percentagens totais.
- *** - Não foi feita uma distinção entre escolas primárias e secundárias.

Fig. 11 - Acesso à Internet nas escolas.

1.3 Problemática: a disponibilização e a localização de informação na Internet

À medida que a quantidade de material disponibilizado na *web* aumenta agudizam-se também os dois seguintes problemas:

- A capacidade de um disponibilizador de informação de tornar conhecidos os seus materiais de forma eficiente;
- A facilidade dos utilizadores interessados em encontrar esses materiais diminui.

Torna-se assim provável que:

- Grupos de utilizadores distintos acabem por disponibilizar materiais com âmbito idêntico por desconhcerem que o mesmo tipo de material já existia disponível na Internet mas tinha sido produzido por outros utilizadores.

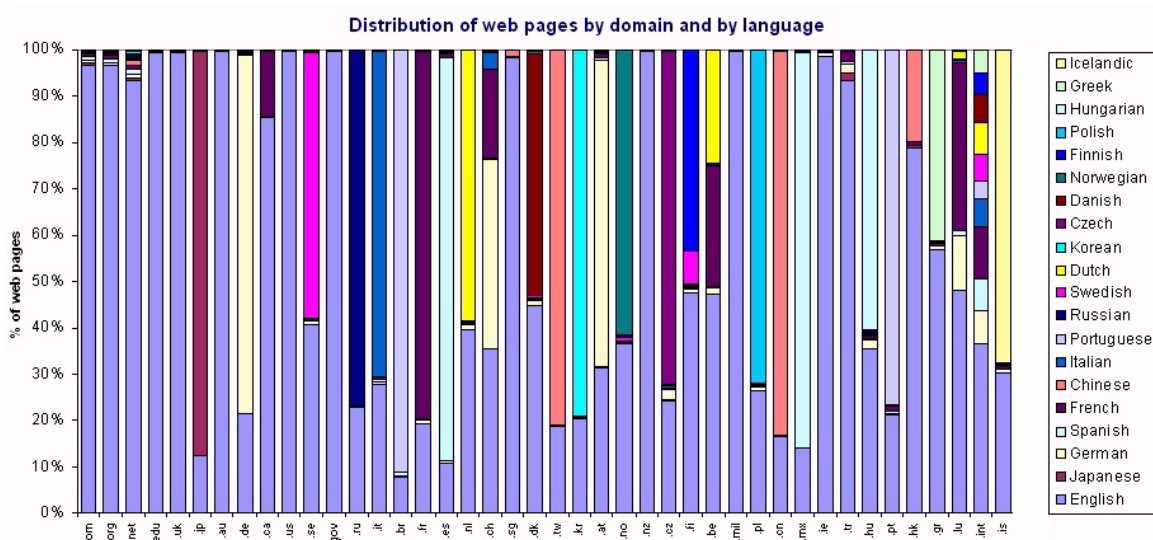
- Utilizadores interessados em determinados materiais acabem por nunca os encontrar como consequência de esses materiais aparecerem submersos por grandes quantidades de outra informação sem relevância para os fins em vista.

Deste modo, para além da eventual duplicação de esforços desnecessária por parte de quem está motivado para produzir material didáctico para disponibilizar através da Internet, os utilizadores interessados poderão continuar a não beneficiar do conhecimento da existência de várias recursos de material didáctico relacionado, dado que hoje em dia, mesmo os melhores serviços de pesquisa genérica existentes um pouco por toda a Internet, dificilmente são capazes de apresentar respostas úteis a este tipo de problemas.

1.3.1 Os conteúdos

É óbvia, para quem navega quotidianamente na Internet, a clara falta de conteúdos temáticos na *web* em língua portuguesa; é ainda problemática a falta de confiança a que os investigadores e académicos portugueses e estrangeiros votam os *publishers* e revistas *on-line*; é dramática a forma como ainda estão desorganizados e dispersos os sites de informação potencialmente importante para um crescimento social da presença portuguesa na *World Wide Web*.

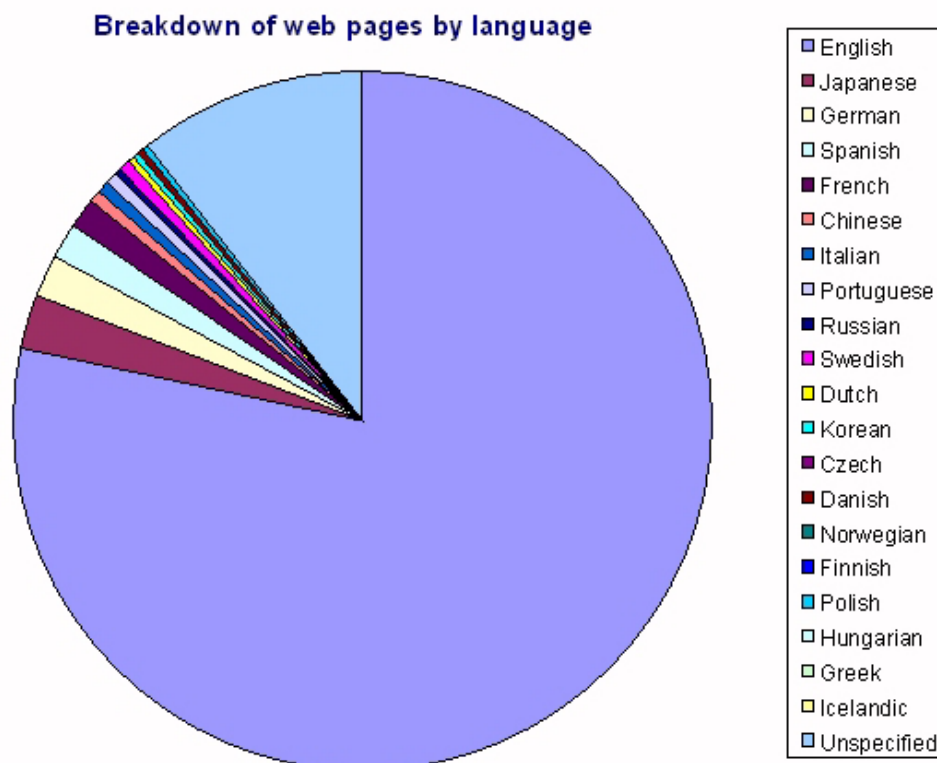
Através de um estudo efectuado pela OCDE, relativo aos conteúdos por língua, é de notar a grande importância que os domínios .pt, .br e .int detém nos conteúdos em língua portuguesa (Fig. 12) [21].



Source: OECD

Fig. 12 - Distribuição de páginas *Web* por domínio e língua.

Do mesmo modo, no presente estudo e, apesar de não serem reveladas percentagens específicas, observamos que a quantidade díspar de documentos em língua inglesa, comparativamente aos conteúdos em português (Fig. 13).

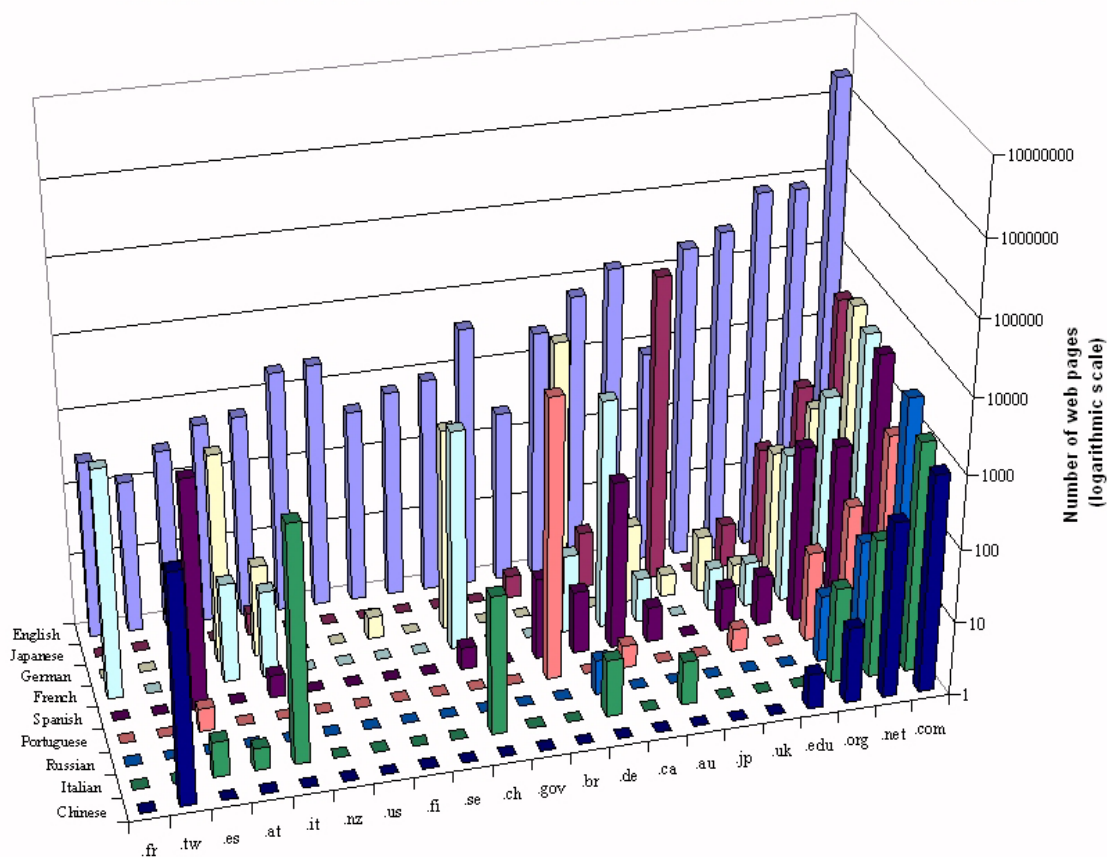


Source: OECD

Fig. 13 - Páginas Web por linguagem.

Através da Fig. 14, poderemos ter uma ideia mais clara, e não menos global, relativa aos conteúdos disponíveis na Internet em língua portuguesa. De salientar, como é óbvio pela densidade populacional e área geográfica, o domínio .br apresenta uma quantidade considerável de páginas e superior aos domínios .com, .net e .org. Outros domínios a referir, pela sua proporção populacional, são o .es (Espanha), provavelmente pela proximidade geográfica, e ainda os domínios .ca (Canadá) e .uk (Reino Unido). O domínio .pt não é referido. Embora apresente uma ideia global de conteúdos lusófonos, a Fig. 14, não nos permite ter uma ideia mais específica e exacta relativa à percentagem de conteúdos em países como a França (.fr), Grécia (.gr) ou Itália (.it).

Breakdown of web pages linking to secure servers by select domains and languages



Source: OECD

Fig. 14 – Páginas Web ligadas a servidores seguros por domínios e linguagens

No que respeita aos sites mais acedidos, através da mesma sondagem Markttest referenciada anteriormente, 10.4% dos entrevistados refere o SAPO²⁴, 6.3% o Yahoo!²⁵, 5.3% o Altavista²⁶, 4.2% o Clix²⁷, 3.6% o Público²⁸, 3.4% o AEIOU²⁹, 3.3% A Bola³⁰, 3.1% O Record³¹ e 2.7% Jornal de Notícias³². Mais uma vez, através destes valores, podemos reflectir sobre a importância que os conteúdos em geral e conteúdos de carácter lúdico e informativo têm.

²⁴ Ver em <http://www.sapo.pt/>.

²⁵ Ver em <http://www.yahoo.com/>.

²⁶ Ver em <http://www.altavista.pt/>.

²⁷ Ver em <http://www.clix.pt/>.

²⁸ Ver em <http://www.publico.pt/>.

²⁹ Ver em <http://www.aeiou.pt/>.

³⁰ Ver em <http://www.abola.pt/>.

³¹ Ver em <http://www.record.pt/>.

³² Ver em <http://www.jn.pt/>.

Apenas como nota de reflexão, relativa a esta sondagem, o site do *BigBrother*³³, obtém uma discreta percentagem de 0.6% o que, de certo modo, não deixa de ser caricato, sendo o *site* mais visitado em Portugal³⁴.

1.3.2 O problema de encontrar informação

Tentemos, por exemplo, encontrar informação acerca dos Centros de Divulgação Científica existentes em Portugal. Imaginemos que não sabemos nada acerca do assunto, nem mesmo os nomes dos mesmos. Tentemos uma busca num motor de pesquisa nacional com o tema ciência ou divulgação. O resultado estende-se desde as páginas das universidades até artigos sobre a Burguesia Mercantil do Século XVII e páginas de Tunas Académicas sem ser possível encontrar, de uma forma clara a página do Centro de Ciência Viva do Algarve por exemplo. Claro que uma navegação ramificada pelos resultados destas buscas poderá, eventualmente, levar o esforço a bom porto mas, torna-se evidente a necessidade urgente da criação de um portal devidamente vocacionado para estes temas. Algo que, de uma forma integrada, forneça de uma forma clara e directa as ligações, comentários e informação relativos a todos os projectos que estão em curso e tomam forma no mundo virtual do ciberespaço.

1.3.3 A necessidade de credibilizar a informação

Lançando mão a um estudo feito por *Rob Kling* e *Lisa Kovi* acerca dos *Publishers on-line*³⁵ podemos encontrar alguns tópicos e fulcros problemáticos do processo de publicação de material científico, credível e de qualidade, no domínio da Internet [22].

O problema, que se esperaria ver resolvido de uma forma milagrosa pela Internet, da passagem dos artigos científicos ou de cariz tecnológico-científico do domínio das revistas e publicações académicas, naturalmente fora do circuito social dominante, para a Internet, de acesso global que se pretende globalizante, tornou-se, à luz da experiência acumulada durante os últimos anos, de uma complexidade bastante maior que a esperada.

Uma característica típica de quem produz e publica este tipo de artigos é a alta sensibilidade à legitimidade e status da publicação em causa.

³³ Ver em <http://www.bigbrother.iol.pt/>.

³⁴ Fonte TVI/IOL.

³⁵ Ver em <http://www.slis.indiana.edu/kling/pubs/ais95dig.html>.

Por outro lado, em campos tecnológicos de rápida evolução é altamente criticada a lentidão que o processo de *peer review*, impressão e periodicidade impõe à saída de novos avanços e descobertas.

Torna-se óbvio que o problema da periodicidade e impressão são eliminados num *publisher on-line*, reduzindo este constrangedor *over-head* temporal ao processo de *peer-review*.

A ideia resultante desta conjugação de factores é apostar na introdução do processo de *peer-review* no processo de aceitação de publicações em *publishers on-line*. E tudo isto a favor de uma credibilização de todo o sistema que conduza ao aproveitamento das imensas potencialidades da Internet neste campo.

A solução não existe senão na experiência e interessa atentar nos casos de sucesso de publicações on-line. Um caso particular é o da revista JAIR – Journal of Artificial Intelligence Research³⁶. Além da integração do *peer-review* na estrutura editorial, a própria comissão de selecção é constituída por elementos cientificamente credíveis e o interface gráfico aproxima-se muito do seu irmão em papel, encurtando o fosso entre as duas versões.

Claro está que a questão da reprodução do material de uma forma incontrolável põe em cheque a questão da Propriedade Intelectual do mesmo. Este é um problema de difícil resolução mas que não poderá ser visto como intransponível já que a evolução das técnicas de protecção de conteúdos on-line avançam diariamente. E não fosse este problema já existente nas versões em papel desde a vulgarização da fotocopiadora.

1.3.4 Conclusões

Como podemos aferir de uma forma bastante clara e objectiva, a relação existente entre a divulgação da cultura científica, a Internet e as infra-estruturas telemáticas é bem mais problemática do que o que poderia parecer numa primeira análise. A utilidade deste tipo de ferramentas é altamente dependente da potenciação das suas funcionalidades. Pode-se dizer que há uma relação de mutualismo que ainda falta (e é necessário) gerar entre as classes sociais

³⁶ Disponível em <http://www.jair.org/>.

dominantes e o domínio Internet, de forma a criar um ambiente realmente integrado de geração, disseminação e aquisição de conhecimento de importância formativa relevante.

A solução não é única nem unidireccional mas pretende-se unificadora. Depende de uma concentração de esforços que se estendem desde o plano educacional até ao plano económico e seus constrangimentos, puros e simples, numa floresta de ramificações e interoperacionalidades concretas mas, no seu objectivo final, dispersantes.

A resposta que damos, fruto da orientação científica e académica do grupo de trabalho, assenta na tentativa de resolver os problemas de credibilização da informação, conteúdos disponíveis e acessibilidade dos mesmos num ambiente on-line.

É essencial agora que as suas funcionalidades sejam potenciadas pelos outros actores envolvidos no processo, dando dimensão social à tecnologia.

2 Uma Proposta de Abordagem aos Problemas da Publicação de Documentos na Internet

2.1 *Serviço de catalogação de material didáctico*

Para resolver o problema de localização de recursos de material didáctico que tenha sido disponibilizado através da Internet, mais concretamente sob a forma de páginas da *Web*, o projecto propôs-se desenvolver um serviço que centralizasse o registo de trabalhos com esse âmbito. Assim, quem quiser saber que material existe sobre um determinado tema, poderá ter o trabalho de pesquisa largamente facilitado.

Por outro, os autores deste tipo de material poderão ser encorajados não só a ajudar a registar o material que disponibilizam através da Internet, mas também poderão já realizar algum esforço de pré-classificação do material divulgado indicando as eventuais categorias do material que pretendem registar ou eventualmente sugerir novas categorias.

2.2 *Serviço de interligação automática entre páginas*

A existência duma base de dados deste género sobre o material didáctico disponível na Internet, poderão ajudar os autores desse tipo de material a enriquecer os seus documentos com ligações (*link:s*) entre partes relevantes dos seus textos com outros documentos catalogados na base de dados que tinham a ver com os tópicos abordados nesses textos.

Assim, por exemplo, uma pessoa que quiser elaborar um documento sobre o desporto nas escolas para disponibilizar através da Internet, poderá usar o serviço de consulta da base de dados de material didáctico, não só para descobrir que material já existe sobre o assunto que lhe possa ajudar a elaborar o seu documento, mas também para fazer referência no documento ao material encontrado sobre assuntos relacionados, fazendo uso de ligações (*link:s*) para as páginas relevantes.

No entanto, os utilizadores que mais poderão beneficiar da informação existente numa base de dados de material didáctico para elaborar documentos desse tipo, frequentemente não dispõem de conhecimentos suficientes para estar à vontade com a ideia de adicionar

sistematicamente as interligações entre os documentos, o que muito os poderia enriquecer em termos de valor didáctico e conteúdo.

Como forma de responder a esta necessidade, o projecto propôs-se incluir no serviço AMIS uma funcionalidade que possa examinar um documento já disponível na Internet e sugerir ao seu autor a adição de algumas ligações para páginas sobre assuntos relevantes, e que com a aprovação do autor, pudesse produzir uma nova versão do documento com as ligações já inseridas nas partes apropriadas do documento.

3 Especificação

Para a criação de uma base de documentos de interesse didáctico é necessário desenvolver uma plataforma que permita a composição e a publicação deste tipo de conteúdos num servidor *Web*.

Assim, a plataforma a ser desenvolvida deve cumprir os seguintes requisitos segundo dois pontos de vista: o da **utilização do sistema** e o da **administração do sistema**:

3.1 *Utilização do sistema*

A forma de utilização do sistema deverá ser fácil de compreender e de usar por utilizadores com poucos conhecimentos técnicos.

3.1.1 Documentação da forma de utilização do sistema

Para editar e publicar documentos os autores deverão ter acesso a documentação escrita em linguagem simples e acessível sobre os diversos passos que cada autor deverá seguir. Deverá estar disponível um manual de utilização sob a forma de páginas directamente acessíveis a partir do servidor de *Web*.

Deverá existir uma versão do manual do utilizador apropriada para ser impressa. Essa versão poderá ser disponibilizada sob a forma de uma página única em formato *HTML* ou como alternativa em formato *PDF* (*Portable Document Format - Adobe Acrobat*).

Em cada uma das páginas que os autores devem aceder para executar os diversos passos de publicação dos seus documentos deverá existir a possibilidade obter ajuda sob cada um dos aspectos dessas páginas sob a forma de ligações para as respectivas páginas do manual de utilização disponibilizado no servidor.

3.1.2 Base de dados do sistema

Todos os autores de documentos devem estar registados para que possam ser identificados pelo servidor sempre que queiram publicar novos documentos ou alterar os seus documentos anteriormente publicados.

Portanto, deverá existir um sistema que permita o registo de novos autores numa base de dados de modo a ser disponibilizada uma conta de acesso única para cada um.

Para assegurar que cada documento publicado só possa ser alterado pelo respectivo autor, cada autor deve possuir um nome de utilizador único e uma senha de acesso individual e secreta.

O nome de acesso de cada utilizador deverá constar apenas de uma palavra escrita apenas com letras minúsculas (sem acentos e sem cedilhas) baseadas no nome do autor de modo a ser mais fácil de memorizar.

No momento do registo o autor deverá sugerir uma senha de acesso e também escolher uma expressão que lhe permita lembrar a senha quando se tiver esquecido dela.

Também deverá existir a possibilidade de alterar a senha de cada autor mesmo depois da conta de acesso ter sido criada, mas sem necessidade de intervenção de um operador humano.

Cada autor deverá ter uma área de acesso restrito na qual serão colocados os documentos que publica. Essa área deverá ter o mesmo nome de utilizador do autor de modo a facilitar a localização no servidor *Web* dos documentos que publica.

3.1.3 Actores envolvidos no Serviço AMIS

Num serviço de disponibilização de informação, como o AMIS, é necessário um espectro de utilizadores envolvidos, não só para manutenção e melhoramento do serviço, como também a própria utilidade, validação e divulgação do mesmo, em comunidades com interesses nos recursos informativos disponibilizados.

Nesta perspectiva, existem três vertentes de utilização do AMIS: Administradores, autores e utilizadores não autores (utilizadores com acesso à Internet).

3.1.3.1 Utilizadores anónimos

Do ponto de vista do utilizador não autor, o acesso aos recursos informativos é válido, i.e., qualquer pessoa com acesso Internet poderá usufruir de toda a informação disponível no portal do serviço, para consulta ou recolha da mesma e também efectuar pesquisas. Esta categoria de utilizadores, não é registada na base de dados do serviço, o que permite a um largo número de pessoas aderirem ao mesmo.

3.1.3.2 Autores

No caso de um utilizador desejar aderir ao serviço como autor, bastar-lhe-á registar-se e aguardar pela aprovação do administrador. Se for aceite, receberá o estatuto de “autor”. Se a sua inscrição não for aprovada poderá receber o estatuto de “inscrição negada” (supondo um utilizador que deseja usufruir do serviço para fins ilícitos) ou inscrição pendente, no caso da mesma necessitar a aprovação de várias pessoas. A ideia base do serviço criar vários tipos de utilizadores é com o objectivo de validar e credibilizar os recursos informativos disponíveis na Internet, tal como já foi referido anteriormente.

O autor é um utilizador com o acesso aos recursos informativos e com a permissão para publicar documentos no AMIS, adicionar ligações (*hyperlinks*) e eventos (e classificar os seus recursos informativos) e efectuar pesquisas . Um autor também poderá apagar os seus próprios documentos do serviço, se assim o desejar. Outra das possibilidades de um autor é o de deixar os seus comentários ou informações no placard do serviço. Este placard informativo suporta comentários diários.

Através do correio electrónico (*e-mail*) o serviço informa o autor das alterações que o administrador pode efectuar aos seus dados. Em estudo encontram-se futuras interfaces de comunicação entre os dois suportes do serviço do AMIS (*Web* e *e-mail*).

3.1.3.3 Administradores

O utilizador com todos os privilégios do serviço são os administradores. De momento, o AMIS apenas possui um administrador, com a possibilidade coexistirem outros no mesmo serviço. O administrador é um autor com o poder de manutenção do site. Tem todas as permissões de utilizadores não registados e de autores, como o acesso à navegação (portal), publicação de documentos e inserção de ligações, eventos e comentários, complementado com a tarefa de manutenção e gerência do serviço. A aprovação de autores depende única e exclusivamente do administrador. Além disso, tem como tarefa a gestão da informação inserida (documentos, ligações e eventos) com funções de a alterar ou apagar.

Esta estrutura de utilizadores, permite que todas as comunidades e grupos científico-tecnológicos, com acesso à Internet, possam aderir ao serviço, num dos três possíveis estatutos de utilização (Fig. 15).

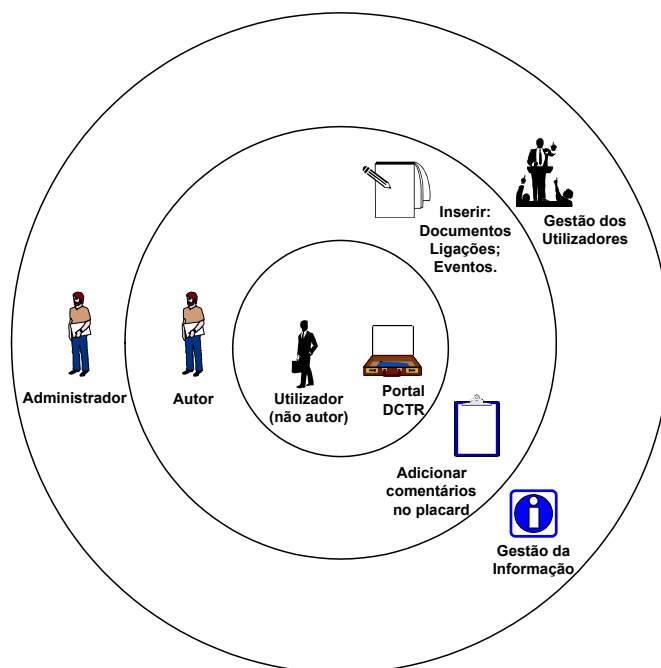


Fig. 15 - Esquema hierárquico de utilizadores.

3.1.4 Criação e publicação de documentos

A criação de um documento implica a criação de um ou mais ficheiros que têm de ser enviados para o servidor de *Web* para serem publicados. A publicação de cada documento deverá ser feita através dos programas de composição já mencionados através das primitivas atrás descritas.

Para evitar a eventual confusão de ficheiros de diferentes documentos, deverá ser criada uma área distinta para alojar os ficheiros que dizem respeito a cada documento dentro da área de trabalho do respectivo autor.

Para tal é necessário que haja a possibilidade de cada autor criar previamente uma área para cada documento que cria antes de o poder publicar.

Essa possibilidade deve ser dada a cada autor através de um formulário solicitando a informação necessária, como o título do documento, uma descrição sucinta do seu contexto e o nome da área a ser criada para o documento.

O nome dessa área deve constar apenas de letras minúsculas sem acentos nem cedilhas de modo a ser mais fácil indicar a localização do documento no servidor, dado que esse nome fará parte do *URL* do documento.

3.1.5 Classificação de documentos publicados

De modo a ajudar a organizar os documentos disponibilizados num servidor, cada documento deverá ser classificado pelo próprio autor. Para assegurar que a classificação de todos documentos é sempre feita, o acesso a cada documento publicado num servidor de *Web* só deve ser permitido depois do autor ter classificado devidamente o documento.

A classificação de um documento deverá ser feita a dois níveis: uma classificação temática e outra através da indicação de expressões chave relacionadas com o contexto do documento de modo a ajudar à sua pesquisa.

A classificação temática consiste na indicação de um tema ou mais temas dentro dos quais o assunto do documento se insere, dentro de uma lista de temas pré-definidos. Se não existir nenhum tema apropriado para classificar um documento, o autor poderá sugerir um novo tema.

O autor também poderá classificar um documento como sendo de um sub-tema abrangido dentro de um ou mais temas já existentes expandindo assim uma hierarquia de temas.

A classificação por expressões chave consiste na indicação de uma ou mais expressões constituídas por uma ou mais palavras que estejam associadas aos assuntos abordados em cada documento.

3.1.6 Interligação de documentos publicados

Depois de um autor ter classificado um documento que já publicou o documento ficará disponível para consulta por qualquer utilizador. Mas para que o documento possa ser enriquecido deverá haver uma funcionalidade que permita estabelecer a interligação do documento publicado com outros documentos existentes sobre temas relacionados.

O estabelecimento de ligações entre documentos deve ser feito a pedido do autor de um documento depois deste ter sido publicado.

O sistema deverá analisar os ficheiros em formato *HTML* de cada documento e tentar encontrar expressões que foram usadas para classificar outros documentos anteriormente publicados.

O sistema deverá apresentar a lista de expressões encontradas assinalando-as directamente no documento para que o autor possa ver claramente que interligações lhe estão a ser sugeridas.

O utilizador terá a possibilidade de aprovar ou não qualquer uma das interligações sugeridas. As interligações aprovadas pelo autor deverão aparecer no documento como qualquer hiperligação num documento de *HTML*, mas o *URL* apontado por essa hiperligação deverá ser de tal forma para que possa ser identificado pelo sistema de modo a ser removido mais tarde se for caso disso.

3.1.7 Pesquisa de documentos

Para facilitar a localização de documentos existentes num servidor, deverá existir uma página que permita a um utilizador fazer pesquisas sobre a base de dados dos documentos.

As pesquisas deverão incidir primeiramente nas expressões chave indicadas pelos autores dos documentos, e depois em outros elementos descritivos dos documentos, como os seus títulos e a suas descrições.

Se não forem encontrados documentos com expressões chave iguais às indicadas pelo utilizador, a pesquisa deverá ser repetida novamente mas tentando encontrar expressões com sonoridade idêntica, de modo a dotar o sistema de pesquisa de alguma tolerância a erros ortográficos que os utilizadores possam cometer.

3.2 Administração do sistema

O sistema deverá ser tão autónomo quanto possível, mas nas partes em que é necessária a intervenção de um operador humano deverão existir páginas através das quais as tarefas de administração do sistema sejam facilitadas.

3.2.1 Gestão de contas de acesso

Para um autor poder publicar documentos precisa de se inscrever no sistema. Uma conta de acesso deverá ser criada a partir dos dados introduzidos por cada autor num formulário de inscrição.

Como os pedidos de inscrição deverão ser sujeitos a aprovação de um administrador do sistema, deverá existir uma página através da qual o administrador pode consultar a lista de pedidos de inscrição pendente e a partir daí poder aprovar os pedidos de autores legítimos.

Por cada pedido analisado deverá ser enviada uma mensagem a notificar o respectivo autor da sua aprovação, tendo o administrador de explicar o motivo das eventuais reprovações.

3.2.2 Gestão de documentos

O administrador do sistema deverá ter a possibilidade de controlar a acessibilidade dos documentos publicados pelos autores. Assim, o administrador poderá inibir o acesso a documentos que não estejam em condições de serem publicados por não estarem de acordo com os propósitos a que se destina o sistema.

3.3 *Estrutura do Repositório de Informação do Serviço AMIS*

O AMIS armazena três tipos de recursos informativos: Documentos, ligações (*hyperlinks*) e Eventos (Fig. 16). Por sua vez, os recursos informativos são classificados em unidades temáticas, definidas pelos utilizadores do AMIS (Fig. 17). O serviço também possui um placard informativo com comentários introduzidos pelos utilizadores e actualizados ao dia corrente.

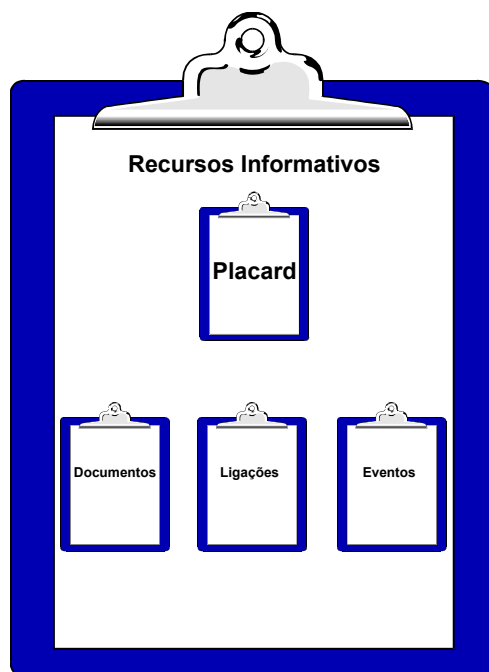


Fig. 16 - Organização de recursos informativos.

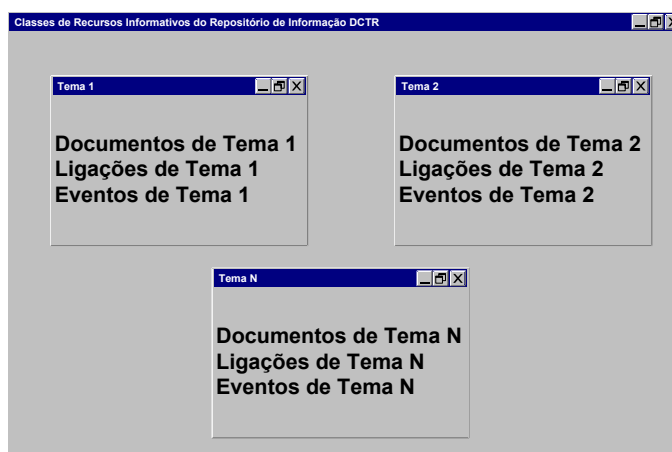


Fig. 17 - Organização temática de conteúdos.

3.3.1 Documentos

Os documentos são qualquer tipo de informação que um autor pode publicar. Podem ser uma dissertação, um texto, *slide-shows*, imagens, gráficos ou qualquer outro tipo de informação que um autor deseje disponibilizar on-line.

Os documentos são introduzidos no AMIS de duas maneiras: Ou pelo programa de edição e publicação de páginas *Netscape Composer* ou através de uma página *Web* do AMIS destinada para esse efeito. Após a introdução do documento no serviço, o documento encontra-se

disponível on-line, inserido no tema “Sem Classificação”. O autor deve classificar o documento, nas classes temáticas existentes ou criando uma nova classe ou sub-classe, e adicionar expressões chave ao documento de modo a que este, seja visualizado ou pesquisado com maior sucesso por parte dos utilizadores do serviço. O autor tem também a possibilidade de apagar os documentos que desejar (propriedade intelectual do indivíduo).

3.3.2 Ligações

Outro recurso informativo, não menos importante que um documento, são as ligações (*hyperlinks*) existentes na Internet. Estas ligações abarcam todo o tipo de temas e assuntos, muitos dos quais, de interesse para adição ao AMIS. Autores e administradores tem permissão para as adicionar e classificar na base de dados do serviço, com uma breve descrição. Se estas não forem classificadas, encontrar-se-ão no portal do AMIS sob a classe “Sem classificação”. Tal como nos documentos, esta fonte de informação deve ser classificada, facilitando a procura no portal e nas pesquisas, de quem pretende recolher ou visualizar informação referente a um assunto inserido no âmbito do projecto DCTR. Apesar dos autores terem a possibilidade de adicionar recursos informativos, apenas o administrador os poderá remover, sempre que essas ligações não sirvam os propósitos do projecto DCTR.

Sempre que um utilizador tentar aceder a estas ligações, antes do seu programa de navegação os redireccionar, são verificadas. Se existirem o programa de navegação redirecciona para o destino. Se o servidor *Web* que disponibiliza as ligações estiver “em baixo”, o serviço redirecciona o utilizador para a página fonte informando-o que não é possível efectuar a ligação. Se essa ligação não existir, o serviço remove esse recurso informativo, informando o utilizador que essa ligação não existe.

3.3.3 Eventos

Os eventos poderão ser conferências, “workshops”, seminários ou qualquer outro tipo de encontro, propícios a debates, informação e divulgação ou confrontação de ideias, típicos e usuais de comunidades científico-tecnológicas. É deste modo que aglomeramos os eventos nos recursos informativos. Do mesmo modo que as ligações, autores e administradores poderão adicionar eventos, somente o administrador os poderá apagar sempre que esses eventos não sirvam os interesses do projecto DCTR. O serviço apaga um determinado evento, quando a data de conclusão desse evento for ultrapassada pela data do dia corrente. A Fig. 18, apresenta a estruturação, em ambiente *Web*, de todos esses recursos.

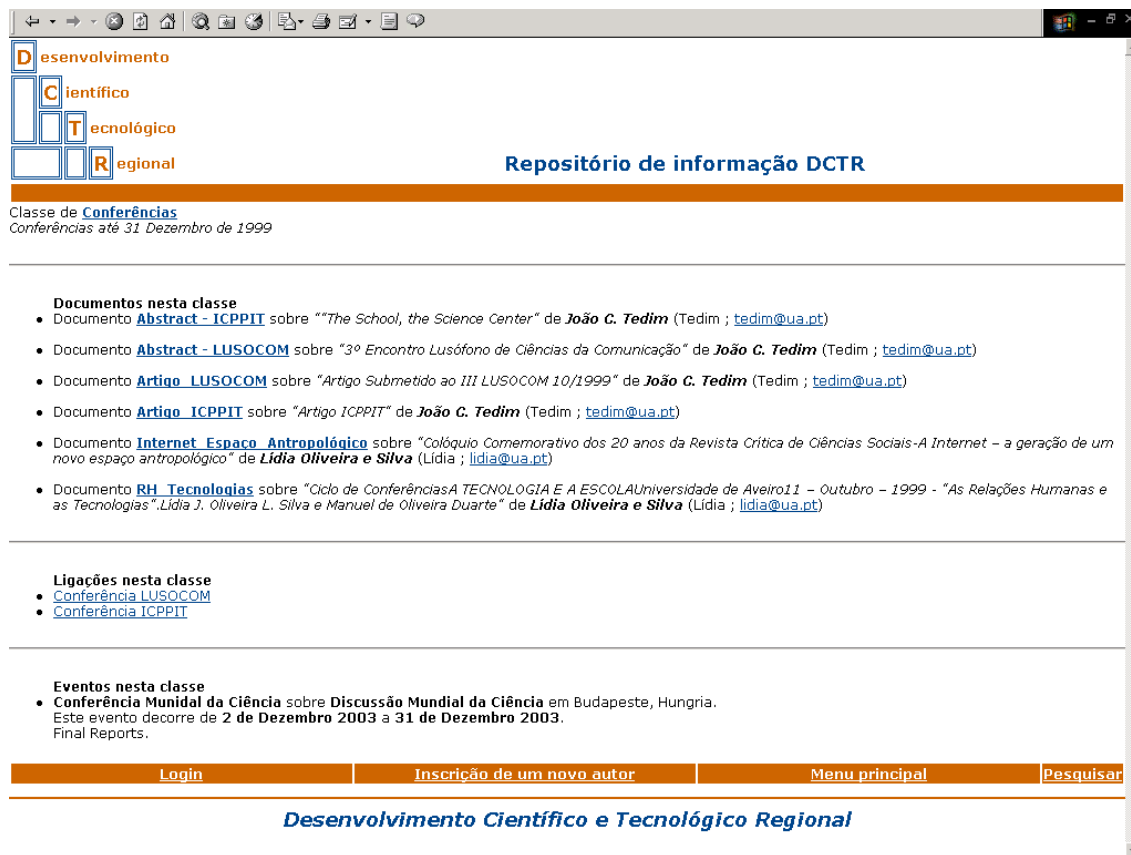


Fig. 18 - Estruturação de conteúdos na Web.

3.3.4 Classes

Todos os recursos informativos, estão associados a classes temáticas que são criadas sempre que nova informação introduzida mereça uma classificação ainda não definida, i.e., a criação de uma nova classe principal, ou uma classificação mais específica, i.e., uma nova classe que é sub-classe de outra de nível hierárquico superior. (Suponha-se a criação de uma nova classe de “sociologia”, dentro da classe de “ciências sociais”).

Outra função apenas permitida a administradores é a alteração de classes existentes (um título mal introduzido) ou remoção das mesmas (no caso de transpor os objectivos do serviço). Ambos autores e administrador tem a permissão para adicionar novas classes e sub-classes.

No caso do administrador remover uma classe de nível hierárquico inferior, todos os recursos informativos da classe removida são transpostos para a classe de hierarquia superior (supondo que a classe de “sociologia” é removida, todos os seus recursos informativos serão movidos para a classe “ciências sociais”).

Na hipótese de remoção de uma classe superior, os recursos informativos dessa mesma classe perderão a sua classificação (voltando ao mesmo exemplo, todos os recursos informativos da classe “ciências sociais” perderiam a sua classificação e seriam disponibilizados no serviço na classe “sem classificação”).

3.3.5 Portal AMIS

O portal é a página principal do AMIS que engloba todos as classes temáticas com recursos informativos, placard e menu de opções.

3.3.5.1 Lista de classes

No portal são disponibilizadas várias classes temáticas que abarcam um leque de recursos informativos e outras sub-classes.

3.3.5.2 Placard informativo

Todos os autores e administrador são convidados, diariamente pelo serviço, a introduzirem um breve comentário diário. Esse comentário estará disponível durante o dia corrente no portal. Poderão ser inseridos nesses comentários informações de carácter lúdico ou profissional.

3.3.5.3 Menu de opções

O menu de opções é dotado de variadas funcionalidades destinadas a acções de cada autor ou administrador:

- **Autores:**
 - Inscrição de um novo autor;
 - Alterar os dados de um autor.
- **Documentos:**
 - Criar um novo documento;
 - Publicar documentos Word, PDF,...
 - Alterar os dados de um documento;
 - Apagar documentos;
 - Listagem de documentos.
- **Recursos Informativos:**
 - Adicionar ligações;
 - Adicionar eventos;
 - Pesquisar.
- **Administração.**
 - **Utilizadores:**

- Listagem dos utilizadores;
- Alterar os dados de um utilizador.
- **Ligações:**
 - Alterar ligações;
 - Apagar ligações.
- **Eventos:**
 - Alterar eventos;
 - Apagar eventos.
- **Classes:**
 - Alterar classes;
 - Apagar classes.

As opções de administração serão apenas visualizadas se o autor tiver o estatuto de administrador. Será pedida uma palavra-passe ao utilizador na tentativa de execução de uma das funções do menu por parte do mesmo. Na Fig. 19, é apresentada a estrutura na *Web* do portal:



Fig. 19 - Portal AMIS/DCTR.

3.3.6 Pesquisas

As pesquisas estão acessíveis a qualquer utilizador (anónimo, autor e administrador). Um utilizador pode efectuar pesquisas, simples ou complexas, de toda a informação armazenada na base de dados do AMIS (documentos, expressões-chave, ligações, classes e eventos). Na pesquisa simples as expressões introduzidas, serão pesquisadas em todas as fontes de informação da base de dados do serviço. Na pesquisa complexa o utilizador pode escolher quais os recursos da base de dados que pretende procurar.

A expressão-chave a analisar de várias formas: se a expressão chave for coincidente com alguns recursos informativos, são retornados resultados; se não retornar resultados é procurada uma expressão-chave similar com aquela que foi introduzida; se mesmo assim a expressão-chave introduzida (entenda-se expressão por várias palavras) não retornar resultados é partida em pirâmide esquerda e direita até que sejam retornados alguns resultados; se não retornar resultados, é procurada palavra a palavra. Em todas as pesquisas são procuradas palavras e expressões idênticas ou similares.

Desta forma é facilitada a pesquisa aos utilizadores que, muitas vezes noutros serviços na *Web*, têm de introduzir palavra a palavra até obterem algum sucesso em procura de informação nos serviços de quem a disponibiliza. A Fig. 20 apresenta uma busca efectuada ao serviço.

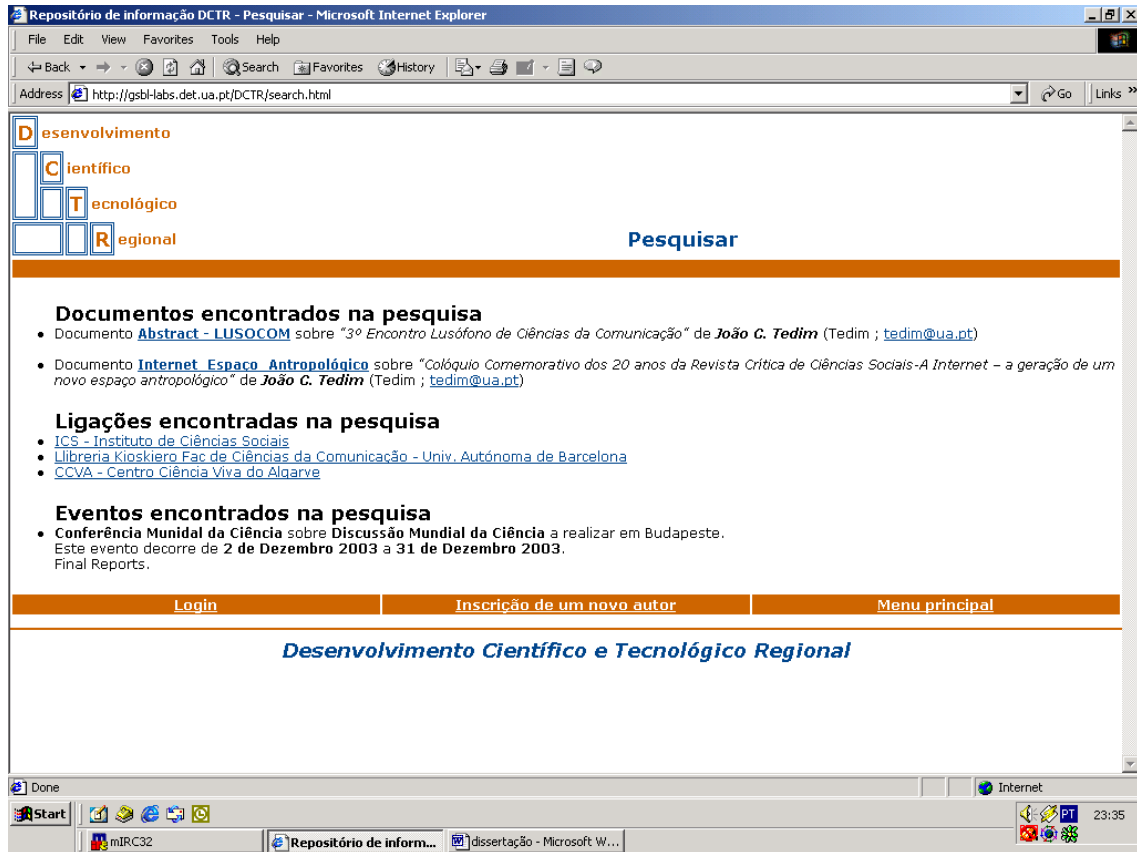


Fig. 20 - Resultados de uma busca sobre “ciência”.

4 Implementação

4.1 *Requisitos de Hardware do sistema*

4.1.1 Uma máquina destinada à função de servidor de informação ligada à Internet.

Idealmente deverá ser um computador baseado num sistema operativo tipo UNIX, particularmente devido à existência de *software* de elevada qualidade que é apropriado para a implementação para aplicações orientadas para a Internet.

4.2 *Requisitos de Software do sistema*

4.2.1 Servidor de Web Apache

*Apache*³⁷ é o nome do servidor de *Web* de longe mais usado em todo mundo, não só porque é gratuito, mas também pela elevada qualidade da implementação das funções para as quais foi concebido (servidor de *HTTP*) e possibilidades de extensão das suas capacidades com módulos de *software* facultativos também disponíveis gratuitamente.

4.2.2 · Servidor de base de dados MySQL

A catalogação e pesquisa de grandes quantidades de informação requer o uso de *software* especializado para a criação e gestão de bases de dados. *MySQL*³⁸ é o nome dum produto de *software* destinado a esse efeito tendo como principais características a elevada conformidade com a norma **ANSI SQL 92** e elevado desempenho na criação e acesso bases de dados de grande complexidade.

Apesar de em termos gerais não ser gratuita, *MySQL* está disponível sem quaisquer encargos para utilizações sem fins lucrativos.

4.2.3 · Linguagem de programação de páginas dinâmicas PHP

A interligação dum servidor de *Web* com um servidor de base de dados deve ser feito através dum componente de *software* que permita criar as diversas partes dos serviços que se pretende implementar. A linguagem *PHP*³⁹ é uma das mais sofisticadas ferramentas disponíveis para este efeito.

³⁷ Disponível em <http://www.apache.org/>.

³⁸ Disponível em <http://www.tcx.se/>.

³⁹ Disponível em <http://www.php.net/>.

A linguagem *PHP* permite criação dinâmica de páginas de *HTML* apropriada ao âmbito deste projecto. Esta linguagem permite estabelecer um interface transparente à aplicação tanto com vários tipos de servidores de *Web* como com mais de uma dezena de produtos de gestão de bases de dados.

A linguagem *PHP* permite desenvolver aplicações de elevada portabilidade. Esta capacidade permite desenvolver aplicações sem ter que presumir a dependência num determinado tipo de servidor de *Web*, base de dados ou mesmo de sistema operativo.

O *software* desta linguagem está disponível de forma completamente gratuita tanto para sistemas tipo *UNIX* como para sistemas *Windows*.

4.2.4 · **Software de gestão de versões de software CVS**

O desenvolvimento de projectos de *software* de tamanho significativo requer uma gestão apropriada das sucessivas versões de cada componente desenvolvido, em particular quando os diversos componentes podem ser desenvolvidos em paralelo por vários membros do projecto.

*CVS*⁴⁰ é nome de um pacote destinado a gestão do desenvolvimento concorrente de *software*. Este *software* permite manter uma base de dados de todos os passos do desenvolvimento de *software* de um projecto, desde a adição de novos componentes até as alterações feitas em cada componente por cada membro envolvido num projecto, impedindo os conflitos derivados de alterações simultâneas no mesmo componente feitas por dois ou mais membros.

Este *software* está disponível gratuitamente tanto para plataformas tipo *UNIX* como para *Windows NT*.

4.2.5 · **Software de análise e processamento de documentos de HTML - SGMLS**

A análise e processamento de documentos *HTML* requer o uso de ferramentas dedicadas à análise sistemática deste tipo de ficheiros. Existem várias ferramentas apropriadas para este efeito em particular e de documentos formatados segundo *SGML* (*Standard Generalized Markup Language*).

⁴⁰ Disponível em <http://www.cvshome.org/>.

SGMLS⁴¹ é o nome de uma das ferramentas conhecidas de análise de documentos. No entanto há necessidade de analisar melhor a existência de outras ferramentas para este propósito menos conhecidas.

4.2.6 · **Software de gestão de projectos - Keystone**

A gestão dum projecto de relativa complexidade, requer o planeamento estruturado, não só das tarefas a executar como o tratamento das alterações aos planos iniciais desde a adição de novas tarefas e cancelamento de tarefas que se tornem desnecessárias, como também as alterações ao calendários previsto para a execução das tarefas do projecto.

Keystone⁴² é o nome de um pacote de *software* que facilita o trabalho de planeamento e gestão das tarefas necessárias à implementação de projectos em particular de *software*.

O acesso aos diversos aspectos do desenvolvimento de projectos é feito através de páginas de *Web* geradas dinamicamente. *Keystone* foi desenvolvido em linguagem *PHP*. Isto significa que a base de dados com a informação de tarefas pode ser criada com vários tipos de produtos de *software* de gestão de base de dados, incluindo o *MySQL*.

Keystone está disponível gratuitamente para actividades sem fins lucrativos. A eventual adopção do *software Keystone* para gestão das tarefas deste projecto carece de análise mais aprofundada da existência de eventuais alternativas com melhores possibilidades.

4.3 **Capacidades implementadas**

A implementação das capacidades do *software* previstas seguiu a especificação detalhada na secção anterior deste documento. No entanto, até à data em que este documento foi escrito não houve possibilidade de implementar completamente todas as capacidades descritas.

4.3.1 **Base de dados do sistema**

Para a implementação do *software* para a publicação de documentos num servidor de *Web*, foi concebido um esquema para base de dados que armazena informação pertinente a funcionamento deste sistema. A representação gráfica do esquema da base de dados é apresentado em *UML (Unified Modeling Language)* no **apêndice A** desta dissertação.

⁴¹ Disponível em <ftp://ftp.jclark.com/pub/sgmls/>.

⁴² Disponível em <http://www.stonekeep.com/keystone/>.

Como se pode constatar pelo esquema, a base de dados do sistema já prevê o armazenamento de informação sobre os utilizadores do sistema, os documentos publicados, os ficheiros de que é composto cada documento, a classificação temática de cada documento e as expressões chaves associadas.

4.3.2 Páginas de acesso ao sistema

Para aceder ao sistema instalado, os utilizadores devem aceder ao um conjunto de páginas criadas para esse efeito. Essas páginas que podem ser acedidas a partir duma página inicial na qual existe um menu de opções.

De momento o menu de opções do sistema está dividido em três partes: autores, documentos e administração (Fig. 21). As diversas páginas disponíveis podem ser acedidas escolhendo as respectivas opções que aparecem sob a forma de ligações (*links*). Apenas autores e administradores têm acesso ao menu de opções embora, todos os utilizadores, incluindo os utilizadores anónimos, têm acesso a todo o conteúdo publicado.



Fig. 21 - Menu de opções activo para um administrador.

As opções que são mostradas dependem dos privilégios de cada utilizador que acede ao sistema. Assim, por exemplo, um utilizador que só tem privilégios de autor não poderá aceder às páginas que dizem apenas respeito à administração do sistema.

O controlo de acessos a páginas que requerem determinados privilégios é feito através do pedido de identificação de cada utilizador através de um nome de acesso e uma senha de acesso secreta (Fig. 22).



Fig. 22 - Páginas de acesso ao menu de opções para autores e utilizadores registados.

4.3.2.1 Utilização de cookies para utilizadores registados

Uma das possibilidades que o serviço oferece, quando a introdução de nome e senha de acesso, é o de guardar, através de *cookies*, o nome e a senha de um utilizador. Quando o utilizador prime a *checkbox* “lembrar o seu acesso” (Fig. 33), o serviço guarda no computador cliente uma *file* contendo um número de identificação único (*UniqueId*). A próxima vez que o utilizador registado aceder ao serviço, já não terá de introduzir o nome e senha de acesso, devido a que o serviço já o identificou, por essa pequena *file* a que tem acesso, no *browser* do cliente. No caso do utilizador registado não escolher esta opção, o menu de opções apenas

estará activo enquanto o *process Id*, ou seja o executável, estiver “vivo”. Para o utilizador desfrutar desta comodidade, terá necessariamente de ter os *cookies* activos nas preferências do *browser*.

4.3.3 Registo do administrador do sistema

Para gerir os diversos recursos do sistema tem de existir uma pessoa que se responsabiliza por essa função. O administrador do sistema é apenas um utilizador com privilégios de acesso especiais.

Inicialmente a base de dados de utilizadores do sistema está vazia. Assim, quando é necessário aceder a uma página cujo o acesso está restrito ao administrador do sistema, o sistema redirecciona o acesso para uma outra página de registo do administrador do sistema.

A página de registo do administrador do sistema contém um formulário o qual deve ser preenchido com os dados da pessoa que se vai responsabilizar pela função de administração.

Depois de ser submetido o formulário só a pessoa inscrita pode aceder às páginas de administração. Apesar da base de dados do sistema estar preparada para poder suportar mais do que um administrador, de momento essa possibilidade não está disponibilizada.

4.3.4 Registo de novos autores

Para um autor poder publicar os seus documentos precisa de se registar no sistema. Existe uma página para registo de novos autores que pode ser acedida seguindo uma opção de rodapé “Inscrição de um novo autor”.

A página de registo de novos autores pode ser acedida por qualquer utilizador ou não do sistema. Nessa página aparece um formulário no qual devem ser preenchidos os campos que respeitam a diversos tipos de dados pessoais (Fig. 23).

The image shows a web browser window with a registration form. The browser's address bar is empty. The page has a header with a logo consisting of the letters 'D', 'C', 'T', and 'R' in a grid, followed by the text 'Desenvolvimento Científico Tecnológico Regional'. The main heading is 'Inscrição de um novo autor'. Below this is a sub-heading: 'Por favor, preencha e submeta este formulário de inscrição.' The form itself is a rectangular box containing several input fields: 'Endereço de correio electrónico:', 'Nome pessoal:', 'Nome pelo qual prefere ser tratado:', 'Nome de acesso:', 'Senha de acesso:', and 'Confirme a senha de acesso:'. There is also a text prompt: 'Escreva uma frase que lhe ajude a lembrar a sua senha de acesso quando se tiver esquecido dela:'. A 'Submeter a inscrição' button is located at the bottom center of the form. At the bottom of the page, there are two buttons: 'Menu principal' and 'Pesquisar', and a footer line with the text 'Desenvolvimento Científico e Tecnológico Regional'.

Fig. 23 - Página de inscrição de um novo autor.

Depois de submetido e validado o formulário, os dados nele inscritos são inseridos na base de dados de utilizadores de sistema criando uma ficha de autor com inscrição pendente.

4.3.5 Aprovação da inscrição de um novo autor

A inscrição de um novo autor só será aprovada quando o administrador do sistema aprovar a inscrição submetida. Aquando a inscrição de um autor pendente, o administrador recebe um *e-mail*, notificando-o que um utilizador pretende obter um estatuto de autor. A aprovação é feita através da página de administração que permite modificar os dados de um utilizador do sistema para autor ou administrador e inscrição negada, pendente ou aprovada.

Um administrador do sistema pode mudar o estatuto de um autor com a inscrição pendente aprovando ou não a sua inscrição. Em qualquer dos casos o administrador tem de entrar um comentário que será enviado numa mensagem de correio electrónico para a pessoa que se inscreveu justificando uma eventual recusa da inscrição ou aprovação.

4.3.6 Publicação de conteúdos

A publicação de conteúdos permite que seja feitos *uploads* de qualquer tipo de conteúdos, suportados por tipos MIME⁴³, para o servidor *Web*. As seguintes subsecções detalham o estudo realizado ao protocolo HTTP, protocolo utilizado para *upload*, e o procedimento necessário para a publicação que o serviço cumpre:

4.3.6.1 Programas de publicação de conteúdos

A *Web* é geralmente conhecida como o meio mais óbvio para disponibilizar documentos e informação em geral através da Internet. No entanto, para que essa informação possa ser disponibilizada é necessário que os ficheiros dos documentos sejam compostos para depois serem guardados no disco de um servidor de *WWW* para, a partir daí, o servidor os fornecer a quem os pretender aceder.

Para os autores que compõem documentos para disponibilizar através da *Web*, existem várias alternativas para compor e publicar esses documentos. A composição de documentos pode ser feita através de dois tipos de programas: os **editores manuais** e os **editores visuais**.

Os editores manuais são simples programas de edição de texto através dos quais um autor pode definir o conteúdo dos seus documentos directamente através da linguagem *HTML*.

Os editores visuais são programas que permitem ao autor ter visualmente uma ideia do aspecto do documento que está a elaborar à medida que o vai compondo.

Os editores visuais são mais apropriados para autores com poucos conhecimentos técnicos dado que não requerem a aprendizagem prévia da linguagem *HTML*.

No entanto, os editores manuais são preferidos pelos autores mais experientes porque ao permitirem escrever directamente na linguagem HTML dão todas as possibilidades a esses autores de tirar o máximo proveito das capacidades desta linguagem.

Depois de compor um documento, um autor necessita de o publicar, ou seja, de enviar o documento de alguma maneira para um servidor. Existem várias maneiras para enviar um

⁴³ *Multipurpose Internet Mail Extensions*. Ver em <ftp://ftp.isi.edu/in-notes/rfc2045.txt>.

documento para um servidor de *WWW*: **correio electrónico** (*E-mail*), **transferência de ficheiros** (*FTP*), **publicação através da *Web*** (*WWW*).

A publicação por correio electrónico não é muito comum porque requer a cooperação de um operador humano ou de um sistema automático que receba mensagens de correio electrónico e delas extraia os ficheiros dos documentos nelas contidos.

A publicação através da transferência de ficheiros para um servidor de *FTP* é mais comum. No entanto requer a existência de um servidor de *FTP* à parte do servidor de *Web* que os vai disponibilizar. Por outro lado, a publicação de documentos através de *FTP* requer por parte dos autores, conhecimento de utilização de programas próprios transferir os ficheiros dos documentos.

A publicação através da *Web* é o método mais simples para utilizadores com poucos conhecimentos técnicos. A maior parte dos editores visuais existentes já são capazes de publicar documentos através da *Web*. Isto permite a um autor poder compor e publicar documentos a partir de um só programa sem ter que tomar conta da quantidade e dos nomes dos ficheiros que devem ser transferidos para o servidor *Web*.

4.3.6.2 O Protocolo HTTP

O trabalho desta parte do projecto AMIS estará centrado essencialmente na publicação de documentos na *Web* através dos métodos descritos nas normas oficiais que definem a comunicação com servidores *Web*, ou seja, o protocolo *HTTP* (*HyperText Transfer Protocol*).

A apresentação da especificação dos protocolos usados na Internet é feita normalmente através de documentos designados genericamente por *RFC* - *Request For Comments*.

A especificação do protocolo *HTTP* foi apresentada através dos documentos *RFC 1945* (*HTTP 1.0*)[23] abaixo e *RFC 2068* (*HTTP 1.1*)[24]. Estes documentos especificam várias **primitivas** que definem como se processa o intercâmbio de informação entre um programa servidor de *Web* e um programa cliente que tanto pode ser um programa de navegação (*browser*), como um programa de composição de páginas ou ainda um *robot* para captura sistemática de páginas.

A comunicação entre um programa cliente e um programa servidor é feita através de **troca de mensagens** em que para cada mensagem enviada pelo programa cliente, o programa servidor retorna uma resposta.

Uma mensagem de *HTTP* é um bloco de informação definido segundo a especificação **MIME** (RFC 2045 a 2049 e outros)²⁵ - *Multipurpose Internet Mail Extensions* - que é tradicionalmente usada para envio de mensagens de correio electrónico.

Normalmente, uma mensagem definida segundo a especificação *MIME* está dividida em duas partes: o **cabeçalho** e o **corpo da mensagem**. O cabeçalho é constituído por várias linhas de texto que servem para definir os valores de parâmetros da mensagem. O corpo da mensagem contém a informação propriamente dita que se pretende transmitir.

A comunicação através de *HTTP* é sempre iniciada pelo programa cliente. Depois de estabelecida a ligação, o programa cliente envia uma mensagem através da qual faz um pedido ao servidor.

Uma mensagem de um pedido de *HTTP* pode ser definida através de vários tipos de **primitivas** de acesso conforme o tipo de pedido. O tipo de primitiva é definido na primeira linha do cabeçalho da mensagem enviada pelo cliente ao servidor *Web*.

4.3.6.2.1 HTTP 1.0

O protocolo *HTTP 1.0* definiu 3 primitivas para comunicação entre um cliente e um servidor de *Web*: GET, POST e HEAD.

4.3.6.2.1.1 Primitiva GET

A primitiva GET serve para fazer um pedido genérico de informação a um servidor de *Web*. Esta é a primitiva que é usada normalmente pelos programas de navegação para aceder a uma página.

Normalmente, o *URL* indicado pelo utilizador define a localização relativa no disco do servidor onde está o ficheiro da página a aceder. Em geral um *URL* define o contexto da página a aceder que pode não ser necessariamente a localização dum ficheiro guardado no disco do servidor.

4.3.6.2.1.2 Primitiva HEAD

A primitiva HEAD é semelhante à primitiva GET, excepto que, como resposta o servidor retorna apenas o cabeçalho da mensagem que seria retornada se o pedido tivesse sido feito através da primitiva GET indicando o mesmo *URL*.

Esta primitiva serve para fazer breves consultas a um servidor como por exemplo para determinar se uma página com um dado *URL* ainda está disponível. Foi através desta primitiva, que se desenvolveu o sistema de validação de links, abordado mais à frente.

4.3.6.2.1.3 Primitiva POST

A primitiva POST serve para fazer um pedido de acesso de acordo com parâmetros definidos pelo programa cliente. Este é a primitiva que é usada normalmente para submeter formulários definidos em páginas de *HTML*. Os valores definidos pelo utilizador nos campos de entrada do formulário são enviados para o servidor como parâmetros no corpo da mensagem.

A primitiva POST também pode ser usada para submeter formulários em que os campos de entrada são ficheiros indicados pelo utilizador, de acordo com a especificação descrita no *RFC 1867*[26].

A primitiva GET também pode ser usada para submeter formulários, em que os valores dos campos do formulário aparecem sob forma codificada como parte do *URL* da página a aceder. No entanto, esta primitiva não pode ser usado para envio de tipos de informação mais complexa, como por exemplo ficheiros.

4.3.6.2.2 HTTP 1.1

O protocolo *HTTP 1.1* acrescenta 4 novas primitivas à especificação 1.0: OPTIONS, TRACE, PUT e DELETE.

4.3.6.2.2.1 Primitiva OPTIONS

A primitiva OPTIONS serve para determinar as opções das comunicações para acesso a um determinado servidor de *Web* ou a um recurso de informação acessível através desse servidor.

4.3.6.2.2.2 Primitiva TRACE

A primitiva TRACE serve para determinar por que servidores intermédios passa um acesso a um servidor *Web* quando esse acesso é feito através de servidores *proxy*.

O cabeçalho da mensagem de resposta de um pedido de acesso a um servidor *Web* feito através desta primitiva mostra uma cópia do cabeçalho da mensagem enviada pelo programa cliente, mas acrescentado de linhas de cabeçalho que denotam os servidores *proxy* por onde eventualmente essa mensagem tenha passado se for caso disso.

4.3.6.2.2.3 Primitiva PUT

A primitiva PUT serve para pedir a um servidor *Web* que coloque um dado ficheiro para ser disponibilizado a partir de um dado *URL*. O ficheiro a ser disponibilizado no servidor é enviado como parte do corpo da mensagem enviada pelo programa cliente.

Normalmente, um servidor que suporte a primitiva PUT delega num programa auxiliar a acção de validar e disponibilizar os ficheiros enviados pelos programas clientes de acordo com a configuração do programa servidor.

4.3.6.2.2.4 Primitiva DELETE

A primitiva DELETE serve para pedir a um servidor *Web* para eliminar ou pelo menos tornar indisponível um dado recurso indicado como parâmetro na mensagem enviada por um programa cliente.

4.3.6.3 Ferramentas de composição e publicação de documentos em servidores *Web*

As ferramentas de composição e publicação de páginas mais conhecidas usam a primitivas POST e PUT para publicar documentos em servidores *Web*.

Concretamente, o programa *Microsoft Frontpage* usa a primitiva POST para enviar para o servidor os ficheiros das páginas compostas pelos utilizadores. Para suportar a primitiva usada por este programa é necessário ter instalado extensões no servidor de *Web* de modo a permitir a descodificação dos ficheiros enviados.

Os programas *W3C Amaya* e *Netscape Composer* (parte do programa *Netscape Communicator*) usam a primitiva PUT para publicar os documentos compostos pelos utilizadores. O suporte à

primitiva PUT só pode ser proporcionado por um servidor *Web* que suporte o protocolo *HTTP 1.1*, como é o caso do programa servidor *Apache 1.3* que é usado neste projecto.

Como a primitiva PUT é a recomendada pela norma do protocolo *HTTP 1.1* para publicação de documentos em servidores *Web*, é sobre esta primitiva que se irá basear a plataforma de publicação de documentos que está a ser desenvolvida.

De notar também que qualquer dos programas de composição e publicação mencionados que suportam a primitiva PUT, podem ser obtidos gratuitamente para a generalidade das plataformas.

4.3.6.4 Criação de pasta

A partir do momento que a sua inscrição é aprovada, uma autor está habilitado a criar pastas para publicar conteúdos. No entanto, para criar uma pasta é necessário criar um registo para identificar um documento perante o sistema.

O registo de uma nova pasta é feita através duma página que pode se acedida usando uma opção da parte respectiva a “Recursos Informativos” do menu de opções.

Na página de registo de criar pastas existe um formulário que pede um título, uma descrição e uma localização para o novo documento a ser alojado no servidor.

Para a localização é apenas pedida a parte final do *URL* do documento criar que na prática corresponde a um subdirectório onde serão alojados os ficheiros do documento dentro da área de documentos respectivos ao autor (Fig. 24).

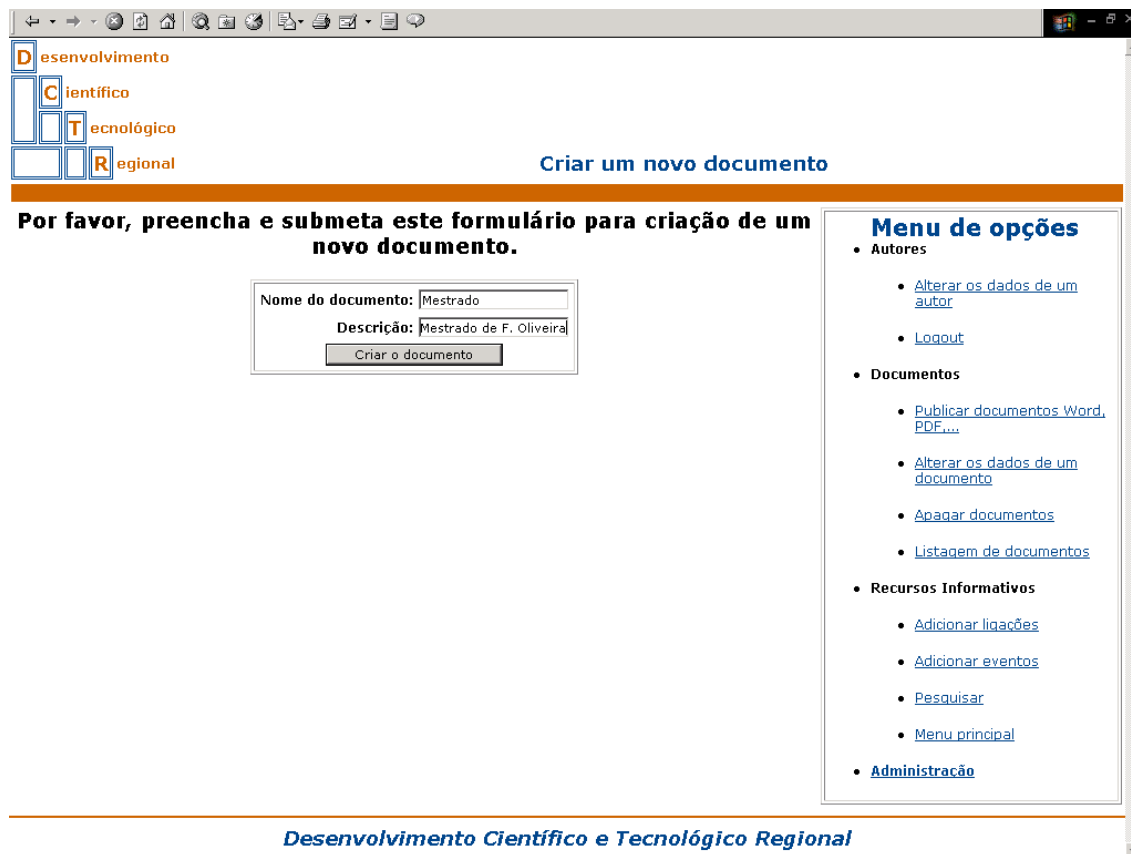


Fig. 24 - Criação de uma pasta.

4.3.6.5 Publicação de conteúdos

Após a criação de uma pasta, o serviço apresentar-lhe-á uma opção de *directory browsing*, ou seja, navegar pelo disco duro do cliente e escolher as *files* a publicar (Fig. 25). Todas as *files* que publicar, serão guardadas na sua pasta, anteriormente criada.

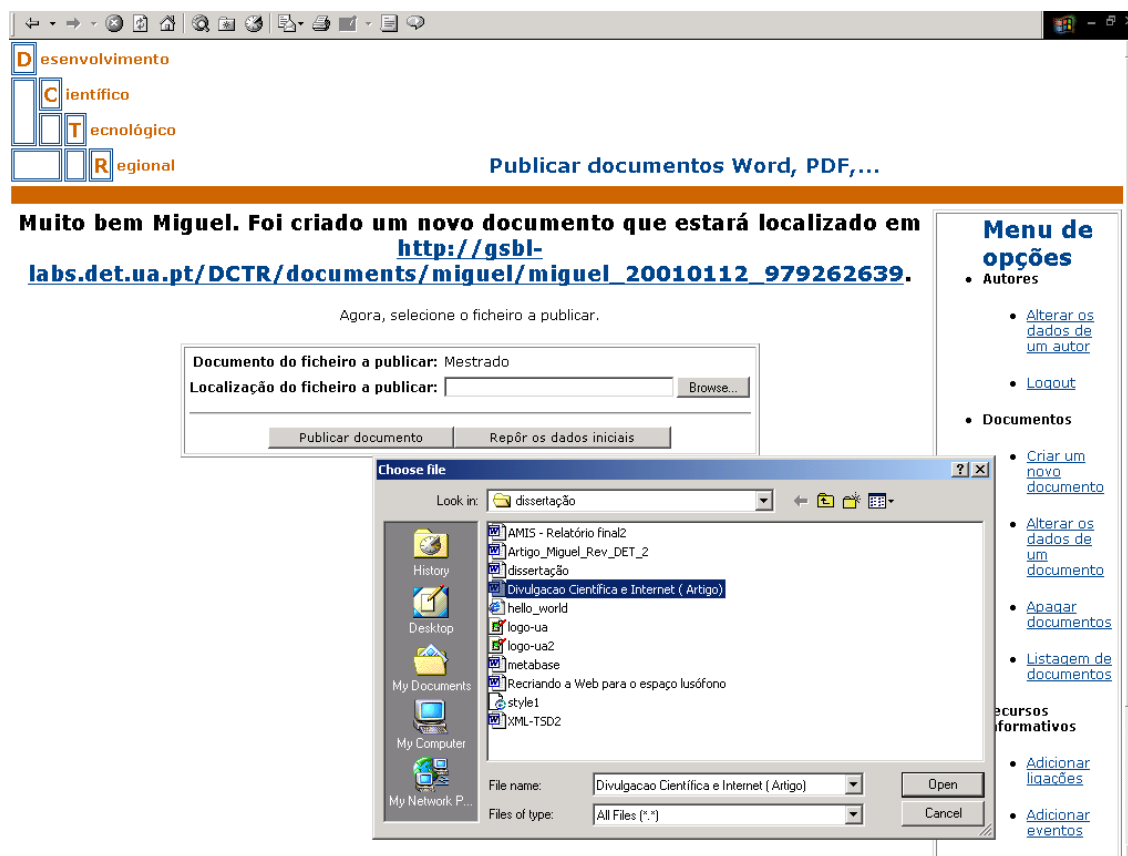


Fig. 25 - Página de publicação de conteúdos com *Directory Browsing*.

4.3.6.5.1 Publicação através do Netscape Composer

A publicação de um documento é feita através de um programa de publicação como o *Netscape Composer*⁴⁴ ou o *Amaya*⁴⁵. Para além serem usados para composição de documentos, este programa são capazes de enviar os ficheiros dos documentos editados usando a primitiva *PUT* do protocolo *HTTP 1.1* descrito atrás.

Como está atrás descrito, um autor precisa de usar um programa de publicação de documentos para enviar para o servidor os ficheiros que compõem esses documentos. Normalmente, os programas de publicação sabem determinar automaticamente que ficheiros devem ser enviados.

⁴⁴ Disponível em <http://home.netscape.com/download/index.html?cp=djucl>.

⁴⁵ Disponível em <http://www.w3.org/Amaya/>.

Para enviar um documento para o servidor, cada autor deve indicar a localização (*URL*) do documento a publicar no servidor, bem como a sua identificação como autor perante o sistema, ou seja, o seu nome de acesso e a respectiva senha.

Apesar das funcionalidades destes programas, a sua utilização não é tão amigável como a que foi descrita na secção anterior. É necessário introduzir alguns campos de autenticação, *URLs*, nome da *file*, o que se torna algo “cansativo”. Por essa razão, apesar de a publicação por programas externos ter sido implementada anteriormente, foi desenvolvida a solução apresentada anteriormente.

4.3.6.6 Classificação de conteúdos

Após o *upload* das files, o autor é questionado com uma página para classificação temática e classificação por palavras chave de conteúdos:

4.3.6.6.1 Classificação de conteúdo por expressões chave

Depois da criação de um documento, o seu autor pode classificar o seu documento definido expressões chave que estão relacionadas com o seu contexto. Esta classificação permite fazer a interligação com outros documentos sobre temas relacionados.

A definição de expressões chave pode ser feita através duma página destinada a fazer alterações aos dados de um documento. Essa página pode ser acedida através duma opção disponível na parte respectiva aos documentos do menu de opções.

Na página para fazer alterações aos dados de um documento existe um formulário que para além de outros aspectos permite adicionar, alterar e remover expressões chave a associar ao documento em questão (Fig. 26).

4.3.6.6.2 Classificação temática de conteúdos

A capacidade de classificar um documento por temas é disponibilizada a partir da página que existe para alteração dos dados de um documento. Essa capacidade permite classificar um documento dentro de temas já existentes numa hierarquia de temas já definidos e também permitir um autor propor novos temas. Um tema novo proposto por um autor deve ser sujeito à aprovação do administrador do sistema.

Após a classificação do documento o serviço analisa as várias expressões-chave do documento e pesquisa expressões iguais ou idênticas noutros documentos. Se existirem, o serviço notifica o autor por *e-mail*, informando-o da possibilidade de adicionar ligações ao documento classificado com outros documentos. Na notificação por *e-mail*, o serviço informa sobre quais as expressões idênticas que encontrou, quais os documentos e *files*, e que autor é que as publicou (Fig. 26).

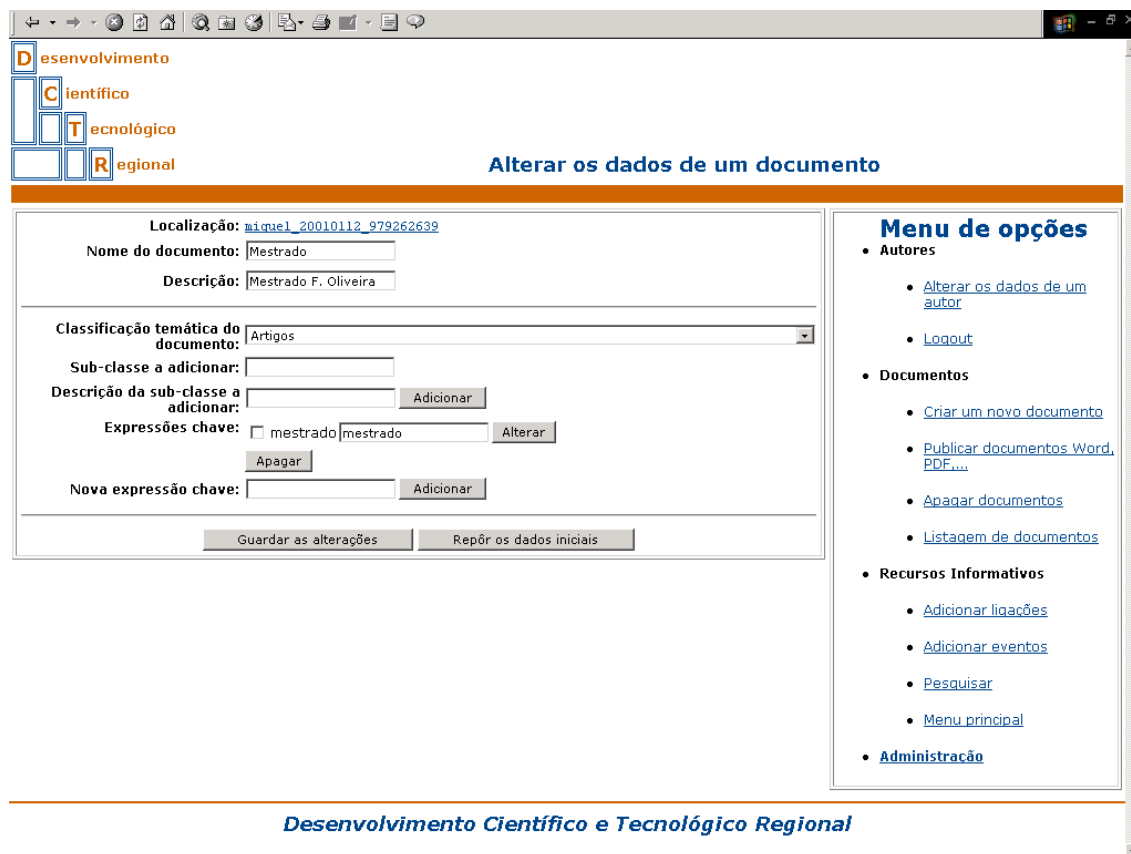


Fig. 26 - Classificação de conteúdos temática e por Expressões-chave.

A partir da classificação temática, o documento fica disponível no portal, sob a classificação temática que o utilizador inseriu. No caso de não classificar tematicamente o documento, este fica disponível numa classe dinâmica⁴⁶ denominada “Sem tema”.

⁴⁶ A classe “Sem tema”, só visualizada quando existem conteúdos que não são classificados por tema.

Todas estes passos, são cumpridos pelo sistema aquando a publicação de novo conteúdo. Mas também estão disponíveis isoladamente para alteração de dados classificativos e adição de novas files numa pasta.

4.3.7 Criação de ligações

A partir do momento que a sua inscrição é aprovada, uma autor está habilitado a adicionar ligações. Na página que adiciona ligações, o autor tem a possibilidade de classificar por tema a ligação, inserindo-a em temas já existentes, ou criando um novo tema (Fig. 27).

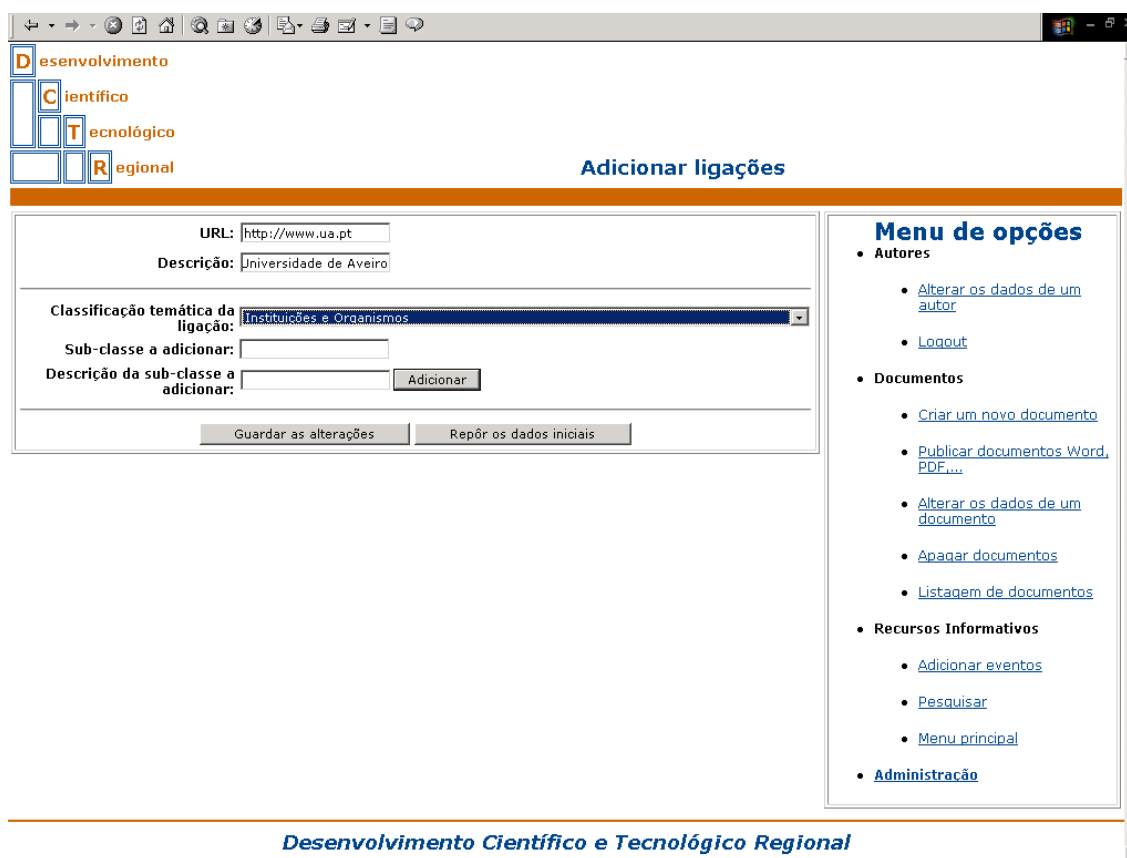


Fig. 27 - Inserção de *links*.

4.3.8 Criação de eventos

A partir do momento que a sua inscrição é aprovada, uma autor está habilitado a adicionar eventos (conferências, *workshops*, seminários). Na página que adiciona eventos, o autor tem a possibilidade de classificar por tema a ligação, inserindo-a em temas já existentes, ou criando um novo tema.

4.3.9 Notificação da aprovação de um autor

Assim que a inscrição pendente de um autor é aprovada ou não, este recebe uma notificação automática por correio electrónico.

Qualquer alteração que seja feita a um autor pelo administrador do sistema, este poderá notificar por correio electrónico o autor.

4.3.10 Interligação de um documento com outros documentos publicados

4.3.10.1 Interligar documentos

A proposta de inserção de ligações para outros documentos é feita numa página especial com três janelas.

Na janela à esquerda é apresentado o documento analisado assinalando claramente as ligações que são propostas para que o autor as possa aprovar ou não como definitivas.

Na janela à direita são apresentadas as várias ligações correspondentes às expressões encontradas, assim como informações correspondentes às várias ligações e apontadores de expressões para expressões do documento a interligar.

Na janela do topo encontram-se várias opções de navegação e de ajuda a interligar o documento.

As ligações inseridas não apontam directamente para outro documento. Quando se tenta aceder a uma ligação que foi inserida por este processo, o documento respeitante a essa ligação é procurado na base de dados do sistema. Se esse documento tiver sido apagado por um autor e não for possível acede-lo, será direccionado para o próprio documento onde se encontra a ligação e o utilizador será informado da inacessibilidade do documento. Se o documento para onde aponta a ligação existir, será direccionado para tal.

4.3.10.2 Adicionar ligações aos documentos

Um autor tem a possibilidade de inserir ligações no fim dos seus documentos. Se um autor seleccionar esta opção, ser-lhe-ão propostas várias ligações de outros documentos com expressões-chave idênticas aquelas que adicionou no documento que pretende interligar. As que seleccionar serão inseridas nas files HTML desejadas do documento.

Tanto a secção 4.3.10.1 como a secção 4.3.10.2 só são válidas para documentos HTML.

4.3.11 Alteração dos dados de um autor

O administrador do sistema e os autores tem acesso a uma página com a função de alterar os seus dados pessoais. Embora o autor possa alterar os seus dados está restringido aos privilégios que normalmente um administrador tem e que o um autor não deve ter, nomeadamente a alteração do estatuto do utilizador.

4.3.12 Alteração das expressões chave de um documento

A possibilidade de alteração ou remoção de expressões chave já associadas é disponibilizada na mesma página que permite alterar os dados de um documento.

4.3.13 Classificação temática de ligações e eventos

A capacidade de classificar ligações e eventos por temas é disponibilizada a partir da página que existe para adicionar ligações e eventos. Essa capacidade permite classificar ligações e eventos dentro de temas já existentes numa hierarquia de temas já definidos e também permitir um autor propor novos temas. Um tema novo proposto por um autor deve ser sujeito à aprovação do administrador do sistema.

4.3.14 Pesquisas

Deverá existir uma página para esse efeito que permita a um utilizador pesquisar na base de dados dos documentos alojados no servidor indicando palavras que podem fazer parte das expressões chave, títulos, descrições ou classificação temática dos documentos existentes.

Existe uma página que permite a pesquisa simples ou complexa de documentos, ligações e eventos no serviço. Na pesquisa simples são procurados no serviço, todos os recursos informativos (eventos, ligações, documentos) e também classes temáticas, expressões-chave e utilizadores). Na pesquisa complexa o utilizador pode escolher quais dos campos anteriores quer pesquisar e também efectuar a pesquisa nos motores sapo e cusco.

Em ambas as pesquisas é procurada a expressão chave completa ou idêntica. No caso da pesquisa não retornar resultados, parte a expressão em pirâmide esquerda e direita até retornar resultados. Se mesmo assim não retornar resultados, efectua pesquisas em cada uma das palavras chaves introduzidas pelo autor. Também neste caso são pesquisadas expressões-chave e palavras idênticas ou iguais.

4.3.15 Apagar documentos

Cada autor tem a possibilidade de apagar os seus documentos, bastando para isso seleccionar aqueles que deseja apagar.

4.3.16 Alterar e apagar ligações

Como as ligações são de domínio público, apenas o administrador pode apagar ou alterar as ligações, bastando para isso seleccionar aqueles que deseja apagar.

Na página de visualização dos recursos informativos, as ligações são testadas pelo serviço, i.e., não direcciona o programa de navegação de imediato para a nova ligação, mas verifica a validade da ligação. Se esta não existir, ou se o servidor estiver em baixo, o autor é notificado no programa que a ligação não está acessível ou que não existe. No caso de não existir, a ligação é apagada pelo serviço. Se existir o serviço redirecciona o programa de navegação para a ligação.

4.3.17 Alterar e apagar eventos

O administrador tem a possibilidade de apagar ou alterar os dados dos eventos. As ligações relacionadas com os eventos também são testadas. Se não existirem o serviço apaga as ligações e actualiza a base de dados.

Os eventos são apagados do serviço quando a data de conclusão do evento é inferior à data corrente.

4.3.18 Alterar e apagar classes temáticas

O administrador tem a possibilidade de apagar ou alterar classes temáticas se assim o entender. No caso de alterar classes dependentes hierarquicamente, actualiza todos os recursos informativos para a classe hierarquicamente superior em relação à classe em que estavam inseridos.

4.3.19 Placard informativo

Os autores registados no serviço podem adicionar comentários diários num placard informativo, de interesse para os utentes. Os comentários são apagados no serviço quando termina o dia.

4.4 Capacidades a implementar

De acordo com a especificação para as capacidades do sistema, houve um conjunto de capacidade para as quais até à data houve tempo para implementar parcial ou completamente.

4.4.1 Controlo do acesso aos documentos

De momento o acesso aos documentos publicados não está a ser feito de forma controlada. A criação dos directórios para alojar os documentos criados só é feita no momento em que o autor publica os documentos, ou seja, quando usa o programa de publicação para enviar os respectivos ficheiros de cada documento.

Quando um utilizador acede a um documento que ainda não contém quaisquer ficheiros publicados, o servidor retorna uma mensagem de erro que habitualmente retorna quando não existe no servidor o recurso associado ao *URL* indicado pelo utilizador que o acede.

Na circunstância descrita deverá haver a possibilidade de mostrar uma mensagem a indicar de uma forma mais simpática que o documento ainda não foi publicado.

Depois do documento ser publicado deverá haver também a possibilidade de indicar ao sistema qual o ficheiro a associar como página inicial quando o utilizador que acede ao servidor apenas indica o *URL* respectivo ao directório onde está alojado o documento. De momento o ficheiro de página inicial de um documento é *index.html* que é definido na configuração do servidor Apache.

4.5 Capacidades em estudo e desenvolvimento

4.5.1 CSS

4.5.1.1 Introdução

A CSS⁴⁷ foi definida e desenvolvida pelo W3C⁴⁸, como um simples mecanismo simples de estilos de páginas que permite a criadores e webdesigners adicionarem estilo (cores, tipos de letra, espaços, etc.) a documentos HTML. A linguagem CSS é facilmente lida e interpretada, tal como o HTML, expressando estilos numa terminologia de publicação comum[27].

⁴⁷ Cascading Style Sheets, ver em <http://www.w3.org/Style/CSS/>.

⁴⁸ World Wide Web Consortium, ver em <http://www.w3.org/>.

Uma das grandes vantagens das CSS é a possibilidade de alterar, mudar, renovar uma página, feita em HTML nativo, e dar-lhe um aspecto completamente diferente. Além disso, permite que uma plataforma como o AMIS, tenha diferentes CSSs, ajustando o serviço para gostos ou falhas humanas e tecnológicas. Actualmente, apesar de toda a tecnologia envolvente ser a mesma, ambas as plataformas RI-DCTR e UUGuide, feitas com o mesmo HTML nativo, diferem uma da outra devido a diferentes CSSs incluídos a cada uma (Fig. 28 e Fig. 29).



Fig. 28 - Página de rosto do DCTR.

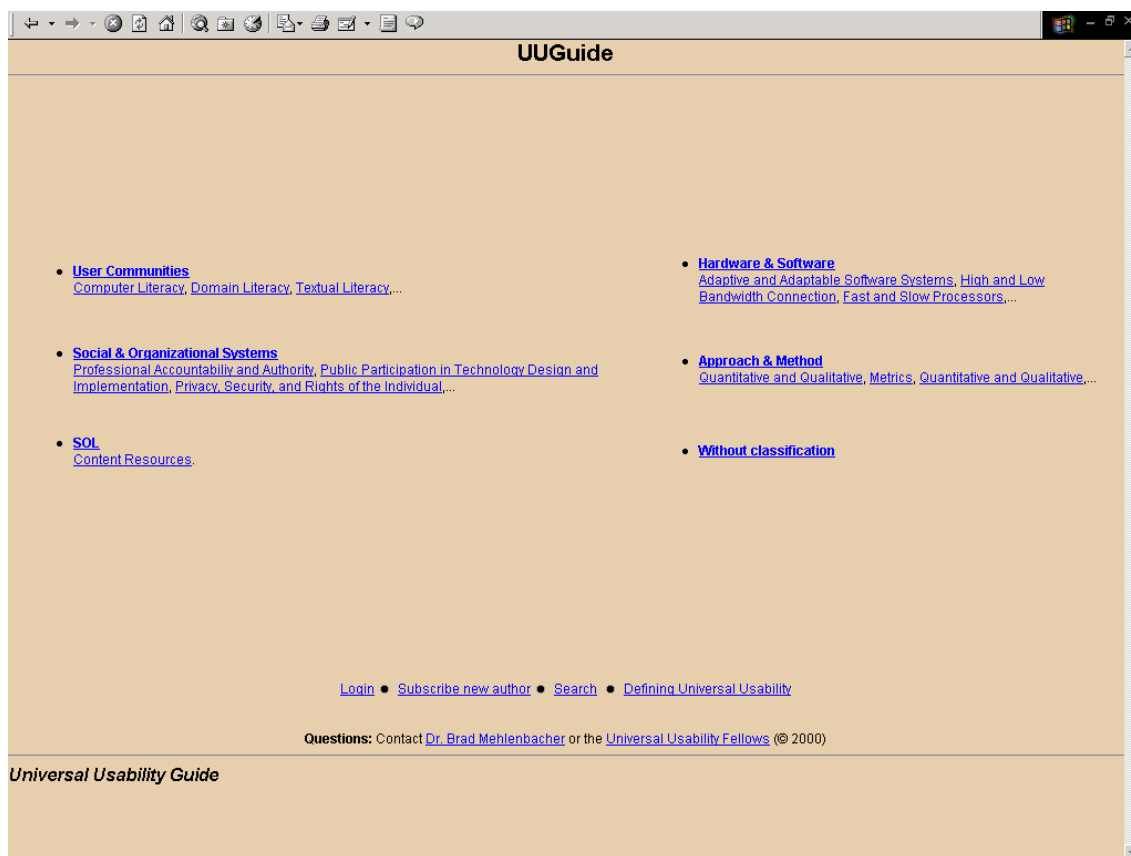


Fig. 29 - Primeira página de rosto do UUGuide. Através das CSSs, é possível apresentar a mesma tecnologia com interfaces diferentes.

Por último, as CSSs têm um grande impacto a nível comercial devido a que, quando uma página é apresentada com uma CSS em *attach* bem definida e estruturada, torna-se impossível a alteração à página, por parte de um utilizador, em cores, tipos de *links*, tipos de letra, cores de fundo, etc., o que significa que para um serviço comercial que pretende transmitir uma determinada imagem, além das possibilidades de integração de publicidade no *site*, impõe a sua própria interface para o consumidor ou publico alvo. Outro dos grandes interesses das CSSs a nível comercial, é a possibilidade de alterar, sempre que o desejarem a interface, bastando para isso alterar a CSS, funcionalidade esta demonstrada na secção seguinte. Como exemplos de *sites* que usam CSSs, podemos referir a página da CNN⁴⁹, SONY⁵⁰, SoundBlaster⁵¹, entre outros.

⁴⁹ Ver em <http://cnn.com/>.

⁵⁰ Ver em <http://www.sony.com/>.

⁵¹ Ver em <http://www.soundblaster.com/>.

4.5.1.2 Exemplos do uso de CSSs

Supondo um simples exemplo em HTML de uma página:

```
<html>
<head>
<title>Hello World</title>
</head>
<body>
<h1>Hello World - Teste de CSS</h1>
</body>
</html>
```

Num *browser*, aparecerá a página desta forma (Fig. 30):

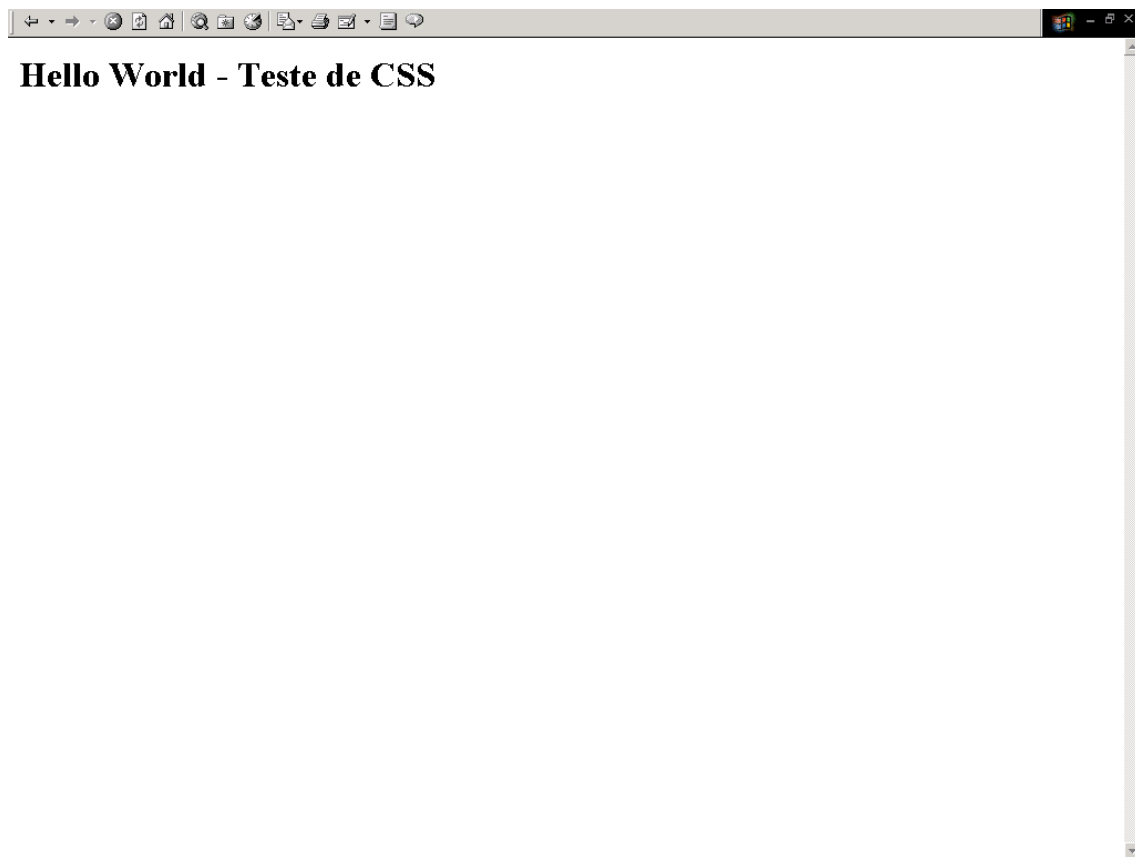


Fig. 30 - Exemplo Hello World em HTML.

Criando agora uma CSS, denominada por HelloWorld.css, definindo uma cor de fundo, um tipo de letra Arial, Helvetica:

```
BODY { font-family: arial, helvetica, sans-serif;
background: #CCCCCC; }
BODY { color: black; }
H1 { display: block }
```

E adicionando à mesma file `HelloWorld.html` o URL da `HelloWorld.css`:

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="/HelloWorld.css"
title="Hello World Style">
<title>Hello World</title>
</head>
<body>
<h1>Hello World - Teste de CSS</h1>
</body>
</html>
```

O mesmo HTML nativo de `HelloWorld.html` será apresentado desta forma (Fig. 31):



Fig. 31 - Exemplo Hello World em HTML com CSS.

Através desta representação de estilo com CSS, podemos concluir que uma página deve ser construída em HTML, no seu formato mais puro e primitivo, evitando variações de *browsers*,

adicionando-lhe sempre uma folha de estilo (CSS). Permite personalizar uma interface para vários utilizadores, quando o HTML nativo é apresentado de uma forma dinâmica.

4.5.1.3 Editor de css

Como podemos observar na secção 9.5.1.1, uma CSS altera por completo a imagem de uma interface nativa em HTML. Uma das tarefas que se encontra em fase de desenvolvimento é um editor de CSSs personalizado a cada utilizador registado (numa fase posterior, será também possível a inclusão de utilizadores anónimos do serviço). Após o registo de acesso dos autores e administradores, ofereceremos a interface HTML *default*. No caso de estes não se identificarem com a interface em questão, poderão alterá-la através de um editor CSS em desenvolvimento, que inicialmente lhes permitirá alterar os tipos de letra, cor de fundo, formato dos links. Mais tarde, a ideia será permitir a alteração de todo a HTML nativo existente na interface. Cada CSS construída através do editor *Web* do serviço, será guardada dentro de cada área do utilizador e invocada quando a página é invocada pelo mesmo, apenas adicionando a seguinte linha ao HTML invocado dinamicamente:

```
<link rel="stylesheet" type="text/css" href="/userXPTO/StyleUserXPTO.css"
title="Hello World Style">
```

4.5.2 HtDig – Motor de indexação e pesquisa de conteúdo estático

4.5.2.1 Observações de motores de indexação

Um motor de indexação, tal como o nome indica, indexa conteúdo estático de páginas alojadas num servidor *Web*, rastreando-as e indexando todo o seu conteúdo e armazenando-o numa base de dados própria, em tempo variável ou não, em palavras e expressões para, posteriormente, serem pesquisadas pelos utilizadores. A vantagem duma ferramenta de indexação é permitir fazer pesquisas a páginas de conteúdos estáticos, que só seria possível se os dados fossem guardados numa base de dados. Usando um motor de indexação, o próprio ao indexar o seu conteúdo está a armazená-lo numa base de dados própria, pronta a ser pesquisada posteriormente.

É necessário ter em conta que os actuais motores de indexação “centrados na *Web*”, em que um utilizador fornece ao motor externo o(s) seu(s) URL(s), não oferecem garantias de segurança, podendo eventualmente indexar *logs*, *passwords* e outras informações privadas sem autorização do utilizador. Além disso, geralmente, não fornecem ajudas e informação para evitar estas situações. Como solução a este problema, o ideal é proceder a uma instalação e

configuração de motores de indexação no próprio *Website*. Entre os mais populares encontram-se o Ht://Dig⁵² da empresa com o mesmo nome, para plataformas UNIX, e o *Microsoft Index Server*⁵³ para plataformas *Windows*.

A escolha do Ht://Dig baseou-se, principalmente, por ser um dos mais populares da plataforma UNIX, ambiente em que este serviço se encontra desenvolvido, e também pela sua facilidade de integração com linguagens interpretadas como o PHP, também usada neste serviço. Por fim, a ideia foi de, quando uma pesquisa à base de dados do serviço não retornar resultados, proceder a uma pesquisa de conteúdo estático, previamente indexado pelo motor externo, neste caso, Ht://Dig.

4.5.2.2 Descrição

O método de indexação do HtDig é simples - captura páginas HTML, através do protocolo HTTP, e une informação, desses documentos, em *word lists* e *expressions lists*. Posteriormente, essa informação é usada, mais tarde para pesquisas sendo assim, denominado também por robot de pesquisa para sites de médio porte e *intranets*. De forma alguma substitui o *Lycos*⁵⁴, *Infoseek*⁵⁵ ou outro qualquer motor de pesquisa. Permite apenas o rastreio, captura e indexação de informação de um domínio WWW, de um directório ou página.

4.5.2.3 Características gerais

Como características principais do HtDig, passo a enumerá-las:

- Pesquisa de informação em *intranets*;
- *FreeWare*;
- Exclusão de robots é suportada (em detrimento de robots.txt e http-meta-equiv);
- Pesquisa de expressões booleanas;
- Resultados de pesquisas configuráveis;
- *Fuzzy* searching (*equal*, *soundex*, *metaphone*, *endings*, sinónimos);
- Pesquisa de HTML/TXT;

⁵² Disponível em <http://www.htdig.org/>.

⁵³ Disponível em <http://www.microsoft.com/ntserver/web/techdetails/overview/IndxServ.asp>.

⁵⁴ Ver em <http://www.lycos.com/>.

⁵⁵ Ver em <http://www.infoseek.com/>.

- Adição de palavras chave às *expression lists* e *word lists*, provenientes de páginas HTML – não visíveis no *browser*, mas existentes em *keywords lists*;
- Notificação por e-mail de documentos expirados;
- Pode usar autenticação de utilizadores para indexar;
- Suporte a ISO-Latin-1 (todo o interesse para pesquisas de documentos em língua portuguesa).

4.5.2.4 Configuração

Actualmente, no serviço AMIS, o HtDig encontra-se em fase de testes (Fig. 32). A primeira tarefa de configuração prende-se com a periodicidade que pretendemos para a indexação das páginas do serviço. Neste serviço específico, optei por uma indexação diária, processo que adicionei à *crontab* do sistema operativo.

Os parâmetros de configuração do htdig, encontram-se descritos na tabela em baixo:

parâmetro	descrição
start_url	Primeira file HTML de indexação
limit_url	Limite de endereços. Normalmente, opta-se por limitar todo o serviço e permitir indexar apenas as files que desejamos.
search_algorithm	Algoritmo de pesquisa. Geralmente, utiliza-se 100% para uma palavra exacta e 50% para a terminações.
exclude_urls	Files que devem ser excluídas nas pesquisas, tais como files de acesso, dados pessoais, etc.
database_dir	Directório onde se encontra a base de dados do HtDig com as <i>expressions lists</i> e <i>word lists</i> .
bad_extensions	Extensões de files que devem ser ignoradas na indexação, tais como, .wav, .zip, .gif, .jpeg, .mpg, etc.
valid_punctuation	Pontuação válida (_-!#\$%).
template	File padrão de apresentação de resultados.
search_result_header	Cabeçalho da file de apresentação de resultados.
search_result_footer	Rodapé da file de apresentação de resultados.
nothing_found_file	File de padrão de resultados nulos.
syntax_error_file	File de padrão de erros de sintaxe.

Na file de configuração, `htdig.config`, optei pela seguinte configuração:

start_url: `http://gsbl/index.html`

limit_urls: `http://gsbl/`

search_algorithm: `exact:1 endings:0.5`

exclude_urls: `? subscribe_new_author.html subscribe_change_author_data.html login.html`

database_dir: `/opt/www/htdig/db`

bad_extensions: `.wav .zip .gif .jpeg .mpg .mov .avi`

valid_punctuation: `: . - _ / ! # $ % ^ & * ()`

template_map: `/home/htdig/htdig_template.html`

search_results_header: `/home/htdig/htdig_header.html`

search_results_footer: `/home/htdig/htdig_footer.html`

nothing_found_file: `/home/htdig/htdig_nomatch.html`

syntax_error_file: `/home/htdig/htdig_syntaxerror.html`

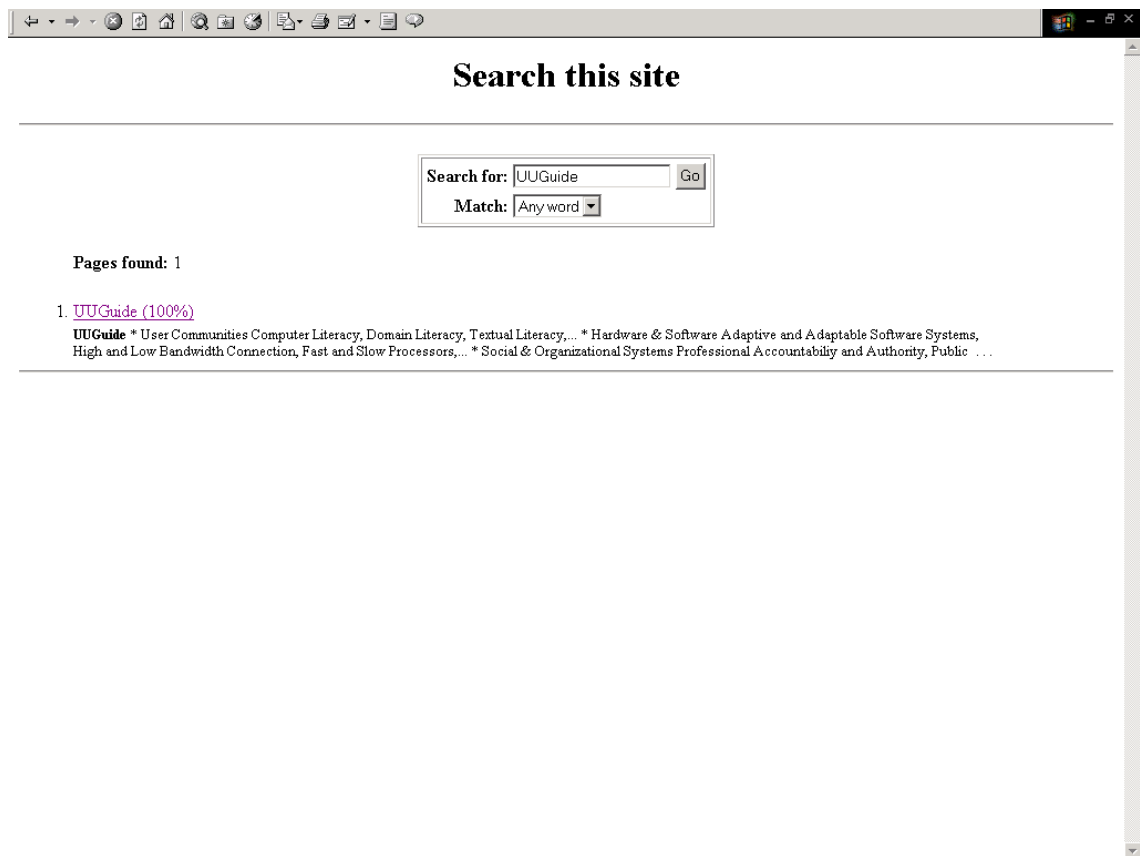


Fig. 32 - Ambiente de testes no *UUGuide* com HtDig.

4.5.2.5 Conclusões

A utilização de motores de indexação de informação, proveniente de páginas estáticas, é sempre útil, quando pretendemos diminuir a probabilidade de uma pesquisa não apresentar resultados. De modo algum, num serviço de disponibilização de conteúdos informativos, numa interface dinâmica, com um sistema próprio de pesquisa, um motor externo de indexação substitui o motor de pesquisa à base de dados. De qualquer maneira, quando as buscas do motor interno não apresentam resultados satisfatórios, é vantajoso tentar uma busca num motor de indexação, dado ser mais uma hipótese de resultados com sucesso. Outra das vantagens prende-se com a própria indexação feita com o HtDig, a nível de palavras, expressões, sonoridade, e meta-informação proveniente de páginas HTML.

5 Validação do projecto AMIS

5.1 Repositório de Informação do DCTR

5.1.1 Enquadramento do sistema de informação no projecto DCTR

Num ambiente de validação de ideias e propósitos a atingir no projecto AMIS, foi criado um Repositório de Informação – Desenvolvimento Científico Tecnológico e Regional (RI-DCTR).

Com base nas ideias apresentadas e definidas anteriormente, foi implementado e concebido um sistema de informação, em progresso e desenvolvimento, com o intuito de albergar informação sobre a oferta de infra-estruturas e serviços telemáticos com incidência nas actividades de ciência e tecnologia.

Com o crescimento exponencial da Internet e de redes telemáticas, as comunidades científico-académica tem demonstrado interesse em disponibilizar os seus conteúdos científicos *on-line*, abrangendo assim não só outras comunidades e grupos científico-académicos, como também pequenos nichos de interesse em diversas áreas. Além disso, é generalizada em Portugal, a falta de confiança, por parte de cientistas e investigadores, em *publishers* e revistas *on-line*, que se reflecte num entrave à consolidação do meio Internet como instrumento credível e de disseminação de conhecimento.

É nesta filosofia que assenta o Repositório de informação DCTR (RI-DCTR⁵⁶) - estrutura disponível e acessível às várias comunidades e nichos científico-académicos com acesso à Internet, incorporando os métodos clássicos de avaliação de conteúdos de publicações científicas (*peer review*), credibilizando o Repositório de Informação DCTR.

O RI-DCTR apresenta, como principais funcionalidades, um repositório de informação produzida e recolhida pelo projecto DCTR, com um mecanismo de difusão e divulgação dos mesmos e de interacção de entidades com interesses afins, acessível e disponível através da Internet.

⁵⁶ Disponível em <http://gsbl-lbas.det.ua.pt/DCTR/>.

O RI-DCTR (Fig. 33) apresenta as mesmas funcionalidades do projecto AMIS acrescido e adaptado de outros requisitos no âmbito do projecto DCTR. Este repositório de informação apoia as actividades de um projecto PRAXIS com o nome “Desenvolvimento Científico Tecnológico e Regional”.

Prestou-se particular atenção à necessidade de qualificar e autenticar informação publicada, aspecto mais crítico devido à falta de confiança por parte dos Cientistas e Investigadores, nos *publishers* e revistas *online* que se reflecte num obstáculo de consolidação do meio Internet como veículo credível de disseminação de conhecimento.

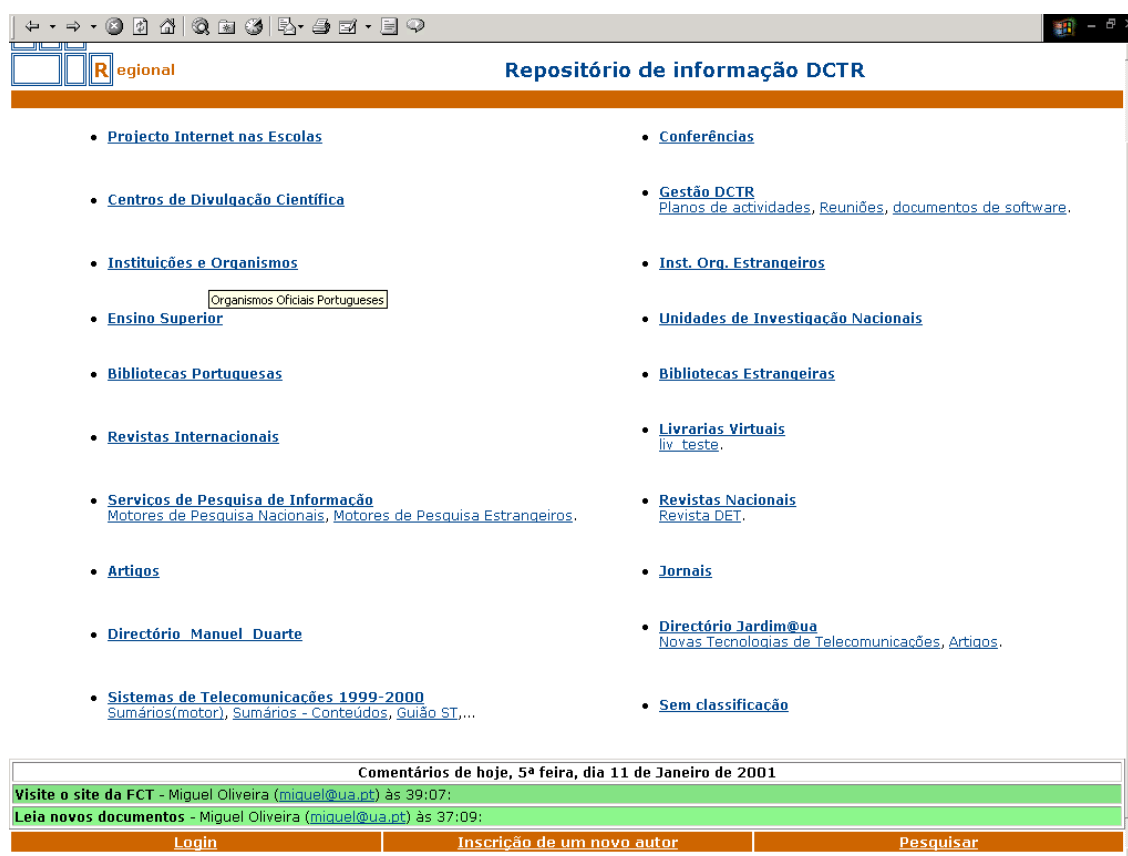


Fig. 33 - Página principal do Repositório de Informação DCTR.

Incidu-se particularmente sobre a problemática da divulgação da cultura científica usando os recursos proporcionados pela Internet e pelas redes telemáticas. Foi dedicada especial atenção à comunicação entre os profissionais da Educação e Investigação (Ensino Superior e Secundário). O RI-DCTR apresenta-se como um sistema de informação *on-line* desenvolvido na Universidade de Aveiro que fazendo uso dos mecanismos de comunicação proporcionados

pela Internet e pelas redes telemáticas, procura incorporar as metodologias clássicas de avaliação de conteúdos de publicações científicas (*peer review*), permitindo o livre acesso à informação armazenada no sistema e também a publicação de documentos, links e eventos de carácter científico por parte de comunidades científicas de interesses comuns.

5.1.2 Objectivos do RI-DCTR

A construção de um sistema de informação deve ser capaz de operacionalizar e enquadrar a passagem para o meio envolvente de resultados e da informação construídos, produzidos ou recolhidos pelo próprio projecto. Este sistema de informação é encarado como um instrumento de trabalho essencial para a prossecução das actividades do projecto DCTR: Base de Dados – Sistema de Informação.

Em termos específicos este sistema de informação deverá proporcionar as seguintes funcionalidades:

- Repositório de informação construída, produzida ou recolhida pelo projecto
- Mecanismo de difusão e divulgação dos resultados do projecto.
- Mecanismo de interacção com entidades com interesses afins.
-

Este sistema de informação será acessível via Internet ficando desta forma disponível a todas as pessoas e instituições com interesses afins aos da problemática do projecto [28].

5.2 *Universal Usability Guide*

5.2.1 Enquadramento do sistema de informação no projecto UUGuide

O principal objectivo do acesso universal à informação e serviços de telecomunicações é envolvente. Entusiastas em telecomunicações, empresas liderantes no sector e governos, neste momento, já conseguem ter uma visão que observa oportunidades e benefícios nesta área embora, mesmo bem sucedidos e com economias de escala apresentem baixos custos, investigadores têm muito trabalho a realizar. Deparam-se com a seguinte questão: Como pode a informação e os serviços de telecomunicações serem utilizados por todos os cidadãos? O desenvolvimento de ferramentas, programas e utilitários para utilizadores experientes e frequentes não é difícil mas o mesmo desenvolvimento para um espectro de utilizadores com poucos conhecimentos técnicos é, provavelmente, muito difícil e poderá ser um grande desafio. Será necessária muita “inspiração” e “transpiração” se pensarmos em 100 engenheiros

de *software* a desenvolverem *software* para 100.000 professores, a ser usado para 100.000.000 alunos.

Os autores de tecnologias antigas, tais como correio postal, telefones e televisão foram bem sucedidos no que diz respeito à usabilidade universal⁵⁷, mas a informática ainda é muito imatura e difícil de usar para muitas pessoas. Numa estatística de 6000 utilizadores de computadores apresenta uma média de 5 horas por semana gastas em tentativas de utilização de computadores, mais tempo gasto do que a navegar na Internet. A frustração e ansiedade dos utilizadores é crescente, e o número de não-utilizadores ainda é elevado. O baixo custo do *hardware*, *software* e das telecomunicações trará muitos utilizadores, mas as interfaces e a estruturação da informação são necessários para adquirir níveis mais altos de sucesso. Desta forma, podemos definir usabilidade universal como tendo mais de noventa por cento de famílias como utilizadores de sucesso de informação e serviços de comunicações pelo menos uma vez por semana[29].

Como agenda principal de investigação, foram desenvolvidos alguns temas como Variedade Tecnológica, ou seja, suporte a uma larga gama de *software*, *hardware* e acesso às telecomunicações; Diversidade de utilizadores, o que significa abranger utilizadores de diferentes níveis, conhecimentos, idade, sexo, disfunções físicas (mobilidade, luz, barulho), cultura, etc.; Falhas no conhecimento do utilizador, i.e., o que é que o utilizador sabe e o que é que o utilizador tem de saber.

5.2.2 Participação no projecto UUGuide

Em Março de 2000, de modo a suportar e desenvolver um *site* que atendesse às especificações e regras de usabilidade universal, para a conferência *Universal Usability*⁵⁸ da ACM⁵⁹, foi apresentado um programa de candidaturas a voluntários entre os quais seriam aceites aqueles que tivessem projectos relacionados com o tema, projectos esses em que o AMIS era evidente. Aos voluntários foi-lhes dada a tarefa “enublada” de criar, desenvolver e desenhar um serviço detalhado de recursos informativos de usabilidade universal. Um desafio a que me prontifiquei a aceitar, colaborar, criar, desenvolver com mais 15 pessoas com quem trabalhava

⁵⁷ Tradução de *Universal Usability*.

⁵⁸ Ver em <http://www.acm.org/sigchi/cuu/>.

⁵⁹ *Association for Computing Machinery*, <http://www.acm.org>.

assincronamente em tele-trabalho, de diferentes idiomas e barreiras culturais, desde Abril de 2000 até Novembro do mesmo ano. Com algum esforço e dedicação, foi apresentado o universalusability.org.

Como responsável pelo grupo técnico, foi possível aplicar no *UUGuide* algumas directrizes e métodos de desenvolvimento na interface do serviço que tinha utilizado no AMIS, entre as quais:

- Método de publicação de documentos através da *Web* pela primitiva POST do HTTP 1.0;
- Classificação dos recursos por árvore temática e palavras-chave;
- Ferramentas externas de pesquisa e indexação de conteúdo estático (SWISH⁶⁰);
- Ferramentas externas de fóruns de discussão (SWIKI⁶¹);
- Cores esbatidas de fundo, títulos em links (TITLE) enormes alternativos em imagens (ALT) e uso de tabelas em detrimento de *frames* para o menu de funções.
-

O site foi apresentado publicamente em no Arlington Towers Hotel, Maryland, Washington D.C., E.U.A, a 16 de Novembro de 2000 (Fig. 34).

⁶⁰ Disponível em <http://homepage.mac.com/paulilucas/software/swish/>.

⁶¹ Disponível em <http://swikis.swiki.net/>.

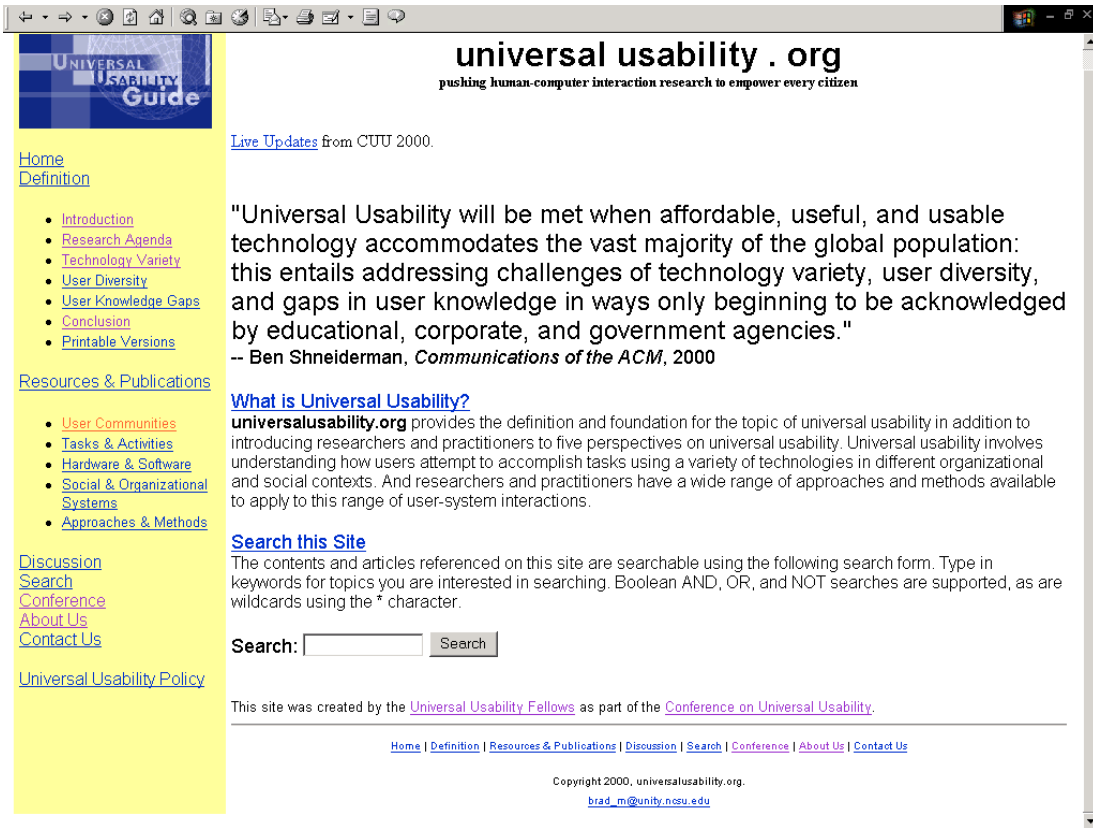


Fig. 34 - UniversalUsability.org

6 Possíveis futuras implementações do Projecto AMIS

6.1 Nota introdutória

Para melhor entender as secções que se seguem, foi efectuado um estudo relativo aos agentes de *software*. As questões e propostas apresentadas, a seguir, baseiam-se em tarefas autónomas, que agem ou reagem consoante determinadas acções, executando tarefas pré-definidas. Para tal, na secção seguinte, é apresentado um estudo com vista a clarificar algumas ideias respeitantes aos agentes de *software*.

6.2 Caracterização dos agentes de software

6.2.1 Definição e especificação de agentes

Em termos gerais, um agente é algo ou alguém que **representa e actua em nome de outros**. Um utilizador poderá utilizar agentes em tarefas comuns (simples ou complexas) sem saber que o está a usar mas, de um modo transparente, conduz o agente a actuar em diversas situações como, por exemplo, tarefas triviais que está habituado a fazer, podendo despende o seu tempo noutras acções não triviais e mais importantes[30].

Os agentes podem ser tão simples como subrotinas ou métodos mas, tipicamente, apresentam-se como entidades com algum poder persistente de controle e selecção (como, por exemplo, processos simples numa só máquina ou processos distintos em máquinas remotas). Uma das principais características de um agente de *software* é a sua expressividade, permitindo a troca de dados, informação lógica, comandos individuais e *scripts* [31].

6.2.1.1 Definição de agente

Numa perspectiva simplista, o termo agente é definido como um sistema de *software* e/ou *hardware* que possui as seguintes propriedades [32]:

- **autonomia:** os agentes operam sem a intervenção directa, humana ou de outros, e que detém algum controlo nas acções que tomam, bem como o seu estado interno de funcionamento;
- **sociabilidade:** os agentes interagem com outros agentes e com humanos através de alguma forma (linguagem) de comunicação;

- **reactividade:** agentes que compreendem o seu ambiente (actividades que se proporcionam) e que actuam quando ocorrem alterações no meio em que estão inseridos;
- **proactividade:** os agentes não só actuam quando há alterações do ambiente em que estão, mas também tomam decisões e executam determinadas tarefas por iniciativa própria;
- **continuidade temporal:** os agentes estão constantemente a executar tarefas para um sistema, utilizador ou grupo de utilizadores;
- **sentido de orientação:** agentes capazes de executar várias tarefas (complexas ou não) e sub-tarefas gerindo da melhor maneira que acharem, aumentando a sua *performance*.

Agentes mais complexos poderão ter uma ou mais das seguintes características:

- **mobilidade:** capacidade de se moverem através de uma rede de computadores (LAN, WAN, Internet,...);
- **tolerância:** agentes que não entram em conflito com aquilo que lhes é destinado ou pedido para fazerem;
- **racionalidade:** agentes que tentam acumular tarefas triviais de utilizadores de modo a que estes não as executem de um modo rotineiro;
- **adaptabilidade:** agentes que se ajustam aos hábitos, métodos de trabalho e preferências dos seus utilizadores;
- **colaboração:** agentes que não só aceitam tarefas, mas que prestam auxílio a utilizadores; agentes que, em casos de falhas ou erros dos utilizadores vitais para a sua actuação, questionam o utilizador ou recorrem ao seu perfil (através de uma base de conhecimento) actuarem com sucesso.

6.2.1.2 Agência

Agência é o grau de autonomia e autoridade investido num agente. Pode ser medido pela qualidade e natureza de acções investidas por um agente e outras entidades no sistema (local ou remoto) onde opera.

Um agente que actua em função de um utilizador é um grau de agência. Outro possível grau de agência é um agente que actua e interage com outras entidades como dados, aplicações ou serviços.

6.2.1.3 Inteligência

A inteligência de um agente pode ser definida como grau de capacidade de aprendizagem e de racionalidade que possui. A inteligência depende de certas características do agente como o perfil do utilizador ou sistema que pretende actuar (base de conhecimento), regras, mecanismos de actuação ou de razão para actuar [33].

6.2.1.4 Agentes de Software

Um agente de *software* é um programa ou uma parte dum programa que serve para auxiliar os utilizadores a realizar tarefas que podem ser executadas através de meios informáticos. É normal designar os agentes de *software* apenas por **agentes**. A diferença entre agentes de *software* e programas vulgares é que os agentes possuem determinadas características, referidas anteriormente, que lhes permite actuarem em nome de outro, através de determinadas regras e preferências.

Os agentes de *software* podem ter um papel importante na maneira de como os utilizadores tiram proveito dos meios informáticos devido essencialmente à complexidade de que muitas das tarefas que, sem qualquer ajuda, por vezes requerem que os utilizadores tenham conhecimentos não triviais sobre o assunto ou pelo menos tenham um treino especial.

6.2.1.4.1 Agentes Inteligentes para Interfaces com utilizador

Os agentes inteligentes para *interfaces* com o utilizador são uma das formas mais conhecidas dos agentes de *software* precisamente porque é a forma que normalmente é mais visível para os utilizadores. Este tipo de agentes actua como meio de facilitar o uso de programas que são controlados através dum interface com o utilizador.

A inteligência dum agente deste tipo deriva da sua **capacidade de se poderem adaptar ao perfil do utilizador** e à maneira de como este pretende realizar as tarefas que estão ao seu alcance através do programa que pretendem usar.

A adaptabilidade do agente inteligente traduz-se na sua capacidade de autonomamente alterar a sua forma de interagir com o utilizador no sentido de maximizar a sua produtividade na realização das tarefas que pretende executar.

Portanto, para além de tentar auxiliar o utilizador, a função do agente inteligente consiste em tentar adquirir informação relevante sobre o perfil do utilizador a partir das suas acções de modo a o poder determinar como ajudar melhor.

Um agente inteligente deverá então:

- Saber que informação relevante poderá obter a partir das acções do utilizador.
- Saber como é que a partir dessa informação poderá determinar o que tarefas o utilizador pretende executar e como pretende que sejam executadas.
- Saber como é que a partir da informação sobre as intenções e preferências do utilizador pode determinar como é que o pode ajudar.

É sobre este tipo de agentes que me debruçarei mais adiante no sentido de avaliar o seu interesse em aplicações que permitam aos utilizadores fazer melhor uso dos recursos da Internet e deles tirar melhor proveito.

6.2.1.4.2 Agentes Distribuídos

Por vezes surgem problemas que devido à sua complexidade torna-se difícil resolver apenas através do recurso a apenas a um agente. Este é caso por exemplo da marcação de viagens cuja rota depende de informação que não só não está disponível no mesmo local, mas também porque o acesso a essa informação está condicionado a autorização de diferentes entidades.

É neste tipo de cenários que o recurso a agentes distribuídos se revela de importância vital. Um conjunto de agentes distribuídos por diversos pontos pode não só ajudar a resolver problemas de elevada complexidade, como também pode minimizar os recursos necessários em cada ponto permitindo resolver os problemas em menor tempo de execução contribuindo para assegurar a fiabilidade de todo o sistema no que toca à sua capacidade para resolver problemas complexos, facilitando a resolução de problemas em que eventualmente nem todos os dados são conhecidos ou os dados que são conhecidos podem não ser rigorosos.

6.2.1.4.3 Agentes Móveis

A resolução de alguns problemas de maior complexidade requer por vezes o acesso e o processamento de grandes quantidades de informação distribuídas por sistemas informáticos localizados em pontos distantes.

O recurso a agentes distribuídos permite a um agente local colaborar com agente em sistemas remotos de modo a aceder a informação disponível nesses sistemas. Mas para isso por vezes é necessário que nos locais remotos existam agentes especializados na extracção de informação necessária que é específica em relação ao problemas que se pretende resolver localmente, de modo a minimizar a quantidade de informação que circula entre os sistemas remotos e o sistema local.

Neste cenário faz todo sentido a existência da possibilidade de fazer deslocar um agente que se possa «deslocar» ao local onde está a informação relevante e aí extrair e processar a informação de modo a obter apenas a informação pertinente à resolução do problema eventualmente já num formato adequado.

A implementação duma plataforma que permita a mobilidade dos agentes impõem alguns requisitos, não só em termos de infra-estruturas de comunicação entre sistemas, mas também de uniformização do *software* do qual são constituídos os agentes propriamente ditos.

Isto significa que os agentes terão de ser constituídos por componentes de *software* que possam executar as tarefas para as quais foram concebidos em sistemas informáticos heterogéneos, ou seja, que possam correr em várias plataformas com sistemas operativos diferentes e processadores diferentes.

6.2.2 Aplicação de Agentes Inteligentes para Internet

6.2.2.1 Agentes de gestão de redes

Uma das aplicações de agentes inteligentes é na gestão de uma rede informática. As actividades entre cliente(es)/servidor(es) intensifica o tráfego numa rede e a complexidade de gestão de uma rede, complexidade que tende a aumentar ao longo do tempo, não só pelo *software* que vai mantendo ou instalando, como também pelo número crescente de utilizadores.

Agentes podem filtrar e tomar determinadas acções e reagir para uma melhor performance do sistema. Também são usados para gerir de uma forma dinâmica uma rede.

6.2.2.2 Agentes de filtragem de informação

Um dos tipos de agentes para interfaces com o utilizador são aqueles que podem assistir a tarefas de filtragem de informação. A sua função é de essencialmente de analisar a informação que o utilizador recebe e classificá-la no seu todo ou em elementos que a compõem.

Como exemplos de aplicação mais comuns temos os filtros de mensagens de correio electrónico (agentes *Anti-Spam*). Na maior parte dos programas de correio electrónico mais sofisticados, existe a possibilidade de configurar o programa de modo que este através dum agente possa classificar e separar as mensagens recebidas de acordo com critérios de selecção definidos pelo utilizador.

Embora já hajam estudos realizados a novos agentes *Anti-Spam*, estes ainda não foram implementados. Os estudos foram realizados de modo a adquirir algum *know-how* e baseiam-se na definição de perfis de utilizadores que através de estatísticas e do grau de agência do agente de modo a que este possa actuar.

A maior parte deste tipo de agentes não são inteligentes porque não se adaptam automaticamente a novos tipos de mensagem que o utilizador possa querer classificar e separar.

No entanto, isso não desmerece o valor deste tipo de agentes, na medida em que frequentemente realizam um trabalho automático de classificação de mensagens que normalmente seria demasiado moroso para ser realizado manualmente pelo utilizador (ex. Selecções de *mailing-lists* para um determinado espaço dedicado).

Dentro deste tipo de agentes, destacam-se aqueles que são capazes de determinar e classificar mensagens que normalmente correspondem a correio electrónico não solicitado (*junk mail*). Normalmente, associado a este tipo de agente existe uma base de dados de conhecimento (*knowledge data base*) que define critérios de selecção para identificação das mensagens que devem ser excluídas e eventualmente eliminadas.

Um problema que este tipo de agentes levanta está relacionado com subjectividade dos critérios de selecção. A análise do conteúdo duma mensagem implica por vezes a análise de texto em linguagem natural (ex: as ferramentas *Thesaurus* dos procesadores de texto) pelo que a sua interpretação pode ser falível.

Um cuidado que deve ser tido em consideração é o de pedir ao utilizador para confirmar a acção a executar associada ao critério de filtragem que o agente julga que foi satisfeito.

Assim, perde-se um pouco o carácter de autonomia do agente e este torna-se **semi-autónomo**. Mas pelo menos não se corre o risco do agente cometer um erro de apreciação que poderia prejudicar o utilizador sem este se ter dado de conta. A utilização de agentes em regime semi-autónomo pode ser uma solução para resolver os problemas de ambiguidade que possa surgir.

Uma alternativa para resolução desses problemas numa forma pelo menos mais automática, é o recurso a técnica de **análise de lógica difusa**, como por exemplo o recurso a **redes neuronais**.

6.2.2.3 Agentes para acesso à informação

Através da Internet, quando um utilizador pretende aceder a informação que não sabe como encontrar, normalmente recorre ao uso de ferramentas de pesquisa (*search engines*) de formas variadas existem abundantemente nesta rede.

No entanto, um problema que surge deriva do facto de por vezes os critérios de pesquisa utilizados serem demasiado vagos. O que acontece é que como resultado das pesquisas, os utilizadores são inundados com inúmeras possibilidades de tal modo que as mesmas se tornam praticamente inúteis e os utilizadores desistem porque não têm nem paciência nem tempo para andar à procura de uma «agulha no palheiro» até porque normalmente sabem que, à partida, não deverão achar a «agulha».

Noutras situações os critérios são tão específicos que as ferramentas de pesquisa não encontram nada. Por vezes isto deriva do facto de que quem pergunta por vezes não conhece o suficiente para saber como se escreve correctamente aquilo pelo qual procura.

O uso de sistemas tolerantes a erros ortográficos poderia ajudar a resolver esta questão, mas normalmente isso também poderia agravar o problema de quando os critérios de pesquisa são demasiadamente vagos, dado poderia contribuir para que fossem retornados ainda mais resultados.

Uma possível solução para tentar resolver o problema do retorno de um número excessivo de resultados seria o recurso de pré-classificação dos recursos de informação existentes de modo a permitir o utilizador escolher dentro de que área classificada quer restringir a sua pesquisa. Isto é feito pela maior parte das ferramentas de pesquisa baseadas em listas classificadas como o Yahoo, SAPO, AEIOU, etc..

Uma possibilidade que poderia ser usada para melhorar este serviços seria a de dar a oportunidade ao utilizador para definir novas áreas de pesquisa que ainda não estavam contempladas, permitindo assim aos sistemas aprender com as necessidades do utilizador.

O trabalho dum possível agente inteligente seria de voltar a analisar a informação classificada existente de modo a determinar que informação corresponde às novas áreas de pesquisa indicadas pelos utilizadores.

Esta tarefa não é trivial, mas com alguma cooperação do utilizador é possível evoluir o sistema de modo que este possa executar pesquisas baseadas em critérios mais personalizados que trariam resultados mais orientados para os interesses dos utilizadores.

Um problema que deriva da necessidade deste tipo de agentes de coleccionar informação sobre o perfil psicológico do utilizador, é o problema da privacidade.

Muitos utilizadores não se sentem confortáveis com a ideia de que estão a ser observados e a ser classificados. Seria sempre necessário informar que a informação recolhida é para beneficiar o seu tipo de utilização.

Mesmo assim há sempre utilizadores que desconfiam, principalmente porque ignoram até que ponto um agente inteligente os pode ajudar. Um exemplo bem conhecido disso é o facto de

que muitos recusam aceitar os *cookies* recebidos dos servidores de *WWW*, normalmente porque não sabem ao certo se é bom ou mau.

Como não compreendem, assumem que se o programa de navegação lhes perguntam se aceitam, é porque pode ser mau e à falta de mais informação negam os *cookies* sem saber o que perdem.

6.2.2.3.1 Exemplos de Agentes de Filtragem de Informação

6.2.2.3.1.1 NetAttaché Pro

O *NetAttaché Pro*⁶² foi desenvolvido pela *Tympani*⁶³ e consegue integrar numa só aplicação diversas tecnologias que estão disponíveis.

Este agente tem incorporado um motor de busca acessível para um utilizador único, móvel ou uma rede intranet que efectua rastreios (*Spider*) internos (na máquina do utilizador ou da rede intranet quando se encontra *off-line*) e externos (na Internet ou noutros domínios, quando está *on-line*). Além disso, é dotado de filtros para inclusão ou extracção de informação e filtros definidos com *keywords* pelo utilizador que extrai informação.

Entre outras características, como a navegação *off-line*, o *NetAttaché Pro* permite adicionar diversos motores de busca, para que a base de conhecimentos tenha um maior leque de fontes de informação.

O *NetAttaché Pro* está disponível em versões *Windows* 16 e 32 bits. Existem duas versões disponíveis, uma para o mercado individual, nomeadamente teletrabalhadores, e outra para grandes e médias empresas.

6.2.2.3.1.2 GoAgent!

Outro agente actualmente disponível no mercado para extrair informação da Internet é o *GoAgent*⁶⁴. Entre várias funções, o programa permite a captura de listas de endereços de *e-mail*, *links* de várias páginas, ficheiros para *download* e catálogos de imagens.

⁶² Disponível em <http://www.tympani.com/>.

⁶³ Disponível em <http://www.tympani.com/>.

As pesquisas são efectuadas de um agente (*Wizard*) que através de vários ecrãs e diversas opções vai conduzindo as escolhas do utilizador. É dotado de um *interface* de simples compreensão, que possibilita a utilização de vários motores de busca e permite que as pesquisas e toda a informação relevante seja guardada em disco, para mais tarde serem consultadas em *off-line*.

O GoAgent! É um programa *shareware* e está disponível para *Windows*.

6.2.2.3.2 Agentes para assistência à realização tarefas conhecidas

6.2.2.3.2.1 Microsoft Bob

Um tipo de agentes para interface com o utilizador que provavelmente é mais utilizado são os agentes para a assistência à realização de tarefas conhecidas. Este tipo de agentes vêm já incorporados em alguns programas conhecidos como *Bob* que é usado no programa *Microsoft Word*⁶⁴.

Como têm o completo conhecimento das capacidades do programa, este tipo de agentes pode estar permanentemente a analisar as sequências de acções que o utilizador pratica para realizar as tarefas que pretende e a partir daí pode intervir pro-activamente, eventualmente fazendo várias tipos de sugestões que podem ajudar o utilizador a realizar as mesmas tarefas de forma mais eficiente, podendo beneficiar o utilizador na medida em que consumiria menos tempo e paciência para realizar as tarefas que pretende.

Uma forma mais reactiva de actuar destes agentes, é aquela que permite ao utilizador interrogar o agente sobre como executar as tarefas que pretende executar.

A análise duma interrogação colocada em forma de linguagem natural levanta os problemas habituais de complexidade e ambiguidade da linguagem humana.

⁶⁴ Disponível em <http://www.goagent.com/>.

⁶⁵ Disponível em <http://www.microsoft.com/catalog/display.asp?subid=22&site=814&cx=35&y=9>.

Em geral os agentes de assistência à realização tarefas conhecidas não são propriamente agentes inteligentes, dado que não aprendem automaticamente com as acções do utilizador.

No entanto, parece haver algum consenso no que toca ao facto de que as tentativas de inclusão de inteligência que se adapta de forma automática, pode levar a efeitos indesejáveis derivados da subjectividade da análise da linguagem natural humana.

Por outro lado, a forma por vezes intrusiva como alguns agentes actuam pode torná-los por vezes importunadores. Esse é o caso de agentes que aparecem sob forma animada como o *Microsoft Bob*, não só porque frequentemente não ajudam grande coisa, como acabam por distrair e desconcentrar o utilizador que tenta realizar as suas tarefas.

6.2.2.3.2.2 Open Sesame!

Outro exemplo deste tipo de agentes é o *Open Sesame!*⁶⁶ Que aprende tarefas virtuais realizadas pelos utilizadores de computadores *Macintosh*⁶⁷. Este agente arquiva tudo o que um utilizador faz e elimina tarefas triviais, poupando tempo ao utilizador, para que cada minuto que este despenda no computador seja produtivo.

O *Open Sesame!* Usa um agente que observa todas as actividades do utilizador e determina (aprende) quais as tarefas que este repete "vezes sem conta", de modo a que essas tarefas repetitivas sejam automatizadas e executadas pelo agente.

O *Open Sesame!* pratica dois tipos de acções: baseado em **tempo** e baseado em **eventos**. Acções baseadas em tempo leva o agente a actuar ao fim de determinado tempo, enquanto acções baseadas em eventos levam o agente a actuar após determinadas tarefas executadas pelo utilizador.

6.2.2.3.3 Outros exemplos de agentes

6.2.2.3.3.1 Newbot

⁶⁶ Disponível em <http://www.opensesame.com/>.

⁶⁷ Ver em <http://www.apple.com/>.

*Newbot*⁶⁸ é um agente de notícias e informação que permite aos utilizadores definirem critérios para receberem informação publicada recentemente. O banner do agente mostra "*What's new on the net today.*".

6.2.2.3.3.2 Jango

*Jango*⁶⁹ é um agente de comércio electrónico disponível no *Newbot*, que permite aos utilizadores encontrarem os melhores preços de livros, CDs, vídeos, etc.. O *Jango* rastreia a Internet e indica sites onde os utilizadores podem fazer as suas compras. O *Jango* foi comprado pelo *Excite* e integrado no *site* na secção de compras .

6.2.2.3.3.3 Firefly

O *Firefly*⁷⁰ é um site que usa a tecnologia de agentes por em contacto utilizadores com os mesmos gostos de música, cinema, e eventualmente noutras áreas de interesse. O *Firefly* foi recentemente comprado pela *Microsoft*⁷¹.

6.2.2.3.3.4 WBI (Web Browser Intelligence)

O *WBI*⁷² (pronuncia-se *Webby*) é uma tecnologia de agentes disponível da IBM. Este agente monitoriza o uso da Internet, sugere "atalhos", captura conteúdo, entre outras funções.

6.2.2.3.3.5 Roboshopper

Este agente é considerado um dos mais úteis na Internet para qualquer tipo de utilizador. É um agente que procura produtos variados e ao melhor preço. Faz compras de uma maneira rápida e eficiente. Procura em várias lojas virtuais e apresenta um relatório completo ao utilizador para que este possa comparar preços, quantidades disponíveis, etc.. *Roboshopper*⁷³ também pode procurar por informações e depoimentos de outras pessoas relativamente ao produto em questão.

⁶⁸ Disponível em <http://www.wirenewbot.com/>.

⁶⁹ Disponível em <http://www.excite.com/>.

⁷⁰ Disponível em <http://www.firefly.com/>.

⁷¹ Ver em <http://home.microsoft.com/>.

⁷² Ver em <http://www.ibm.com/news/>.

⁷³ Disponível em <http://www.roboshopper.com/>.

Roboshopper é um serviço grátis que vive da publicidade que mostra no *site*. É bastante "user-friendly", possui uma lista com as suas funções e capacidades e apresenta diversos motores de busca para procurar informação.

6.2.2.3.3.6 BargainFinder

É uma página da *Web* da *Accenture*⁷⁴ em que os utilizadores podem aceder para comprarem CDs de música ao melhor preço. Podemos também ver os quarenta álbuns mais vendidos assim como a lista de lojas virtuais de música que o *BargainFinder*⁷⁵ procura.

6.2.2.3.3.7 BottomDollar

*BottomDollar*⁷⁶ procura os melhores preços em lojas virtuais de livros, música, filmes, etc.. Os preços são actualizados continuamente dado serem extraídos directamente das lojas.

6.2.3 Uso de agentes em Classificação, Indexação, Rastreo e Interligação de Diversas Fontes de Informação.

6.2.3.1 Panorama actual

Nos últimos anos, temos observado um crescimento exponencial de *sites* na Internet que, na grande maioria, são criados para disponibilizarem informação (fonte: *Network Wizards*⁷⁷). Enquanto antigamente as fontes de informação eram limitadas, as pesquisas e acções para encontrar essas informações era ligeiramente mais acessível, devido a que a grande maioria dessas fontes estavam catalogas.

Hoje em dia, os actuais motores de pesquisa (*search engines*, ex: *Lycos*⁷⁸, *Metacrawler*⁷⁹, etc.) e as listas classificadas (ex: *Yaboo*⁸⁰, *SAPO*⁸¹) não conseguem, por vezes, satisfazer as necessidades de um utilizador. Um exemplo é o caso em que é feita uma pesquisa normal e somos

⁷⁴ Ver em <http://www.accenture.com/>.

⁷⁵ Ver em <http://bf.cstar.ac.com/bf/>.

⁷⁶ Disponível em <http://www.bottomdollar.com/>.

⁷⁷ Disponível em <http://www.nw.com/>.

⁷⁸ Ver em <http://www.lycos.com/>.

⁷⁹ Ver em <http://www.metacrawler.com/>.

⁸⁰ Ver em <http://www.yahoo.com/>.

⁸¹ Ver em <http://www.sapo.pt/>.

"inundados" em informação que não desejamos ou, pelo contrário, quando fazemos uma pesquisa mais restrita e não nos é retornada alguma informação.

O número de motores de pesquisa aumenta de dia para dia, assim como das listas classificadas, que num futuro próximo, não interessará **quem é que fornece** essa informação mas sim **quem a procura** (existem centenas de motores de busca, directórios e listas classificadas por todo o mundo, desde os motores e listas que já nos habituámos a consultar, até aqueles dedicados a uma determinada área ou função[34]).

6.2.3.2 Problemas relacionados com as pesquisas de informação (quem a procura)

Como já referi, a quantidade de informação tem crescido exponencialmente e sem nenhum critério definido. Num sentido figurativo, se já era difícil "encontrar uma agulha num palheiro" no momento, mais difícil é "encontrar algumas agulhas em muitos palheiros".

Os actuais métodos de pesquisa não estão a dar solução para o problema o que torna, por vezes, inútil e desmotivante procurar informação. Esses métodos partem do princípio que um utilizador sabe **o que é** que exactamente procura e **onde é** que essa informação está definida. Uma alternativa a isto é procurar se **existe** ou não tal informação, e **onde** (se disponível) é que se poderá encontrar.

As pesquisas na Internet têm falhado completamente neste sentido por várias razões:

- ***A própria estrutura dinâmica da Internet.*** Não há nenhuma supervisão do crescimento e desenvolvimento da rede. Qualquer pessoa que queira usar ou dispor de informação ou serviços, é livre de o fazer. Por esta razão é que as actuais estatísticas da Internet são baseadas em estimativas;
- ***A natureza dinâmica da informação disponível.*** Informação que hoje está disponível, amanhã pode não estar e vice-versa;
- ***A informação e serviços de informação são heterogéneos.*** A informação é disponibilizada de várias maneiras e em vários formatos. Uma falha nos actuais motores de busca na Internet é que estes actualmente fazem rastreio de informação em servidores *WWW*, pondo à margem todos os outros serviços disponíveis (*FTP, Telnet,...*).

6.2.3.3 Motores de Busca e Agentes

Actualmente, os meios disponíveis de indexação e rastreio de informação é feito por motores de busca. Embora diferentes, o princípio básico da maioria destes é o mesmo: usam ferramentas que rastreiam meta-informação (geralmente consiste em títulos, cabeçalhos e sub-cabeçalhos, cerca de 100 palavras do corpo do documento e das 20 primeiras linhas, armazenados num número restrito de *bytes*) que são armazenados numa base de dados. Para alguém que tente localizar informação, terá de introduzir um número de palavras-chave num formulário, que irá ser procurado nos campos de meta-informação das base de dados.

Apesar dos motores de busca possuírem várias vantagens, também detém várias desvantagens quando comparados com possíveis implementações de agentes:

- Numa pesquisa de informação, pressupõe-se que o utilizador introduz as palavras chave-correctas. Se não o fizer, está sujeito a ser inundado com informação irrelevante. Um agente pode, de uma forma inteligente, procurar informação com ferramentas como o Thesaurus (processadores de texto) analisando assim o contexto da(s) palavra(s)-chave(s). Também pode tolerar erros ortográficos num determinado nível de pesquisas;
- Num motor de busca, a informação é armazenada numa base de dados, concentrando meta-informação de cada fonte encontrada. Por vezes uma consulta consome demasiado tempo, origina muito tráfico e, por vezes, é ineficiente. Por sua vez, um agente pode criar uma base de dados de conhecimento dinâmica, que é actualizada após cada pesquisa. Um exemplo disto é quando um motor de busca retorna informação que já não está disponível, que por vezes é desagradável e desencorajador para quem procura. Um agente, após retornar informação que já não está disponível, deve actualizar a sua base de dados. Actualmente para evitar esta situação, quem disponibiliza informação, mantém uma página de redireccionamento para o novo local onde a informação se encontra.;
- As pesquisas de informação em motores de busca são limitadas à pesquisa na *WWW*. Um agente poderá "poupar" um utilizador de "detalhes hierárquicos" e efectuar rastreios em serviços como *news*, *Telnet*, *FTP*,...;

- Um motor de busca é individualista, apenas procura informação na sua base de dados ou em algumas listas classificadas. Um agente deve comunicar com outros agentes de diferentes serviços e com vários motores de busca.

6.3 Personalização do serviço para utilizadores

Apesar dos seus primeiros passos no mundo da indústria de *software*, as tecnologias de personalização começam a ganhar algum relevo no que toca à promoção de serviços adaptados ao perfil de cada utilizador.

Embora seja crescente a oferta de serviços na Internet, estes não tem desenvolvido e integrado sistemas que consigam cativar, não só utilizadores com poucos conhecimentos técnicos, como também aqueles que, quando se registam num determinado serviço, desejam um leque de funcionalidades mais ou menos abrangente, pretendem ou tem por hábito aceder aos mesmos assuntos temáticos e que cometem por defeito um conjunto de acções consecutivas quando acedem ao serviço. Assim, é “marginalizado” um certo grupo de possíveis e potenciais utilizadores de determinado serviço. Além disso, muitas empresas que disponibilizam este género de serviços limitam-se a criar sistemas idênticos a outros concorrentes e o panorama nacional não é excepção como se tem vindo a verificar.

Mas existem também alguns exemplos de sucesso com as tecnologias de personalização. Um destes caso é o *My Yahoo!*⁸², introduzida a Julho de 1996 no portal *Yahoo!*⁸³ para uma larga escala de utilizadores. Entre várias funcionalidades, o *My Yahoo!* permite a personalização do portal *Yahoo!*, personalização que vai desde a interface de utilização (cores, fundos, modo como a informação deve estar agrupada) e de conteúdos (notícias prioritárias que devem ser visualizadas, informação de TV's, meteorologia, etc.)[35].

⁸² *My Yahoo!*. Uma aplicação, em ambiente *Web*, que permite a personalização da interface do portal *Yahoo!* e de conteúdos. Disponível em <http://my.yahoo.com/?myHome>.

⁸³ *Yahoo!*. Um dos mais conhecidos portal de conteúdos. Disponível em <http://www.yahoo.com/>.

6.3.1 Registo de gostos, experiências e acções comuns dos utilizadores do serviço AMIS

Pretende-se implementar no serviço AMIS uma base de conhecimento relativa aos vários utilizadores que armazene gostos e preferências dos mesmos e registe as acções mais comuns praticadas por estes.

Do ponto de vista esquemático, um perfil paradigmático para o serviço AMIS deverá ser assim (Fig. 35):

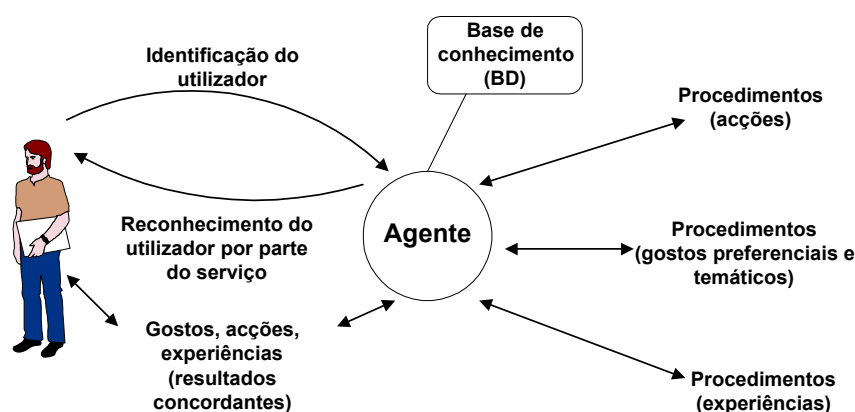


Fig. 35 - Esquema de um possível modelo de personalização do serviço AMIS.

O primeiro passo será a identificação do autor por parte do sistema. Este processo é comum por um nome de acesso e palavra-passe (*login, password*), mas limita o serviço a utilizadores registados (autores). Uma possível implementação será através de *cookies*, em pormenor na secção seguinte.

Um agente de *software* deve registar na base de conhecimento acções, gostos temáticos e experiências concordantes, ou seja, gestos rotineiros comuns praticados pelo utilizador. O agente serve-se da base de conhecimento para apresentar ao utilizador resultados e ou acções que podem ser novos recursos temáticos ou tarefas rotineiras que este tem por hábito praticar (suponha-se o caso de um utilizador que tem por hábito diário publicar um boletim informativo e ver novos recursos informativos na classe de “comunicação social”).

O serviço poderia direccionar-se, quando este se identifica, para a página de publicação de documentos, e ao mesmo tempo apresentar-lhe os últimos recursos informativos dentro da classe “comunicação social” que foram introduzidos desde a última vez que utilizou o serviço).

É nesta linha de pensamento que o agente “aprenderia” com o utilizador. Outro paradigma possível seria o agente aprender as acções do utilizador através de redes neuronais, mas não o vamos fazer porque não se encontra no âmbito deste projecto.

O serviço deve assumir sempre por defeito um perfil geral para todos os novos utilizadores (não autores, autores ou administradores) e deve permitir ao mesmo, definir o seu próprio perfil, no caso de este não desejar um perfil elaborado pelo próprio serviço. Um perfil genérico pré-definido (perfil *default*) é o primeiro passo, ao nível da programação e implementação, para a personalização por parte de um serviço, sendo o ponto de partida para uma robusta e consistente implementação.

Outra possibilidade que poderia ser usada para melhorar este serviço seria a de dar a oportunidade ao utilizador para definir novas áreas de pesquisa que ainda não estavam contempladas, permitindo assim aos sistemas aprender com as necessidades do utilizador.

O trabalho dum possível agente inteligente seria de voltar a analisar a informação classificada existente de modo a determinar que informação corresponde às novas áreas de pesquisa indicadas pelos utilizadores.

Esta tarefa não é trivial, mas com alguma cooperação do utilizador é possível evoluir o sistema de modo que este possa executar pesquisas baseadas em critérios mais personalizados que trariam resultados mais orientados para os interesses dos utilizadores.

Um problema que deriva da necessidade deste tipo de agentes de coleccionar informação sobre o perfil psicológico do utilizador, é o problema da privacidade.

Muitos utilizadores não se sentem confortáveis com a ideia de que estão a ser observados e a ser classificados. Seria sempre necessário informar que os dados recolhidos são para beneficiar o seu tipo de utilização.

Mesmo assim há sempre utilizadores que desconfiam, principalmente porque ignoram até que ponto um agente inteligente os pode ajudar. Um exemplo bem conhecido disso é o facto de

que muitos recusam aceitar os *cookies* recebidos dos servidores de *WWW*, normalmente porque não sabem ao certo se é bom ou mau.

Como não compreendem, assumem que se o programa de navegação lhes perguntam se aceitam, é porque pode ser mau e à falta de mais informação negam os *cookies* sem saberem as facilidades e funcionalidades que lhes podem permitir.

6.3.2 Cookies como ferramenta de reconhecimento de utilizadores do serviço AMIS.

Um *cookie* é uma file de texto (não pode exceder os 4K) que é enviada pelo servidor *Web* para ser armazenada pelo *browser*, para que mais tarde possa ser lido do *browser*, na próxima vez que este utilizador “único” volte a navegar no mesmo servidor *Web*.

Os *cookies* são bastante úteis, para permitir a um *browser* ter informação sobre o seu utilizador tal como, último acesso ao servidor, tempo despendido, preferências do utilizador, etc.. O número limite de *cookies* é de 300 numa pilha *FIFO*.

Os *cookies* são a única forma corrente para um serviço identificar utilizadores anónimos e de “compreender” a sua navegação durante a visita. Além disso dão a possibilidade de oferecer funcionalidades semelhantes às dos utilizadores registados (suponha-se o caso de um utilizador que não deseje registar-se, através de um *cookie*, se o serviço achar conveniente, pode permitir a publicação de documentos ao utilizador anónimo).

Na maioria dos casos um utilizador anónimo continua a ser um utilizador anónimo. Um sistema como o serviço AMIS deve sempre garantir a confidencialidade dos dados do utilizador.

6.4 XML – o grande ícone da informação na Internet

6.4.1 Resumo

Pretende-se nesta secção documentar um estudo teórico-experimental sobre XML⁸⁴, uma derivação de SGML⁸⁵ para enviar, receber e processar informação na *Web*, do mesmo modo

⁸⁴ *eXtensible Markup Language*. Ver em <http://www.w3c.org/XML/>.

⁸⁵ *Standard Generalized Markup Language*. Ver em <http://www.w3c.org/MarkUp/SGML/>.

que o HTML⁸⁶ permite. Neste documento, serão apresentados conceitos e definições de como é que XML deve ser implementado, focando a sua utilidade na criação de meta-informação e de representação de informação. Serão também apresentadas algumas aplicações baseadas em XML para a criação e distribuição de informação (protocolos, ferramentas de permuta de conteúdos, representação de formatos de informação, etc.) e discutidos os vectores tendenciais de desenvolvimento requeridos e pretendidos pelas actuais comunidades de Investigação e Desenvolvimento (I&D).

6.4.1.1 Linguagens de marcas e linguagens de programação

Para melhor entendimento do estudo descrito neste documento, são descritas as diferenças entre linguagens de marcas e linguagens de programação.

As linguagens de programação são dinâmicas, o que significa que processam dados através de cálculos, *queries*, ordenação, *input*, *output* e análise. Os dados e/ou informação introduzida são transformados e apresentados ao utilizador, de maneira a que este os possa usar e manipular. Como alguns exemplos poderemos referir o C, Pascal e Basic.

As linguagens de marcas são estáticas, i.e., não processam informação e um documentos não pode realizar nenhuma operação. No entanto, uma linguagem de programação pode processar e analisar (*parse*) informação num documento com marcas. A principal diferença reside no facto de que uma linguagem de marcas identifica e define parcelas de informação, e oferece alguma inteligência a um documento, para que variadas aplicações possam ler e processar informação de uma forma mais eficiente. Como exemplos podemos referir o HTML, XML e SGML.

6.4.2 Evolução do XML

XML descreve uma classe de objectos de dados, denominados documentos XML, descrevendo parcialmente as funções dos vários programas que processam esses documentos [36]. Embora o nome XML transmita uma ideia de “linguagem de marcas”, a sua aplicação não serve apenas esse propósito: é uma metalinguagem (linguagem de dados sobre dados) que permite a criação de uma linguagem de marcas, específica e personalizada, destinada a vários propósitos e aplicações. Uma linguagem de marcas, por exemplo HTML, permite definir informação de uma determinada maneira e aplicada a um determinado contexto [37]. De um

⁸⁶ *HyperText Markup Language*. Ver em <http://www.w3c.org/MarkUp/>.

ponto de vista simplista, XML permite definir e caracterizar o “nosso próprio HTML” ou outra linguagem de marcas, graças ao seu desenvolvimento e criação ter sido realizado em SGML, *standard* internacional de metalinguagem para linguagem de marcas [ISO8879]. A criação de documentos XML estão de acordo com as normas dos documentos SGML.

Os documentos XML armazenam entidades que contêm dados formatados e não formatados. Os dados formatados são constituídos por informação no seu estado original e por marcas que concentram dados. As marcas contêm descrições de como é que os documentos devem ser apresentados e estão inseridas numa estrutura lógica de dados no documento. A especificação de XML define mecanismos que impõem regras à construção, apresentação e estrutura lógica destes documentos.

A análise de documentos de XML é feita por processadores de XML (*XML parsers*), que actuam em função de linguagens de programação já existentes, na maioria orientadas para a *Web*. Como exemplo podemos referir o Perl, PHP, Java, entre outras. Estes processadores de XML acedem a documentos e extraem o conteúdo e estrutura do documento.

6.4.2.1 História e Objectivos do XML

A criação de um tipo de documentos que se interligava com outros e que poderia ser lido em qualquer parte do mundo, através da Internet pelo programa *Mosaic*⁸⁷, por comunidades científicas deu origem ao HTML. A sua finalidade era a de partilhar informação entre cientistas por todo o mundo. O HTML generalizou-se devido à sua simplicidade, mas ao longo dos anos chegou-se à conclusão que era pouco flexível, o que levou a companhias que produziam programas de navegação (*browsers*) criassem as suas próprias derivações do HTML. Através da experiência acumulada ao longo dos anos, o W3C⁸⁸ concentrou-se na criação de uma nova linguagem de marcas destinada à *Web*.

A criação e desenvolvimento de XML teve início do *XML Working Group*, seguido por *Jon Bosak* (*Sun Microsystems*) e com participação de *XML Special Interest Group* [36] da W3C [38]. Os principais objectivos deste grupo na criação do XML eram:

1. Tornar o XML como padrão para disponibilização de informação na Internet.
2. O XML deve suportar de largo número de aplicações.
3. O XML deve ser compatível com SGML.

⁸⁷ Primeiro programa de navegação na *Web* orientado a documentos HTML, antecessor do *Netscape Navigator*. Ver em <http://www.mozilla.org/> e <http://home.netscape.com/>.

⁸⁸ *World Wide Web Consortium*. Entidade responsável por especificações, desenvolvimento e criação de protocolos comuns que promovam a evolução e interoperabilidade na *Web*. Ver em <http://www.w3.org/>.

4. Deve ser fácil a criação de aplicações de XML para programadores.
5. O número de funções opcionais deve ser mínima, de preferência zero.
6. Os documentos XML devem ser legíveis, do ponto de vista humano.
7. A criação de um documento XML deve ser fácil, imediata, formal e concisa.

6.4.2.2 SGML: A mãe de todas as linguagens de marcas

O SGML surge nos anos 60 de modo a automatizar a definição e formatação de documentos que, até à data, necessitavam de grandes quantidades de tempo para serem processados. O principal objectivo era criar uma *standard* para definir descrições de estruturas, e de conteúdos de diferentes tipos, de documentos em formato electrónico. Desenhado para ser uma linguagem de meta-informação de marcas, o SGML, não contém entidades ou marcas específicas. Em vez disso, é uma linguagem usada para criar e definir essas marcas (*tags*). Por exemplo, criando um DTD⁸⁹, posso definir a marca <FRUTA> para indicar que uma parcela de texto de um documento se refere a fruta: <FRUTA>Laranja</FRUTA>. Outro exemplo é a especificação do tipo de fruta: <FRUTA tipo="citrino">Laranja</FRUTA>.

6.4.2.3 XML versus SGML

Como já foi referido anteriormente, o SGML é o expoente máximo das linguagens de marcas. Permite-nos criar linguagens de marcas como HTML, MathML⁹⁰, MoDL⁹¹ e outras.

Referindo o tipo de documento genérico na Internet, o HTML, podemos afirmar que este é apenas um dos muitos tipos de documentos criados através do SGML: é definido como um único e fixo tipo de documentos com marcas, permitindo criar documentos com cabeçalhos, parágrafos, listas, ilustrações, etc., e também conteúdo multimédia.

O XML é uma abreviação do SGML que facilita a criação e definição de tipos de documentos, tal como o SGML, facilitando o desenvolvimento de aplicações suportadas por parte dos programadores. Além disso, o XML omite as funcionalidades mais complexas e menos usadas

⁸⁹ *Document Type Definition*. Na criação de uma linguagem de marcas através de SGML, é necessário definir um DTD, uma *file* que define as regras que a linguagem a ser criada deve usar. Por exemplo, o HTML foi criado em SGML e é detentor de um DTD que define as regras a usar na criação de um documento HTML. Veremos mais à frente alguns exemplos que clarificam o uso dos DTD's. Ver em <http://www.w3.org/XML/1998/06/xmlspec-report.htm>.

⁹⁰ *Mathematics Markup Language*. Aplicação de SGML para apresentação e definição semântica de modelos matemáticos. Disponível em <http://www.w3.org/TR/REC-MathML/>.

⁹¹ *Molecular Dynamics Language*. Aplicação de SGML para criar e gerir informação molecular. Geralmente a informação é criada em MoDL e visualizada em XML-VRML. Disponível em <http://www.ch.ic.ac.uk/cml/>.

do SGML, de maneira que a criação de aplicações se torne mais fácil de perceber, de distribuir e de operar por parte dos programadores. A validação e análise de conteúdo em XML segue os mesmos requisitos que SGML.

6.4.3 Descrição

Como já foi referido anteriormente, existe um pensamento generalizado de que XML é uma versão poderosa do HTML o que, de facto, não passa de uma ideia errada. Nesta secção descreveremos o XML como uma metalinguagem que permite a criação e definição de linguagens de marcas únicas e personalizadas, orientadas para determinados efeitos, gerais ou específicos. Citando Julia Flanders, directora de projecto do Women Writers Project, “O SGML detém funcionalidades que o XML não possui, mas a maior parte dessas funcionalidades, únicas do SGML, não são usadas”[39]. Nesta secção, apresentaremos exemplos e analogias de documentos SGML e HTML com XML, mas devemos ter em mente que, não só para documentos e informação a ser visualizada, é que beneficiam do XML. Exemplos disso são, por exemplo, estruturas de base de dados em XML, sistemas de pesquisa distribuída em conteúdo XML e protocolos baseados em mensagens XML, discutidos mais à frente.

6.4.3.1 Vantagens e funcionalidades

Associadas às funcionalidades do XML estão vantagens que passamos a enumerar:

1. XML é extensível.
2. XML é construído sobre uma estrutura precisa e aprofundada.
3. XML possui dois tipos de documentos (documentos válidos e documentos bem formados).
4. XML abarca extensões bastante úteis e poderosas.

6.4.3.1.1 A importância da extensibilidade

Do ponto de vista técnico, SGML e XML são meta-linguagens, ou seja, permitem criar linguagens de marcas. O HTML é uma linguagem de marcas criada através do SGML. Os documentos criados em HTML são enriquecidos com *tags*, denominadas por elementos e o autor terá sempre de usar o mesmo conjunto de *tags* para criar documentos em HTML. Por exemplo, para criar um parágrafo um autor terá sempre de usar <p>. Isto é bastante útil e pratico mas vamos supor que um autor precisa de definir novos elementos - HTML não o permite. Com XML isto é possível, devido à sua extensibilidade, e os documentos podem ser adaptados de acordo com o tipo de informação que precisa de ser processada. Por exemplo, se uma loja de comércio electrónico precisa de um documento de encomenda de um produto,

podemos criar um tipo de documento que suporte elementos como <ENCOMENDA>, <NOME_CLIENTE>, <MORADA_CLIENTE> e <VALOR>. Outro exemplo: se tivermos um documento relacionado com química, podemos criar um tipo de documento com elementos como <ATOMO>, <MOLECULA> e <FORMULA>. Em XML, tal como já referimos, não interessa se o documento é um livro, uma encomenda ou outro tipo de estrutura – o que importa é que essa informação pode ser processada. Extensibilidade permite ao autor a possibilidade de enriquecer a estrutura do documento de acordo com a sua necessidade.

Como o HTML falha, em termos de extensibilidade, o criador do documento não pode definir novos elementos que não existam no HTML e que seria úteis em defini-los para serem processados.

6.4.3.1.2 Minimização de marcas

Uma das diferenças do XML e SGML em relação ao HTML é a minimização de marcas[40]. A minimização de marcas significa que alguns elementos não necessitam de ser inseridos por serem reconhecidos implicitamente. Por exemplo, se quisermos adicionar um parágrafo a um documento HTML, adicionamos o elemento <p>, escrevendo o texto, não tendo necessidade de adicionar o elemento </p> para fechar esse parágrafo. Embora facilite a criação de documentos HTML para os autores, a minimização de marcas dificulta o processamento da informação aos programas (*browsers, parsers, spiders*⁹²) e aumenta os custos de desenvolvimento do *software*. Uma das intenções do XML é o de facilitar o processamento da informação a quem cria este tipo de *software*, fornecendo uma estrutura precisa do documento, tal como referido anteriormente em *vantagens e funcionalidades*. Assim, cada elemento de abertura introduzido num documento terá obrigatoriamente de ter o seu elemento de fecho:

```
<p>Isto é um parágrafo</p>
```

Não havendo minimização de marcas, os documentos são criados com estruturas bem definidas, facilitando o trabalho de criação e manutenção de aplicações. Um exemplo desta dificuldade existiu no *Projecto AMIS*⁹³. Uma das funcionalidades a implementar no projecto, era a inserção de *links* em documentos HTML, após um agente analisar expressões em documentos. Devido à minimização de marcas e à tolerância dos *browsers*, muitos documentos

⁹² *Spiders* (aranhas) são agentes que procuram páginas pessoais na *Web*, *Web sites* ou outro conteúdo HTML. São usados por um grande número de motores de pesquisa, tais como *Webcrawler*, *Lycos* ou *Infoseek* para encontrarem novo conteúdo para bases de dados robustas.

⁹³ Projecto AMIS (Agentes Móveis e Inteligentes de *Software*). Projecto financiado pelo Instituto de Telecomunicações – Pólo de Aveiro, PT Inovação e Universidade de Aveiro.

não possuíam marcas de finalização, o que obrigou, ao repensar o algoritmo de análise de expressões, que tolerasse essas falhas. Para que este agente funcionasse na perfeição foi gasto muito mais tempo do que o previsto na implementação e em testes. Quando as estruturas de documentos são bem definidas e combinadas com a extensibilidade, os documentos são mais flexíveis e reutilizáveis. Um exemplo de reutilização é o de um documento que pode fornecer diferente informação a diferentes utilizadores conforme as suas necessidades e permissões. Por exemplo, o mesmo documento de uma encomenda de um livro pode dar diferentes informações ao cliente (livro, descrição) e ao vendedor (*id* do cliente, livro, ISBN).

6.4.3.1.3 Dois formatos de especificação de documentos

O XML permite a criação de dois tipos de documentos:

1. Documentos válidos.
2. Documentos bem-formatados.

Em SGML, um documento necessita sempre de ser definido por um DTD, i.e., uma definição do tipo de documento. O DTD não é nada mais do que as regras a que um documento está sujeito para que este seja válido. O DTD define os tipos de elementos, atributos e entidades a que o documento se deve cingir, aquando a adição de marcas, e que tipo de dados, que não são SGML, podem ser importados para o documento. Além disso, um DTD define a ordem de ocorrência dos elementos. Em documentos com grandes quantidades de informação, como livros e manuais técnicos, um DTD pode ser bastante complexo. Cada vez que um documento é processado, o seu DTD deve ser acedido para que o documento seja analisado para validação. Um documento que respeite todas as regras do seu DTD e da especificação do SGML é um documento válido. Por exemplo, o HTML respeita o seu próprio DTD, incorporado nos *browsers* disponíveis e, apesar de não ser referenciado nos seus documentos, está implícito que esse DTD tem de ser respeitado.

Ao contrário do HTML, o XML restringe-se às regras e definições dos seus DTD's. Se não respeitarmos qualquer uma das regras, o documento deixa de ser válido. O XML não permite que as suas regras eram quebradas.

Enquanto em SGML é sempre necessário a criação de um DTD, em XML é facultativo. Podemos optar por utilizar um DTD para a criação de documentos válidos, ou um documento sem DTD para documentos bem-formatados. Um documento bem-formatado não tem de necessariamente possuir um DTD, mas tem de respeitar duas regras fundamentais. Em primeiro, cada marca de início tem de ter uma marca de finalização. Desta forma, o

documento, por mais simplista que seja, respeita o conceito de estrutura bem definida e robusta. Segundo, cada documento com marcas tem de ter uma marca primária, ou marca de raiz (*root tag*), que contém todos os outros elementos. Elementos como <CLIENTE>, <ENCOMENDA> ou <MORADA> têm de estar obrigatoriamente inseridos na marca primária. Por outras palavras, os elementos e os dados têm de estar bem encadeados. Desta forma, criamos documentos bem-formatados.

6.4.3.1.4 Conversão HTML/XML

Programas de análise de documentos XML bem-formatados sem DTD são fáceis de serem implementados. Se um documento é simples e não precisa de um DTD, ou é uma estrutura simples para ser distribuída na Internet e a largura de banda⁹⁴ é limitada, devemos ter em conta documentos sem DTD. A ideia de documentos bem-formatados surge na altura em que se pensa transferir o conteúdo HTML para XML. Para isso, os documentos HTML terão de ser compostos por:

1. Marcas de início e de finalização.
2. Substituir a declaração DOCTYPE e os seus argumentos (ex: <!DOCTYPE HTML...>) pela declaração XML <?xml version="1.0" standalone="yes"?>.
3. Alterar todos os elementos EMPTY, i.e., elementos de uma só marca, de modo a que acabem com /> (ex: por ou).
4. Assegurar que todos os restantes elementos têm uma marca inicial e uma marca final (ex: <p> e </p>).
5. Os caracteres < e & devem ser substituídos por < e &, respectivamente.
6. Todos os atributos de uma marca têm de estar “entre aspas”.
7. Todas as marcas devem estar encadeadas de uma forma lógica e concisa.

⁹⁴ Largura de banda – em termos técnicos – a “unidade de medida” da capacidade de transporte de informação num meio de comunicação (por exemplo, fibra óptica). Um serviço na Internet que faça uso de dados gráficos (e, por vezes, áudio ou vídeo) necessita de uma quantidade considerável de largura de banda enquanto, por exemplo, um simples serviço de Webmail necessita de uma quantidade consideravelmente inferior.

Para que o conteúdo HTML possa ser lido num browser XML é necessário respeitar as regras anteriores. Além disso existem algumas ferramentas de análise e conversão de documentos para XML como, por exemplo, *The SGML/XML Oasis Page*⁹⁵ e *James Clark's Home Page*⁹⁶.

Os documentos em XML também podem ser convertidos para HTML através do uso de *stylesheets*⁹⁷ como, por exemplo XSL⁹⁸. Através do uso de XSL um documento poderá estar definido em XML e ser convertido para HTML. Segundo o W3C, esta linguagem de estilos será importante para a conversão a longo prazo de informação HTML para XML, pois permite que novo conteúdo criado em XML, possa ser visualizado nos *browsers* da generalidade em HTML.

6.4.3.1.5 Extensões em XML: Estilos e Links

O XML é uma linguagem extensível, como já referimos, e permite a congregação de outras aplicações, descritas em baixo.

6.4.3.1.5.1 Estilos em XML

O conteúdo e estilo em XML são dados separados, o que faz com que o conteúdo seja mais fácil de processar. Para especificar um estilo em XML temos de usar um estilo de página, por exemplo CSS, e XSL. Enquanto CSS⁹⁹ permite definir como é que um documento deverá ser apresentado, XSL permite que o mesmo documento possa ser transformado, por exemplo, em HTML. Um documento pode ter mais do que um estilo (todos do mesmo formato), de modo a ser rentabilizado e aproveitado, de maneira que apenas haja necessidade de alterar o seu estilo e não o seu conteúdo. Isto possibilita que um documento possa ter vários estilos consoante os seus utilizadores. Se um utilizador tem alguns problemas em ler um tipo de letra normal, o estilo poderá providenciar um documento com uma fonte maior.

6.4.3.1.5.2 Links em XML

Embora o mecanismo de interligação de páginas HTML tenha sido implementado de forma simples para a *Web*, poderia ter sido criado de modo a possuir capacidades de interligação mais

⁹⁵ *The SGML/XML Oasis Page*. Disponível em <http://www.oasis-open.org/cover/xml.html>.

⁹⁶ *James Clark's Home Page*. Disponível em <http://www.jclark.com/>.

⁹⁷ *Stylesheets* definem o modo como documentos são apresentados no écran, em papel ou como são pronunciados. Ver em <http://www.w3.org/Style/>.

⁹⁸ *eXtensible Style Language*. Disponível em <http://www.w3.org/Style/XSL/>.

⁹⁹ *Cascading Style Sheets*. Disponível em <http://www.w3.org/Style/CSS/>.

poderosas e práticas. Em XML, os *links* são apresentados na forma de *Xlink*¹⁰⁰. Os *links* podem descrever vários tipos de recursos assim como múltiplas ligações em vez de apenas uma, como é feito em HTML. O HTML detém simples e fracos mecanismos de procura dentro de um documento enquanto que, ao contrário do HTML, XML apresenta a funcionalidade *Xpointer*¹⁰¹ que permite a busca de elementos a qualquer granularidade do documento que autor requeira [36].

6.4.3.2 Manipulação e gestão de documentos e conteúdo

Um documento é composto por *entidades* e *elementos*, podendo estes ser reutilizados sempre que tal seja necessário. Elementos são parcelas de informação que compõem a estrutura lógica do documento, e são de extrema importância para a manipulação e gestão do mesmo. Entidades são a estrutura física do documento. A manipulação de um documento XML, pode ser feita através de DOM¹⁰², através de linguagens como *Javascript*¹⁰³. Através de DOM, o conteúdo de um documento, a sua estrutura lógica e o estilo, pode ser processado usando elementos e atributos como objectos.

O princípio base de manipulação de conteúdo é simples e o XML sustenta esta necessidade: Escrever uma vez, usá-lo em qualquer sistema de informação, independentemente dos métodos usados. Estando “marcado” o conteúdo em XML, a manipulação poderá ser feita de modo intermédio com aplicações proprietárias como, por exemplo, processadores de texto. A seguinte secção, dá-nos um exemplo de manipulação e gestão de um documento em XML.

6.4.3.2.1 Um exemplo simples de manutenção e gestão de documentos em XML

Nesta secção é apresentado um exemplo simples de uma empresa que vende CD's de música pela Internet.

```
<?xml version="1.0" standalone="no" encoding="UTF-8"?>
<DOCTYPE encomenda SYSTEM "Encomenda.dtd">
<Encomenda>
```

¹⁰⁰ XML Linking Language. Disponível em <http://www.w3.org/TR/WD-xlink/>.

¹⁰¹ XML Pointer Language. Disponível em <http://www.w3.org/TR/WD-xptr/>.

¹⁰² Document Object Model. Disponível em <http://www.w3.org/DOM/>.

¹⁰³ Linguagem de programação que define elementos, atributos, entidades e conteúdo de um documento como objectos. Uma das suas fortes aplicações é a validação de informação de um documento do lado do cliente. Ver em <http://developer.netscape.com/docs/manuals/index.html?content=javascript.html>.

```
<Cliente>
  <NomeCliente>
    <PrimeiroNome>Miguel</PrimeiroNome>
    <UltimoNome>Oliveira</UltimoNome >
  </NomeCliente>
  <MoradaCliente>
    <Rua>Delírio dos bytes, 128</Rua>
    <Cidade>Baitólândia</Cidade>
    <CodigoPostal>9600-256</CodigoPostal>
    <Pais>Portugal</Pais>
  </MoradaCliente>
  <Email>miguel.oliveira@acm.org</Email>
</Cliente>
<CD SerialNo="102394586" STOCK="C7423">
  <Artista>Carl Orff</Artista>
  <Titulo>Carmina Burana</Titulo>
  <Valor>1900</Valor>
  <MetodoPagamento Tipo="Visa"/>
</CD>
</Encomenda>
```

Embora seja um documento simples, contém informação relevante que pode ser facilmente processada pela razão de estar implementada em XML. Por exemplo, o serviço pode, eventualmente, verificar que género de cd's o cliente costuma comprar, para informá-lo automaticamente de novos cd's de música do mesmo estilo vão entrando no armazém. Isto pode ser feito através de processos automatizados em XML, *Javascript* ou até através de agentes inteligentes. Através do uso de CDF¹⁰⁴, uma linguagem de marcas criada em XML, é possível implementar a funcionalidade descrita. Outra possibilidade é a de automatizar o processo de encomenda. Devido ao CD possuir os atributos de número de série e de *stock*, é mais fácil de implementar um sistema de encomenda automático.

A primeira marca do documento define a versão XML corrente (1.0 é a corrente) e especifica o DTD a ser referenciado. Devido ao documento possuir um DTD, é que o documento não é

¹⁰⁴ *Channel Definition Format*. Especificação que permite a um *Web publisher* oferecer conteúdos de informação actualizados, ou canais de informação, a partir de qualquer servidor *Web* para programas de recepção compatíveis com o objectivo de entrega automática de informação. Disponível em <http://www.w3.org/TR/NOTE-CDFsubmit.html>.

standalone, e dá a conhecer ao programa de análise o esquema de codificação utilizado, neste caso, “UTF-8”.

Diferentes documentos podem ser enviados a diferentes actores que interagem na encomenda do CD. Por exemplo, o empregado encarregue do envio do CD poderá apenas receber os dados do cliente (nome, morada), enquanto o empregado responsável pelo pagamento recebe o método de pagamento e o nome. Outros elementos podem ser adicionados como a <DataEnvio>, <DataPrevistaRecepção> e <Transporte>.

Uma questão pendente é relativa aos tipos de dados são introduzidos no documento. Neste exemplo, tudo tem de ser especificado num DTD.

```
<!--DTD criado para encomendas de uma loja virtual de CDs-->
<!ELEMENT Encomenda (Cliente, CD+, MetodoPagamento)>
<!ELEMENT Cliente (NomeCliente, MoradaCliente, Email)>
<!ELEMENT PrimeiroNome (#PCDATA)>
<!ELEMENT UltimoNome (#PCDATA)>
<!ELEMENT MoradaCliente (Rua, Cidade, CodigoPostal, Pais)>
<!ELEMENT Rua (#PCDATA)>
<!ELEMENT Cidade (#PCDATA)>
<!ELEMENT CodigoPostal (#PCDATA)>
<!ELEMENT Pais (#PCDATA)>
<!ELEMENT Email (#PCDATA)>
<!--Informação relativa ao CD-->
<!ELEMENT CD (Artista+, Titulo, Valor)>
<!ATTLIST CD SerialNo CDATA #REQUIRED STOCK CDATA #REQUIRED>
<!ELEMENT Artista (#PCDATA)>
<!ELEMENT Titulo (#PCDATA)>
<!ELEMENT Valor (#PCDATA)>
<!ELEMENT MetodoPagamento EMPTY/>
<!ATTLIST MetodoPagamento Tipo (MasterCard|AmericanExpress|Visa)
#REQUIRED>
<!--End DTD-->
```

Todos os elementos dentro do documento têm de ser declarados. O elemento principal ou raiz é Encomenda, e todos os outros elementos têm de estar inseridos neste elemento. O conteúdo de um elemento têm de estar dentro de parênteses. O sinal “+” significa que um determinado elemento ocorre pelo menos uma vez e pode ocorrer várias vezes. Existem

outros sinais tais como: o asterisco “*”, que significa que um elemento ocorre zero ou mais vezes; o ponto de interrogação “?”, que significa que um elemento ocorre zero ou uma vez. Por exemplo, um livro pode ou não ter um prefácio (?) e pode ter um ou mais apêndices (*). Como podemos observar no exemplo anterior, o elemento CD possui dois atributos: O `SerialNo` e o `Stock`. Estes atributos fornecem informação relevante ao elemento e, embora não sejam geralmente visualizados, são processados pelo sistema de informação. O atributo “#PCDATA” é o texto ou informação XML a ser inserida como dados. Geralmente esta informação é visualizada pelo utilizador.

Os atributo detém tipos, tais como texto, *tokens*, *id* ou enumerações. Por exemplo, o elemento `MetodoPagamento` possui o atributo `Tipo` que é uma enumeração de vários tipos possíveis de pagamento. O utilizador tem de obrigatoriamente (tipo #REQUIRED) escolher um tipo de pagamento *Visa*, *American Express* ou *MasterCard*. Alguns atributos possuem o valor IMPLIED, i.e., o utilizador não tem de os inserir podendo estes serem fornecidos por outra aplicação. O valor FIXED indica a um utilizador a necessidade de usar sempre o mesmo dado. Por último o valor DEFAULT que indica que há sempre um determinado valor por defeito num elemento.

Os elementos e atributos são a estrutura lógica de um documento XML. Um DTD bem concebido pode permitir um processamento eficiente de informação e dissemina dados como resultados.

Outra vantagem do XML é a possibilidade de selecção de vários métodos de codificação. Se uma companhia internacional necessita de usar Grego, isto não é problema porque XML usa *Unicode*¹⁰⁵, *standard* para criar informação em qualquer linguagem existente.

6.4.3.2.2 A estrutura física

Outra das potencialidades do XML é, tal como programação orientada a objectos, o documento pode ser organizado em componentes. Em XML a estrutura física ou componentes são denominados por *entidades*. Entidades, tais como elementos e atributos, têm de ser declarados no DTD. Entidades podem ser qualquer tipo de informação como texto ou gráficos. A entidade geral é uma funcionalidade bastante útil em XML (e SGML) e bastante

¹⁰⁵ O *standard Unicode* contém um vasto número de caracteres, de maneira a suportar todos os caracteres existentes no mundo. Também possui caracteres compatíveis com antigos caracteres de codificação e caracteres de controlo de funções., assim como a especificação dessas funções. Ver em <http://www.w3.org/TR/1999/WD-unicode-xml-19990928/>.

poderosa que nos pode poupar tempo e dinheiro. Entidades gerais são representações simbólicas de parcelas de texto reutilizadas noutros blocos de texto. Por exemplo, em repartições públicas e outras instituições onde a informação burocrática é uma constante, muitos trechos de informação são comuns à maioria de impressos e declarações a preencher por contribuintes. Estas parcelas de texto podem ser reproduzidas usando uma entidade geral. Veremos como isto pode ser feito:

Em primeiro, a entidade geral tem de ser declarada num DTD. Vamos supor um contrato de arrendamento em que é declarado que o inquilino aceitou o contrato.

Declarando a entidade num DTD:

```
<!ENTITY CA "Após a aceitação de todas as condições do contrato de arrendamento, _____ o _____ inquilino _____ compromete-se a aceitar todos os requisitos e condições do mesmo, para efeitos de usufruto da moradia _____, Rua _____, número _____, _____ em _____ a partir de ____ de _____ de _____.">
```

A sigla "CA" é apenas para lembrar que a entidade define um contrato de arrendamento. A entidade poderia ter sido definida como "<!ENTITY ContratoArrendamento \"...\">".

A partir de agora, o funcionário que opera o sistema de informação, em vez de rescrever todo o texto, apenas necessita de introduzir os dígitos:

&CA;

Quando a aplicação do sistema de informação ler a entidade &CA apresenta:

```
"Após a aceitação de todas as condições do contrato de arrendamento, o inquilino _____ compromete-se a aceitar todos os requisitos e condições do mesmo, para efeitos de usufruto da moradia _____, Rua _____, número _____, _____ em _____ a partir de ____ de _____ de _____."
```

A entidade geral também pode estar gravada numa *file* externa e ser referenciada como tal. A entidade geral também pode ser bastante útil quando é distribuída por muitos documentos e através de uma rede e, além disso, poupa alguma largura de banda. Vamos supor que temos de

disseminar o contacto e endereço de uma empresa para uma enorme quantidade de documentos. Inicialmente teríamos de alterar cada um desses documentos “à mão”, gastando tempo, dinheiro e recursos humanos. Com uma entidade geral é bastante mais fácil de realizar esta operação e automatizar o processo. Se o contacto de uma empresa for armazenado como uma entidade geral, poderíamos implementar desta forma:

```
<!ENTITY ContactoEmpresa "Delírio dos bytes, 128. Baitólândia, 9600-256.
Telefone 234126256">
```

Assumindo que é feita uma mudança do endereço, alterando apenas a entidade geral no DTD:

```
<!ENTITY ContactoEmpresa "Delírio dos bits, 256. Baitólândia, 9600-128.
Telefone 234126256">
```

O novo contacto aparecerá em todos os documentos do sistema. Outra vantagem do uso de entidades gerais é o de não haver duplicados do endereço. Vamos supor que uma ou mais pessoas necessitam de trabalhar num mesmo documento. Se o documento for dividido em secções ou capítulos, dependendo da informação a que se refere, estas podem ser declaradas como entidades, permitindo aos autores trabalharem no mesmo documento em secções individuais, sem necessitarem do documento na totalidade. A secção pode ser declarada no DTD como:

```
<ENTITY Sec1 SYSTEM "Sec1.xml">
```

Para aceder à secção 1 do documento:

```
<Documento>
<Secção>&Sec1</Secção>
</Documento>
```

Assim que o processador ler a entidade Secção 1 envia ao autor o texto da Secção 1 enquanto outro autor pode trabalhar noutra secção.

6.4.3.3 Exemplos de consolidação de ideias

6.4.3.3.1 MoDL

Uma das primeiras comunidades a utilizar as potencialidades do XML foi a comunidade de investigação de química. A linguagem de marcas CML, que antecedeu o XML, é uma ferramenta genérica para manipular informação técnica e molecular.

A combinação de XML com VMRL¹⁰⁶ já foi usada previamente para a visualização de dados do mercado de acções e mapas de navegação em *sites Web*. Neste ultimo caso, as páginas são descritas com um conjunto de marcas e de dados, convertidas para VRML oferecendo possibilidades de navegação em 3D.

MoDL permite construir documentos com elementos como átomo, molécula, radical, fronteira, etc.. Usando esta estrutura, cientistas podem construir macromoléculas complexas de uma maneira bastante fácil. A entidade DEFINE permite aos utilizadores definir tipos particulares de átomos, moléculas e fronteiras e usá-los posteriormente para os instanciar como objectos desse tipo. Por exemplo, Oxigénio e Hidrogénio podem ser definidos desta forma:

```
<DEFINE type='atom' name='O' radius='0.3' color='0 0 1' />
<DEFINE type='atom' name='H' radius='0.2' color='1 0 0' />
```

Neste caso o átomo de hidrogénio é azul e tem raio 0.3 e o hidrogénio é vermelho e tem raio 0.2.

Usando a definição anterior em MoDL, podemos definir, por exemplo, uma molécula de água (H₂O):

```
<DEFINE type='molecule name='Water' />
<atom type='O' id='o1' position='0 0 0' />
<atom type='H' id='h1' position='-2 10 0.5' />
<atom type='H' id='h2' position='2 1 0.5' />
<bond from='h1' to='o1' />
<bond from='h2' to='o1' />
</DEFINE>
```

A especificação de bond, define fronteiras entre átomos, bastando para isso especificar os id's entre átomos. A simulação VRML é a imagem em baixo (Fig. 36):

¹⁰⁶ *Virtual Reality Modelling Language*. Standard ISO para representação de conteúdo tridimensional na *Web*. Ver em <http://www.vrml.org/Specifications/VRML97/index.html>.

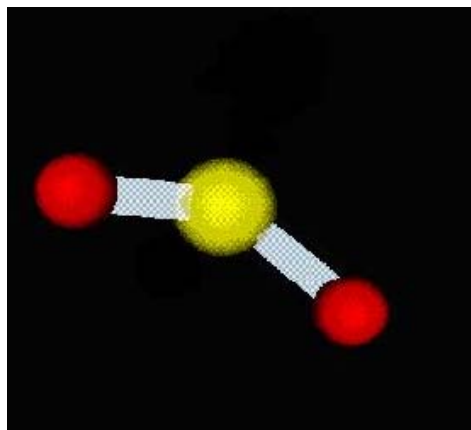


Fig. 36 – Molécula da água. Descrição em MoDL, convertida em VRML

A conversão de XML (MoDL) para VRML é feita através da linguagem de interpretação de *scripts* Perl¹⁰⁷, com um script que usa o módulo *XML::Parser*. *XML::Parser* é a interface em Perl para *expat*, uma biblioteca em C escrita por *James Clark*.

A combinação de XML com VRML demonstra um meio eficiente de transporte de informação química na Internet. A visualização de simulações químicas melhorou e progrediu com o aparecimento do VRML, permitindo observar simulações em tempo real [40]. Outras áreas como arquitectura e planeamento também podem melhor desta tecnologia.

6.4.3.3.2 Generic XML Parser Class¹⁰⁸ para PHP3

O PHP é uma linguagem de programação interpretada que gera páginas dinâmicas. Esta linguagem surgiu como um *script* de *Perl* por *Rasmus Lerdorf*¹⁰⁹ para criar as suas páginas dinâmicas. Ao longo de quatro anos foi desenvolvida pela comunidade *OpenSource*¹¹⁰ de PHP, tornando-se independente, já estando disponível na versão 4.0beta [41].

Na versão 3.0 foi implementada uma extensão de suporte do *expat* de *James Clark*. Este conjunto de funções de *handlers* possibilitam a análise de documentos XML sem validação. Além disso suporta os esquemas de codificação US-ASCII, ISO-8859-1 e UTF-8. A extensão XML do PHP foi implementada para facilitar a criação de *parsers* para posteriormente definir *handlers* para diferentes eventos de XML.

¹⁰⁷ *Practical Extraction Report Language*. Linguagem de programação interpretada. Nesta linguagem, os algoritmos não são compilados mas sim validados por um interpretador. Disponível em <http://www.perl.com>.

¹⁰⁸ Disponível em <http://phpclasses.upperdesign.com/browse.html/package/4>.

¹⁰⁹ Engenheiro de *Software* na IBM, criador do PHP.

¹¹⁰ Grupos de desenvolvimento de *software* em que o código encontra-se disponível a qualquer programador que o queira melhorar ou actualizar.

Através das funcionalidades XML implementadas no PHP, *Manuel Lemos*¹¹¹ desenvolveu uma classe em PHP, *Generic XML Parser Class (GXMLPC)*, para análise arbitrária de documentos XML. Após a análise, é criado um *array* com toda a estrutura de marcas e de dados dos elementos. Opcionalmente, armazena as posições dos vários elementos para determinar a sua localização, no caso de estarem errados contextualmente.

A melhor aplicação desta classe é para criação e gestão de bases de dados. Geralmente para criar uma base de dados em UNIX, são utilizados *scripts*, por vezes bastante extensos. Quando surgem alterações nessa base de dados, para as alterar somos obrigados a alterar o *script* ou editar manualmente a(s) tabela(s) no programa cliente. Usando o *parser* GXMLPC e criando apenas um documento XML é possível poupar bastante tempo, transferindo acções manuais para acções automatizadas. Um exemplo de um documento XML para ser analisado pelo GXMLPC é o seguinte:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<DATABASE>
  <NAME>test</NAME>
  <CREATE>1</CREATE>
  <TABLE>
    <NAME>users</NAME>
    <DECLARATION>
      <FIELD>
        <NAME>user_id</NAME>
        <TYPE>INTEGER</TYPE>
        <DEFAULT>0</DEFAULT>
        <NOTNULL>1</NOTNULL>
      </FIELD>
      <FIELD>
        <NAME>user_name</NAME>
        <TYPE>TEXT</TYPE>
      </FIELD>
      <FIELD>
        <NAME>password</NAME>
        <TYPE>TEXT</TYPE>
```

¹¹¹ Engenheiro de Software da *UpperDesign*. Ver em http://www.e-na.net/the_author.html.

```
</FIELD>
<FIELD>
  <NAME>name</NAME>
  <TYPE>TEXT</TYPE>
</FIELD>
<FIELD>
  <NAME>email</NAME>
  <TYPE>TEXT</TYPE>
</FIELD>
<INDEX>
  <NAME>users_id_index</NAME>
  <UNIQUE>1</UNIQUE>
  <FIELD>
    <NAME>user_id</NAME>
  </FIELD>
</INDEX>
</DECLARATION>
</TABLE>

<SEQUENCE>
  <NAME>user_id</NAME>
  <START>1</START>
  <ON>
    <TABLE>users</TABLE>
    <FIELD>user_id</FIELD>
  </ON>
</SEQUENCE>

<TABLE>
  <NAME>groups</NAME>
  <DECLARATION>
    <FIELD>
      <NAME>name</NAME>
      <TYPE>TEXT</TYPE>
    </FIELD>
    <FIELD>
      <NAME>owner_id</NAME>
      <TYPE>TEXT</TYPE>
    </FIELD>
```

```

<FIELD>
  <NAME>description</NAME>
  <TYPE>TEXT</TYPE>
</FIELD>
</DECLARATION>
</TABLE>
</DATABASE>

```

O formato do documento é simplificado para usar apenas marcas sem atributos. Está a surgir um novo movimento para simplificar os formatos de XML nesse sentido sob o argumento que as pessoas entendem melhor como se constrói a informação. Não são precisos DTDs nem validadores. Este é um exemplo básico apenas com três tabelas, uma das quais com um índice e também existe uma sequência associada para gerar números identificadores únicos para usar no campo `user_id` da tabela `users` e outras relacionadas.

Através da extensão *expat*, a GXMLPC processa toda a informação do documento e analisa todos os dados a processar. Através do esquema em baixo (Fig. 37) podemos ter uma noção de como é que o processamento é feito:

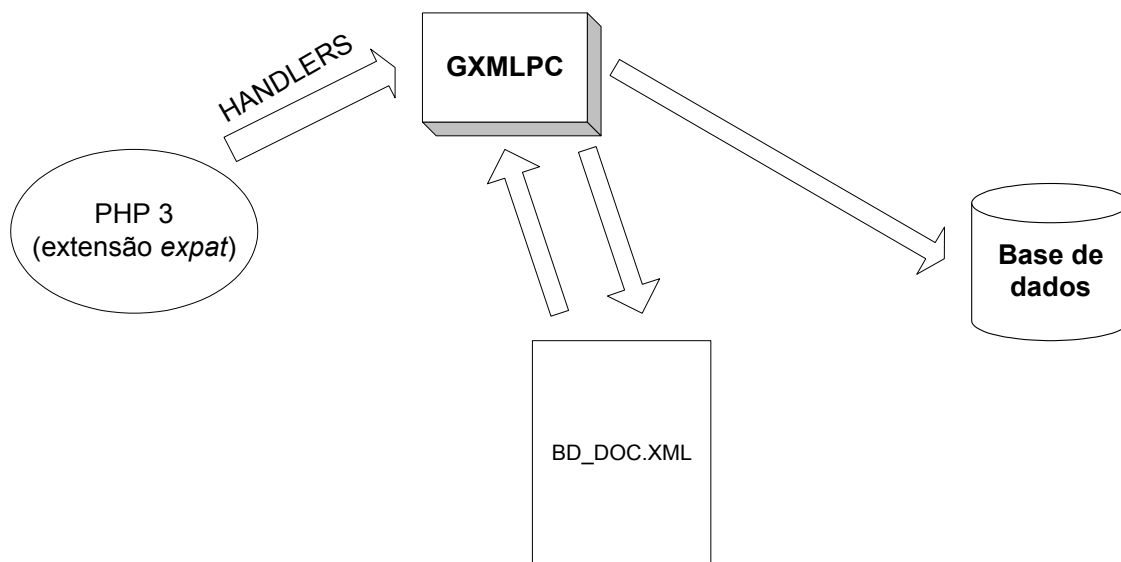


Fig. 37 – Esquema genérico de processamento GXMLPC-PHP3

Podemos concluir que, através da GXMLPC, a criação de bases de dados é muito mais simples, bastando para isso um único documento de XML para definição da sua estrutura. No caso de alterarmos ou adicionarmos um campo, bastará alterarmos ou adicionarmos o elemento correspondente no documento XML.

6.4.4 Aplicações

Nesta secção serão discutidas algumas aplicações correntes derivadas de XML. Embora já tenham sido apresentadas algumas aplicações de XML, estas eram destinadas à criação de informação em novo formato (MoDL e GXMLPC). As tecnologias descritas em baixo, relacionam o XML com vista à implementação de sistemas distribuídos, o que significa que a partilha, colecta e distribuição de dados também pode ser melhorada implicando que os actuais paradigmas de tecnologias de sistemas distribuídos não se tem comportado à altura. Isto significa que, não só que as tecnologias intituladas de “plataforma independente” para partilha de dados, como também os métodos de implementação têm de ser bastante específicos, de máquina para máquina, restringindo assim código a aplicações específicas. Outro problema é que as actuais distribuições de *software*, por exemplo CORBA¹¹², variam consideravelmente de distribuição para distribuição dificultando a implementação de algoritmos e obrigando a alterações de código.

6.4.4.1 XML-RPC

6.4.4.1.1 Especificação

XML-RPC é um protocolo RPC¹¹³ que é usado na Internet [42]. Existe um vasto número de formatos possíveis para implementar RPCs, sendo XML-RPC um destes formatos possíveis. A vantagem do XML é que tanto humanos como computadores a podem entender facilmente. O XML-RPC usa XML, como formato padrão de partilha de dados para invocar RPCs e, permite invocar procedimentos de máquinas de plataforma, aplicação e linguagem diferente. Além de ser fácil a sua implementação, também mais fácil de ser processada.

Uma mensagem XML-RPC é um *request* HTTP¹¹⁴ usado através da primitiva POST¹¹⁵. O corpo da mensagem é definido em XML. Um procedimento que seja executado num servidor, retorna valores formatados em XML. Os parâmetros de um procedimento podem ser

¹¹² *Common Object Request Broker Architecture*. Linguagem de definição de interfaces que permite a comunicação de aplicações, dados e métodos, onde quer que estas se encontrem. Disponível em <http://www.omg.org/corba/whatiscorba.html>.

¹¹³ *Remote Proceeding Call*. Extensão de procedimento que permite a comunicação entre procedimentos de diferentes aplicações e diferentes máquinas. Ver em <http://davenet.userland.com/1998/07/14/xmlRpcForNewbies>.

¹¹⁴ *HyperText Transfer Protocol*. Protocolo de transferência de dados e partilha de informação na *Web*. Ver em <ftp://ftp.isi.edu/in-notes/rfc1945.txt>.

¹¹⁵ Método usado para transferir informação na *Web* através do HTTP. Permite enviar/receber múltiplos dados. Ver em <ftp://ftp.isi.edu/in-notes/rfc1945.txt>.

escalares, números, *strings*, etc. e podem ser complementados com *records* e estruturas. O exemplo em baixo mostra um *request* em XML-RPC a um servidor:

```
POST /RPC2 HTTP/1.0
User-Agent: Frontier/5.1.2 (WinNT)
Host: deliro.bytes.bl
Content-Type: text/xml
Content-length: 181

<?xml version="1.0"?>
<methodCall>
  <methodName>examples.getStateName</methodName>
  <params>
    <param>
      <value><i4>41</i4></value>
    </param>
  </params>
</methodCall>
```

O URI na primeira linha do cabeçalho da mensagem não é especificado. O *User-Agent*, *Host* e *Content-Type* têm sempre de ser definidos, e o *Content-Length* têm de ser especificado e estar correcto.

Neste exemplo é feito um pedido de invocação de um procedimento denominado *examples.getStateName* (determinar o nome de um estado norte-americano através de um id) com os parametros *ia=41*. Em baixo é apresentada a resposta:

```
HTTP/1.1 200 OK
Connection: close
Content-Length: 158
Content-Type: text/xml
Date: Fri, 17 Jul 1998 19:55:08 GMT
Server: UserLand Frontier/5.1.2-WinNT

<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value><string>South Dakota</string></value>
    </param>
```

```
</params>  
</methodResponse>
```

O cabeçalho cumpre as regras padrão do protocolo HTTP, já referido. A resposta do procedimento invocado vêm formatada XML.

Como já referido anteriormente, o XML-RPC, através do formato em XML, facilita o processamento de dados devido à simplicidade de reconhecimento do seu conteúdo, e permite que os actuais *parsers* de XML das várias linguagens beneficiem deste protocolo.

6.4.4.1.2 Vantagens e Objectivos

Um dos principais objectivos deste protocolo é o de fornecer uma especificação de permuta de dados independente da plataforma e aplicação, sem necessitar de actualizações de *software* nas várias aplicações e plataformas. Além disso, *proxies* e *firewalls* podem permitir a transferência de dados, tendo apenas em conta se o Content-Type é text/xml.

Outra vantagem é a simplicidade de envio e recepção de dados em XML, como já foi referido. Deverá ser fácil para um codificador de HTML procurar uma *file* com a invocação de um procedimento XML-RPC, determinar o que é pretendido e, modificá-lo se for necessário, executando-o na próxima vez em que este for invocado.

Por último, é um protocolo que permite a implementação de aplicações de uma forma bastante fácil e simples. Alguns sistemas de pesquisa distribuída, com uma federação de repositórios, apresentam resultados de pesquisas colectadas a partir de vários repositórios. Um exemplo é o serviço de pesquisa *Metacrawler*¹¹⁶, um motor de pesquisa, que colecta pesquisas através de outros motores de pesquisa e listas indexadas. O *Projecto AMIS* pretende utilizar esta tecnologia para obter dados colectados de uma federação de repositórios.

6.4.4.2 ICE

6.4.4.2.1 Introdução

Um dos problemas que tem surgido até hoje é relativo à maneira como jornais *on-line*, lojas virtuais e portais da *Web* poderão actualizar e disponibilizar conteúdo actualizado. Até ao momento, este problema era solucionado criando conteúdo original ou adquirindo conteúdo

¹¹⁶ Disponível em <http://www.metacrawler.com/>.

de várias fontes de informação, demarcando o conteúdo com o nome próprio [43]. Por exemplo, o jornal *Público* adiciona na última uma pequena banda desenhada “Calvin & Hobbes” que compra à *Marvel*. A loja virtual Inforlândia cria as suas páginas de produtos incluindo fotografias e especificações técnicas dos fabricantes. Podemos concluir que a disponibilização de informação actualizada em “tempo real” é importante para o consumidor e para quem disponibiliza o serviço. Através da organização de conteúdos¹¹⁷, lojas virtuais, portais e jornais poderão ter uma vantagem maior, em relação à concorrência, onde a manutenção da informação na *Web* requer custos elevados.

6.4.4.2.2 O Problema

Até ao momento, as organizações de informação *on-line*, providenciam conteúdo através de soluções próprias. Algumas disponibilizam informação adicionando *links* às sua páginas da *Web*, enquanto outras efectuam *downloads* “à noite” em *sites* FTP de sindicatos¹¹⁸. Outras vezes, os sindicalistas criam e mantêm vários *sites* diferentes onde os cibernautas podem aceder a ligações seguras.

Destas formas *ad hoc* de sindicalização, tem resultado muitos problemas técnicos. Muitas das vezes, a informação não se encontra disponível, pode não estar actualizada e obriga a tratamento de informação periódico por parte de programadores e *Web designers*. Do lado do sindicalista, um novo utilizador têm de receber formação sobre o modo de funcionamento do sistema, ficando dependente do sistema de sindicalização.

As aproximações *ad hoc* que referimos requerem a intervenção humana para negociar conteúdos de informação simples ou complexos. Em suma, sem um protocolo *standard*, a sindicalização não se torna escalável, em termos práticos.

6.4.4.2.3 Sindicalização em ICE

Para resolver todos estes problemas, um consórcio de empresas de aplicações em servidores e que disponibilizam conteúdo, foi criado um protocolo de sindicalização baseado em XML denominado por ICE (*Information and Content Exchange*). Já referido anteriormente, o XML é um *standard* simples de representar dados e hierarquias num estilo idêntico ao HTML. O ICE segue os seguintes requisitos [44]:

1. Um potencial assinante requer um catálogo de ofertas de subscrição;

¹¹⁷ Optei por usar organização de conteúdos para definir *Syndication*.

¹¹⁸ Um sindicato é um centro de informação que garante conteúdo actualizado de várias fontes de informação.

2. Um sindicato responde com a oferta de um catálogo de subscrição, cada oferta é descrita em detalhe com o tipo de conteúdo, restrições, tempo e frequência de entrega de conteúdo;
3. Um assinante subscreve uma ou mais ofertas, negociando o método e tempo de entrega de conteúdo específico;
4. Um assinante carrega um pacote de conteúdo registado de um sindicato;
5. Um sindicato entrega um pacote de conteúdo a um assinante;
6. Um assinante ou sindicato cancela ou altera uma subscrição.

A inovação deste protocolo reside no facto de usar XML. Enquanto muitos protocolos que usam XML para dados, o ICE não usa o XML para especificar o formato dos dados enviados do sindicato. Em termos específicos, o XML é utilizado dentro de uma entidade *ice-item*. Os dados dentro de um *ice-item* são estruturados usando especificações XML existentes como BizTalk, RosettaNet, CommerceOne, WDDX.

Outros esforços de standardização são englobados no ICE devido a só agora a publicação de informação na Web começa a estar automatizada. O *Resource Description Framework*¹¹⁹ (RDF) da W3C e outros *standards* relacionados, especificam como a descrição pode ser anexada ao conteúdo de files, permitindo que o conteúdo seja identificado e seleccionado com filtros ou critérios de ordenação. Todo este enquadramento pode ser usado com ICE, permitindo aos assinantes seleccionar ofertas e usando critérios mais ou menos complexos.

O modelo de sindicato/assinante em ICE define tanto a publicação como a subscrição, demonstrando equivalência entre os standards de protocolos de publicação e validação como o CORBA e DCOM. As mensagens em ICE são entregues em XML através de uma ligação típica do protocolo HTTP em oposição aos protocolos de baixo nível já referidos. Tal como nos outros exemplos já referidos, o ICE é bastante fácil de entender e usar mas muito mais completo a nível da semântica. Também é permitido o compactamento dos pacotes ICE, no caso da limitação da largura de banda ser uma realidade.

6.4.4.2.4 Especificação (briefing)

O ICE define um conjunto de pares pedido/resposta em XML. Embora não especifique o protocolo de transferência de dados, sugere a implementação da resposta usando a primitiva

¹¹⁹ RDF integra uma variedade de actividades de metadata como *sitemaps*, taxas de conteúdo, definições de canais de *streaming*, dados colectados de motores de pesquisa, usando XML como sintaxe de interligação. Ver em <http://www.w3.org/RDF/>.

POST do HTTP, denominada ICE/HTTP. O corpo da mensagem de um POST em HTTP contém o `ice-request` e a resposta em HTTP vêm formatada em ICE.

Todos os `ice-requests` estão dentro de um `ice-payload`, que identifica a versão e o mensageiro, fornecendo um `request-id`. O exemplo em baixo apresenta um pedido de informação. A maior parte do cabeçalho é destinada a informação de *debug*:

```
<?xml version="1.0"?>
<!DOCTYPE ice-payload SYSTEM "http://www.ice-ag.org/ICE.dtd">
<ice-payload payload-id="12984751223" timestamp="13:31:21"
ice.version="1.0">
<ice-header>
<ice-sender sender-id="5fde2359-439c-1201-56ab-1964dca39034"
name="Andromedia Aria 3.0"
        role="subscriber"/>
</ice-header>
<ice-request request-id="1998-07-05T02:02:23@xyz.com-1">
. . .
...ICE Directivas...
. . .
</ice-request>
</ice-payload>
```

Após a recepção do pedido, o servidor cria uma resposta de informação, que contém aquilo que o assinante pediu, se esta estiver disponível. A resposta contém um cabeçalho de informação idêntico como o pedido de informação.

6.4.4.2.5 Conclusão

Existem outras especificações do ICE que não serão descritas por se sobreporem ao âmbito deste documento entre as quais o pedido e recepção de catálogo, regras de entrega de informação e de subscrição, distribuição e sequência de pacotes em ICE. Para melhor informação deverá consultar as referências adjacentes e notas de rodapé.

O uso do protocolo ICE torna mais fácil a congregação, controlo e entrega de informação, venha esta donde vier, poupando tempo e dinheiro. Algumas empresas já fornecem serviços como o *iSyndicate*¹²⁰ que reúne conteúdo de páginas *Web*, pacotes noticiosos e portais, através

¹²⁰ Disponível em <http://www.iSyndicate.com/>

da subscrição dos interessados. O *Projecto AMIS* encontra-se em cooperação com a *iSyndicate* para a distribuição de conteúdo em língua portuguesa dos seus repositórios para o sindicato. Graças à possibilidade de negociação de largura de banda das redes e infra-estruturas existentes, um sindicato permite que uma vasta colecção de informação, uniformemente estruturada, esteja disponível a múltiplos assinantes como, por exemplo, fabricantes de peças, coleccionadores, *software developers*, casas de fotografia, entre outros. Com o amadurecimento da *Web* e estando a comunidade da Internet ciente que “se está a inundar em informação”, obter qualidade e não quantidade implica usar esta tecnologia ou seja, assinantes, congregam informação, controlar e estar disposto a pagar pela mesma.

6.4.4.3 RDF

O RDF (*Resource Description Framework*) é um formato de organização de dados sobre dados (metadata), oferecendo interacção entre aplicações que permutam dados na *Web*, facilitando processamento automático de recursos. O RDF serve alguns propósitos como melhorar as capacidades de motores de pesquisa, catalogação de recursos para a descrição de conteúdos e relações entre os mesmos, agentes inteligentes de *software* para a partilha e troca de conhecimento, direitos de propriedade intelectual, entre outros. O RDF com assinaturas digitais será a chave para a “*Web* de confiança” para comércio electrónico, colaboração e outras aplicações [45]. Até ao momento, o RDF oferece as seguintes funcionalidades:

1. Interoperabilidade de metadata;
2. Dados semânticos legíveis para metadata;
3. Pesquisas mais precisas na descoberta de recursos.

Em geral, o RDF fornece funcionalidades básicas para a criação de ferramentas de autorização, manipulação e pesquisas na *Web*, promovendo a transformação da *Web* num repositório de informação processado por computadores.

Para melhor compreensão do RDF, é apresentado um pequeno exemplo, que deve ser incluído dentro do <HEAD> de um documento HTML.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.0/">
  <rdf:Description rdf:about=http://doc
    dc:creator="Miguel Oliveira"
    dc:title="My document"
    dc:description="Miguel derando com bytes."
    dc:date="1999-09-10" />
```

</rdf:RDF>

Em `http://doc` substitui-se pelo URL do documento em causa. Para Miguel Oliveira, substitui-se pela pessoa ou organização que criou o documento.

6.4.5 Investigação e Desenvolvimento

Nesta secção discutiremos algumas ideias e possíveis paradigmas tendo em vista o XML. Serão apresentados conceitos e alguma investigação que está a ser feita a nível de distribuição de dados, classificação de informação e inteligência artificial.

6.4.5.1 Federações de Agentes

Os agentes fornecem funcionalidades vantajosas para a criação de aplicações de negócios. Faz sentido entender comunidades de agentes como componentes orgânicas e naturais em aplicações lógicas [46].

Agentes dedutivos e agentes que aprendem, negociam com grandes quantidades de dados em tempo real e, precisam de agentes de decisão para assegurar a quantidade de informação que os sistema de informação desejam. A aprendizagem e dedução de agentes, comunicam dados intelectuais entre eles. Obviamente, esta forma de cooperação usa agentes de mensagens para interagir com comunidades.

Esta interacção entre agentes implica uma manutenção mais fácil e envolvente devido à sua flexibilidade. Os actuais *standards* da *Web*, como XML e DOM, e linguagens orientadas a objectos, como o Java, permitem-nos criar multicomunidades de agentes, sistemas de multiagentes que respondem às necessidades do mundo real.

Um dos assuntos que mais tem sido discutido, refere-se à definição de uma linguagem *standard* de agentes, baseada em XML, para a partilha de informação e definição de padrões de comunicação de agentes, de modo a que agentes em aplicações independentes possam comunicar o seu conhecimento entre eles. Este parece ser o próximo passo da indústria de *software*, que segue os mesmos fundamentos da sociedade humana, ou seja, comunidades abertas.

6.4.5.2 XML/EDI

Um interessante desenvolvimento, baseado em XML, é a definição de um *standard* de XML/EDI¹²¹ para promover o comércio electrónico.

¹²¹ *Electronic Data Interchange.*

A transferência de dados electrónicos (EDI) encontra-se em uso à cerca de 25 anos. Apesar das numerosas vantagens do EDI, estimam-se apenas 125000 organizações em todo o mundo que usam o sistema. Estes números reflectem cerca de 0.5% do mundo empresarial, primariamente devido aos custos e complexidade destes sistemas, que tornam difícil a pequenas e médias empresas a implementação e manutenção dos tradicionais sistemas EDI [47]. Por estas razões, a maioria de empresas não adere à eficiência que um sistema destes propõe. Apesar desta situação, graças ao advento XML/EDI o panorama está a mudar.

A partir de Setembro de 1997, foi proposto um novo estilo de linguagem XML, o XSL, como já foi referido, que incorpora variantes de *javascript*. Esta especificação de XML/EDI beneficia da combinação avançada desta linguagem, para definir o controlo de processos de documentos XML/EDI, para serem armazenados de uma forma distribuída.

Em Outubro de 1997, foi publicada a especificação de DOM para XML. Este modelo fornece uma API¹²² para ferramentas baseadas em XML. Combinando XML com EDI para desenvolver XML/EDI implica a criação de *software* para capturar e codificar informação EDI através de formulários electrónicos codificados em XML. Até ao momento, as características dos formulários em XML, encontram-se em fase de aprovação.

6.4.6 Conclusões

Através deste trabalho podemos verificar que os utilizadores da Web e outros *trabalhadores de conhecimento*¹²³, poderão beneficiar das potencialidades do XML. Várias linguagens de marcas têm sido desenvolvidas para criar informação e automatizar processos de entrada e saída de dados em ambientes distribuídos e em sistemas de informação colectada.

O sucesso do XML varia de duas características muito importantes: em primeiro, a adaptação de uma linguagem poderosa, o SGML, para uma versão *light*, mantendo as principais funcionalidades e limitando-a às funcionalidades mais usadas no SGML; em segundo, uma linguagem de dados sobre dados que permite criar qualquer tipo de documentos, para qualquer que seja o destino do mesmo.

Neste trabalho foram descritos os conceitos, para uma clarificação das ideias e criação uma base sólida daquilo que se pretendia documentar, foram apresentadas algumas aplicações, para

¹²² *Application Programming Interface*.

¹²³ *Knowledge Workers*. Expressão utilizada por Bill Gates em “Negócios @ velocidade do pensamento” para definir aqueles que desenvolvem, concebem e utilizam sistemas de informação. Ver em referências.

demonstrar as vantagens e funcionalidades, e algum desenvolvimento que se pretende que dê frutos num futuro próximo.

Na minha opinião pessoal, o desenvolvimento não acaba aqui. Tudo aquilo que se refira a informação distribuída, partilhada e colectada, não esquecendo informação móvel e inteligente, terá de passar sempre por XML.

6.5 Federação de repositórios e sistema de pesquisa distribuída.

O crescimento exponencial da Internet implica um maior número de utilizadores e por sua vez a uma maior quantidade de recursos informativos.

Um servidor *Web* que albergue documentos e recursos informativos não deve abarcar uma quantidade “colossal” de informação não só pelas limitações de unidades de armazenamento, nem pela sobrecarga que poderia causar aos vários servidores do sistema. Além destas limitações, as diferenças das redes de comunicação são limitadas e a dependência de *proxies* e *firewalls* atrasa o acesso a determinados *sites*. O potencial do serviço AMIS pode ser ampliado, se minimizarmos estas dificuldades existentes em qualquer infra-estrutura de telecomunicações.

Múltiplos repositórios de informação disponibilizavam informação distribuída por vários servidores a pouca ou grande distância física, partilhando recursos informativos e minimizando tempos de acesso (Fig. 3).

Supondo vários repositórios de informação regionais, os utentes do serviço poderiam aceder ao serviço da sua área e, não só acederem a informação desse repositório como também obterem resultados de pesquisas colectados a partir de repositórios através de um sistema de pesquisa distribuída.

A principal vantagem é um serviço com uma federação de repositórios que partilha bases de dados entre os vários sistemas.

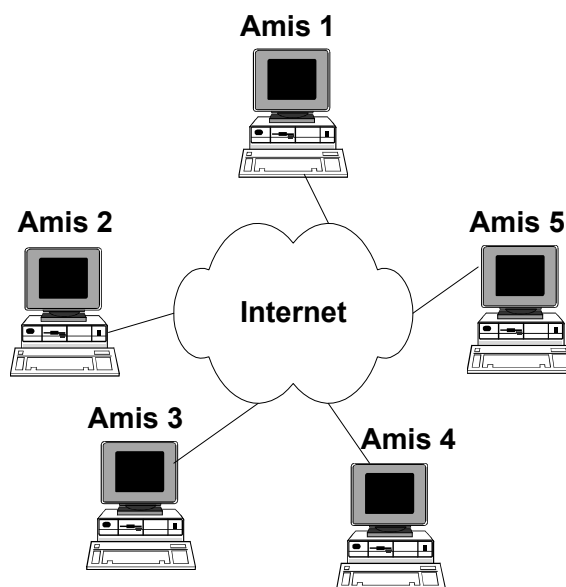


Fig. 3 – Federação de repositórios do serviço AMIS

6.5.1 Utilização de XML-RPC para pesquisas numa federação de repositórios.

XML-RPC é uma especificação e conjunto de implementações que permite a execução de código RPC (*Remote Procedure Calling*) em vários sistemas operativos e em diferentes ambientes na Internet [8]. XML-RPC pode ser visto como um RPC usando o protocolo HTTP como transporte e XML como código.

Uma mensagem em XML-RPC usa a primitiva *POST* do HTTP, com o corpo da mensagem em XML. Após a execução do pedido num servidor, são retornados resultados em XML. A principal vantagem deste protocolo reside no facto dos pedidos e respostas terem como formato XML, compreendido por computadores e por programadores, que torna mais fácil a elaboração de código e interpretação de resultados [48].

Para testar a eficiência de um sistema de pesquisa distribuída em repositórios de informação, uma das possibilidades de implementação será com XML-RPC para procura e pesquisa de informação nos vários repositórios. Após uma fase inicial, poderão ser implementados agentes móveis para minimizar os tempos de resposta.

Definindo um *standard* de comandos e padrões de comunicação baseado em XML para a partilha de informação permitiria que agentes comunicassem entre servidores (ou mesmo entre agentes) minimizando tempos de acesso e garantindo o acesso à informação distribuída.

6.6 METABASE PHP Package¹²⁴ – instalação, acesso e gestão independente de SGBDs

6.6.1 Introdução

O propósito desta secção é o de documentar o pacote de *software* **METABASE**, em PHP, que permite o desenvolvimento de aplicações, em modelo *three-tier* (Fig. 1), com total independência do sistema de gestão de base de dados (SGBD) a utilizar.

Os procedimentos a tomar por programadores para a construção de aplicações baseadas em METABASE, com suporte em SGBDs, são idênticos aos procedimentos nativos usados para comunicar directamente com as bases de dados. A grande vantagem do METABASE, é que permite aos programadores que aprendam e usem apenas um conjunto de comandos para desenvolvimento e implementação das suas aplicações.

6.6.1.1 O que é o METABASE?

METABASE é uma aplicação em PHP, constituída na generalidade por classes, cuja a sua estrutura de classes e funções permite a instalação (criação, alteração, *database dumping*), acesso, gestão e manutenção de bases de dados SQL.

6.6.1.2 Objectivos

São dois os principais objectivos do METABASE. O primeiro é o de disponibilizar uma interface programável unificada, para aceder a SGBDs SQL distintos. Usando a mesma interface, programadores podem desenvolver aplicações para bases de dados, que correm em diferentes SGBDs comerciais: *Oracle, Sybase, MySQL, Microsoft SQL Server*, etc..

O segundo objectivo é o de disponibilizar uma ferramenta para instalação e manutenção do esquema da base de dados, usando mesma interface de base de dados independente dos SGBDs. Desta forma, programadores podem desenvolver aplicações, que envolvem esquemas de bases de dados, que servem as necessidades e propósitos das aplicações, sem terem de se preocupar em como instalar ou como fazer *upgrades* em cada SGBD diferente que é usado (Fig. 38).

¹²⁴ Disponível em <http://phpclasses.upperdesign.com/browse.html/package/20/>.

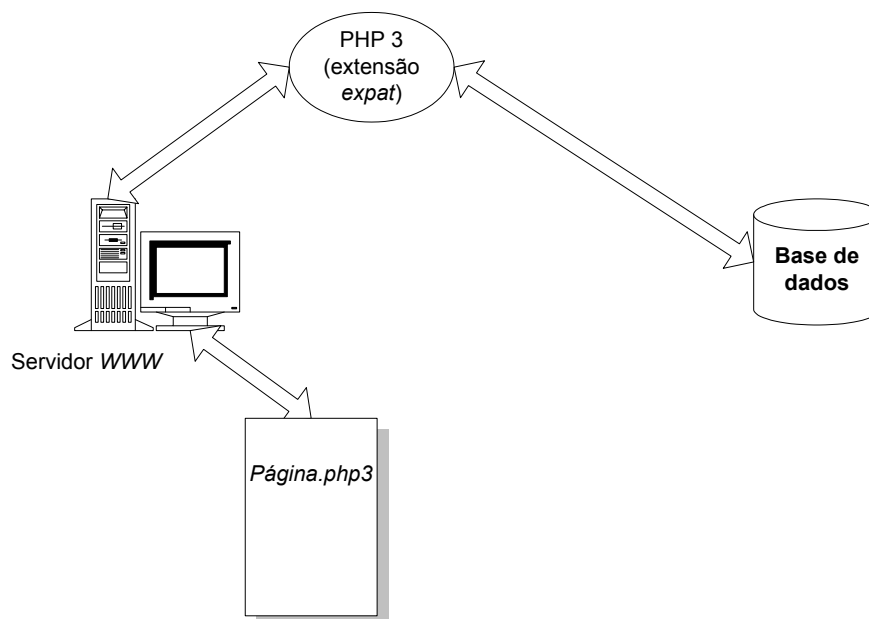


Fig. 38 - Modelo *Three-Tier*. Sistema de informação orientado à *Web*.

6.6.1.3 Estrutura do pacote METABASE

O METABASE encontra-se dividido em várias partes:

6.6.1.3.1 Interface de programação independente dos SGBDs

Consiste numa classe capaz de instanciar classes de *drivers* (*Oracle*, *Sybase*, *MySQL*, *Microsoft SQL Server*, etc.). É também disponibilizado um conjunto de funções que as aplicações devem invocar para aceder às bases de dados.

6.6.1.3.2 Drivers específicos de SGBDs

Consiste num conjunto de *drivers* que implementam métodos chamados pelas funções da interface para implementar funções de acesso às bases de dados.

6.6.1.3.3 Analisador (parser) de descrições de esquemas

Classe que analisa descrições de esquemas de bases de dados, definidas em formato XML simples (marcas sem atributos). A classe produz a estrutura de dados, contendo diferentes informações de objectos de bases de dados, descritos na *file* de descrição de esquema.

6.6.1.3.4 Gestor de bases de dados

Classe que, através da estrutura de dados gerada pela classe que analisa descrições de esquemas (1.3.3), cria e actualiza bases de dados com o seu SGBD, usando funções da interface de programação (1.3.1).

Esta classe, é capaz de comparar estruturas de dados de duas versões da mesma descrição de um esquema (o esquema que está instalado e o que pretende actualizar) e executar as operações necessárias para alterar a base de dados, sem perturbar a integridade e dados armazenados, após a instalação da mesma ou desde a sua última actualização.

6.6.2 Descrição

6.6.2.1 Desenvolvimento de aplicações baseadas no modelo *tree-tier*

Os sistemas de informação, baseados no modelo *tree-tier*, diferem do modelo *two-tier*, no servidor de aplicações intermédio entre o cliente e servidor. Esta evolução natural que tem vindo à acontecer, reflecte-se transferência de tecnologias que os sistemas de informação têm tomado. Poderemos mesmo afirmar que o “típico terminal preto e verde” está a ser progressivamente substituído por *browsers* na maioria dos sistemas de informação.

Os servidores de aplicações tornam os sistemas de informação mais independentes, permitindo que a comunicação entre cliente e servidor, seja feita através do servidor de aplicações, e não directamente entre cliente e servidor como acontecia no modelo *two-tier*. Além disso, facilita o desenvolvimento dessas aplicações para suportar vários clientes, através de *intranets* ou da *internet*, e o *software* necessário encontra-se disponível para várias plataformas.

Por sua vez, os servidores de aplicações, catalisam e desenvolvem melhores tecnologias, que permitem a integração dessas tecnologias em aplicações e sistemas de informação. Como exemplo, podemos referir os *parsers* em XML, suportados por várias linguagens de programação (*PHP*, *Perl*, etc.) e bases de dados (*Oracle*, *DB2*), os diferentes *drivers* de bases de dados, comunicações através das primitivas HTTP, definidas pela W3C, etc.

6.6.2.1.1 Sistemas de Gestão de Bases de Dados

No momento, existem vários *drivers* disponibilizados nesta aplicação para SGBDs, embora alguns não estejam, ainda, completos. Novos *drivers* estão a ser implementados pela *Upperdesign*, entidade responsável pelo desenvolvimento do METABASE, e por programadores de comunidades *Open Source*. Os *drivers* completos i.e., que o desenvolvimento está paralelamente associado à última versão do SGBD, são para *MySQL*, *Mini-SQL*, *ODBC*, *Oracle* através da *Oracle Using Interface (OCI)* e *PostgreSQL*.

6.6.2.2 Criação de uma base de dados com PHP/METABASE

É descrito, nesta secção, a implementação de esquemas de bases de dados, usando o METABASE, a criação e instalação da base de dados.

6.6.2.2.1 Esquema da base de dados

Quando se pretende criar uma base de dados, o primeiro passo a tomar é a esquematização da mesma, que vai suportar toda a informação que é necessária para a aplicação a ser desenvolvida.

O METABASE simplifica este passo, porque permite que programadores implementem estes esquemas de uma forma independente da base de dados. Apenas é necessário escrever uma *file* de texto em formato genérico XML que descreve as tabelas e campos que a base de dados terá. Vamos supor que pretendemos criar uma base de dados chamada teste (Fig. 39):

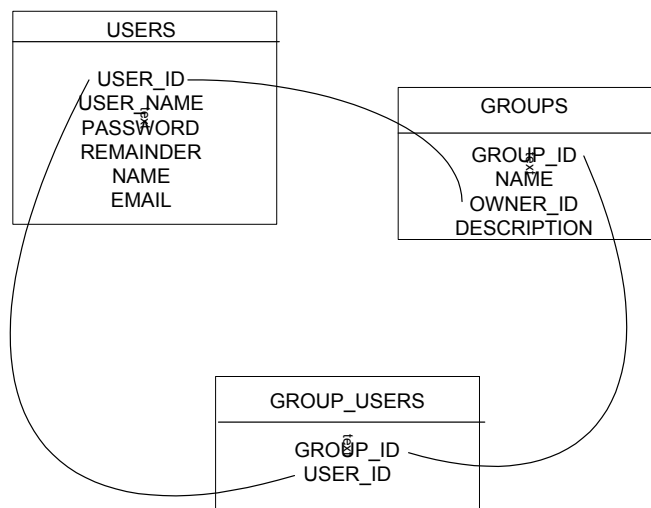


Fig. 39 - Esquema UML da base de dados teste.

Será agora apresentado um exemplo de descrição de um esquema da base de dados: Todas as files que descrevem esquemas em XML do METABASE terão, obrigatoriamente, de ter este cabeçalho:

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

De seguida, é necessário inserir a marca de início (*start tag*) do METABASE:

<database>

As propriedades gerais da base de dados são inseridas como marcas. A propriedade/marca `name` é necessária e, a marca `create`, embora não seja necessária, pretendemos que o METABASE crie a base de dados por nós, quando instalada pela primeira vez. Logo, esta propriedade tem de ser definida como 1.

<name>teste</name>

<create>1</create>

Para criar uma tabela, é necessária a inserção da marca `table` e `name`, ou seja, criação da tabela e nome, respectivamente.

<table>

<name>Users</name>

As definições das tabelas podem ter duas secções, além das suas propriedades: `declaration` e `initialization`. A marca `declaration` é necessária e tem de incluir as declarações de todas as tabelas, campos provenientes dessas tabelas e índices. A marca `initialization`, embora não seja necessária, é usada para inserir valores iniciais nas tabelas que são criadas.

A declaração da tabela, tem de ter, pelo menos, um campo (`<field>`). Na definição do campo, i.e., da marca `<field>`, é obrigatória a introdução das propriedades nome do campo, `<name>`, e tipo, `<type>`. Os tipos mais usados nos campos são o `integer` e `text`, embora o tipo `timestamp` também seja muito utilizado.

<declaration>

<field><name>user_id</name><type>integer</type>

<default>0</default><notnull>1</notnull></field>

<field><name>user_name</name><type>text</type></field>

<field><name>password</name><type>text</type></field>

<field><name>reminder</name><type>text</type></field>

<field><name>name</name><type>text</type></field>

<field><name>email</name><type>text</type></field>

O uso de índices, embora facultativo, é muitas vezes utilizado para minimizar tempos de acesso à base de dados, sendo na generalidade criados em campos que pertencem às cláusulas de procura de *queries* mais utilizados onde a aplicação corre. Obviamente, os campos em que os índices devem ser adicionados dependem do propósito da aplicação em desenvolvimento. De qualquer forma, existem campos que, geralmente, são declarados como chaves primárias ou estrangeiras numa tabela. O METABASE não suporta a declaração de chaves, pela razão de que alguns SGBDs não o suportam. As razões são diversas, sendo a principal a manutenção de uma aplicação robusta e modularidade. No universo do desenvolvimento de *software*, opta-se cada vez mais por garantir robustez e fiabilidade às aplicações, dando-se cada vez mais importância à portabilidade e standardização, em detrimento da velocidade e processamento. No entanto, suporta a declaração de índices únicos em campos individuais, que na prática podem funcionar como chaves primárias desses campos.

De notar, que campos onde seja criado um índice não pode ter entradas nulas (NULL). Por essa razão, o *parser* em XML da descrição de esquemas do METABASE, força o campo a ser criado com a propriedade de não nulo (`notnull`). Requer, também, que a propriedade *default* seja definida com um valor não nulo. No exemplo da base de dados deste será definida como no exemplo em baixo:

```
<index>
<name>users_id_index</name>
<unique>1</unique>
<field><name>user_id</name></field>
</index>
```

Sequências de objectos são frequentemente usadas para gerar valores inteiros únicos, sempre que é necessário inserir uma nova entrada numa tabela com um novo número identificador (único). Devido a esta necessidade, muitos programadores declaram campos com a propriedade `auto_increment`, dando o mesmo efeito a identificadores únicos. O METABASE não suporta esta propriedade, porque nem todos os SGBDs a suportam. Neste caso, recomenda-se o uso de sequências (`sequence`) em vez de `auto_increment`. Uma das

vantagens do uso de sequências consiste no facto da reutilização dos valores das sequências para inserir noutras tabelas que o requirem.

As sequências são esquemas de objectos *standalone* em bases de dados. No entanto, devemos especificar, no gestor da base de dados, qual a sequência de valores que são pretendidos usar. Isto ajuda, na medida em que determina um valor inicial da sequência, se adicionarmos posteriormente a sequência à base de dados, e então, teremos adicionado o campo da tabela com que deve estar sincronizada:

```
</declaration>
</table>

<sequence>
<name>user_id</name>
<start>1</start>
<on><table>users</table><field>user_id</field></on>
</sequence>
```

Outras tabelas e sequências podem ser declaradas, tantas quantas forem precisas:

```
<table>
<name>groups</name>
<declaration>
<field><name>name</name><type>text</type></field>
<field><name>owner_id</name><type>text</type></field>
<field><name>description</name><type>text</type></field>
</declaration>
</table>

<table>
<name>group_users</name>
<declaration>
<field><name>group_id</name><type>text</type></field>
<field><name>user_id</name><type>text</type></field>
</declaration>
</table>
```

Os esquemas XML do METABASE têm de ser finalizados com a marca coincidente com a marca inicial:

```
</database>
```

6.6.2.2.2 Instalação da base de dados com o METABASE

Uma vez definida a *file* com o esquema da base de dados, procedemos à instalação do servidor SGBD, através da classe de gestão do METABASE. Como exemplo, vamos supor que gravamos o esquema XML, apresentado anteriormente, numa *file* designada por `MyDatabase.schema`.

Para a instalação da base de dados, pela primeira vez, por exemplo, MySQL, necessitamos apenas de criar um objecto da classe de gestão, que invoca o método `UpdateDatabase function`, e usar um interpretador *standalone* em PHP para executar um *script*, similar com o apresentado em baixo:

```
<?php
$schema_file="MyDatabase.schema";
```

A variável `$schema_file` é a definição do nome do esquema XML da base de dados.

```
$variables=array();
```

Na necessidade de definir e instalar valores de variáveis temporais (tipo de SGBD, *username*, *password*, etc.), definimo-las em `$variables`. Na declaração em cima, não declaramos variáveis, vamos então defini-las:

```
$arguments=array(
    "Type"=>"mysql",
    "User"=>"mysql_user_name",
    "Password"=>"mysql_user_password"
);
```

Para a configuração da ligação ao servidor DB, são necessários alguns argumentos específicos dos SGBDs que, por sua vez, são passados para o método `SetupDatabase` da classe de gestão METABASE. Para criar a classe de gestão:

```
$manager=new metabase_manager_class;
```

No método `UpdateDatabase` da classe de gestão, passamos como parâmetro o nome da *file* do esquema XML, os argumentos de configuração da base de dados e o *array* de variáveis do esquema. O segundo parâmetro é de especial importância, pois define o nome da *file*, do esquema a ser instalado, em que será copiado, após uma instalação com sucesso. A *file* copiada, será utilizada posteriormente, sempre que quisermos actualizar o esquema XML. Esta *file* não deverá ser apagada para que o gestor determinar o esquema instalado.

```
$SUCCESS=$MANAGER->UPDATEDATABASE (
    $schema_file,
    $schema_file.".before",
    &$arguments,
    &$variables);
```

Se a instalação falhar, poderemos visualizar o error retornado:

```
if (!$success)
echo "Error: ".$manager->error."\n";
```

Além disso, mesmo no caso da instalação ter sido completada com sucesso, devemos sempre visualizar os *warnings* resultados:

```
if (count ($manager->warnings)>0)
echo "WARNING:\n", implode ($manager->warnings, "!\n"), "\n";
?>
```

6.6.2.3 Manipulação de dados

Um dos propósitos do METABASE é o de comunicar com SGBDs através do SQL. A sua função é o de permitir a construção e execução de comandos SQL nas suas aplicações que sejam necessárias para enviar e retornar dados das bases de dados.

6.6.2.3.1 Configuração do acesso à base de dados

O primeiro passo a tomar é a configuração da ligação do servidor SGBD aos nossos scripts de aplicações.

A ligação à base de dados é configurada invocando a função `MetabaseSetupDatabase` que, não inicia a comunicação, mas define os parâmetros que o *driver* SGBD necessita, para saber como comunicar com o servidor de base de dados que quer. Geralmente, a ligação só fica estabelecida quando é efectuado o primeiro *query* à base de dados.

A função `MetabaseSetupDatabase` aceita como argumento um *array* associado com opções de ligação. A entrada `Type` do *array* é requerida e tem de conter a designação do *driver* específico para as chamadas do interface ao servidor SGBD utilizado. Por exemplo, se o SGBD utilizado é o servidor *MySQL DBMS*, a entrada do elemento `Type` é `mysql`.

Outros tipos de entrada no *array* prendem-se com o *driver* e o tipo de SGBD utilizado. Como não é do interesse para este documento, não as vamos documentar. Para isso dever-se-á consultar o manual, relativo aos diferentes *drivers* de SGBDs suportados.

O segundo argumento na função mencionada, `MetabaseSetupDatabase`, toma, por referência, uma variável que armazena o valor de acesso à base de dados. Este valor é de extrema importância, pois é utilizado em todas as funções necessárias ao acesso da base de dados.

Por ultimo, a função `MetabaseSetupDatabase` pode falhar por inconsistência de valores dos argumentos especificados. Para isso, no caso de falha, retorna uma *string* contendo o erro que provocou a falha. Além disso, esta função é bastante importante para funções de *debug* nas aplicações desenvolvidas. Em baixo, encontra-se um exemplo da função referida nesta secção:

```
$error=MetabaseSetupDatabase (array (
    "Type"=>"mysql",
    "User"=>"mysql_user_name",
    "Password"=>"mysql_user_password"
), &$database);

if ($error!="")
{
echo "Database setup error: $error\n";
```

```
exit;  
}
```

6.6.2.3.2 Execução de queries à base de dados

O METABASE disponibiliza duas maneiras de construir e executar *queries* às bases de dados: directos e indirectos.

6.6.2.3.2.1 Queries directos

Queries directos a uma base de dados, são aqueles que são executados passando declarações SQL para a base de dados. Para tal, bastará recorrer à função `MetabaseQuery`. Esta função, recebe como argumentos o acesso à base de dados e o statment SQL:

```
$query="SELECT name,password FROM users";  
$result=MetabaseQuery ($database, $query) ;
```

A função retorna um inteiro em função do resultado do *query*. O retorno deve ser usado para certificar os resultados e retornos de linhas da base de dados.

6.6.2.3.2.2 Queries preparados

Queries preparados são *statements* SQL que são implementados antes de serem executados. A execução de *queries* preparados leva menos tempo do que um *query* directo pela razão de que já foi feito o *parse ao script*, antes de serem executados. Além disso, os *queries* preparados devem ser usados para é necessário o usar o mesmo *query* mais do que uma vez no mesmo *script*.

No exemplo em baixo, apresentamos um *query* directo de acesso à base de dados:

```
$query="SELECT name,password FROM users WHERE  
user_name=".MetabaseGetTextFieldValue ($database, "admin") ;  
$result=MetabaseQuery ($database, $query) ;
```

Poderá ser reescrito desta maneira, como um *query* preparado:

```
$query="SELECT name,password FROM users WHERE user_name=?";  
$prepared_query=MetabasePrepareQuery ($database, $query) ;
```

```
if ($prepared_query)
```

```
{  
MetabaseQuerySetText ($database, $prepared_query, 1, "admin");  
  
$result=MetabaseExecuteQuery ($database, $prepared_query);  
}
```

Um *query* preparado pode ter tantos argumentos como o numero de ? que definir. Este tipo de *queries* pode ser invocado várias vezes, no mesmo *script*, permitindo a mudança de dos valores dos argumentos, antes de serem executados.

6.6.2.3.3 *Retorno de resultados dos queries*

Um SELECT executado com sucesso retorna um valor designado por *result*. Este valor tem de ser passado por cada função do METABASE que possamos vir a precisar para aceder aos dados retornados.

O *result* é apenas um número, que serve como uma referência, retornado pelo SGBD para a aplicação que executou o *query*, que devolve uma tabela de linhas e colunas preenchidos com os dados pedidos. Para aceder a esses dados, é necessário usar a função `MetabaseFetchResult`. Como parâmetros, esta função requer a base de dados onde o *query* foi feito (`$database`), o result da base de dados (`$result`), o campo (name) e a linha da tabela (0) onde os dados estão. Em baixo, encontra-se um exemplo:

```
$result=MetabaseQuery ($database, "SELECT name, email FROM users");  
if ($result!=0)  
{  
$name=MetabaseFetchResult ($database, $result, 0, "name");  
$email=MetabaseFetchResult ($database, $result, 0, "email");  
}  
else  
$error=MetabaseError ($database);
```

As colunas podem ser acedidas pelos seus números correspondentes, em vez dos seus nomes, o que torna a aplicação mais rápida:

```
$name=MetabaseFetchResult ($database, $result, 0, 0);  
$email=MetabaseFetchResult ($database, $result, 0, 1);
```

6.6.2.4 *Updates à base de dados*

O procedimento de *updates* a bases de dados é simples e idêntico ao procedimento da primeira instalação, sendo apenas necessário editar e alterar o esquema XML (`MyDatabase.schema`), e usar o mesmo *script* de *instalação* usado na primeira instalação.

A classe de gestão da base de dados verificará todas as diferenças do esquema instalado previamente (`MyDatabase.schema.before`), comparando-o com o novo esquema a fazer o *update* e criar a lista de alterações.

Após este procedimento, pratica todas as acções necessárias para instalar e alterar a base de dados, garantindo a integridade e preservação dos mesmos.

6.6.2.5 *Portabilidade de dados (Database Dumping)*

Uma das grandes vantagens deste *software* é o de permitir a portabilidade dos dados, de um SGBD para outro SGBD.

O mecanismo utilizado é idêntico ao da criação da base de dados. Através das funções `DumpDatabase`, `DumpDatabaseChanges` e `DumpDatabaseContents`, é possível sem grandes esforço, portar estruturas e conteúdos sem grandes perdas de tempo e recursos humanos. A vantagem reside no facto de que os dados são portados para um esquema em XML, que é único e igual para qualquer base de dados, seja qual for o SGBD de origem e SGBD de destino. Além disso, o esquema de migração é idêntico ao esquema de instalação de uma nova base de dados, com a diferença de que um esquema de migração detém os dados em bruto, além do esquema da base de dados.

6.6.2.6 *METABASE versus ODBC*

O ODBC, *Open Database Conectivity*, é um conjunto de funções idênticas ao Metabase mas é específico para plataformas *Windows*. Existem vários pacotes de emulação de ODBC também para *Unix* embora apresentem grandes dificuldades de instalação, nomeadamente na compilação e configuração de *handlers* e *drivers*, e não são munidos de *drivers* para todas as bases de dados.

O ODBC não é tão rápido, quanto aceder à base de dados através de funções específicas de cada base de dados, mas tal como no Metabase o interesse é usar o mesmo código em

aplicações para aceder bases de dados diferentes. Daqui, podemos concluir que se pudermos usar código nativo, devemos fazê-lo.

Além da independência de bases de dados e plataformas computacionais, o METABASE garante a portabilidade de dados de SGBD para SGBD através de funções de *dumping* que o ODBC não permite.

6.6.3 Conclusão – Um exemplo: Sistemas de informação comerciais

Cada vez mais, o mundo empresarial tende a converter os seus sistemas de informação cliente - servidor (*two-tier*) para cliente-servidor de aplicações – servidor (*three-tier*) orientados para a *Web*. Esta conversão permite o uso de aplicações em redes intranet e até Internet, sem grande dificuldade de instalação e desenvolvimento de aplicações, minimizando custos, tempo e recursos humanos.

O desenvolvimento de aplicações em sistemas de informação, usando pacotes de *software* como o METABASE, permite que o "core" de um sistema de informação seja centrado apenas na aplicação, oferecendo independência dos SGBDs a usar, graças à interface única de acesso a diferentes SGBDs. Não só é viável para uma empresa em crescimento, permitindo que um SGBD possa ser substituído por outro no caso do serviço prestado ser rentável, como também permite migração de informação de SGBD para SGBD consoante custos, fiabilidade e eficácia dos mesmos SGBDs. Outra das vantagens é a facilidade de gestão, *updates* e *dumps* às bases de dados, sendo apenas necessário manipular o esquema XML da base de dados. Por último, em equipas de desenvolvimento de aplicações, permite a integração de módulos de desenvolvimento de *software* para produção em pequenas quantidades de tempo.

6.7 Adaptação do projecto AMIS para a publicação de conteúdos na Universidade de Aveiro

Em discussão com o GAGI (Gabinete de Gestão de Informação) encontra-se uma proposta de adaptação do projecto AMIS para um possível portal de edição, publicação e procura de recursos informativos da Universidade de Aveiro.

7 Conclusões

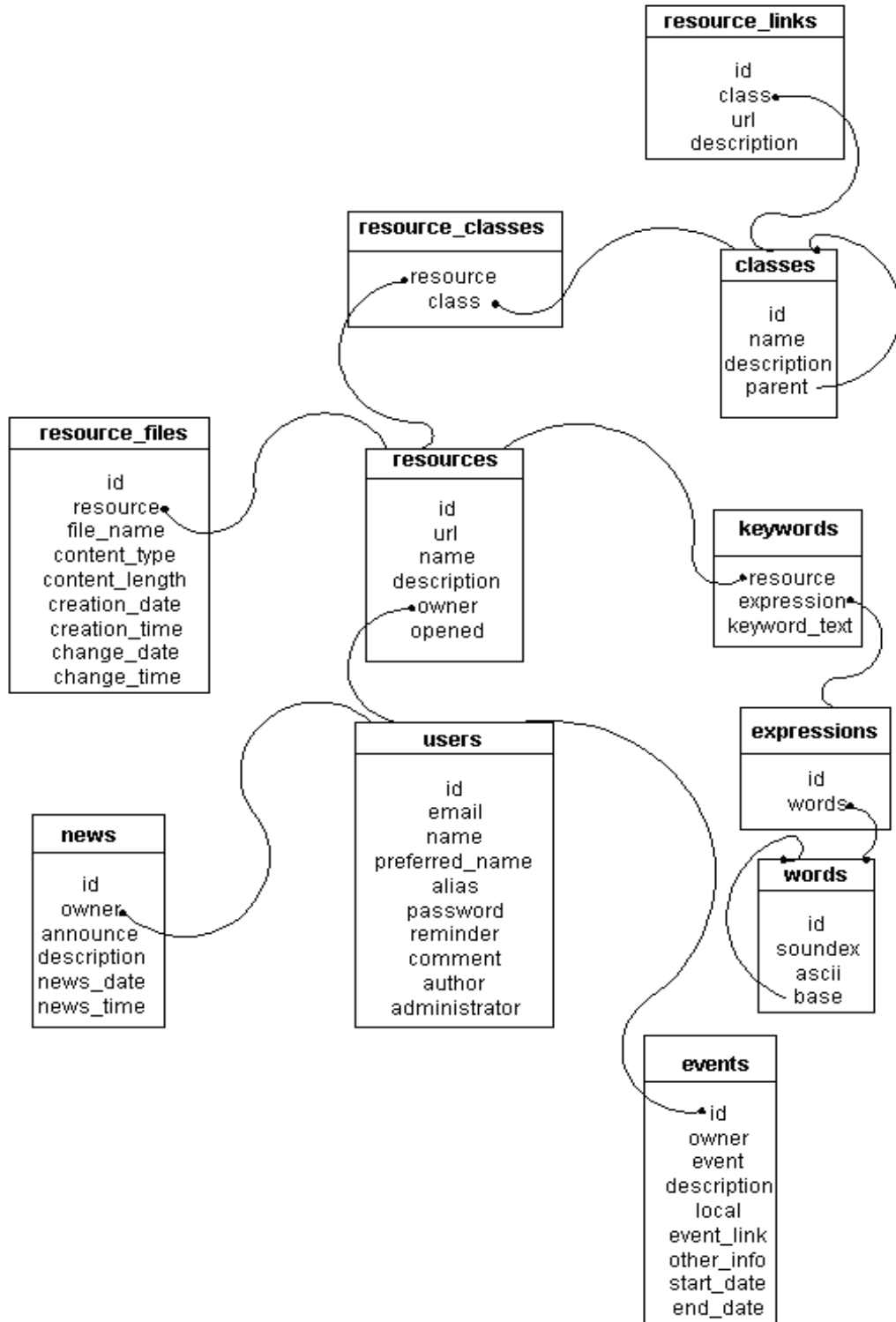
O sistema de informação desenvolvido disponibiliza um repositório de informação a autores que desejem publicar documentos e adicionar outros recursos informativos e permite o acesso aos mesmos por utilizadores anónimos.

Para *hardware* de pequeno porte, torna-se necessário não sobrecarregar o serviço existente, desperdiçando tempo de acesso e capacidade de armazenamento. Além disso, é de interesse para qualquer serviço que disponibiliza informação, tanto no campo empresarial como no ramo académico, a expansão do mesmo para um maior número de possíveis interessados em publicar ou adquirir informação, sejam utilizadores anónimos ou registados.

Este artigo descreveu, por ordem cronológica, os passos de elaboração e planeamento do serviço, a execução de ideias e do projecto proposto, um projecto piloto como ambiente de validação do serviço e, por último, futuras implementações e possíveis paradigmas que se encontram em estudo para um melhor desempenho e eficiência do serviço.

Apêndices

A. Esquema UML da base de dados do sistema



B. Participação na III LUSOCOM no âmbito do projecto AMIS

A Direcção da LUSOCOM e o Departamento de Ciências da Comunicação da Universidade do Minho organizaram em Braga, de 27 a 30 de Outubro de 1999, do 3º Encontro Lusófono de Ciências da Comunicação.

Depois de Lisboa e de Sergipe (Brasil), este 3º Encontro teve como tema específico *Investigação: Convergências e Desafios*. Com ele pretendeu-se congregar docentes e investigadores de Ciências da Comunicação para identificar campos de interesse comuns, sublinhar singularidades e idiosincrasias e explorar eficazmente caminhos de cooperação e de intercâmbio, sempre com a preocupação de promover o rigor e a qualidade.

Além de três painéis de especialistas e individualidades convidadas, funcionou um conjunto de sessões centradas em torno de um tema, as quais se propõem como um espaço de apresentação e debate de projectos e de resultados de pesquisas em curso ou recentemente terminadas, no campo das ciências da comunicação.

B.1 Abstracto da comunicação apresentada no âmbito do projecto AMIS - A problemática da divulgação da cultura científica usando os recursos proporcionados pela Internet e pelas redes telemáticas

A divulgação da cultura científica usando a Internet e as redes telemáticas é ainda um paradigma de comunicação por resolver. Os aspectos mais críticos deste processo são a qualidade e a autenticidade da informação publicada. A falta de confiança, por parte dos Cientistas e Investigadores, nos *publishers* e revistas *on-line* é, portanto, um obstáculo à consolidação do meio Internet como veículo credível de disseminação de conhecimento.

Este artigo pretende reflectir sobre a problemática da divulgação da cultura científica usando os recursos proporcionados pela Internet e pelas redes telemáticas. É dedicada especial atenção à comunicação entre os profissionais da Educação e Investigação (Ensino Superior e Secundário). Apresenta-se um sistema de informação *on-line* desenvolvido na Universidade de Aveiro que fazendo uso dos mecanismos de comunicação proporcionados pela Internet e pelas redes telemáticas, procura incorporar as metodologias clássicas de avaliação de conteúdos de publicações científicas (*peer review*), permitindo o livre acesso à informação

armazenada no sistema e também a publicação de documentos, links e eventos de carácter científico por parte de comunidades científicas de interesses comuns.

Neste trabalho toma-se também em consideração a experiência adquirida em iniciativas de utilização da Internet e das redes telemáticas com fins educacionais de interacção entre comunidades científico-académicas, tais como o Projecto Internet nas Escolas (e a sua unidade de gestão uARTE ...), o recente aparecimento de Centros de Divulgação Científica e a Rede de Cálculo Científico Nacional.

Acrónimos

- [ADSL] *Asynchronous Digital Subscriber Line*
- [ACM] *Association for Computing Machinery*
- [AMIS] *Agentes Móveis e Inteligentes de Software*
- [ANSI] *American National Standard Institute*
- [ARPANET] *Advanced Research Projects Administration Networks*
- [CATV] *Cable TV*
- [CORBA] *Common Object Request Broker Architecture*
- [CSS] *Cascade Style Sheet*
- [CV] *Curriculum Vitae*
- [CVS] *Concurrent Versions system*
- [DCTR] *Desenvolvimento Científico Tecnológico e Regional*
- [DCOM] *Distributed Component Object Model*
- [DNS] *Domain Name Server*
- [DOM] *Document Object Model*
- [DTD] *Document Type Definition*
- [EDI] *Electronic Data Interchange*
- [FCCN] *Fundação para a Computação Científica Nacional*
- [FEDER] *Fundo Europeu de Desenvolvimento Regional*
- [FIFO] *First in First Out*
- [FTP] *File Transfer Protocol*
- [GAGI] *Gabinete de Gestão de Informação*
- [GXMLPC] *Generic XML Parser Class*
- [HTML] *HyperText Markup Language*
- [HTTP] *HyperText Transfer Protocol*
- [I&D] *Investigação e Desenvolvimento*
- [ICANN] *Internet Corporation for Assigned Names and Numbers*
- [IRC] *Internet Relay Chat*
- [ICE] *Information and Content Exchange*
- [ISDN] *Integrated Services Digital Network*
- [ISP] *Internet Service Provider*
- [IST] *Information Society Technologies*
- [JAIR] *Journal of Artificial Intelligence Research*

[MIME] *Multimedia Internet Mail Extensions*
[MODL] *Molecule Design Language*
[ODBC] *Open DataBase Connectivity*
[OECD] *Organization for Economic Co-Operation and Development*
[PDF] *Portable Document Forma*
[PERL] *Practical Extraction Report Language*
[PHP] *Professional Home Page*
[RCTS] Rede Ciência, Tecnologia e Sociedade
[RDBMS] *Relational DataBase Management Systems*
[RDF] *Resource Description Framwork*
[RFC] *Request For Comments*
[RI-DCTR] Repositório de Informação DCTR
[RPC] *Remote Procedure Call*
[SGBD] *Sistema de Gestão de Bases de Dados*
[SGML] *Standard Generalized Markup Language*
[SGBD] *Sistema de Gestão de Bases de Dados*
[SQL] *Standard Query Language*
[TI] Tecnologias de Informação
[UML] *Unified Modeling Language*
[UMTS] *Universal Mobile Telecommunications System*
[URL] *Unifjed Resource Locator*
[UUGuide] *Universal Usability Guide*
[VRML] *Virtual Reality Modeling Language*
[W3C] *World Wide Web Consortium*
[WWW] *World Wide Web*
[XML] *eXtended Markup Language*

Lista de Figuras

Fig. 1 - Fonte <i>Network Wizard</i> – número de <i>hosts</i> registados em DNS.....	23
Fig. 2 - Fonte <i>Network Wizard</i> – contador de <i>hosts</i> registados de Janeiro de 1991 a Julho de 2000.....	23
Fig. 3 - Fonte <i>Netcraft</i> - Estatística de servidores Web de Dezembro de 2000 a Janeiro de 2001.	24
Fig. 4 - População Global na Internet pelo centro de pesquisa da <i>CommerceNet</i>	24
Fig. 5 - Número de <i>hosts</i> por 1000 habitantes na zona OCDE.	26
Fig. 6 - Penetração da Internet em lares da união Europeia.....	26
Fig. 7 - Evolução dos domínios em PT.	29
Fig. 8 - Hierarquias PT.....	30
Fig. 9 - Total de domínios activos.	31
Fig. 10 - Perfil dos utilizadores de computador.	33
Fig. 11 - Acesso à Internet nas escolas.	37
Fig. 12 - Distribuição de páginas <i>Web</i> por domínio e língua.	38
Fig. 13 - Páginas <i>Web</i> por linguagem.....	39
Fig. 14 – Páginas <i>Web</i> ligadas a servidores seguros por domínios e linguagens	40
Fig. 15 - Esquema hierárquico de utilizadores.....	50
Fig. 16 - Organização de recursos informativos.....	54
Fig. 17 - Organização temática de conteúdos.....	54
Fig. 18 - Estruturação de conteúdos na <i>Web</i>	56
Fig. 19 - Portal AMIS/DCTR.....	58
Fig. 20 - Resultados de uma busca sobre “ciência”.	60
Fig. 21 - Menu de opções activo para um administrador.....	64
Fig. 22 - Páginas de acesso ao menu de opções para autores e utilizadores registados.....	65
Fig. 23 - Página de inscrição de um novo autor.	67
Fig. 24 - Criação de uma pasta.	74
Fig. 25 - Página de publicação de conteúdos com <i>Directory Browsing</i>	75
Fig. 26 - Classificação de conteúdos temática e por Expressões-chave.....	77
Fig. 27 - Inserção de <i>links</i>	78
Fig. 28 - Página de rosto do DCTR.	83

Fig. 29 - Primeira página de rosto do UUGuide. Através das CSSs, é possível apresentar a mesma tecnologia com interfaces diferentes.	84
Fig. 30 - Exemplo Hello World em HTML.....	85
Fig. 31 - Exemplo Hello World em HTML com CSS.....	86
Fig. 32 - Ambiente de testes no <i>UUGuide</i> com HtDig.....	91
Fig. 33 - Página principal do Repositório de Informação DCTR.....	94
Fig. 34 - UniversalUsability.org.....	98
Fig. 35 - Esquema de um possível modelo de personalização do serviço AMIS.....	115
Fig. 36 – Molécula da água. Descrição em MoDL, convertida em VRML.....	133
Fig. 37 – Esquema genérico de processamento GXMLPC-PHP3.....	136
Fig. 38 - Modelo <i>Three-Tier</i> . Sistema de informação orientado à <i>Web</i>	149
Fig. 39 - Esquema UML da base de dados teste.	151

Referências

[1] M. Lemos, *Estar na Internet*, Mc-Graw Hill, Lisboa, Portugal, 1998, p.20, ISBN: 972-8298-94-3.

URL: <http://www.e-na.net/> (current Jan. 2003).

[2] B. M. Leiner *et al.*, “A Brief History of the Internet,” In *Histories of the Internet*, <http://www.isoc.org/internet-history/brief.html> (current Jan. 2003).

[3] “*Internet Domain Survey: Number of Hosts Advertised in the DNS July 2000*”, Internet Software Consortium.

URL: <http://www.isc.org/ds/WWW-200007/index.html>. (current Jan. 2003).

[4] “Netcraft Webserver Survey,” *Netcraft*, <http://www.netcraft.com/Survey/> (current Jan. 2003).

[5] “Worldwide Internet Population,” *CommerceNet Research Center*, <http://www.commerce.net/research/stats/wwstats.html> (current Jan. 2000).

[6] *Internet and the Future Policy Framework for Telecommunications - study on the development of new telecommunications services, in particular those exploiting Internet, and their impact (economic and otherwise) on the European Union regulatory and policy framework for telecommunications - A report for the European Commission*, tech. report, Fischer & Lorenz, Copenhagen, Jan. 2000.

URL: <http://europa.eu.int/ISPO/infosoc/telecompolicy/en/Fischer31a.pdf> (current Jan. 2003).

[7] “Internet and Electronic Commerce Indicators Update,” *OECD*, <http://www.oecd.org/dsti/sti/it/cm/stats/newindicators.htm> (current Oct. 2000).

[8] *Sociedade da Informação em Portugal – indicadores de Progresso*, tech. report, Observatório das Ciências e Tecnologias, Lisboa, Portugal, Dez. 2000.

[9] *eEurope 2002 - Comunicação da Comissão ao Conselho e ao Parlamento Europeu*, tech. report, Comissão das Comunidades Europeias, Bruxelas, Nov. 2000.

[10] *Idem*.

[11] *Estatísticas – Evolução dos domínios em PT*, tech. report, Fundação para a Computação Científica Nacional, Lisboa, Portugal, Jan. 2001.

URL: <http://www.fccn.pt/estatisticas/PT/index.shtml> (current Jan. 2003).

[12] *Hierarquias PT*, tech. report, Fundação para a Computação Científica Nacional, Jan. 2001.

URL: <http://www.fccn.pt/DNS/Hierarquias/index.shtml> (current Jan. 2003)

[13] *Estatísticas – Total de domínios activos*, tech. report, Fundação para a Computação Científica Nacional, Jan. 2001.

URL: <http://www.fccn.pt/estatisticas/PT/index.shtml> (current Jan. 2003).

[14] *Resultados da Terceira Vaga de 2000 do Bareme Internet*, tech. report, Departamento de Estudos e Meios Markttest, Lisboa, Portugal, Fev. 2001.

URL: http://www.markttest.pt/ASP/go.asp?file_content=/Informacao/Bareme-Internet/B-Internet/contents.htm&file_main=/Informacao/Bareme-Internet/Resultados/index2000.htm (current Jan. 2001)

[15] *Inquérito à Utilização das Tecnologias de Informação e Comunicação pela População Portuguesa – 2000*, tech. report., Observatório das Ciências e das Tecnologias, Dez. 2000.

[16] *Iniciativa Internet: Acções executadas, em curso ou programadas*, tech. report 1-14-12-2000, Comissão Interministerial para a Sociedade da Informação, Dez. 2000.

[17] A. Grönlund, “Democracy in an IT-Framed Society,” *Communications of the ACM*, vol. 44, no. 1, Jan. 2001.

[18] *Livro Verde para a Sociedade da Informação em Portugal*, Missão para a Sociedade da Informação / Min. da Ciência e Tecnologia, Lisboa, Portugal, 1997, ISBN: 972-97349-0-9.

URL: <http://www.aceso.mct.pt/docs/lverde.htm> (current Jan. 2003).

[19] M. Cardoso e S. Sá , “Como a Internet está a revolucionar Portugal,” *Visão Digital, Revista Visão*, nº 420, Mar. 2001.

[20] *Strategies for Jobs in the information Society*, tech. report, European Commission, Nov. 2000.

URL: <http://europa.eu.int/scadplus/leg/en/cha/c11705d.htm> (current Jan. 2003).

[21] *Internet Indicators*, tech. report, OECD, Dec. 2000.

URL: <http://www.oecd.org/dsti/sti/it/cm/stats/indicators.htm> (current Oct. 2000).

[22] R. Kling, L. Covi, “Electronic Journals and Legitimate Media in the Systems of Scholarly Communication,” *The Information Society*, vol. 11, no. 4, Nov. 1995, pp. 261-271.

URL: <http://www.slis.indiana.edu/TIS/articles/klingej2.html> (current Jan. 2003).

[23] *Request For Comments 1945, Hypertext Transfer Protocol – HTTP/1.0*, Network Working Group, SRI International, May 1996.

[24] *Request For Comments 2068, Hypertext Transfer Protocol – HTTP/1.1*, Network Working Group, SRI Standards Track, Jan. 1997.

[25] *Request For Comments 2045-2049, Multipurpose Internet Mail Extensions (MIME)*, Network Working Group, SRI Standards Track, Nov.1996.

[26] *Request For Comments 1867, Form-based File Upload in HTML*, Network Working Group, SRI Experimental, Nov. 1995.

[27] *W3C Recommendation 17-12-1996, Cascading Style Sheets, Level 1*, W3C, Dec. 1996.

[28] A. M. de O. Duarte, *Explicitação da Actividades do projecto DCTR*, Grupo de Sistemas de Banda Larga, Dept. Electrónica e Telecom., Univ. Aveiro, Portugal, Apr. 1999.

[29] B. Shneiderman, “Introduction & Definition,” *Universal Usability Guide*, <http://www.universalusability.org/definition/introduction.html> (current Jan. 2003).

[30] S. Green *et al.*, *Software Agents: A review*, tech. report, Trinity Colledge, Broadcom Éireann Research Ltd., Dublin, Ireland, May. 1997.

URL: http://www.cs.tcd.ie/research_groups/aig/iag/ (current Jan. 2003).

[31] M. R. Genesereth, S. P. KetchPel, “Software Agents,” *Communications of the ACM*, vol.37, no. 7, 1994, pp. 48-53.

[32] B. Hermans, “Intelligent Software Agents on the Internet: an Inventory of Currently Offered Functionality in the Information Society and a Prediction of (Near) Future Developments,” *First Monday – Peer-Reviewed journal on the Internet*, http://www.firstmonday.dk/issues/issue2_3/ch_123/index.html (current Jan. 2003).

[33] A. Gilbert *et al.*, “The Role of Intelligent Agents in the Information Infrastructure”, IBM, <http://activist.gpl.ibm.com:81/WhitePaper/ptc2.htm> (current Dec. 2001).

- [34] G. Notess, "Search Engine Showdown Analysss," Search Engine Showdown, <http://searchengineshowdown.com> (current Jan. 2003).
- [35] U. Manber, A. Patel, J. Robison, "Experience with Personalization on Yahoo!," *Communications of the ACM*, vol. 37, no. 7, 1994, pp. 48-53.
- [36] *W3C Recommendation 10-Feb-98, eXtensible Markup Language (XML) 1.0*, W3C, Feb. 1998.
- [37] P. Flynn *et al.*, *The XML-FAQ*, FAQ, W3C XML Special Interest Group, Oct. 1998.
URL: <http://www.ucc.ie/XML/> (current Jan. 2003).
- [38] *World Wide Web Consortium*, <http://www.w3c.org/> (current Jan. 2003).
- [39] J. Flanders, S. Bauman, "Women Writers Project", *ACM Crossroads Student Magazine*, vol. 6, no. 2, Winter 1999.
- [40] J. B. Bedunah, "XML: the Future of the Web", *Crossroads: The ACM Student Magazine*, vol. 6, no. 2, Winter 1999.
- [41] S. S. Bakken *et al.* "A brief history of PHP", *PHP3 Manual*, The PHP Documentation Group, USA, Jul. 1999.
URL: <http://www.php.net/manual/> (current Jan. 2003).
- [42] "XML-RPC Specification", *UserLand Software*, <http://www.xml-rpc.com/spec> (current Jan. 2003).
- [43] D. R. Greening, "Self-Service Syndication with ICE", *WEBTechniques*, <http://www.webtechniques.com/archives/1999/11/greening/> (current Jan. 2003).
- [44] N. Webber *et al.*, "The Information and Content Exchange (ICE) protocol", *W3C*, <http://www.w3.org/TR/NOTE-ice> (current Jan. 2003).
- [45] T. Berners-Lee, R. R. Swick, "Frequently Asked Questions about RDF", *W3C*, <http://www.w3.org/RDF/FAQ> (current Jan. 2003).
- [46] G. Barnea, "Intelligent Agent Communities", *WEBTechniques*, <http://www.webtechniques.com/archives/1999/09/barnea/> (current Jan. 2003).
- [47] R. Murch, T. Johnson, *Intelligent Software Agents*, Prentice Hall, Upper Saddle River, New Jersey, November 1998, ISBN: 0130110213.

[48] UserLand Software, “XML-RPC Specification”, Friday, September 03, 1999. Disponível em “XML-RPC Specification”, *UserLand Software*, <http://www.xml-rpc.com/spec> (current Jan. 2003).