



Departamento de Ciências e Tecnologias da Informação

Sistema de Notificações Automáticas de Informações
Provenientes de Fontes *Web*, através de Mensagens Escritas para
Telemóveis

Hélder da Corte Pestana

Dissertação submetida como requisito parcial para obtenção do grau de
Mestre em Estudos de Informação e Biblioteca Digitais

Orientador:

Professor Doutor Pedro Faria Lopes, Professor Associado,
ISCTE-IUL

Janeiro, 2010

Resumo

O presente trabalho aborda o desenvolvimento e teste de um sistema de informação baseado na plataforma *Web*, que permita aos utilizadores, possuindo apenas conhecimentos na óptica de utilizador, criar e gerir fontes de informação provenientes de *feeds* da *world wide Web*, para posterior recepção no telemóvel, através de *SMS* (*Short Message Service*), de notificações de actualizações de informação verificadas nas referidas fontes.

Palavras chave: Internet, Site, Aplicação *Web*, World Wide *Web*, Gestão de Informação, Notificações Automáticas, SMS, XML, RSS, Atom

Abstract

This study focus the development of an information system based on a Web platform, allowing given users, having only basic computer knowledge, to create and manage information sources from world wide web feeds, to further receive on the cellular phone, through SMS (Short Message Service), alerts based on new information found in these sources.

Keywords: Internet, Site, Web Application, World Wide Web, Information Management, Web Alerts, SMS, XML, RSS, Atom

Agradecimentos

O meu profundo agradecimento ao Professor Doutor Pedro Faria Lopes, pela sua orientação, incentivo e apoio ao longo da prossecução deste trabalho.

A todos os meus colegas, especialmente o Eng. Carlos Mora e Eng. Sérgio Rodrigues, do Departamento de Tecnologias de Informação e Comunicação da Escola Superior de Tecnologia de Abrantes - Instituto Politécnico de Tomar, pelo apoio e incentivo.

À minha família e amigos, pelo apoio e incentivo, principalmente pela compreensão dos momentos, ao longo desta caminhada, em que não pude estar presente.

Índice:

1	Introdução	10
1.1	Introdução ao tema.....	10
1.2	Estrutura da Tese.....	11
2	Estudo e síntese da literatura relevante.....	12
2.1	A <i>World Wide Web</i>	12
2.1.1	Preocupações, Implicações, e Critérios de qualidade da <i>Web</i>	15
2.2	Sindicação e agregação de conteúdos na <i>Web</i>	17
2.2.1	História da Sindicação e Agregação de Conteúdos	17
2.2.2	O que são as RSS ou Atom	20
2.2.3	Arquitectura de uma <i>Feed</i>	23
2.2.4	Agregadores de <i>Feeds</i>	24
2.3	As SMS – <i>Short Message Service</i>	26
2.3.1	O que são as SMS?	26
2.3.2	As SMS nos dias de hoje	27
2.3.3	Arquitectura das SMS	29
2.3.3.1	SMSC – Short Message Service Center.....	30
2.3.3.2	Envio de SMS através de Agregadores de SMS.....	31
2.3.4	Interacção através de SMS.....	32
3	Sistema de Notificações Automáticas de Informações Provenientes de Fontes <i>Web</i> , através de Mensagens Escritas para Telemóveis	34
3.1	Descrição do problema	34
3.2	Análise	37
3.2.1	Levantamento dos Requisitos	40
3.2.1.1	Requisitos Funcionais	40
3.2.1.2	Requisitos Não Funcionais	41
3.2.1.3	Definição de <i>Use Cases</i>	42
3.2.1.4	Descrição Estruturada dos <i>Use Cases</i>	43
3.2.2	Desenho do Diagrama de classes	45
3.2.3	Transposição para o Modelo Relacional.....	46
3.2.3.1	Regras de transposição para o modelo relacional	47
3.2.3.2	Transposição para o Modelo Relacional.....	48
3.2.4	Definição do Diagrama de Actividades	49
3.2.5	Diagramas Físicos	51
3.2.6	Composição do <i>layout</i> para o sistema.....	53
3.2.6.1	Esboço.....	53
3.2.6.2	Logomarca	55
3.2.6.3	<i>Layout</i> gráfico	56
4	Desenvolvimento do Sistema.....	58
4.1	Tecnologias adoptadas no desenvolvimento do sistema.....	58
4.1.1	Tecnologias envolvidas no lado do cliente	59
4.1.1.1	O HTML - <i>Hyper Text Markup Language</i>	59
4.1.1.2	As CSS - <i>Cascade Style Sheets</i>	61
4.1.1.3	Javascript - dinamismo do lado do cliente.....	62
4.1.2	Tecnologias envolvidas no lado do servidor: linguagem dinâmica.....	63
4.1.2.1	PHP - <i>Hypertext Preprocessor</i>	65
4.1.3	A SGBD MySQL - Sistema de Gestão de Bases de Dados.....	67
4.1.4	O serviço de envio de SMS.....	69
4.2	Descrição do sistema.....	70

4.2.1	A arquitectura da aplicação <i>Web</i>	70
4.2.2	Configuração e instalação do <i>Hardware</i> e <i>Software</i> base	71
4.2.2.1	<i>Hardware</i>	72
4.2.2.2	<i>Software</i>	75
4.2.3	Edição da estrutura base da interface <i>Web</i> , através de HTML e CSS	76
4.2.4	Utilização de dinamismo no lado do cliente com Javascript	79
4.2.5	Criação da base de dados	81
4.2.6	Dinamismo no lado do servidor	81
4.2.6.1	A aplicação <i>Web</i> de interface ao utilizador.....	82
4.2.6.2	Os processos automatizados	87
5	Apresentação e discussão dos resultados	91
5.1	Teste de avaliação quanto à acessibilidade do protótipo	91
5.2	Teste de desempenho do sistema	93
5.2.1	Apresentação e análise de resultados do questionário inicial	94
5.2.1.1	Apresentação de resultados.....	94
5.2.2	Estatísticas sobre a utilização do protótipo.....	99
5.2.3	Apresentação e análise de resultados do inquérito final	100
5.2.3.1	Apresentação de resultados.....	100
6	Conclusões	102
7	Bibliografia e Referências.....	106
	Apêndices.....	115
A1.	Diferentes Especificações de <i>Feeds</i>	115
A2.	Questionário inicial e final.....	119
A3.	Resultados da validação de acessibilidade.....	123

Índice de Figuras:

Fig. 3-1	Ciclo de vida no desenvolvimento de um sistema de informação	39
Fig. 3-2	Diagrama de <i>Use Case</i> do Sistema de Notificações de <i>feeds</i> RSS para SMS.....	43
Fig. 3-3	Diagrama de Classes	46
Fig. 3-4	Modelo Relacional do Sistema de Informação	49
Fig. 3-5	Diagrama de Actividades: leitura de informação de <i>feeds</i>	50
Fig. 3-6	Diagrama de Actividades: envio de informação das <i>feeds</i> por SMS	51
Fig. 3-7	Diagrama de componentes para a aplicação <i>Web</i>	52
Fig. 3-8	Diagrama de instalação do sistema	52
Fig. 3-9	<i>Wireframe</i> da interface gráfica do sistema	55
Fig. 3-10	<i>Feed icon</i> - ícone representativo de uma <i>feed</i>	55
Fig. 3-11	Logomarca para o sistema.....	56
Fig. 3-12	<i>Layout</i> gráfico desenvolvido para o sistema	56
Fig. 4-1	Diferentes camadas na arquitectura de uma aplicação <i>Web</i>	59
Fig. 4-2	Exemplo de validação de um campo de formulário, com recurso a Javascript	63
Fig. 4-3	Diferenças de processamento de páginas dinâmicas e páginas estáticas	64
Fig. 4-4	Linguagens de programação mais populares	66
Fig. 4-5	Comparação de performance entre SGBD na devolução de páginas <i>Web</i> por segundo	68
Fig. 4-6	Arquitectura da aplicação <i>Web</i> , de interface ao utilizador.....	71
Fig. 4-7	Devolução de páginas <i>web</i> por segundo	73

Fig. 4-8 Configuração de mapeamentos estáticos, para o reencaminhamento de tráfego nas portas 80 e 21 do exterior para o interior.....	74
Fig. 4-9 Painel de controle do XAMPP	76
Fig. 4-10 Ping efectuado ao domínio r2s.ath.cx	76
Fig. 4-11 Exemplo de cabeçalho HTML de nível 1, sem recurso a CSS	77
Fig. 4-12 Exemplo de cabeçalho com uma imagem, usando a técnica tradicional	78
Fig. 4-13 Criar uma nova tarefa automatizada no pycron	89
Fig. 4-14 Executar ficheiros PHP via linha de comandos	90
Fig. 4-15 Leitura das <i>feeds</i> subscritas através de uma tarefa automatizada num intervalo regular	90
Fig. 5-1 Formação académica dos inquiridos	93
Fig. 5-2 Locais em que os inquiridos têm acesso à Internet	94
Fig. 5-3 Dispositivos que utiliza para aceder à Internet.....	95
Fig. 5-4 Dispositivos que os inquiridos levam para qualquer lugar	95
Fig. 5-5 Forma de recepção / leitura das notificações	96
Fig. 5-6 Tempo médio decorrido entre o envio e a leitura da notificação	96
Fig. 5-7 Grau de satisfação da relação entre o tempo de envio e leitura da notificação	97
Fig. 5-8 Aferição sobre a vontade de receber notificações em qualquer lugar.....	97
Fig. 5-9 Existência de sindicacão de <i>feeds</i> nos sites habituais dos inquiridos.....	97
Fig. 5-10 Dispositivos utilizados na leitura de <i>feeds</i>	98
Fig. 5-11 Disponibilidade dos inquiridos em receber notificações por SMS	98
Fig. 5-12 Opinião dos inquiridos sobre se a utilização das SMS melhoraria o tempo dispendido entre o envio e a leitura da informação.	98
Fig. 5-13 Apreciação do protótipo quanto à sua usabilidade.....	100
Fig. 5-14 Qualidade do conteúdo dos SMS enviados, face ao seu limite de 160 caracteres.....	101
Fig. 5-15 Resultados do questionário final sobre a utilização do protótipo.....	101

Índice de Tabelas:

Tabela 2-1 Quadro cronológico da evolução dos formatos de sindicacão.....	18
Tabela 2-2 Quadro comparativo de vantagens e desvantagens entre os suportes <i>Web</i> , <i>Email</i> e <i>Feeds</i>	22
Tabela 2-3 Exemplos de Agregadores de <i>Feeds</i>	26
Tabela 3-1 Actores e <i>Use Cases</i>	42
Tabela 3-2 Adicionar uma <i>feed</i> ao perfil de utilizador	43
Tabela 3-3 Ler informação das <i>feeds</i>	44
Tabela 3-4 Enviar SMS aos utilizadores com os itens novos das <i>feeds</i> subscritas.....	44
Tabela 3-5 Quadro de classes e atributos do sistema.....	45
Tabela 3-6 Transposiçã para o Modelo Relacional.....	48
Tabela 3-7 Padrão de Posicionamento de elementos no <i>layout</i> de uma interface gráfica de um site	54
Tabela 4-1 Quadro comparativo entre a utilização de linguagens dinâmicas e linguagens estáticas.....	64
Tabela 4-2 Resultados dos testes de pedidos e devoluçã de páginas <i>web</i>	72
Tabela 4-3 Quadro resumo dos ficheiros PHP desenvolvidos no sistema.....	86
Tabela 4-4 Ficheiros PHP relativos aos processos automatizados do sistema	88
Tabela 5-1 Resultados da avaliaçã do protótipo quanto à acessibilidade	92

Tabela 5-2 <i>Feeds</i> subscritas pelos utilizadores	99
Tabela 5-3 Estatísticas sobre as <i>feeds</i> subscritas pelos utilizadores	99
Tabela 5-4 Estatísticas de envio de SMS para os utilizadores	100

Índice de Abreviaturas e Siglas

AJAX	<i>Asynchronous Javascript And XML</i>
ATOM	Formato de syndicação semelhante ao RSS
ASP	<i>Active Server Page</i>
CDF	<i>Channel Definition Format</i>
CSS	<i>Cascade Style Sheet</i>
DBMS	<i>Database Management System</i> (o mesmo que SGBD)
DDL	<i>Data Definition Language</i>
DNS	<i>Domain Name System</i>
FTP	<i>File Transfer Protocol</i>
GSM	<i>Global System for Mobile Communications</i>
HTML	<i>Hypertext Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
HTTPS	<i>Secure Hypertext Transfer Protocol</i>
IIS	<i>Internet Information Server</i>
IP	<i>Internet Protocol</i>
MAC	<i>Media Access Control</i>
MCF	<i>Meta Content Framework</i>
ODBC	<i>Open Data Base Connectivity</i>
PDA	<i>Personal Digital Assistant</i>
PHP	<i>Hypertext Preprocessor</i>
RDF	<i>Resource Description Framework</i>
RSS	<i>Rich Site Summary; RDF Site Summary; Really Simple Syndication</i>
SGBD	Sistema de Gestão de Bases de Dados
SGML	<i>Standard Generalized Markup Language</i>
SMS	<i>Short Message Service</i>
SMSC	<i>Short Message Service Center</i>
SSL	<i>Secure Sockets Layer</i>
SQL	<i>Structured Query Language</i>
TCP	<i>Transmission Control Protocol</i>

UDP	<i>User Datagram Protocol</i>
UML	<i>Unified Modeling Language</i>
UMPC	<i>Ultra Mobile Portable Computer</i>
UP	<i>Unified Process</i>
URL	<i>Uniform Resource Locator</i>
WEB	<i>World Wide Web</i>
W3C	<i>World Wide Web Consortium</i>
XHTML	<i>Extensible Hypertext Markup Language</i>
XML	<i>Extensible Markup Language</i>

1 Introdução

1.1 Introdução ao tema

A crescente utilização massiva da Internet, como uma das principais formas de comunicar e partilhar conhecimento, conduziu a que a informação digital tenha uma maior importância para os cidadãos: a chegada da era digital veio derrubar barreiras físicas que antes se impunham, permitindo o fácil acesso, por parte dos utilizadores, à informação. O aumento exponencial da largura de banda provocou um fluxo crescente de publicação de informação *online*, a troca de informação através de meios electrónicos e, conseqüentemente, aos utilizadores estarem conectados cada vez mais tempo à Internet.

Sob a forma de resposta, a (atenta) indústria de electrónica e telecomunicações começou a produzir dispositivos móveis como os PDA (*Personal Digital Assistant*), UMPC (*Ultra Mobile Personal Computer*), *Smartphones*, *Netbooks*, etc., com dimensões cada vez mais reduzidas e com maiores capacidades de se conectarem à rede. No entanto, a aquisição destes terminais ainda é cara e o acesso à Internet a partir de conexões móveis é cobrada ainda com valores proibitivos, muitas vezes taxada pelo volume de dados transitado (maioritariamente por acessos desnecessários ou redundantes, no processo de procura por informação actualizada).

Neste processo, é o utilizador que assume um papel activo na procura de informação actualizada. A presente tese procura estudar a possibilidade de inversão dos papéis: o desenvolvimento de um serviço automático que possibilite que a própria informação, assim que disponível, seja canalizada automaticamente e rapidamente para o mais comum dos cidadãos (que assume um papel passivo), em qualquer lugar a qualquer hora. Para atingir esse objectivo, a hipótese colocada em evidência, consiste em verificar a possibilidade de adoptar o mais comum dos dispositivos que qualquer cidadão "carrega" insistentemente para qualquer lugar: o telemóvel e a uma das suas formas

mais básicas de comunicação, as SMS. O presente trabalho propõe-se desenvolver um sistema de informação que permita a notificação, com recurso ao envio de SMS para o utilizador, de conteúdos provenientes de *feeds* (canais de sindicância e agregação de conteúdos, cuja adopção / presença é cada vez mais comum nos sites e portais) personalizadas e parametrizadas inicialmente pelo utilizador.

1.2 Estrutura da Tese

A presente tese encontra-se organizada em 6 capítulos que passamos a apresentar:

No Capítulo “Introdução”, são apresentados os motivos e pertinências que levaram à realização deste estudo, os objectivos a que este se propõe alcançar, bem como as suas limitações.

No Capítulo 2, “Estudo e síntese da literatura relevante”, pretende-se fazer uma revisão bibliográfica / estado de arte sobre a *Internet*: fontes de informação na Internet; formas de acesso; a sindicância de conteúdos. Abordamos também a revisão bibliográfica sobre o estado de arte das mensagens SMS.

No Capítulo 3, “Sistema de Notificações Automáticas de Informações Provenientes de Fontes *Web*, através de Mensagens Escritas para Telemóveis”, descrevem-se as circunstâncias do problema colocado em evidência, que motivaram o desenvolvimento desta tese, bem como enumerar e descrever as fases posteriores que culminaram na resolução do problema proposto.

No Capítulo 4, “Desenvolvimento do Sistema”, pretende-se descrever as fases de desenvolvimento do sistema proposto. O capítulo encontra-se estruturado em duas partes: a primeira aborda as tecnologias envolvidas no desenvolvimento do sistema; a segunda descreve o sistema desenvolvido.

No Capítulo 5, "Apresentação e discussão dos resultados", pretende-se analisar os resultados obtidos, através da recolha de dados junto de uma amostra de utilizadores potenciais do sistema, com o objectivo de avaliar o desempenho do protótipo.

No Capítulo 6, “Conclusões”, resume-se o contexto da tese em estudo e descrevem-se os resultados alcançados durante o desenvolvimento deste trabalho.

2 Estudo e síntese da literatura relevante

Pretende-se neste capítulo efectuar uma revisão bibliográfica sobre as vertentes envolvidas neste estudo: a *World Wide Web*, sua informação e formas de acesso; a sindicância e agregação de conteúdos na *Web*; a utilização de mensagens escritas para telemóveis (SMS).

2.1 A *World Wide Web*

A Internet surgiu em 1968, a partir de uma iniciativa do Departamento de Defesa dos Estados Unidos da América, com fins militares. Pretendia-se construir uma rede de comunicações fiável, que permitisse que a troca de informações ainda que alguns dos postos de comunicações fossem destruídos. Trata-se de uma rede universal constituída por um conjunto de redes universitárias, científicas, comerciais e militares interconectadas através de um - TCP/IP (*transmission control protocol / Internet protocol*) (SILVA, et al., 2002).

Mais tarde, nos finais da década de 80, Tim Berners-Lee e Angers Berglund no CERN (*European Particle Physics Laboratory*) desenharam um sistema de comunicações, a *World Wide Web* (designada a partir de agora por *Web*), constituída por servidores *Web* e clientes (*browsers*). Para disponibilizar a informação nos servidores *Web*, criaram uma linguagem de anotação, o HTML – *Hipertext Markup Language*, baseada numa simplificação do SGML - *Standard Generalized Markup Language* (uma linguagem de anotação criada pela IBM que consistia na utilização de um conjunto de palavras que descreviam partes de um documento). O objectivo primordial consistia no arquivo de conhecimento, contendo informações de diversas fontes, acessíveis através da Internet pelos utilizadores espalhados pelo mundo, guardados em diversos formatos (incluindo texto, imagens, sons e vídeos), interligados através de hiperligações (W3C, 2009) (BERNERS LEE, 1996).

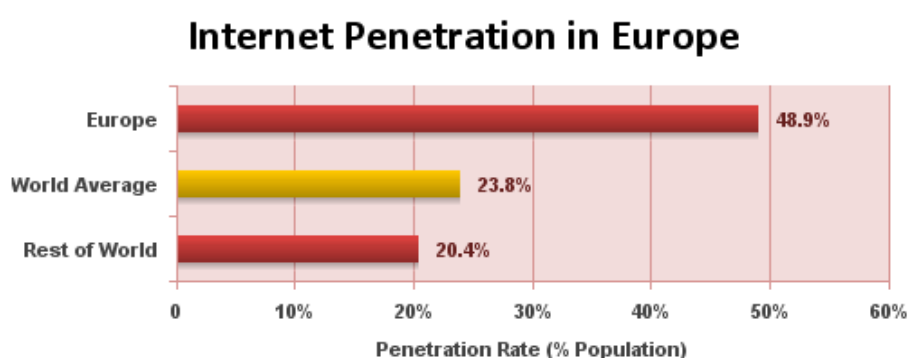
Este facto marcou o início de uma nova era, impulsionando fortemente a publicação de enormes quantidades de informação em linha, alterando para sempre os comportamentos e procedimentos habituais do mais normal dos cidadãos, ainda que indirectamente.

Sempre pautada com um forte crescimento, a *Web* passou a constituir um canal vital de comunicação global e um importante veículo para a disseminação e pesquisa de informação, para um crescente número de utilizadores, pela instantaneidade com que os documentos podem ser produzidos, actualizados e recuperados (MARTZOUKOU, 2004) (SILVA, et al., 2002).

Uma das grandes vantagens associadas à utilização da *Web* é a possibilidade que o receptor da informação tem de interagir directamente com o fluxo de informação, sem intermediários (OLIVEIRA, et al., 2001), tendo com principal propriedade a sua universalidade, residindo o seu poder na hiperligação (BERNES-LEE, et al., 2001).

A informação digital passou a assumir um importante valor social, num mundo onde o poder aquisitivo assume cada vez mais uma maior importância. “A informação em si mesma, é capital, representa valor, dinheiro, poder” (ALENCAR, 2004).

Segundo os dados, deste ano, da Internet World Stats (IWS, 2009), a Internet tem uma taxa actual de penetração de 23.8% na população do mundo, 48.9% no que diz respeito à Europa.



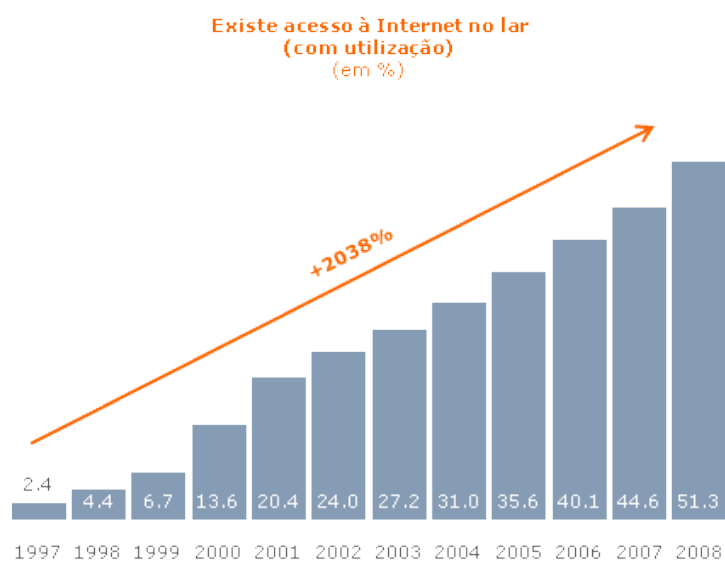
Source: Internet World Stats - www.internetworldstats.com
Based on 1,596,270,108 world Internet users for March, 31 2009
Copyright © 2009, Miniwatts Marketing Group

Fig. 2-1 Penetração da Internet na Europa

No que respeita aos países europeus, a Alemanha assume a primeira posição, com 55.2 milhões de utilizadores, seguidos do Reino Unido, com 43.8 milhões.

Relativamente à taxa de penetração da Internet na população as posições invertem-se com a Islândia a ocupar o primeiro lugar com uma taxa 90%, seguida da Noruega com 86%.

No que concerne ao caso português, Portugal possui 4.249.200 de utilizadores, registados em Dezembro de 2008, representando uma taxa de penetração 39.8%. Estes dados são confirmados no relatório anual do Barome Internet (MARKTEST, 2008), indicando a existência de um acesso à Internet na maior parte dos lares portugueses. Segundo o relatório, existiu um crescimento de 15% em relação ao ano anterior e de 2038% em relação a 1997, contabilizando-se cerca de 3,9 milhões de utilizadores de Internet em Portugal. Outro estudo no mesmo ano (MARKTEST, 2008), revela que 61% dos portugueses informam-se pela Internet.



Fonte: Marktest, Barome Internet 2008

Fig. 2-2 Evolução do acesso à Internet nos lares portugueses

Estas estatísticas demonstram a importância que a Internet passou a assumir no quotidiano dos cidadãos, mostrando um crescimento vertiginoso na última década.

No entanto nem todos os aspectos associados à Internet são considerados positivos. Existem preocupações relativamente à qualidade proporcionada pelo serviço *Web*, como por exemplo a consistência e validade da informação na *Web*.

2.1.1 Preocupações, Implicações, e Critérios de qualidade da Web

A chegada Internet, fortemente impulsionada pela facilidade de produção/publicação e de informação na rede, criou por outro lado obstáculos à procura de informação válida que satisfaça as necessidades do utilizador. Uma das maiores preocupações debatidas neste âmbito é a qualidade e consistência da informação encontrada.

A Internet está, por vezes, associada a alguns aspectos negativos como aponta Cardoso (CARDOSO, 2003) ao assinalar que “a Internet está, porventura, muito mais próxima da Biblioteca de Babel, sonhada pelo escritor argentino Jorge Luís Borges, do que da mítica Biblioteca de Alexandria”. Uma clara alusão às gigantescas quantidades de informação disponíveis na Internet, de uma forma pouco organizada e sistematizada.

“We begin with the assumption that Web pages and Web sites can be collected and categorized” (KOEHLER, 1999).

Um dos problemas, associados à Internet e à discussão da validade das fontes de informação na Internet, reside no facto da pesquisa (e a indexação) ser pouco fiável. “Não é raro navegar-se na Internet e retornar-se sem encontrarmos o que procurávamos, ou mesmo, quando se encontra, o conteúdo está incompleto ou de modo fragmentado, sobretudo quando não se conhecem bem os mecanismos de busca, e sua eficaz utilização” (ALENCAR, 2004). Segundo Yamaoka (YAMAOKA, 2003) “(...) o volume de informação disponível na Web é um dos problemas que actuais sistemas de busca são limitados para trabalhar. Os usuários encontram grandes dificuldades para realizar buscas com a precisão desejada, normalmente retornando grande quantidade de ruídos” (informação irrelevante para a pesquisa).

São apontados como motivos para essa deficiência o facto dos motores de busca ordenarem os resultados em relação ao número de termos de pesquisa que estes contêm, em detrimento da importância do documento (NIELSEN, 2005), onde os utilizadores esperam que essa informação seja devolvida de um ordem clara e ordenada (POLLOCK, 1997). A informação, de uma forma geral, é adquirida através de navegação casual, canais temáticos, motores de busca, onde na maior parte das vezes os utilizadores presumem que existe algum tipo de selecção da informação que é devolvida (BRANDT, 1996). Este problema acentua-se quando a informação tem uma validade, isto é, quando é válida por um período de tempo específico (MOODY, et al.,

1999). Por exemplo: informação financeira, notícias, informação para concursos públicos, entre outros. Um exemplo mais dramático pode encontrar-se na informação médica, onde os avanços científicos por vezes invalidam conhecimentos anteriores.

Estes factores ficam a dever-se à facilidade da disponibilização de informação em linha na Internet, por parte de qualquer utilizador. Brandt (BRANDT, 1996) aponta como grande vantagem da Internet, e ao mesmo tempo uma das suas maiores fraquezas, o facto de qualquer pessoa poder publicar informação na Internet, “ignorando-se” os benefícios das publicações tradicionais, tais como a revisão por uma equipa editorial, ou de especialistas, da informação antes de ser publicada. As fontes de informação na Internet estão a proliferar, contudo apenas uma pequena porção da informação disponibilizada é de boa qualidade (OLIVER, et al., 1997). Estes argumentos levantaram uma discussão relativa à qualidade de informação disponibilizada.

Outra das desvantagens apontada à informação disponibilizada é a sua volatilidade (persistência) e também os problemas relativos aos direitos de autor, que são constantemente ignorados (OLIVEIRA, et al., 2001).

Brandt (BRANDT, 1996) assume a informação como sendo algo importante para todos os indivíduos. No entanto, afirma ainda que, mais importante do que a própria informação é saber se esta é válida, fiável, credível e pertinente. Quando a informação é revista os utilizadores finais confiam na fonte e aceitam de uma forma natural como sendo válida. Quando não existe essa revisão são os próprios utilizadores que deverão efectuar a revisão. Tillman (TILLMAN, 1995) segue a mesma orientação, indicando a necessidade de uso dos mesmos critérios de avaliação da informação que faríamos em relação a um livro, um índice, uma pauta de música ou uma base de dados comercial. Adianta ainda que a Internet apenas difere das anteriores devido ao seu potencial de interacção com diversos tipos de media.

A verificação da fonte é uma das fases mais importantes no processo de validação de critérios de qualidade. É fundamental identificar a entidade intelectual (individual ou colectiva), procurando credenciais referentes ao autor, bem como, verificar a actualidade da informação através das datas de publicação e actualização indicadas (e também se a informação é actualizada periodicamente) (TOMAÉL, et al., 2001).

Os pré-requisitos para o uso efectivo da informação são: conhecer a sua existência; saber onde está armazenada; como aceder à informação (forma de acesso); saber como utilizá-la (MOODY, et al., 1999).

Os dez itens essenciais para a revisão de critérios de qualidade das fontes de informação (TOMAÉL, et al., 2001) são:

- **Identificação**, identificando os detalhes da entidade responsável pelo *site* (sítio da Internet), indicando elementos como contactos, objectivos da fonte ou a identificação da tipologia da fonte.
- **Consistência**, verificando a cobertura, validade e coerência da informação, e origens dos dados.
- **Autoria**, verificando a qualidade e competências da entidade intelectual responsável para produção da informação.
- **Adequação da fonte**, confirmando o tipo de linguagem utilizada e os propósitos da fonte de informação.
- **Ligações**, internas e externas, atestando a qualidade das informações prestadas.
- **Facilidade de uso**, através de uma análise de usabilidade e acessibilidade, coerência de navegação e recursos auxiliares à navegação (pesquisa...)
- **Composição** gráfica da fonte, verificando a existência de harmonia de utilização dos diversos elementos (texto, imagens...)
- **Restrições apercebidas**, através de restrições de acesso à informação, custos associados, direitos de autor, etc.
- **Suporte** ao utilizador.
- **Outras funcionalidades**, como por exemplo a disponibilização da informação em diversos formatos, ou em diversas línguas.

2.2 Sindicação e agregação de conteúdos na *Web*

O crescimento verificado na utilização da Internet, na publicação de novos conteúdos, levantou, como vimos, problemas na recuperação e consulta de informação existente na *Web*. Por outro lado a demanda de acesso à informação, mais concretamente a necessidade de obter descrições das colecções digitais, passíveis para serem processados por dispositivos, levaram ao surgimento de algumas propostas com os mais variados fins. Uma das propostas consistia na sindicação de conteúdos na *Web* através de *feeds* (canais) constituídas de elementos, descritos através de meta dados, com vista à sua distribuição automática na Internet.

2.2.1 História da Sindicação e Agregação de Conteúdos

A história da sindicação e agregação de conteúdos é particularmente difícil de descrever e sintetizar devido aos vários avanços e recuos nos diferentes formatos desenvolvidos pelos diversos actores envolvidos neste processo. Em tom de resumo, para uma

melhor compreensão, começamos por verificar na Tabela 2-1 a evolução da sindicacão de conteúdos nos diferentes formatos e versões.

Tabela 2-1 Quadro cronológico da evolução dos formatos de sindicacão

Formato	Ver.	Data	Companhia	Estado
scriptingNews		Dez./1997	UserLand	Obsoleto
RSS – RDF Site Summary	0.9	Mar./1999	Netscape	Obsoleto
scriptingNews	2.0b1	Jun./1999	UserLand	Obsoleto
RSS – Really Simple Syndication	0.91	Jul./1999	Netscape/UserLand	Obsoleto, mas ainda assim muito utilizado
RSS – RDF Site Summary	1.0	Dez./2000	RSS-DEV Working Group	Estável
RSS – Really Simple Syndication	0.92	Dez./2000	UserLand	Obsoleto, mas ainda assim muito utilizado
RSS – Really Simple Syndication	2.0	Set./2002	UserLand / Harvard	Estável
Atom	0.3	Dez./2003	Mark Nottingham	Estável

O precursor das *feeds* actuais, enumeradas no quadro anterior, foi o *Meta Content Framework* (MCF), desenvolvido pela Apple, que tinha o objectivo de providenciar informação e documentar as relações entre objectos existentes *sites Web*, *sites FTP* (*File Transfer Protocol*) e bases de dados. O MCF foi usado na altura em mais de 500 *sites*, para descrever a sua estrutura, entre os quais se incluía o famoso Yahoo! (GUHA, 2005). Entretanto uma mudança de estratégia interna na Apple, que culminou no abandono de alguns elementos-chave da equipa de trabalho, levou com que o projecto perdesse força (AYERS, et al., 2005). Um dos elementos-chave da equipa de trabalho da Apple era Ramanathan Guha, que transferiu-se em 1997 para a Netscape onde, em conjunto com Tim Bray e com a W3C, desenvolveu uma nova versão para o MCF baseado em XML, já com um nova designaçã: RDF (*Resource Description Framework*) (GUHA, 2005).

Paralelamente ao desenvolvimento da Netscape, a Microsoft desenvolveu o CDF (*Channel Definition Format*), que partilhava de uma estrutura similar, mas usando

uma lista de canais previamente fixados pela Microsoft, factor este que levou a que os utilizadores não aceitassem esta tecnologia (ELLERMAN, 1997).

Em 1997, Dave Winer (considerado por muitos como o pai do RSS, fundador da empresa UserLand) desenhou uma estrutura utilizando XML, conhecida por *scriptingNews format* (Winer, 1997), com as actualizações dos artigos mais recentes do seu *site* (Scripting News), incluindo os títulos e hiperligações para os seus artigos, despidos de *layouts* e formatações gráficas.

Esta ideia veio a originar dois anos mais tarde o RSS - *RDF Site Summary*, especificação RSS 0.90, adoptada e redesenhada pela Netscape nos seus *Software (compatível com a scriptingNews format)*.

Em tom de resposta, no mesmo ano, face ao RSS, Dave Winer melhorou a sua especificação inicial para a versão 2.0b1, onde compara a sua especificação com a do RSS (da Netscape), onde melhora acrescenta novas opções (WINER, 1999). Pouco depois a Netscape fez surgir a nova especificação RSS 0.91, incluindo já as modificações perpetuadas por Dave Winer, num esforço de standardização do formato. No entanto, devido à sua perda de interesse pela continuidade no desenvolvimento de negócios com portais, a Netscape cedo abandonou o projecto, sendo este entretanto adoptado pela Userland Software (fundada em 1988 por Winer, que tinha como objectivo o desenvolvimento de produtos de *Web logging*) e por um grupo de desenvolvimento livre chamado RSS-Dev Working Group que viria mais tarde a publicar a especificação RSS 1.0 ao mesmo tempo que a UserLand publicava a especificação RSS 0.92.

A especificação RSS 2.0 surgiu em 2002 (seguindo a linha do RSS 0.91) pela UserLand, mas em 2003 a empresa transferiu os direitos para a Berkman Center for Internet & Society at Harvard Law School. No entanto, as duas versões RSS 2.0 e RSS 1.0 mantiveram-se em funcionamento paralelo.

Esta divisão do desenvolvimento por diversos interessados levou a uma confusão, em termos de entendimento, levando com que surgissem várias especificações paralelas, com números de versão desordenados.

Mais tarde em 2003 surgiu uma nova especificação, o Atom (que não é uma sigla), como uma alternativa ao RSS 1.0 e 2.0., introduzindo melhorias. Muito embora tenha

sido adoptado por grandes empresas, como o Google, o Atom nunca foi massivamente adoptado pela comunidade, devido aos seguintes motivos:

- Os *sites* por todo o mundo continuaram a fornecer *feeds* RSS
- O surgimento dos *podcasts* (sindicação de ficheiros multimédia, distribuídos através da Internet para computadores e dispositivos de bolso) usando maioritariamente como base o RSS 2.0 (embora também exista em formato Atom)
- Muitos *sites* publicam apenas em um formato, normalmente em RSS 2.0.
- Os *sites* que publicam em Atom publicam também em RSS.
- O termo RSS passou a abranger todo o tipo de *feeds*.
- O aparecimento de um certo espírito “clubístico” em redor de cada um dos formatos.

Segundo os dados da Syndic8 (SYNDIC8, 2009), existem 566.986 *feeds* indexadas, distribuídas por RSS e Atom.

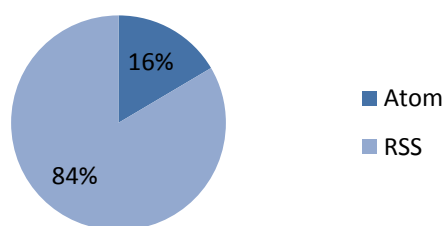


Fig. 2-3 Distribuição das *feeds* pelos formatos

Este quadro anterior demonstra que apesar de mais antigo o RSS continua mais popular que o Atom.

2.2.2 O que são as RSS ou Atom

“*First there was e-mail. Then there was the Web. Now there’s RSS.*” Greg Reinacker

Os RSS (e Atom) estão relacionados com informação, mais concretamente com fluxos de informação. Os RSS podem definir-se como “(...) um dialecto ou vocabulário da família XML, voltado especialmente para a captura automática e distribuição de conteúdos de sítios *Web*. Normalmente é usado para distribuição de conteúdos actualizados com frequência” (ALMEIDA, 2008).

A informação contida no *feed* é descrita através de meta dados, normalmente contendo um título, hiperligação para a fonte, descrição e os elementos individuais (itens),

também com os mesmos descritores (AYERS, et al., 2005). Esta forma de sindicacão providencia formas de publicar a informacão de uma forma estruturada, permitindo o desenvolvimento de novas aplicacões.

Segundo Dare Obasanjo (AYERS, et al., 2005), criador do Software RSS Bandit, afirma que a forma como os utilizadores obtêm, guardam e manipulam a informacão proveniente da *Web* está a mudar, fundamentalmente, devido ao aparecimento e crescimento de *feeds* como o RSS e o Atom.

Esta afirmacão é complementada por Greg Reinacker (AYERS, et al., 2005), fundador da NewsGator, afirmando que apesar da *Web* constituir um grande recurso, a informacão que nós queremos implica que tenhamos de ser nós a procurar por ela e não o contrário.

Podemos apontar algumas vantagens na sindicacão de conteúdos, para editores e subscritores:

- Meta dados, estruturando a informacão.
- Velocidade – As *feeds* são normalmente pequenas, permitindo uma rápida descarga
- Poupança de tempo – As *feeds* podem ser lidas em agregadores (leitores) de *feeds*, podendo ser descarregadas inúmeras *feeds* apenas com um único clique do rato.
- Privacidade – A subscricão não implica o fornecimento de dados pessoais (como o correio electrónico)
- Cancelamento e Protecção de SPAM – Como não deixamos o nosso correio electrónico inscrito em qualquer lista, podemos em qualquer altura cancelar a subscricão da *feed*.
- Organizacão – As *feeds* vêm organizadas por tópicos ou entradas, normalmente descritas numa linha temporal.
- Podem ser lidas por outras aplicacões, sem ser exclusivamente através de um *browser*.
- As *feeds* RSS e Atom estão a ser cada vez mais adoptadas. A indisponibilizacão de uma *feed* num *site* trará desvantagens competitivas.

- Permite aumentar o tráfego do *site*.
- Poderá influenciar os motores de busca, em termos de melhoria de posicionamento.

Em termos comparativos das *feeds* com as restantes tecnologias clássicas na Internet, podemos estabelecer o quadro comparativo, patente na Tabela 2-2.

Tabela 2-2 Quadro comparativo de vantagens e desvantagens entre os suportes *Web, Email e Feeds*

Suporte	Vantagens	Desvantagens
<i>Web</i>	<ul style="list-style-type: none"> • Facilidade de utilização • Quantidade de informação disponível • Hiperligação • Funcionalidades • Popularidade 	<ul style="list-style-type: none"> • Dificuldade na recuperação da informação / pesquisa • Desorganização • Implica (re)visitar inúmeros <i>sites</i> à procura de nova informação. • Mistura de elementos de <i>design</i> com os conteúdos. • Ausência de metadados • Indexação
<i>Email</i>	<ul style="list-style-type: none"> • Facilidade de utilização • Popularidade • Pode ser segmentado • Baixo custo • Possibilidade de responder ou reenca-minhar • Relevância 	<ul style="list-style-type: none"> • SPAM (lixo electrónico). • Intrusivo • Privacidade • Possibilidade de o nosso correio ser considerado SPAM. • Relevância
<i>Feeds</i>	<ul style="list-style-type: none"> • Aproximação dos conteúdos aos utilizadores. • Velocidade • Automático • Privacidade / Anonimato • Inexistência de SPAM • Permite trazer os utilizadores ao site • Podem ser lidas por diversos <i>Software</i> • Metadados • Agregação de conteúdos • Linha temporal. • Filtragem • Codificação em formato XML. • Podemos parar a subscrição em qualquer altura. • Separação do design do conteúdo • Inclusão de elementos multimédia • Eficiência 	<ul style="list-style-type: none"> • Desconhecimento por parte do público, por ser muito mais recente que as anteriores. • Normalmente não contém a informação integral mas sim uma hiperligação para esta. • Necessidade de utilização de Software de leitura

Analisando atentamente o quadro anterior podemos concluir que as *feeds* oferecem

vantagens comparativamente às formas de acesso clássicas ainda que estas sejam largamente muito mais populares e utilizadas pelo grande público.

2.2.3 Arquitectura de uma *Feed*

Verificamos anteriormente que a especificação RSS (e Atom) deriva do formato XML. Consiste na codificação de um ficheiro XML simples, usando na sua codificação das *tags* (anotações, etiquetas ou rótulos) a especificação em causa (RSS ou Atom).

De uma forma geral uma *feed* é constituída por um cabeçalho e por uma lista de elementos alinhados numa linha temporal.

No cabeçalho é definida a informação geral da *feed*, tal como por exemplo o título, hiperligação para a fonte, descrição, autor, língua, data de publicação entre outros. Na listagem de elementos, em cada elemento define-se o seu título, descrição, hiperligação, data de publicação, fonte e autor, entre outros.

Para uma melhor compreensão, a figura 2-4 descreve a estrutura genérica de uma *feed*.



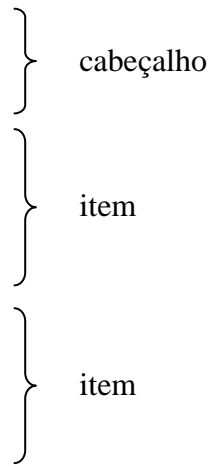
Fig. 2-4 Estrutura conceptual genérica de uma *feed*

Exemplificando em termos práticos, tendo por base a especificação RSS 2.0 e, considerando tratar-se de uma *feed* de informação noticiosa poderíamos obter o seguinte resultado:

```

<rss version="2.0">
  <channel>
    <title>Notícias ISCTE</title>
    <link>http://iscte.pt/</link>
    <description>Notícias do ISCTE</description>
    <pubDate>Tue, 09 Jun 2009 09:39:21 GMT</pubDate>
  </channel>
  <item>
    <title>Concurso Nacional Acesso ao Ensino Superior 2009</title>
    <link> http://iscte.pt/news.jsp?locallink=mainNews.xml#news170</link>
    <description>Como candidatar-se (...)</description>
    <pubDate>Tue, 09 Jun 2009 09:39:21 GMT</pubDate>
    <guid> http://iscte.pt/news.jsp?locallink=mainNews.xml#news170</guid>
  </item>
  <item>
    <title>INOV Contacto</title>
    <link> http://iscte.pt/news.jsp?locallink=mainNews.xml#news169</link>
    <description> Programa INOV Contacto - Estágios (...)</description>
    <pubDate>Tue, 09 Jun 2009 09:39:21 GMT</pubDate>
    <guid> http://iscte.pt/news.jsp?locallink=mainNews.xml#news169</guid>
  </item>
</channel>
</rss>

```



A estrutura da *feed* anterior poderá, no entanto, diferir consoante a especificação e respectiva versão, uma vez que a estrutura de *tags* difere em cada uma destas. Para um maior detalhe sobre as diferenças entre especificações, consulte os exemplos no apêndice A1.

2.2.4 Agregadores de Feeds

O objectivo final das produção/utilização de *feeds* consiste no consumo da sua informação por parte do utilizador. O termo de agregador (muitas vezes conhecido por leitor de RSS ou agregador de notícias, muito embora esta última denominação seja diminutiva relativamente às suas capacidades) é referido neste contexto como o aplicativo *online* ou local, ou *plugin*, que adquire, processa e mostra as informações de uma (ou mais) *feed* ao utilizador (AYERS, et al., 2005).

Os agregadores instalados localmente, de *desktop*, executam as suas tarefas em modo silencioso, de uma forma semelhante a um cliente de correio electrónico (como por exemplo o Outlook), procurando automaticamente por novas entradas nas *feeds* indicadas pelo utilizador. Quando o agregador notifica o utilizador assim que encontra novas entradas na numa determinada *feed*, disponibilizando a sua informação para leitura e oferecendo a possibilidade de reencaminhar o utilizador para a fonte.

Os agregadores *on-line*, de âmbito *Web* (sob a forma de um *site*), como por exemplo o Google Reader, oferecem funções semelhantes. A diferença reside em apenas ser necessário utilizar um *browser* para navegar na aplicação *on-line* (não é necessário instalar nenhuma aplicação no computador).

Os agregadores de *plugin* expandem/complementam a funcionalidade de uma determinada aplicação (como por exemplo um *browser* ou um cliente de correio electrónico), por forma a permitir que estes possam, complementarmente à sua função principal, efectuar a leitura de *feeds* (OLA, et al., 2005).

Cada agregador possui as suas próprias valências que os distinguem dos restantes. De uma forma geral, normalmente, todos proporcionam as seguintes funcionalidades:

- Gerir as subscrições de *feeds*, permitindo adicionar e remover *feeds* consoante o interesse do utilizador.
- Procurar por actualizações nas *feeds*, automaticamente (através de intervalos regulares) ou manualmente (por acção do utilizador), verificando se existem novas entradas desde a última actualização.
- Mostrar ao utilizador o conteúdo de cada *feed*, permitindo a que este possa seleccionar a hiperligação do item se assim o entender.
- Pesquisar nos conteúdos.

A interacção entre o agregador (cliente) e a fonte (servidor) é estabelecida através de pedidos HTTP, conforme se pode observar na Fig. 2-5.

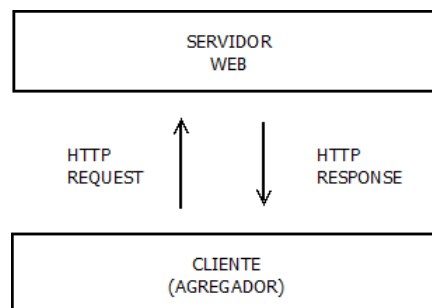


Fig. 2-5 Arquitectura de comunicação entre um agregador e servidor

Esta forma de interacção é idêntica à existente entre um *browser* e um servidor *Web*. Assim que a *feed* é publicada ela fica disponível para ser lida pelos clientes, neste caso os agregadores.

A Tabela 2-3 mostra alguns exemplos, entre as inúmeras possibilidades, de agregadores disponíveis para utilização, na maior parte de forma gratuita, disponíveis para os mais variados dispositivos e plataformas.

Tabela 2-3 Exemplos de Agregadores de Feeds

Agregador	Tipologia
Google Reader	Aplicação <i>Web</i>
NetVibes	Aplicação <i>Web</i>
<i>FeedDemon</i>	Aplicação de Desktop
Sage	Plugin para Firefox
Outlook v.2007	Funcionalidade da Aplicação de Desktop
NewsGator	Aplicação de Desktop
RSS Bandit	Aplicação de Desktop
SPB Insight	Aplicação para PDA

Para além dos agregadores também os *browsers* mais recentes como o Internet Explorer 8.0, Firefox 3, Safari ou Opera 10 conseguem já, de base, ler *feeds* nos mais variados formatos, sem recurso a *plugins*. Surgiram também aplicações que transformam os *feeds* para outros formatos, como por exemplo serviços conversão automática de *feeds* em mensagens de correio electrónico.

2.3 As SMS – *Short Message Service*

Uma das formas de comunicação mais populares, com cariz até fenomenal, nos dias de hoje consiste na utilização de mensagens escritas para telemóveis (SMS) como forma de comunicação entre indivíduos.

Pretende-se com este estudo abordar a criação de um serviço que permita receber a informação contida nas *feeds* e reencaminhar por SMS o respectivo conteúdo ao utilizador interessado. Neste âmbito pretendemos nos pontos seguintes estudar as características das SMS e da sua utilização.

2.3.1 O que são as SMS?

Short Message Service (SMS), conhecido popularmente por mensagens escritas de telemóvel, é um serviço de comunicação definido no sistema de comunicações para telemóveis (e outros terminais) GSM (*Global System for Mobile Communications*) desde o seu início. O serviço SMS permite a troca de mensagens escritas entre telefones tendo como limite de texto 160 caracteres (latinos).

“Developed as part of the GSM Phase 1 ETSI technical specifications, the Short Message Service (SMS) allows mobile stations and other network-connected devices to exchange short text messages.” (LE BODIC, 2005)

Apesar desta limitação (apesar de ser possível actualmente a técnica de concatenação de mensagens), os SMS oferecem algumas vantagens que as tornam ainda assim muito apetecíveis:

- Podem ser enviadas e lidas em qualquer altura.
- É relativamente muito simples enviar/receber uma SMS.
- Podem ser enviadas para destinatários indisponíveis / não conectados (entrega automática), sendo as mensagens entregues no momento em que o destinatário se conecta à rede.
- São “silenciosas”.
- As SMS são suportadas por todos os telefones GSM.
- Podem conter dados binários.
- Podem ser utilizadas como forma de pagamento.
- Permite a notificação de recepção ao emissor.
- Permitir um ambiente multi-tarefas (o receptor/emissor poder fazer outras coisas ao mesmo tempo, como por exemplo conduzir)
- A sua utilização não é exclusiva no terminal, isto é, não impece o terminal esteja a aceder a outros serviços (por exemplo uma chamada de voz).
- Disponibilidade do receptor, comparativamente às restantes formas de comunicação escrita (email, carta, fax, etc.).

As aplicações das SMS são diversas e ilimitadas: comunicação interpessoal, alertas de serviços de informativos, comércio electrónico, publicidade, etc.

2.3.2 As SMS nos dias de hoje

Actualmente as SMS ocupam significativamente um papel activo na comunicação entre indivíduos, sendo uma dos meios de comunicação mais utilizados hoje em dia.

Segundo (MARKTEST, 2008), os Portugueses enviam diariamente em média 12 SMS, semanalmente 84 SMS, registando-se um acréscimo de mais de 70% relativamente a 2006 (49 mensagens enviadas). Apesar deste valor os dados registam comportamentos diferenciados, consoante a faixa etária, onde os jovens ocupam uma grande parte significativa.

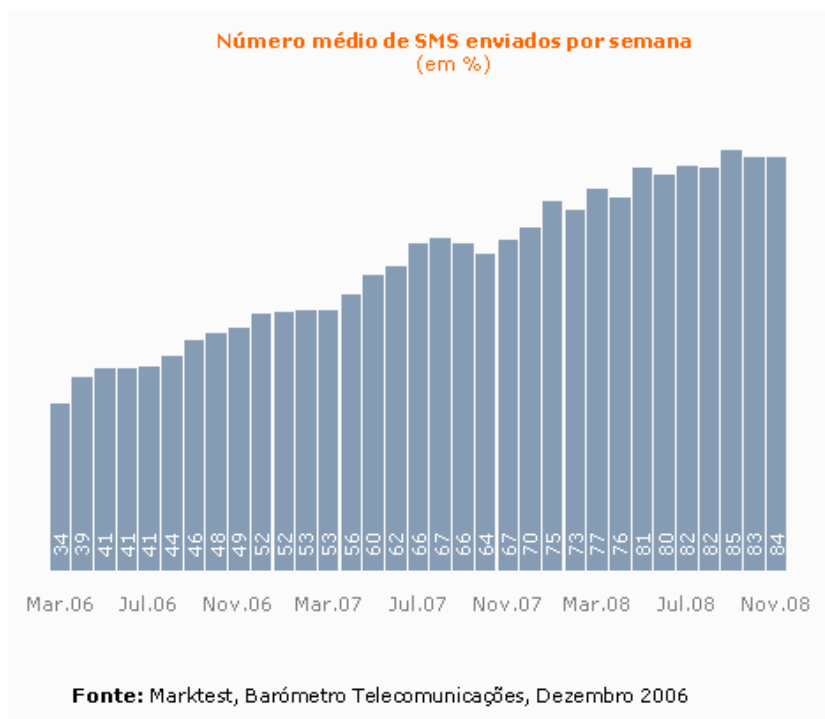


Fig. 2-6 Evolução do número de SMS enviados por semana de 2006 a 2008

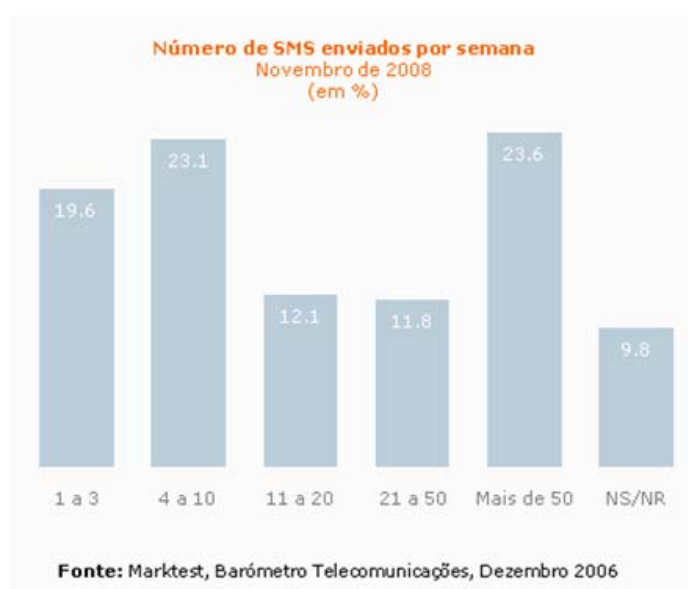


Fig. 2-7 Número de SMS enviados por semana em 2008

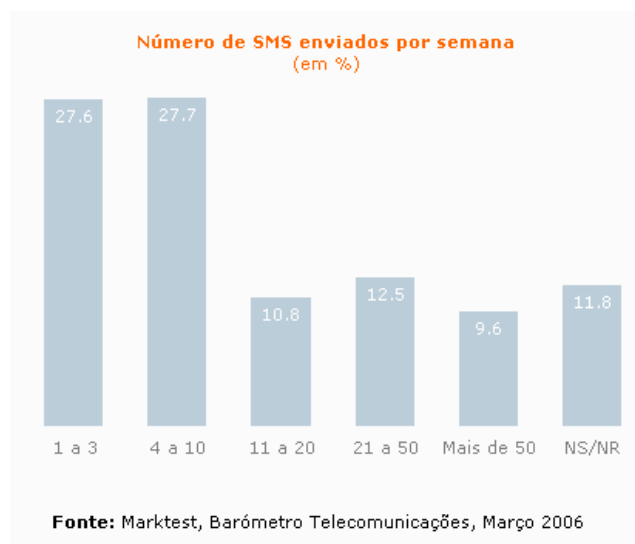


Fig. 2-8 Número de SMS enviados por semana em 2006

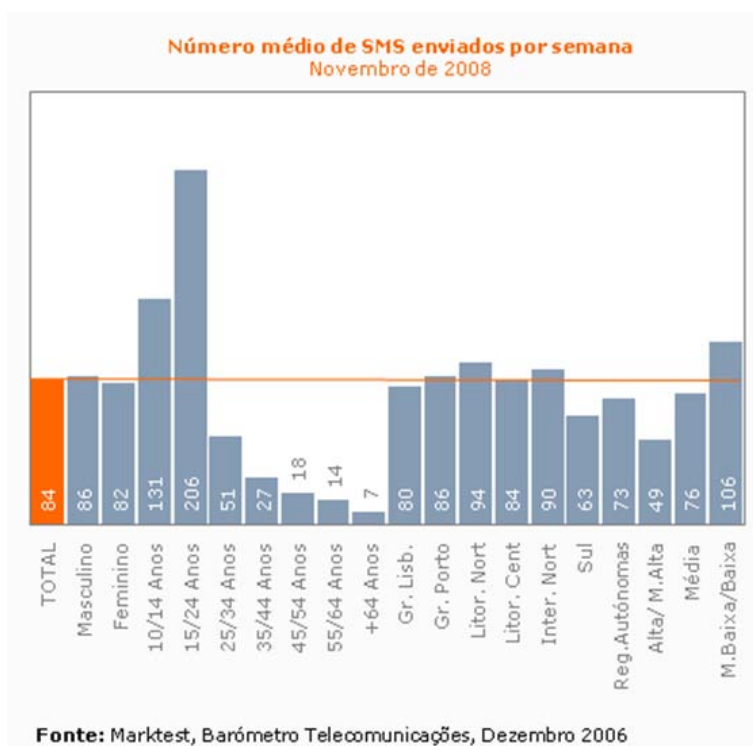


Fig. 2-9 Número médio de SMS enviados por semana, em 2008, por idades

Esta tendência de utilização das SMS não se verifica apenas em Portugal mas também no resto do mundo.

2.3.3 Arquitectura das SMS

O envio dos SMS não é efectuado directamente entre o emissor e o receptor, mas através de um centro de mensagens (SMSC) ou, em alguns casos, através de um agregador de SMS.

2.3.3.1 SMSC – Short Message Service Center

O envio dos SMS não é efectuado directamente entre o emissor e o receptor, mas sim através de um centro de mensagens (SMSC – *Short Message Service Center*), que gere a recepção e entrega das mensagens e notificações de submissão e recepção. O endereço do SMSC é configurado directamente em cada terminal e depende de cada operador de telecomunicações, vindo normalmente pré-configurado nos cartões destes.

“The SMS is a store and forward service. In other words, short messages are not sent directly from sender to recipient, but via an SMS Center. Each mobile telephone network that supports SMS has one or more messaging centers to handle and manage the short messages” (MOBILECOMMS, 1992).

Esta interacção é possível verificar com maior detalhe através do seguinte diagrama:

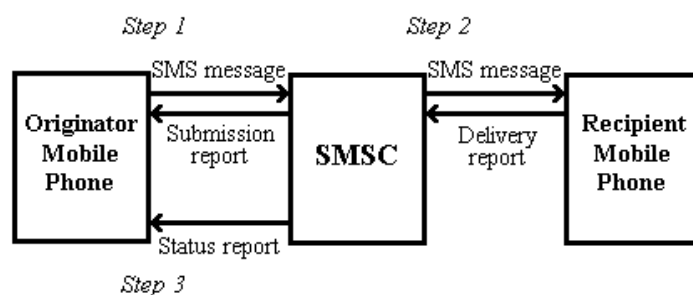


Fig. 2-10 Arquitectura de comunicação no envio de SMS através de um SMSC

Fonte: http://www.developershome.com/sms/sms_tutorial_html_m780f3998.png

“(…) the main functions of the SMSC are the relaying of short messages between SMEs and the store-and-forwarding of short messages (storage of messages if the recipient SME is not available)” (LE BODIC, 2005).

Muita das vezes o destinatário e emissor não se encontram a funcionar no mesmo operador de telecomunicações tendo que existir uma comunicação entre os SMSC de ambas as operadoras envolvidas, descrito no esquema seguinte:

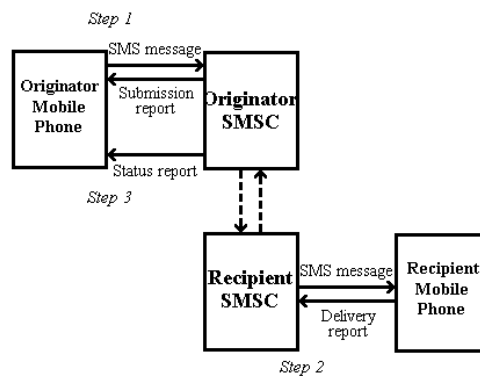


Fig. 2-11 Arquitectura de comunicação entre SMSC no envio de um SMS

Fonte: <http://www.developershome.com/sms/intraInterInternationalSMS2.asp>

O acesso à SMSC poderá ser efectuado sem ser necessariamente através de um terminal GSM: normalmente as operadoras permitem o acesso (por exemplo através de protocolo http) à SMSC, mediante um contrato com custo associado, de aplicações externas à SMSC, com vista ao envio dinâmico de SMS. No entanto este contrato, configuração e instalação tem que ser visto particularmente com cada operadora, tornando muito custosa a instalação.

2.3.3.2 Envio de SMS através de Agregadores de SMS

O agregador de SMS (algumas vezes conhecido por SMS *gateway* na *Web*) é uma das formas mais convencionais (e fáceis) de implementar um serviço de envio de SMS. O agregador possui acordos e interfaces de acesso às SMSC estabelecidos com as operadoras de telecomunicações e disponibiliza aos seus clientes a possibilidade de conectar através de um *Webservice*, API ou outros serviços e protocolos.

Esta entidade permite assim a descentralização de recursos na contratualização de serviços com as diferentes operadoras de telecomunicações e no desenvolvimento acrescido de interfaces para cada uma destas.

Existem no mercado vários agregadores disponíveis, diferenciando-se pelo tipo de serviço prestado e aos públicos a que se destinam (pouco ou muito tráfego).

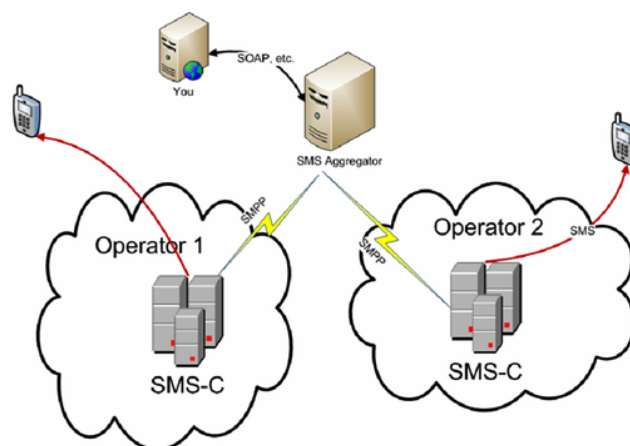


Fig. 2-12 Arquitetura de um Agregador de SMS

Fonte: (RETFORD, et al., 2007)

Em alternativa à utilização de uma gateway *Web* para o envio de SMS, existe também a possibilidade de utilização de uma solução simples utilizando um *smartphone* normal e um contrato normal com uma operadora de telecomunicações. Este procedimento torna o processo menos custoso em termos de implementação, uma vez que precisamos apenas de um contrato simples (cartão pré-pago por exemplo) mas, por outro, estamos limitados à capacidade de processamento deste *smartphone*, conforme adiantam :

“Scale will likely be your biggest limiting factor when using a smartphone-based SMS solution. In our experience, a single phone can process only 30 messages per minute. If your service succeeds, you may wish to support orders of magnitude more messages than this at peak use.” (RETFORD, et al., 2007)

2.3.4 Interação através de SMS

As SMS possuem características singulares que as distinguem de qualquer outra forma de comunicação. Os 160 caracteres de extensão máxima, no melhor dos casos, de uma SMS são porventura a sua característica mais limitadora, devendo adoptar-se uma escrita cuidada e adequada a essa missão:

- Capacidade de síntese
- Barreiras no *input*, devido ao tipo de teclado existentes nos terminais
- Interactividade lenta
- Utilização de abreviaturas

- Limite na utilização de alguns conjuntos de caracteres.
- Múltiplos SMS, em última alternativa, uma vez que os operadores muitas vezes cortam a mensagem sem o nosso controle.

Por outro lado deverá haver um grande cuidado no desenvolvimento de um serviço utilizando por base as SMS, uma vez que estas são intrusivas para os utilizadores. O envio em excesso de SMS poderá provocar um mal-estar perante o receptor. Para isso deverão ser desenvolvidas funções que permitam controlar a periodicidade de recepção e filtragem de conteúdos.

Nos E.U.A., entre outros exemplos, existe já leis que protegem o consumidor de SPAM (lixo electrónico) de SMS. De uma forma geral:

- Não enviar SMS não solicitadas
- Certificar-se que o utilizador pode cancelar ou modificar o serviço quando entender.
- Tentar definir períodos de recepção (para evitar perturbações em horas de repouso, por exemplo receber SMS a meio da noite).

A disponibilidade dos indivíduos para a recepção das SMS constitui ao mesmo tempo um perigo:

“SMS is intrusive. Part of the attractiveness of SMS lies in its direct access to user attention. Email is filtered, snail mail glugged, but who doesn't reach immediately for their phone when the familiar chirp of an incoming message sounds?” (RETFORD, et al., 2007)

Por outro lado as SMS podem operar simultaneamente com outros serviços no telemóvel, como a recepção de chamadas de voz ou de dados, sem que exista para isso um decréscimo de cobertura ou qualidade no serviço.

“Short messages can be sent and received simultaneously with GSM voice, data and fax calls. This is possible because whereas voice, data and fax calls take over a dedicated radio channel for the duration of the call, short messages travel over and above the radio channel using the signaling path” (RETFORD, et al., 2007).

Os SMS constituem uma extensão natural das notificações por email e outros serviços on-line (AGUADO, et al., 2008). Os principais motivos para a sua utilização consiste no divertimento (da sua utilização), disponibilidade (a qualquer a hora), funcionalidade, entre outros (PETERS, et al., 2003).

3 Sistema de Notificações Automáticas de Informações Provenientes de Fontes *Web*, através de Mensagens Escritas para Telemóveis

Pretende-se neste capítulo apresentar, descrever e analisar o problema a que esta tese se propõe resolver.

3.1 Descrição do problema

A crescente utilização da Internet como meio primordial para pesquisa de informação, exige às organizações e aos indivíduos a necessidade constante de disponibilizar “canais” abertos e actualizados que possibilitem a troca de informação, de forma cada vez mais rápida e espontânea. Segundo os dados estatísticos da Markest (MARKTEST, 2008), a utilização da Internet aumentou 7 vezes desde os últimos 10 anos, tendo registado um acréscimo médio anual na ordem dos 27%, apontando a existência de cerca de 4 milhões de utilizadores de Internet.

Neste novo panorama, os meios de comunicação, nomeadamente os digitais, assumem um papel fundamental na sociedade. A chegada da era digital veio derrubar barreiras físicas que antes se impunham, face aos meios tradicionais de acesso, permitindo o fácil acesso, por parte dos utilizadores, à informação.

O surgimento da Internet desencadeou, no final do século passado, a procura de novas formas de trocar informação. Uma das formas mais básicas proporcionadas pela rede global, consiste na consulta e disponibilização de informação em sítios (*site*) na Internet. Os sítios na Internet possibilitam às entidades marcarem uma presença constante, disponível para consulta durante 24 horas por dia, 365 dias por ano, em condições

muito vantajosas comparativamente a outras formas, como escritórios de atendimento, lojas, *call centers*, publicações em papel, etc..

Ao publicar os seus sites na Internet, estas entidades depararam-se com uma nova realidade: a imperatividade de possuir os seus sites actualizados, a crescente exigência por parte dos visitantes do site em obter informação actualizada. Normalmente esta acção de procura por informação actualizada é despoletada pelo próprio utilizador. Para além disso pressupõe que este esteja “munido” de *Hardware* e *Software* necessário para aceder e consultar a essa informação na rede.

Esta procura conduziu a uma resposta por parte da indústria com a produção de novos produtos, oferecendo ao mercado equipamentos portáteis com funcionalidades de acesso à Internet. Entre estes podemos encontrar os PDA (*Personal Digital Assistant*), UMPC (*Ultra Mobile Personal Computer*), *Smartphones* (telemóvel com funcionalidades de *Software* avançadas) ou mais recentemente com os *Netbooks* (computador portátil, normalmente, de baixo custo tendo em vista a utilização em âmbito *Web*) entre outros.

No entanto, nem todos têm a possibilidade de ter acesso a um destes terminais. Na maior parte das vezes devido ao seu alto custo de aquisição. Outras das vezes pela necessidade de competências e aptidões adicionais (do que é normal) para saber configurar e operar esses serviços plenamente. Mesmo para aqueles que têm acesso a este tipo de terminais, e que dispõem de competências necessárias para operá-los, o acesso à Internet através da rede móvel é ainda caro, muitas vezes cobrado pela quantidade de dados transaccionada. Ou ainda pela simples falta de vontade de transportar insistentemente estes dispositivos para qualquer lugar.

Por outro lado, a procura incessante de informação por parte do utilizador leva com que este, muitas das vezes, aceda à Internet e não encontre a informação actualizada que procura (havendo neste caso lugar a um tráfego de informação desnecessária), uma vez que obtém uma elevada taxa de ocorrências para uma baixa taxa de persistência.

Uma solução para este cenário consiste na inversão dos papéis: desenvolver um serviço que possibilite, em detrimento do papel activo do utilizador na pesquisa de informação, ser a própria informação (assim que disponível) ser canalizada automaticamente e rapidamente para o utilizador. Dentro das diversas possibilidades para atingir

este objectivo, o cenário ideal para a possível solução para este problema, será usar uma forma de comunicação muito simples, acessível a qualquer indivíduo, disponível através de um dispositivo que já possua actualmente e que o transporte para qualquer lugar: os telemóveis e as SMS.

As SMS oferecem vantagens inegáveis na concepção de serviços:

«Unlike a PC, people tend to carry their mobile phones with them everywhere they go, and they never turn them off. Because this is the primary distinguishing characteristic of an SMS service, you need to ask yourself "what do people need access to everywhere they go, at any moment?» (RETFORD, et al., 2007)

Pretende-se com esta tese abordar a criação, desenvolvimento e avaliação de um sistema de notificações automáticas, recorrendo a mensagens SMS, que permita a que um dado utilizador, utilizando apenas um telemóvel GSM “normal”, tenha a possibilidade de ser notificado regularmente sobre informações ou actualizações verificadas em fontes de informação *Web* (em que este tenha manifestado interesse em seguir). Por outras palavras, o utilizador, após definir as suas fontes de informação *Web*, assume um papel passivo sendo notificado através de mensagens escritas para telemóveis, informando-o de novas informações publicadas nas referidas fontes.

Através de uma pesquisa sobre sistemas com modelos semelhantes já projectados, verificou-se que este tipo de funcionalidade (notificação por SMS) existe apenas em alguns casos isolados, proporcionados individualmente por alguns sites (normalmente portais de renome) ou organizações, e de formas muito distintas entre estas. Normalmente com um elevado custo associado, com apenas os seus conteúdos, implicando que o utilizador tenha de efectuar uma adesão e configuração individual em cada um destes serviços. Não temos conhecimento actualmente sobre um sistema que permita agregar globalmente diversas fontes de informação e que as converta em SMS.

Desta forma, pretendemos reduzir as barreiras de acesso ao conhecimento, oferecendo uma forma de alargar as possibilidades aos indivíduos e entidades de poderem receber informação, com recurso à agregação de conteúdos, de uma forma muito cómoda, personalizada e sem grandes exigências em termos de equipamentos necessários.

«Most of the information needs today can be satisfied by searching and browsing the Web. However, repetitive tasks such as monitoring information on Web sites should be done automatically on behalf of the user. Likewise, important events such as new

information available on a monitored Web site should be actively reported to the user.» (BAUMGARTNER, et al., 1999)

Com vista à resolução do problema proposto nesta tese, pretende-se desenvolver um sistema que permita ao utilizador:

- Criar ou editar a sua conta pessoal de utilizador.
- Adicionar, editar e organizar as suas fontes de informação (agregação de *feeds*).
- Ter a liberdade de adicionar filtros nas fontes de informação.
- Definir a periodicidade das notificações.
- Permitir o acesso à fonte original de informação.

Nesta ordem de ideias, nos tópicos seguintes iremos abordar e descrever os episódios de desenvolvimento com vista à prossecução da solução para o problema proposto.

3.2 Análise

O desenvolvimento de um sistema de informação deve obedecer a um conjunto de regras, etapas e actividades de modelação devidamente sistematizadas, usando para esse objectivo uma metodologia (O'DOCHERTY, 2005). Esta necessidade, de uso de uma metodologia, torna-se imperativa quando as equipas de desenvolvimento são numerosas ou o projecto é de médias ou grande dimensões.

O processo de modelação de um sistema de informação consiste no desenho (de modelos) da aplicação antes de iniciar o desenvolvimento de código. Deriva de outras formas de engenharia, como por exemplo a Construção Civil, onde a modelação é uma técnica aceite. No entanto esta analogia tem defeitos óbvios, na medida em que as actividades são distintas (construção civil e desenvolvimento de sistemas).

«*A model is a simplification of reality*» (BOOCH, et al., 1998). A modelação é a parte central de todas as actividades que levam ao desenvolvimento de bom *Software*. Os modelos são construídos para visualizar, especificar, documentar e construir a arquitectura do sistema a ser desenvolvido, e principalmente para gerir o risco.

A adopção de uma metodologia (ou até de algumas etapas básicas de uma metodologia) produz os seguintes benefícios no desenvolvimento de um sistema de informação:

- Impor disciplina e conseqüentemente poupar de tempo e poupar código.
- Proporciona um maior conhecimento sobre o problema e conseqüentemente um maior controlo e qualidade na solução.

- Melhora a comunicação entre a equipa.
- Prevê sempre a etapa seguinte.
- Produz código mais flexível e extensível.
- Reduz riscos.

O *Unified Process*, proposto por Jacobson, Rumbaugh e Booch ("*the three amigos*" - como ficaram conhecidos), é uma metodologia de desenvolvimento iterativa e incremental, criada a partir de metodologias existentes (como a metodologia em cascata, espiral, iterativa e incremental) (O'DOCHERTY, 2005).

O *Unified Process* assenta em duas dimensões: temporal e processual. Cada uma destas dimensões está segmentada. A dimensão temporal assenta em 4 fases:

- Início (definição da ideia, visão geral do projecto).
- Elaboração (levantamento de requisitos para o sistema, planeamento de actividades e recursos).
- Construção (desenvolvimento iterativo).
- Transição (disponibilização gradual aos utilizadores).

A dimensão processual consistem em disciplinas (actividades) técnicas de análise e modelação do negócio, levantamento de requisitos, análise, desenho, programação, teste e instalação. Estas fases e disciplinas são articuladas através de iterações, como podemos observar na figura 3-1.

Em cada uma destas actividades podem ser usados modelos produzidos em UML (*Unified Modeling Language*) ou técnicas de gestão de projecto.

A UML permite identificar e integrar os aspectos de natureza organizacional e elementos de natureza tecnológica, determinando as regras, processos e fluxos de informação. Através dos seus modelos podemos visualizar, especificar, construir e documentar um sistema.

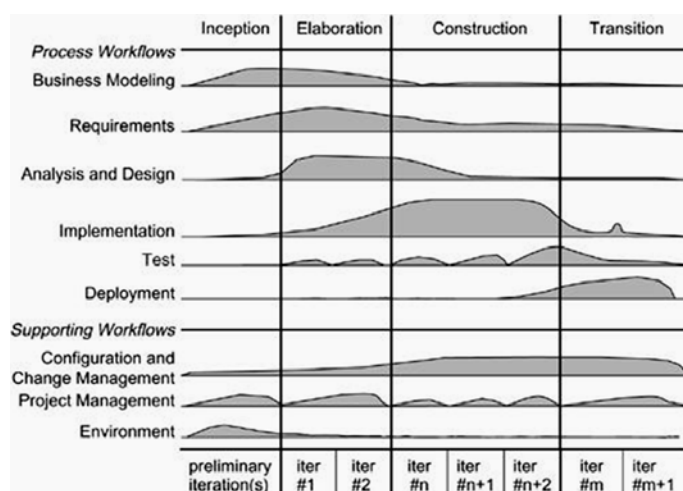


Fig. 3-1 Ciclo de vida no desenvolvimento de um sistema de informação

(BOOCH, et al., 1998)

Segundo (NUNES, et al., 2004) "a UML é uma linguagem que utiliza uma notação padrão para especificar, construir, visualizar e documentar sistemas de informação orientados por objectos", permitindo identificar e integrar os aspectos de natureza organizacional e elementos de natureza tecnológica, determinando as regras, processos e fluxos de informação.

(BOOCH, et al., 1998) refere quatro princípios básicos de modelação:

1. A escolha dos modelos tem uma profunda influência na forma de como o problema é solucionado.
2. Cada modelo pode ser desenhado em diferentes níveis de precisão.
3. Os melhores modelos estão conectados à realidade.
4. Um único modelo é insuficiente. Qualquer sistema deve ser analisado através de modelos independentes.

A análise e desenvolvimento do protótipo do sistema de informação a que se propõe esta tese, de Notificações Automáticas de Informações Provenientes de Fontes *Web*, através de Mensagens Escritas para Telemóveis, será produzido através de elementos de modelação em UML, usando para atingir esse objectivo os princípios e boas práticas previstas de algumas das fases e disciplinas da metodologia *Unified Process*, uma vez que pretendemos apenas desenvolver um projecto de pequena dimensão, um pequeno protótipo. Nas secções seguintes proceder-se-á à pré-visualização do sistema a desenvolver, através do desenvolvimento/estudo do problema com recurso a alguns dos modelos/ferramentas de UML.

3.2.1 Levantamento dos Requisitos

O levantamento de requisitos decorre da descoberta dos objectivos a atingir no desenvolvimento e decorre de dois aspectos: modelação do modelo de negócio e especificação dos requisitos de sistema (O'DOCHERTY, 2005).

Um requisito é uma funcionalidade, comportamento, característica ou serviço considerada relevante para o utilizador na utilização do sistema (NUNES, et al., 2004). A falta de requisitos, ou a existência de requisitos deficientes, são a maior causa para falhas nos projectos de *Software* (GASTALDO, et al., 2003).

Podemos classificar os requisitos em funcionais e não funcionais. Os requisitos funcionais são aqueles que expressam as funcionalidades ou serviços que é esperado que o sistema possa produzir, onde são descritas as entradas (*inputs*), processamento e saídas (*outputs*). Os requisitos não funcionais declaram a qualidade, desempenho, segurança, facilidade de uso do sistema (usabilidade), eficácia, eficiência e satisfação, que são esperados dos requisitos funcionais, desempenhando um papel crítico no desenvolvimento de sistemas.

3.2.1.1 Requisitos Funcionais

No desenvolvimento do sistema para a resolução do problema proposto, identificaram-se os seguintes requisitos funcionais:

- O sistema deverá possibilitar o registo individual dos utilizadores, bem como possibilitar a sua autenticação no sistema.
- O utilizador autenticado deverá poder editar o seu próprio perfil, adicionar, editar, organizar ou até eliminar as suas fontes de informação (*feeds*) e também adicionar filtros de pesquisa nas mesmas.
- O utilizador poderá definir ou alterar a periodicidade das notificações, bem como a escala horária em que o envio poderá ser processado.
- O utilizador deverá poder consultar o histórico de envios.
- O sistema deverá verificar continuamente a existência de alterações nas *feeds*, isto é, a existência de novos itens nas *feeds* e, em caso afirmativo, registar e marcar os itens para envio para os respectivos subscritores.
- O sistema deverá seleccionar e processar o texto para envio, por SMS aos utilizadores, utilizando abreviaturas que possibilitem a economia de caracteres, devido à restrição de 160 caracteres.
- O sistema deverá poder enviar os SMS, registando o sucesso de envio, para evitar hipotéticos envios duplicados.

- No processo de registo do utilizador deverá ser verificado a validade dos dados introduzidos (email, número de telemóvel, etc.). Deverá confirmar-se a autenticidade do número de telemóvel introduzido recorrendo a um envio de uma cadeia aleatória de texto para posterior comparação e validação no *site*.
- Quando o utilizador adiciona uma nova *feed* ao seu perfil, o sistema deverá verificar se a mesma já se encontra registada (por este ou por outro utilizador) a fim de evitar redundância. Se a *feed* não existe no sistema, o sistema deve então verificar se é válida (se a sua estrutura é válida, se consegue ser processada). O sistema deverá marcar inicialmente todos os itens existentes na *feed* como processados com a finalidade de o utilizador apenas receber os novos itens a partir desse preciso momento.
- No processo de captura da *feed* o sistema deverá registar o tempo decorrido entre o início e final do processamento da mesma. O objectivo é saber qual o tempo médio de captura, bem como o tempo médio decorrido entre publicação de itens, com o objectivo de cálculos que levem a que o sistema possa poupar recursos e ao mesmo tempo responder com eficiência e eficácia em termos de tempo de resposta.
- Durante o procedimento de envio de SMS o sistema deverá verificar se a hora actual se encontra entre o período definido para cada utilizador a fim de evitar envios a horas consideradas inconvenientes pelo utilizador (por exemplo o período de sono). Adicionalmente deverá registar o tempo decorrido no envio do SMS, bem como o sucesso ou insucesso do envio, com o objectivo de se poder avaliar qual a capacidade de envio do sistema e conseqüentemente saber quando potencialmente atingirá os seus limites.
- O SMS deverá incluir no final um endereço *Web* (http) reduzido para possibilitar que o utilizador possa consultar a fonte original da informação se assim o entender.

3.2.1.2 Requisitos Não Funcionais

Os requisitos não funcionais referem a qualidade do desempenho dos requisitos funcionais e conseqüentemente do sistema, bem como requisitos que proporcionam a boa experiência do utilizador. Foram identificados os seguintes requisitos não funcionais:

- Os novos conteúdos deverão ser entregues até 24H da data/hora do seu registo, uma vez que o utilizador poderá limitar a escala de disponibilidade horária para a recepção da informação.
- Concepção da aplicação em paradigma *Web*.
- O sistema deve ser fácil de aprender e de memorizar.
- A interface deverá ser agradável para o utilizador.
- A interface deverá ser construída seguindo as directrizes da W3C para a acessibilidade.

3.2.1.3 Definição de *Use Cases*

Definidos os requisitos do sistema, os *Use Cases* (casos de utilização) constituem uma técnica em UML para representar o levantamento de requisitos de um sistema com o objectivo de obter um entendimento comum dos mesmos, podendo ser definidos numa perspectiva de negócio ou de sistema (NUNES, et al., 2004).

Um *Use Case* define a forma de uma parte do negócio ou sistema é utilizado (JACOBSON, et al., 1992). É constituído por actores (uma entidade que interage com o sistema) e por relações entre este e os diversos *cases* do sistema.

O levantamento de requisitos, para o sistema proposto nesta tese, permitiu identificar os seguintes actores e *Use Cases* associados. Os actores "leitores de *feeds*" e "estafeta de SMS" constituem automatismos que o sistema deverá efectuar.

Tabela 3-1 Actores e *Use Cases*

Actor	<i>Use Cases</i>
Utilizador Final	Gerir (adicionar, configurar, remover) <i>feeds</i> ao seu perfil pessoal
Leitor de <i>Feeds</i>	Ler as <i>feeds</i> e capturar todos os itens novos
Estafeta de SMS	Seleccionar os itens ainda não enviados, por cada <i>feed</i> subscrita pelo utilizador, e entrega-os por SMS.

O diagrama seguinte, na Fig. 3-2, representa a visão geral do sistema proposto nesta tese, abordando os principais *cases* identificados no levantamento de requisitos.

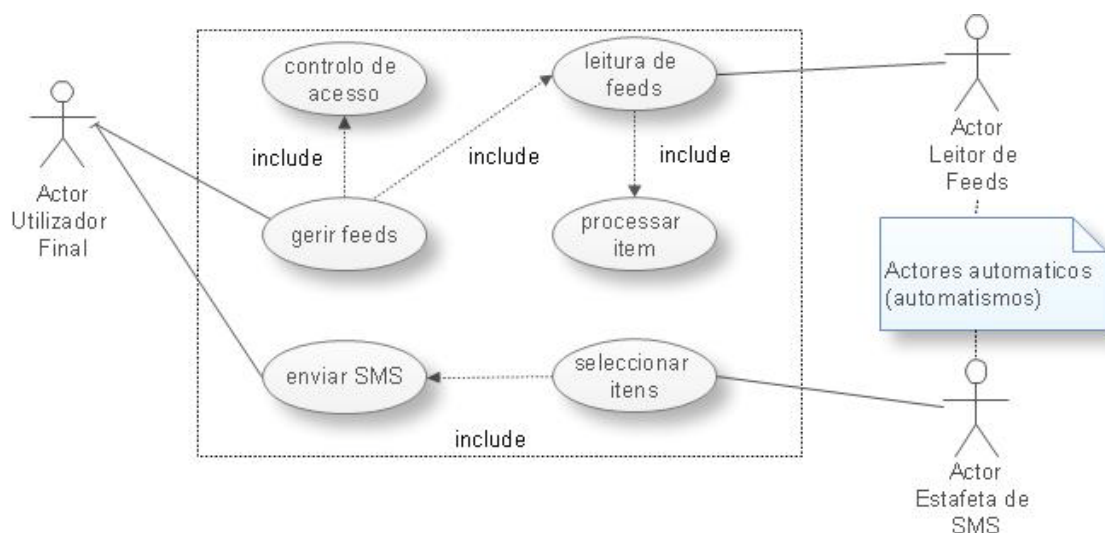


Fig. 3-2 Diagrama de Use Case do Sistema de Notificações de feeds RSS para SMS

Identificados os actores do sistema, e respectivos *Use Cases* associados, partimos para uma descrição mais pormenorizada sobre as acções incluídas em cada um destes.

3.2.1.4 Descrição Estruturada dos Use Cases

As seguintes descrições estruturadas aprofundam os principais *Use Cases* do sistema.

Tabela 3-2 Adicionar uma feed ao perfil de utilizador

<i>Use Case</i>	<i>Gerir Feeds -Adicionar uma feed</i>
Pré-condição	O utilizador está autenticado no sistema.
Descrição	<ol style="list-style-type: none"> 1. O <i>Use Case</i> inicia quando o utilizador introduz o URL da <i>feed</i> a adicionar. 2. Através da URL o sistema verifica se o mesmo já existe no sistema (introduzido por outro utilizador) <ol style="list-style-type: none"> a. Caso já exista, o sistema avança para o ponto 6. b. Caso não exista, o sistema avança para o ponto 3. 3. O sistema tenta obter o conteúdo do URL. <ol style="list-style-type: none"> a. Se o tempo da captura esgotar (<i>timeout</i>) sem que seja capturada o conteúdo do URL, o sistema avisa o utilizador, volta ao passo 1. 4. O sistema tenta efectuar o <i>parsing</i> do XML, através da especificação RSS ou Atom. <ol style="list-style-type: none"> a. Se não conseguir efectuar o <i>parsing</i>, avisa o utilizador e volta ao passo 1. 5. O sistema regista a <i>feed</i>. 6. O sistema associa a <i>feed</i> ao utilizador. 7. O sistema regista todos os itens da <i>feed</i>. 8. O sistema assinala os itens já registados para esta <i>feed</i> como não disponíveis para entrega para este utilizador. 9. O sistema informa o utilizador o sucesso da operação.
Pós-condição	A <i>feed</i> é adicionada ao perfil pessoal do utilizador

Tabela 3-3 Ler informação das feeds

Use Case	Leitura de <i>feeds</i>
Pré-condição	Existir <i>feeds</i> para serem lidas.
Descrição	<ol style="list-style-type: none"> 1. O sistema regista o início da operação. 2. O sistema obtém a lista de <i>feeds</i> subscritas pelos utilizadores. 3. O sistema tenta obter uma das <i>feeds</i> da lista. <ol style="list-style-type: none"> a. Se o tempo da captura esgotar (<i>timeout</i>) sem que seja capturada o conteúdo do URL, o sistema regista o insucesso avança para o ponto 7. 4. O sistema tenta efectuar o <i>parsing</i> do XML, através da especificação RSS ou Atom. <ol style="list-style-type: none"> a. Se não conseguir efectuar o <i>parsing</i>, o sistema regista o insucesso e avança para o ponto 7. 5. O sistema regista os itens da <i>feed</i> que ainda não existem no sistema usando o identificador GUID da <i>feed</i> (unívoco). O sistema regista os <i>nodes</i> "Title", "Description", "Guide" e "PubDate" do item da <i>feed</i>. 6. O sistema associa a <i>feed</i> ao utilizador. 7. O sistema verifica se existe mais <i>feeds</i> na lista do passo 2. <ol style="list-style-type: none"> a. Se existir mais <i>feeds</i> para ler, volta ao passo 4. b. Se não existir mais <i>feeds</i> para ler, o sistema avança para o ponto 8. 8. O sistema regista o fecho da operação.
Pós-condição	Os itens novos das <i>feeds</i> são registados.

Tabela 3-4 Enviar SMS aos utilizadores com os itens novos das feeds subscritas

Pré-condição	Existir itens novos por enviar.
Descrição	<ol style="list-style-type: none"> 1. O sistema regista o início da operação. 2. O sistema obtém a lista de itens a entregar para cada utilizador. 3. O sistema tenta obter um dos itens a enviar da lista. 4. O sistema produz um URL unívoco para o item (para que o utilizador possa depois seguir a hiperligação no SMS para obter informação completa). 5. O sistema verifica o número de caracteres do URL produzido no ponto anterior e subtrai a 160 (caracteres possíveis num SMS), obtendo os caracteres disponíveis. 6. O sistema obtém uma mensagem com o número de caracteres calculados no ponto anterior. 7. O sistema comunica com o serviço de envio de SMS, enviado o destinatário e mensagem e aguarda o resultado. <ol style="list-style-type: none"> a. Se o resultado for positivo, o sistema regista o envio com sucesso e avança para o ponto 8. b. Se o resultado for negativo, ou esgotar o tempo (<i>timeout</i>), o sistema regista o insucesso avança para o ponto 8. 8. O sistema verifica se existe mais itens na lista do passo 2. <ol style="list-style-type: none"> a. Se existir mais itens para enviar, volta ao passo 3. b. Se não existir mais itens para enviar, o sistema avança para o ponto 9. 9. O sistema regista o fecho da operação.
Pós-condição	Os itens novos das <i>feeds</i> são registados.

3.2.2 Desenho do Diagrama de classes

Um diagrama de classes identifica os objectos do sistema e a forma de como estes interagem entre si (O'DOCHERTY, 2005). O processo de análise, com o objectivo de produzir o diagrama de classes, consiste na identificação das classes que descrevem os objectos relevantes para o sistema, descrevendo as suas propriedades e relações (por associação, agregação, composição ou herança) com outras classes do sistema. A relação é ainda caracterizada quanto à sua multiplicidade, indicando quantos objectos participam na mesma.

Importa, antes de identificar as classes e objectos do sistema em questão, perceber o significado de classe e de objecto: entende-se por objecto como uma entidade ou conceito, caracterizado por um conjunto de propriedades (que indicam um estado), comportamentos (que indicam as operações que pode descrever) e identidade (distinção relativamente aos restantes objectos); uma classe representa uma abstracção de um conjunto de objectos (NUNES, et al., 2004).

O *uses cases*, verificados nos pontos anteriores, permitiram identificar as seguintes classes, identificadas na Tabela 3-5.

Tabela 3-5 Quadro de classes e atributos do sistema

Classe	Atributos	Observações
Feed	FeedId , Title, Link, Description	Regista dados sobre as <i>feeds</i>
Item	GuidId , Title, TitleAbbrev, Description, DescriptionAbbrev, Link, Data	Regista dados sobre os itens das <i>feeds</i>
User	UserId , UserLogin, UserPwd, Phone-Number, TimeReceiveMin, TimeReceiveMax	Regista dados sobre os utilizadores
Language	LangId , Language	Línguas
Abbreviation	AbbrevId , Word, Abbrev	Lista de abreviaturas por idioma. Composição de Language.
Status	StatusId , Status	Estados (aberto, fechado, concluído...)
History	CHistoryId , Date, TimeTaken, Orientation	Histórico de pedidos externos ao sistema (contacto externos) para fins estatísticos / de controle
FeedHistory	FHistoryId , Date, TimeTaken, ItemCaptured	Histórico de capturas a <i>feeds</i> , para fins estatísticos / de controle
UserFeed	CreationDate, AppendRedirect, UseAbbrev	Composição entre <i>feed</i> e user (<i>feeds</i> subscritas pelo user)
UserFeedFilter	Filter	Composição de <i>UserFeed</i>
ItemSentUser	Date, TimeTaken	Associação entre <i>Feed</i> , <i>Item</i> , <i>Status</i> , <i>History</i>

A nomenclatura de todas as classes e atributos será construída, no diagrama de classes e para os restantes diagramas, usando a convenção *lower camel case*, que teve a sua origem na linguagem Smalltalk.

Consiste, em termos práticos, em retirar o espaço que separa as palavras e capitalizar a primeira letra de cada palavra, à exceção da primeira (PORTER, 2008). Esta notação permite uma melhor identificação das palavras em oposição em escrever tudo em maiúsculas ou minúsculas.

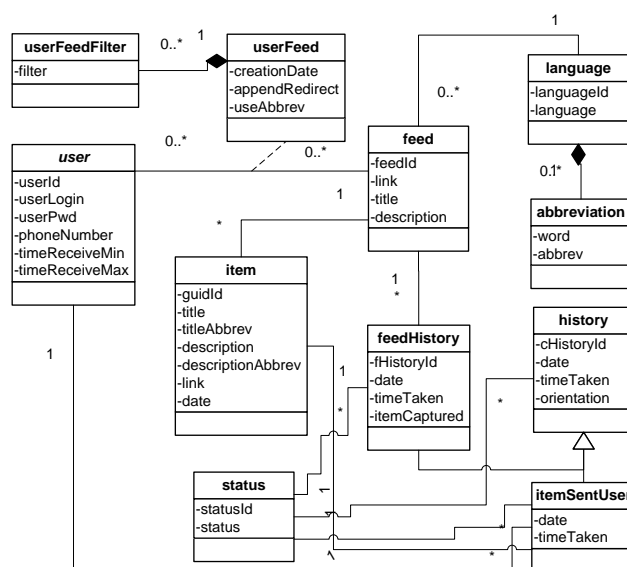


Fig. 3-3 Diagrama de Classes

Identificadas as classes do sistema, atributos das classes e as relações entre classes podemos iniciar a transposição para o modelo relacional.

3.2.3 Transposição para o Modelo Relacional

O modelo relacional baseia-se no princípio que a informação é representada através de tabelas (ou relação), contendo dados agrupados/ relacionados pela natureza do seu domínio (CODD, 1970). As tabelas são constituídas por atributos (ou colunas), cuja função é armazenar dados através de tuplos, isto é, linhas de uma tabela. Todas as tabelas devem possuir uma chave primária, isto é, um ou mais atributos que em conjunto identificam univocamente cada tuplo numa tabela. Quando não for possível obter uma chave primária através dos atributos disponíveis deve ser criado um atributo para o efeito, por exemplo com um número incremental. A relação entre tabelas é conseguido através de chaves estrangeiras, isto é atributos numa tabela que são chaves primárias em outra tabela (relação).

O modelo relacional é actualmente utilizado na maioria dos SGBD (Sistemas de Gestão de Bases de Dados) e será adoptado no desenvolvimento deste projecto.

3.2.3.1 Regras de transposição para o modelo relacional

A transposição do diagrama de classes deve obedecer a um conjunto de regras, de forma a garantir que não existam perdas no processo de transição (NUNES, et al., 2004):

- As tabelas apenas derivam das classes do sistema e das associações de multiplicidade "muitos para muitos". Todas as tabelas devem ter uma chave primária.
- Nas relações de multiplicidade "um para um" entre duas classes, transita a chave da classe principal para a outra (criando assim uma chave estrangeira).
- Nas relações de multiplicidade "um para muitos" entre duas classes, a classe da parte "muitos" (em que a informação repete) deverá receber a chave da outra classe.
- Nas relações de multiplicidade "muitos para muitos" entre duas classes dão origem a uma terceira tabela, constituída pelas chaves primárias de ambas as classes (podendo a terceira tabela ter atributos próprios decorrentes da relação).
- A transposição da generalização decorre da identidade das subclasses. Se tiverem identidade própria, cada subclasse tem o seu próprio identificador. Se a subclasse só tem identidade quando associadas à classe, herdam a chave da classe principal. Em ambos os casos, na classe principal deverá ser criado um atributo que identifica a que subclasse pertence.
- A transposição de agregações usa a mesma regra das multiplicidade.
- A transposição de composições, a tabela que equivale à classe da composição herda a chave da outra classe, fazendo parte da sua chave primária.

Aos pontos anteriores, no que diz respeito às chaves primárias, podem ser adicionados algumas optimizações, com vista à melhor performance da SGBD, como a criação de um única chave primária.

Esclarecidas as regras, podemos proceder à transposição do diagrama de classes para o modelo relacional.

3.2.3.2 Transposição para o Modelo Relacional

A análise no diagrama de classes, mediante as regras descritas no ponto anterior permitiu efectuar a transposição patente na Tabela 3-6.

Tabela 3-6 Transposição para o Modelo Relacional

Tabela	Atributos
<i>Feed</i>	kFeedId , fLangId, fCategoryId, aLink, aTitle, aDescription
language	kLangId , aLanguage
abbreviation	kAbbrevId , kLangId , aWord, aAbbrev
Item	kGuidId , <i>kFeedId</i> , aTitle, aDescription, aLink, aDate, aTitleAbbrev, aDescriptionAbbrev
User	kUserId , aUserLogin, aUserPwd, aPhoneNumber, aTimeReceiveMin, aTimeReceiveMax
<i>userFeed</i>	kUserId , kFeedId , aCreationDate, aAppendRedirect, aUseAbbrev
<i>userFeedFilter</i>	kUserId , <i>kFeedId</i> , kFilter
Status	kStatusId , aStatus
History	kCHistoryId , aDate, aTimeTaken, aOrientation { <i>feedHistory</i> , <i>itemSentUser</i> }
<i>feedHistory</i>	kFHistoryId , fCHistoryId, aDate, aTimeTaken, fStatusId, aItemCaptured
<i>itemSentUser</i>	kUserId , <i>kFeedId</i> , kGuidId , KstatusId , aDate, fCHistoryId, aTimeTaken

A esta descrição do modelo relacional, descrita na Tabela 3-6, no desenho do diagrama do modelo relacional, visível na Fig. 3-4, indicamos também:

- os tipos de dados para cada atributo.
- na transposição, os nomes dos atributos de cada classe foram precedidos com um dígito: "k" (*key*) para uma melhor identificação dos atributos que são chave primária da tabela; "f" (*foreign key*) quando constituem chaves estrangeiras (e não são primárias na própria tabela); "a" (*attribute*) para os restantes atributos.

As actividades que descrevem acções são identificadas s no diagrama através de um identificador inscrito num rectângulo com cantos arredondados.

Os diamantes (de decisão), indicam um comportamento condicional baseado numa condição (mediante a análise de uma expressão booleana). Os diamantes podem descrever pontos de divergência ou de convergência.

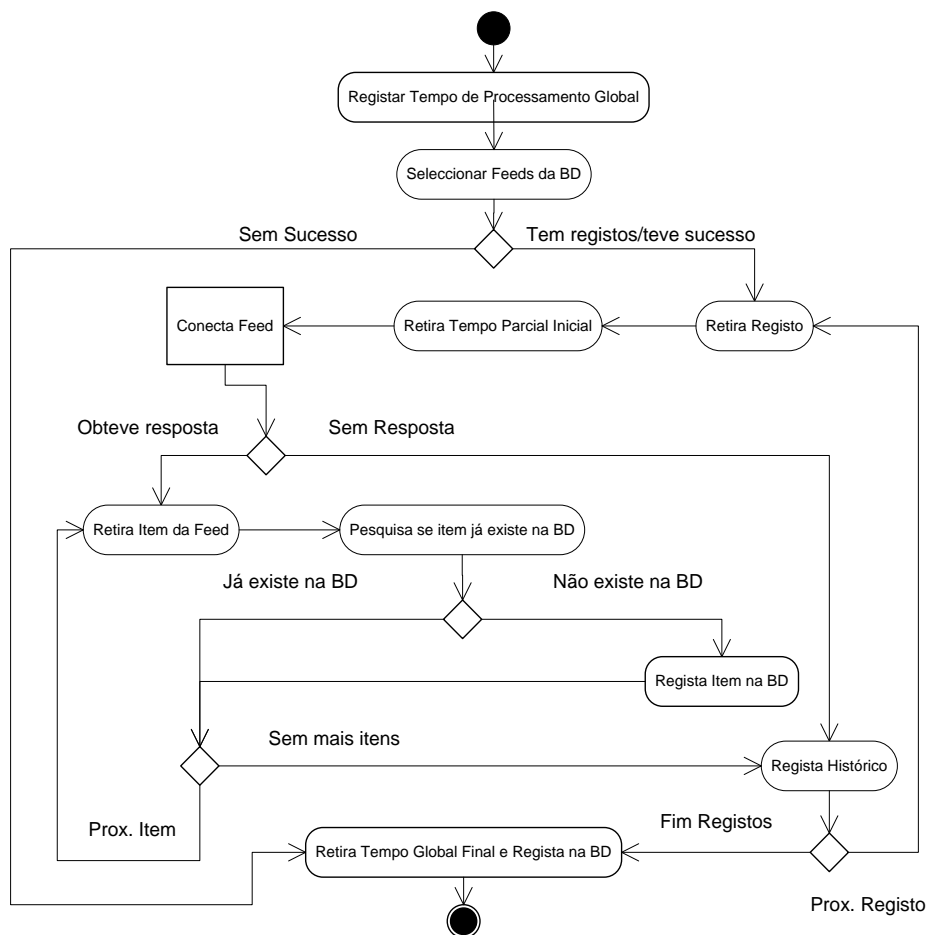


Fig. 3-5 Diagrama de Actividades: leitura de informação de feeds

O diagrama de actividade da Fig. 3-5 descreve os procedimentos que o sistema deverá operar com vista a capturar informação de uma lista de *feeds*, previamente subscritas pelos utilizadores do sistema.

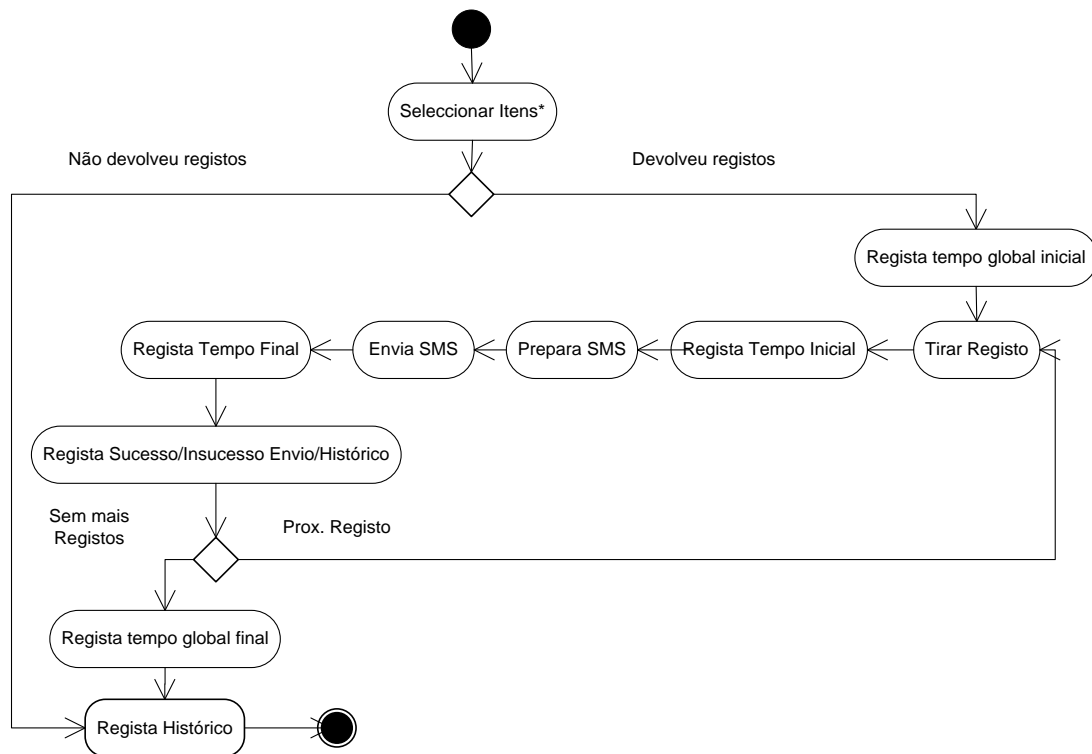


Fig. 3-6 Diagrama de Actividades: envio de informação das *feeds* por SMS

Na Fig. 3-6, podemos observar o fluxo de acções que o sistema opera no envio de informação actualizada das *feeds*, lidas previamente (descrito na actividade da Fig. 3-5). Na acção "Seleccionar Itens", os itens (entradas de informação contida nas *feeds*) serão seleccionados mediante as seguintes condições:

- Serem recentes (nas últimas 48H).
- Não terem sido enviados anteriormente.
- São segmentados por utilizador, mediante a sua subscrição na *feed*.

3.2.5 Diagramas Físicos

Os diagramas físicos servem para descrever as características físicas (arquitectura física) de um sistema (NUNES, et al., 2004). A UML disponibiliza dois tipos de diagramas para este efeito: o diagrama de componentes e o diagrama de instalação (ou distribuição).

Os diagramas de componentes permitem descrever a relação entre os diversos componentes do sistema. Um componente é uma unidade modular ou parte física do sistema, substituível, que proporciona a realização de um conjunto de interfaces (FARIA,

2001). A Fig. 3-7 descreve fisicamente a aplicação *Web* (concretizada através de páginas HTML, com hiperligações entre si), demonstrando as relações entre os diversos componentes da aplicação.

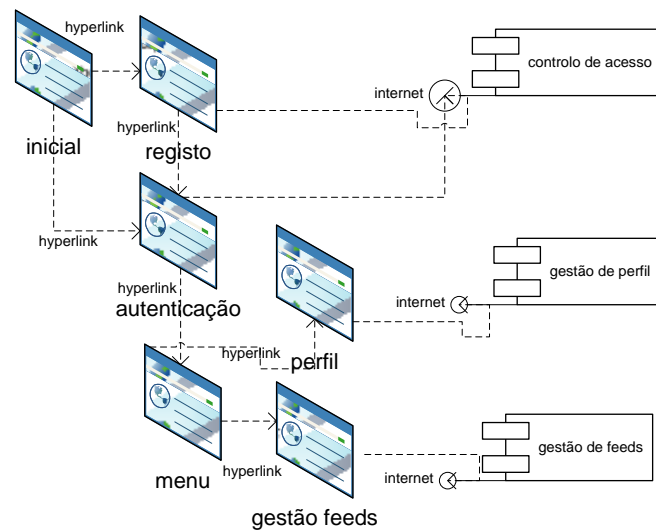


Fig. 3-7 Diagrama de componentes para a aplicação *Web*

O diagrama de instalação documenta a relação física entre componentes de *Hardware* e *Software* no sistema. Na Fig. 3-8 apresenta-se o diagrama físico de instalação do sistema de informação, combinado com o diagrama de componentes, descrevendo os recursos (e a relação entre estes) onde os diversos componentes estarão localizados.

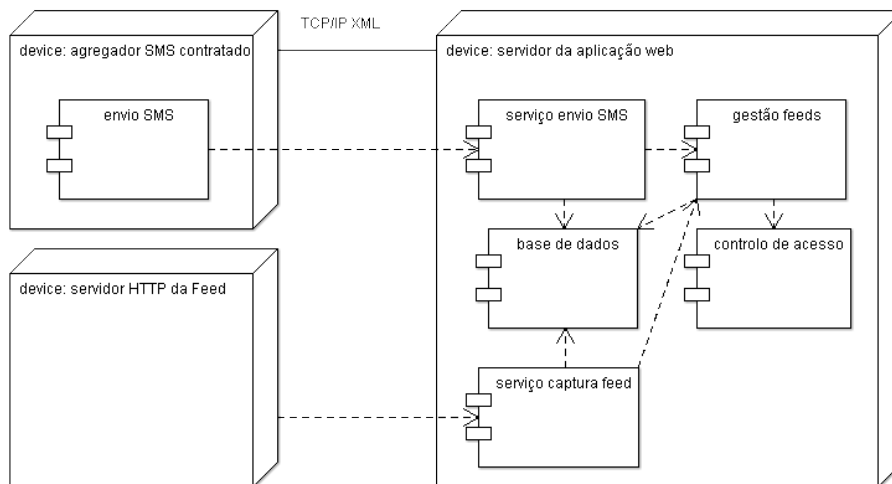


Fig. 3-8 Diagrama de instalação do sistema

Esta forma de combinação de diagramas físicos fornecem uma visão mais abrangente do sistema (NUNES, et al., 2004).

3.2.6 Composição do *layout* para o sistema

O *Web design* (desenho para a *Web*) deve ser pensado do ponto de vista de desenho de conteúdos, isto é, deve consistir na boa arrumação de conteúdos por forma a ser correctamente interpretado pelo utilizador. "O princípio número um do *design* para a *Web* é a simplicidade" (MCGOVERN, et al., 2002). O *design* deve ser fácil e intuitivo.

No estudo para o desenho de um *layout* de uma interface *Web* (distribuição física numa página *Web* de elementos físicos como texto, imagem, hiperligações, etc.) podemos usar ferramentas como *wireframes* (esboços ou rascunhos de uma página *Web*) e *storyboards* (sequência de *wireframes*) (MEMÓRIA, 2006).

3.2.6.1 Esboço

A construção de *wireframes* decorre da análise prévia produzida no arranque do projecto. Na sua concepção devem ser seguidas os seguintes pontos-chave: partir de *wireframes* com pouco detalhe; procurar o equilíbrio e encaixe lógico; procurar uma organização estética; estruturar a navegação; organizar a informação por hierarquias bem definidas.

Para além destas ideias-chave anteriores, o *layout* de uma página *Web* deve seguir as convenções a que os utilizadores já estão habituados.

"Muitas pessoas olham para a *Web* instintivamente como um único meio. Gostam de utilizar noutros sites os conhecimentos de navegação que adquiriram num determinado site. Neste sentido, quanto mais semelhante for o seu sistema de navegação em relação a outros sites, mais fácil será para o leitor navegar no seu site, baseando-se na respectiva experiência" (MCGOVERN, et al., 2002).

A Tabela 3-7, adaptado de (MEMÓRIA, 2006), sintetiza alguns das convenções existentes nos sites, seguindo a análise de investigadores renomeados no estudo da usabilidade em *Websites*.

Tabela 3-7 Padrão de Posicionamento de elementos no *layout* de uma interface gráfica de um site

Elemento	Posicionamento	Investigador
Identificação/Logotipo	Canto superior esquerdo	Nielsen, Adkisson e Bernard
Pesquisa	Parte superior	Nielsen, Adkisson e Bernard
Navegação principal	Parte superior com links na horizontal	Nielsen, Adkisson e Krug
Navegação secundária	Esquerda	Nielsen, Adkisson e Bernard
Navegação de rodapé	Parte inferior	Nielsen, Krug, Lynch e Horton
Conteúdo	Centro	Bernard

Para além da disposição dos elementos descritos anteriormente, o desenho do *layout* deve seguir um conjunto de orientações:

- Obter a compatibilidade máxima com os mais diversos dispositivos (ter cuidado com as tecnologias utilizadas e também o espaço ocupado em termos de ecrã).
- Apenas usar *scroll* (elevadores ou barras de deslocação) verticais, evitando ter páginas com mais de 3 ecrãs em altura. Os conteúdos mais importantes devem estar no topo.
- Estruturar o *layout* em colunas.
- Identificar bem as páginas e usar menus persistentes.
- Utilizar tipos de letra sem serifa, isto é, usar apenas fontes com figuras geométricas simples, sem "rendilhados", uma vez que são mais fáceis de ler.

O estudo dos *Use Cases*, levantamento de requisitos e diagramas do modelo físico permitiu-nos efectuar o levantamento das seguintes áreas que o sistema deverá apresentar na interface gráfica:

- Identificação, através da informação do nome do sistema e o seu objectivo.
- Menu de navegação, com as opções principais que remetem para cada funcionalidade.
- Zona de conteúdo, apresentando o conteúdo da opção seleccionada.
- Zona para ferramentas e acessórios do sistema, para inclusão de eventuais funcionalidades acessórias.

- Rodapé, com hiperligações para conteúdos de contexto geral do sistema.

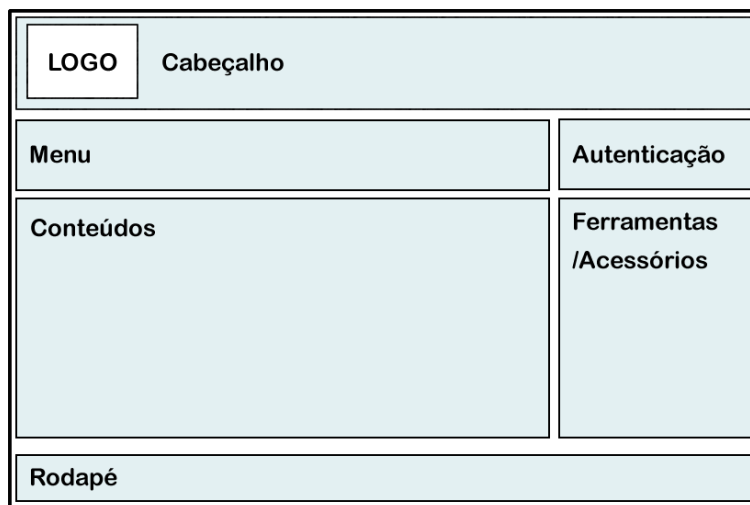


Fig. 3-9 Wireframe da interface gráfica do sistema

Após termos identificado a disposição da informação no ecrã, podemos partir para a concepção do *layout* em si. Esta tarefa foi segmentada em duas partes: o desenho da logomarca e o desenho gráfico do *layout*.

3.2.6.2 Logomarca

O desenho da logomarca (termo largamente confundido com logotipo, que se baseia apenas em tipos), baseou-se no símbolo tradicional das RSS, criado pela Mozilla Foundation e adoptado pela Microsoft, que representa a difusão de informação através de um ponto emissor.



Fig. 3-10 Feed icon - ícone representativo de uma *feed*

O ícone desenvolvido, uma aproximação/proposta para o sistema, baseia-se numa inversão do ícone tradicional da *feed* RSS (visível na figura 3-10), numa tentativa de associar e basear o conceito do sistema ao conceito das *feeds* RSS.



Fig. 3-11 Logomarca para o sistema

O ícone, visível na figura 3-11, representa a agregação de conteúdos num único ponto. As cores foram mantidas relativamente ao ícone do RSS. Na composição da logomarca foram acrescentados os tipos "RSS2SMS", uma composição onde o 2 representa a palavra em inglês "to" (para), formando a expressão "RSS para SMS". A utilização do dígito 2 para representar a transposição ou transformação de informação de um formato para outro é já utilizada em muitas aplicações disponíveis no mercado.

3.2.6.3 *Layout gráfico*

A produção do *layout* gráfico segue as orientações produzidas nos *wireframes*. Na sua produção optou-se por colocar um fundo opaco, em contraste com a zona central do sistema, com o objectivo de a realçar, desviando a atenção do utilizador para a zona que interessa realmente. Para a escolha do fundo optou-se pela cor complementar da logomarca (em laranja), o azul. As cores complementares ajudam a equilibrar e contrastar a composição gráfica. Optou-se pela colocação de um *degradé* (sequência entre tons) vertical entre azuis de tons semelhantes, onde o azul mais claro assume a responsabilidade de realçar a identidade do sistema.

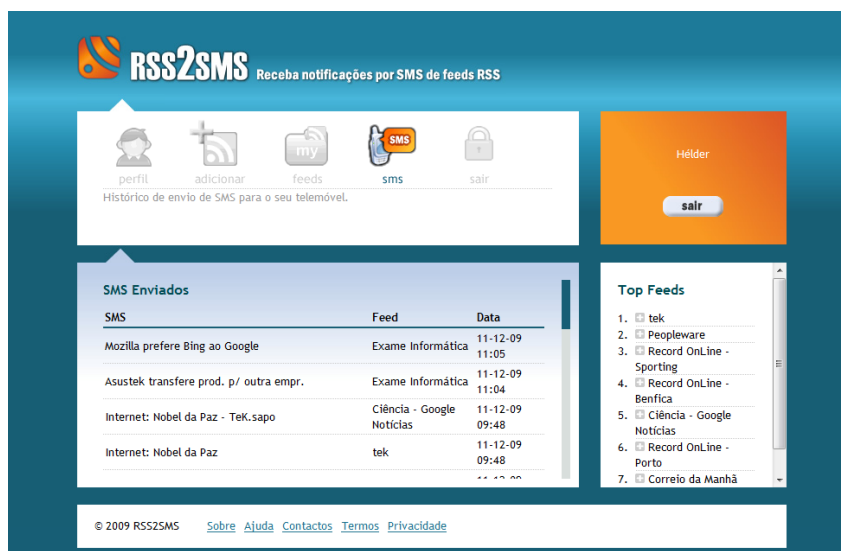


Fig. 3-12 *Layout* gráfico desenvolvido para o sistema

As hiperligações dos menus foram concebidas com recurso a ícones com as respectivas legendas, realçados através da selecção ou *roll-over* (acto de posicionar o ponteiro do rato em cima do botão).

Segundo os dados da W3Counter (W3Counter, 2009), 91.4% dos utilizadores possuem ecrã com resolução igual ou superior a 1024 por 768 pixéis, razão pela qual o *layout* da aplicação *Web* será desenvolvido para essa resolução. No entanto serão adoptadas medidas/técnicas que permitam a acessibilidade dos utilizadores que estiverem a usar dispositivos móveis, com ecrãs menos generosos, possam igualmente poder aceder à aplicação *Web* sem limitações.

4 Desenvolvimento do Sistema

Neste capítulo pretende-se descrever as fases de desenvolvimento do sistema proposto, após a análise do problema realizado no capítulo anterior. O capítulo encontra-se estruturado em duas partes:

- a primeira parte (secção 4.1) aborda as tecnologias envolvidas/adoptadas para o desenvolvimento do sistema;
- a segunda parte (secção 4.2) descreve-se o sistema desenvolvido.

O sistema foi desenvolvido ao longo de 8 meses, sendo constituído por cerca de 3000 linhas de código (PHP, HTML, CSS e Javascript) num total de 6300 linhas com a integração de bibliotecas *open-source*.

4.1 Tecnologias adoptadas no desenvolvimento do sistema

Para um maior esclarecimento sobre as diferentes tecnologias adoptadas no desenvolvimento do sistema proposto, é necessário entender as camadas envolvidas na arquitectura.

Na Fig. 4-1 podemos observar as camadas envolvidas na arquitectura de uma típica aplicação *Web* (WILLIAMS, et al., 2004), mostrado a interacção entre o cliente (*browser*), servidor *Web* e o servidor do SGBD. Este diagrama resume de uma forma geral todas as partes envolvidas no sistema a desenvolver à excepção da camada que representa a comunicação com o *gateway*, que permitirá o envio das SMS que não se encontra retratada no mesmo.

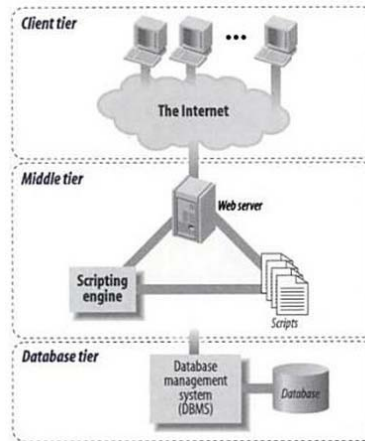


Fig. 4-1 Diferentes camadas na arquitectura de uma aplicação Web

Pretende-se nas subsecções seguintes descrever as tecnologias envolvidas, no desenvolvimento do sistema, em cada uma destas camadas.

4.1.1 Tecnologias envolvidas no lado do cliente

A camada do lado do cliente é constituída pelo *Software* a que o utilizador recorre para navegar na *world wide Web*, isto é, o *browser*. Este *Software*, na maior parte das vezes disponibilizado gratuitamente, tem como função enviar pedidos a um servidor *Web*, interpretar e mostrar no ecrã os resultados devolvidos pelo servidor. Existem no mercado vários *browsers* que o utilizador pode usar, para os mais diversos sistemas operativos e dispositivos, entre os quais (por ordem de popularidade) o Internet Explorer (com mais de 50% do mercado por razões óbvias, uma vez que vem pré-instalado com o sistema operativo Windows), o Firefox, Safari, Opera, Chrome, etc (W3Counter, 2009).

Uma das principais funções de um *browser*, que remota a criação do Nexus (BERNERS-LEE, 2009), é a possibilidade de poder interpretar documentos (devolvidos pelo servidor *Web*) com a especificação HTML (*hyper text markup language*) e renderizar o resultado no ecrã (para que o utilizador o possa visualizar) (ORGERA, 2009).

4.1.1.1 O HTML - *Hyper Text Markup Language*

O HTML é a linguagem universal que todos os *browsers* conseguem interpretar com o objectivo de mostrar texto, gráficos e outros elementos ao utilizador, mas também possibilitar as hiperligações ou até recolher informação do utilizador (W3C, 2004).

A especificação HTML é constituída por um conjunto de *tags* (anotações, também conhecidas por etiquetas HTML) que têm como função descrever o documento e a sua informação. É através destas anotações HTML que o *browser* saberá como mostrar a informação ao utilizador. Por exemplo:

```
<p>Isto é um parágrafo e <b>isto está a negrito</b>.</p>
```

Desde a sua criação, o HTML sofreu várias revisões e encontra-se actualmente na versão 4 ou XHTML 1.0 (*Extensible Hyper Text Markup Language*), sendo esta última uma reformulação da versão 4 para poder ser interpretada também como um documento XML (*Extensible Markup Language*).

É incontornável a utilização desta tecnologia no desenvolvimento do sistema, tendo em conta que o sistema será desenvolvido para uso em paradigma *Web*. No processo de edição do HTML serão usadas as orientações da W3C (através de uma iniciativa intitulada WAI - *Web Accessibility Initiative*) com vista a tornar as páginas acessíveis para pessoas com dificuldades de acesso (W3C, 2008). A WAI define 3 níveis de conformidade, "A", "AA" e "AAA" sendo o "A" o nível de conformidade mais baixo. Podemos resumir essas orientações nos seguintes pontos:

- Fornecer conteúdos alternativos (em texto) para elementos como imagens, animações, som e vídeo.
- Estruturar o documento de uma forma consistente usando títulos, listas.
- Usar sempre que possível *Cascade Style Sheets* (descritas no ponto seguinte) para a concepção do *layout* e estilo da página, em detrimento do uso de tabelas para a sua concepção.
- Providenciar legendas e transcrições em texto de vídeos e sons.
- Tornar as hiperligações mais fáceis de interpretar, evitando fazer hiperligações em expressões como por exemplo "clique aqui".
- Validar o trabalho desenvolvido através dos motores de validação.

Estas orientações foram adoptadas pelas autoridades portuguesas, com a publicação da Resolução do Conselho de Ministros n.º 155/2007 (GOVERNO, 2007), onde indica que os sites de Internet do Governo devem estar em conformidade com as orientações para a acessibilidade da W3C, devendo respeitar a norma "A", e que entre estes os sites que ofereçam serviços transaccionais devem estar em conformidade com a norma "AA".

A validação destas orientações, no protótipo a desenvolver, será conseguinte mediante a utilização do motor de validação da UMIC - Agência para a Sociedade do Conhecimento, disponível em linha no site "<http://www.aceso.unic.pt>", que para além da sua análise agrega também resultados de motores de validação da própria W3C.

4.1.1.2 As CSS - *Cascade Style Sheets*

As CSS (*Cascading Style Sheets*), também conhecidas por folhas de estilo, foram criadas em 1996 pela *World Wide Web Consortium*. Em poucas palavras são um conjunto de regras de formatação que controlam a aparência de uma ou mais páginas HTML (e também XML).

O exemplo seguinte mostra uma regra CSS para a formatação de parágrafos:

```
<style type="text/css">
p {
    color: red; font-size: 15px;
}
</style>
```

Até então os documentos HTML continham no seu interior, para além das *tags* que descreviam o conteúdo, *tags* que unicamente descreviam a forma de como esse conteúdo surgia no ecrã (formatação). O principal objectivo das CSS é separar o conteúdo do documento da formatação do documento, tornando os documentos mais leves, flexíveis, universais (uma vez que é possível especificar diferentes estilos para diversos *outputs* da página, como a disponibilização no ecrã, *browsers* para invisuais, leitura em dispositivos móveis, impressão, etc.), fáceis de interpretar e até poupar tempo no desenvolvimento/actualização dos documentos HTML (W3C, 1998).

As CSS sofreram alterações desde a primeira publicação, encontrando-se actualmente na versão 2 (estando a ser ultimada a publicação oficial da versão 3, já suportada por muitos *browsers*). Todos os *browsers* suportam o uso de CSS muito embora alguns, dependendo da sua versão, não possuam suporte para versões mais recentes de CSS (UGHETTO, 2006).

As CSS vêm deste modo complementar o uso do HTML, tornando-se assim também uma das tecnologias incontornáveis no desenvolvimento deste projecto, sendo usadas para esse objectivo na formatação da interface gráfica do sistema.

O *layout* do protótipo, com vista a seguir as orientações da W3C com vista à acessibilidade das páginas, será construído exclusivamente através de formatações com recur-

so a CSS, usando a técnica *tableless*. A técnica *tableless* consiste na montagem de *layouts* com recursos a posicionamentos e formatações avançadas usando CSS, normalmente usando elementos DIV (divisores), em detrimento da técnica tradicional de tabelas aninhadas (utilização de tabelas dentro de outras tabelas para conseguir a divisão do espaço no ecrã). Hazaël (HAZAËL-MASSIEUX, 2005) defenque o uso das tabelas deve ser só restringido à apresentação de dados e não para fins de montagem de *layouts*.

4.1.1.3 Javascript - dinamismo do lado do cliente

O código que constitui um documento HTML devolve sempre o mesmo resultado no *browser*. Isto é, o documento HTML não se altera dinamicamente em função de algum tipo de variável, mantendo-se inalterado até que volte a ser modificado pelo editor. Deste modo, não permite uma grande interactividade com o utilizador, para além das habituais hiper-ligações. Podemos considerar o HTML como um documento estático.

Para colmatar esta debilidade do HTML, existem linguagens de programação dinâmicas, como o ECMAScript (conhecido universalmente como Javascript, termo que usaremos também a partir de agora) ou o VBScript (suportado apenas por *browsers* da Microsoft). Através destas linguagens podemos criar programas que podem ser processados e executados pelo próprio *browser*, permitindo outras formas de interacção do utilizador com a própria página, como reagir a determinadas interacções do utilizador com elementos da página (eventos). Por exemplo, preencher um campo de um formulário, modificar o conteúdo da página sem a recarregar, etc (W3C, 2009). A junção das duas linguagens, o HTML e uma linguagem dinâmica do lado do cliente, ficou conhecida inicialmente por (*Dynamic HTML*) e mais recentemente por AJAX (*Asynchronous Javascript and XML*).

O código Javascript é embebido no código HTML através de uma *tag* própria, conforme podemos observar no exemplo seguinte:

```
<p> Escrever dinamicamente na página uma contagem de 1 a 100:<br/>
<script type="text/javascript">
for (i=0; i<101; i++) {
  document.write(i + "<br/>");
}
</script>
</p>
```

A utilização neste projecto de uma linguagem dinâmica do lado do cliente torna-se necessária uma vez que nos permitirá:

- Validar o preenchimento correcto dos formulários antes de a informação ser enviada ao servidor, evitando que a página tenha que ser recarregada, poupando-se assim tempo e tráfego desnecessário para o servidor *Web*.
- Obter informações sobre o *browser* e o sistema operativo do utilizador, permitindo adequar a visualização em função dessas características.
- Reagir a eventos (interacção do utilizador com a página),
- Mostrar alertas ao utilizador.

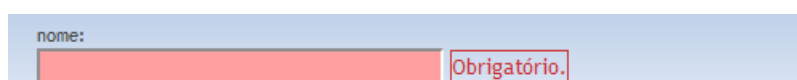


Fig. 4-2 Exemplo de validação de um campo de formulário, com recurso a Javascript

A escolha do Javascript como linguagem de programação a ser executada no lado do cliente neste projecto deve-se à sua universalidade. Qualquer *browser* consegue interpretar e executar Javascript nativamente, sem recurso a instalação de *plugins*, ao contrário de outras linguagens. Segundo os dados de (COUNTER, 2009) a presença de Javascript activo nos *browsers* era de 93%, o que revela um factor de (relativa) segurança na sua utilização.

4.1.2 Tecnologias envolvidas no lado do servidor: linguagem dinâmica

A tecnologia HTML, descrita no ponto anterior, permite apenas descrever e formatar um documento. A sua principal lacuna é que o seu conteúdo, só por si, não se altera ao longo do tempo, mantendo-se inalterado até intervenção/nova publicação por parte do editor responsável (SERRÃO, et al., 2001). Por outras palavras, o HTML é um documento estático, que não se actualiza dinamicamente. Por outro lado, o Javascript também é limitado na sua actuação, não podendo aceder por exemplo a uma base de dados. Se no passado estas características eram largamente satisfatórias para o desenvolvimento de um *site Web* comum, hoje em dia a exigência aumentou, na medida em que os utilizadores exigem que os *sites* se mantenham actualizados ao segundo.

As linguagens dinâmicas do lado do servidor, isto é, linguagens que são interpretadas directamente pelo próprio servidor *Web* (antes do resultado ser enviada ao *browser*), vêm colmatar essa lacuna e permitir a que praticamente quase tudo seja possível com

recurso às mesmas, potenciando desta forma uma interacção mais profunda com o utilizador. Existe inclusive a possibilidade de gerar páginas HTML instantaneamente, que são devolvidas ao *browser*, podendo mostrar, por exemplo, informação actualizada/registada de uma base de dados instalada localmente ou até remotamente, entre outras funcionalidades possíveis.

Na Fig. 4-3 podemos ver e comparar o funcionamento das páginas estáticas e as páginas dinâmicas.

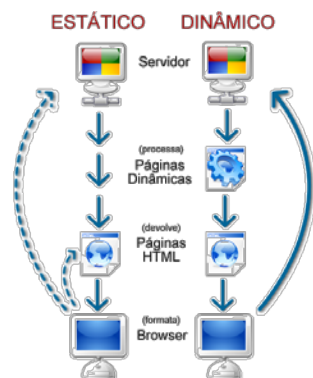


Fig. 4-3 Diferenças de processamento de páginas dinâmicas e páginas estáticas

Em termos práticos, as páginas dinâmicas incluem no seu interior um conjunto de instruções (muitas vezes misturadas no próprio código HTML) que serão previamente executadas pelo servidor *Web* sempre que existir um pedido. O resultado dessas instruções é depois enviado pelo servidor ao *browser*, que o irá interpretar e mostrar ao utilizador.

Tabela 4-1 Quadro comparativo entre a utilização de linguagens dinâmicas e linguagens estáticas

	Página Estática	Página Dinâmica
Vantagens	<ul style="list-style-type: none"> - A sua concepção é mais rápida, pelo que o seu custo será menor; - Não necessita de um serviço de alojamento muito caro; - É ideal para sites que não necessitem de alterações de conteúdos frequentes. - A sua devolução por parte do servidor é mais rápida, uma vez que este não a processa. 	<ul style="list-style-type: none"> - Permite um maior grau de interactividade com os utilizadores; - Possibilita a criação de mecanismos automatizados para receber informação de diversas fontes, tais como, bases de dados, outros sites, etc.; - Ideal para sites que necessitem de alterações de conteúdos frequentes (exemplos: portais, sites informativos, lojas on-line, aplicações on-line diversas, entre outros); - Possibilidade de inserir (agendar) pre-

		<p>viamente as informações/conteúdos que se pretendam publicar e de programar o momento exacto da sua publicação.</p> <p>- Permite criar mecanismos de actualizações dinâmicas do site por terceiros, sem que estes tenham que possuir qualquer conhecimento técnico.</p>
Desvantagens	<ul style="list-style-type: none"> - Não permite grande interactividade com os utilizadores, para além da típica hiperligação; - É necessário recorrer à codificação em HTML para alterar os conteúdos pretendidos num determinado momento; - É extremamente trabalhoso e pouco eficiente proceder a alterações frequentes de conteúdos, uma vez que implica editar fisicamente o ficheiro e envia-lo posteriormente para o servidor. - Sempre que pretenda alterar conteúdos terá alguma dependência dos serviços quem lhe desenvolveu o <i>Web</i> site; 	<ul style="list-style-type: none"> - Tipicamente necessita de um alojamento com um maior custo associado, pois o alojamento terá que permitir a execução de determinadas linguagens de programação utilizadas ou por exemplo a utilização de bases de dados; - Exige normalmente um tempo de desenvolvimento mais alargado, o que implica um maior custo associado;

Para o desenvolvimento do sistema proposto, iremos necessitar de linguagens dinâmicas, na medida em que iremos necessitar de aceder a bases de dados, para recolher/registar informação do utilizador e das *feeds*. Existem diversas linguagens dinâmicas, processadas do lado servidor, tais como as ASP (*Active Server Pages*) da Microsoft, JSP (*Java Server Pages*) da Oracle, PHP (*Hypertext Preprocessor*), C#, Pearl, etc.

A tecnologia adoptada para o desenvolvimento do sistema de informação proposto será o PHP, que iremos ver com maior detalhe, justificando a sua escolha.

4.1.2.1 PHP - *Hypertext Preprocessor*

A tecnologia PHP, que surgiu em 1994 como um projecto pessoal de Rasmus Lerdorf com o intuito de controlar acessos à sua página *Web*, é fortemente baseada nas linguagens C, Java e Perl (MEDEIROS, et al., 2002). É usada por aproximadamente 50% dos 21 milhões de sites no mundo inteiro (NIEDERAUER, 2007) (PHP-Stats, 2007).

Em 2008, a Nexen publicou estatísticas mostrando que o PHP era utilizado em cerca de 35% dos sites *Web*, acima do ASP na ordem dos 20% (SEGUY, 2008).

Segundo a Tiobe, conforme podemos observar na Fig. 4-4, o PHP assumia em Dezembro de 2009 como a terceira linguagem de programação mais popular com 29% (TIOBE, 2009).

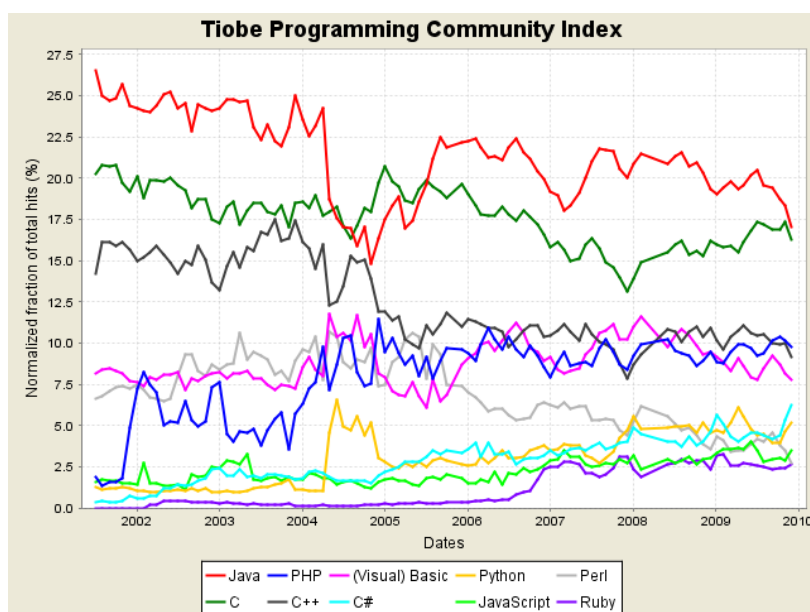


Fig. 4-4 Linguagens de programação mais populares

Segundo o seu site oficial (PHP, 2009), o PHP permite fazer tudo. É, no entanto, maioritariamente vocacionado para o processamento do lado do servidor, permitindo também alternativamente a concepção de aplicações de linha de comandos e até aplicações tradicionais de desktop, sendo suportado pelos principais sistemas operativos e servidores *Web*. Entre a maior parte dos servidores *Web* que suportam o PHP encontra-se o Apache, o servidor mais utilizado na *World Wide Web* segundo as estatísticas da Netcraft (NETCRAFT, 2009).

A escolha do PHP como linguagem de desenvolvimento do lado do servidor para este projecto deve-se aos seguintes factores:

- Ser de acesso gratuito e *open-source*.
- Popularidade, permitindo uma maior confiança e facilidade de acesso a documentação técnica.
- Disponibilidade da linguagem nas empresas de *hosting* (alojamento *Web*).
- Flexibilidade, na medida em que pode ser usado com os principais servidores *Web*, como o Apache, IIS (Internet Information Server), Lighthttpd, etc, bem como um vasto leque de SGBD.

- Possibilidade de ser executado através de linha de comandos, uma vez que, deste modo, permite a criação de serviços que executam automaticamente em períodos determinados (*cron job*).

O PHP oferece suporte a uma longa lista de sistemas de gestão de bases de dados (SGBD), tal como MySQL, SQL Server, Oracle, Sybase, etc, bem como possui suporte para comunicar usando o ODBC (*Open Database Connection Standard*), e consequentemente com todos os SGBD por este suportados.

A utilização de PHP consiste em colocar as suas instruções numa etiqueta própria que é identificada pelo servidor, que processa as páginas PHP. O código PHP pode estar misturado com código HTML entre outros.

4.1.3 A SGBD MySQL - Sistema de Gestão de Bases de Dados

Podemos definir uma base de dados, de uma forma genérica, como um conjunto de dados armazenados num suporte (papel, agenda, livro, lista), que podem ser recuperados. No entanto, quando aplicamos a expressão base de dados referimo-nos a sistemas de gestão de bases de dados (SGBD), os programas informáticos que gerem a interacção de uma ou mais base de dados.

O sistema proposto nesta tese necessita de uma SGBD para registar e manipular, de forma geral:

- Dados sobre o perfil de cada utilizador.
- Dados sobre as *feeds* (gerais e de cada item capturado).
- Registar o envio de SMS.
- Registar informações estatísticas sobre os tempos de envio e captura de informação para posterior tratamento.

Existem disponíveis no mercado vários SGBD, comerciais e gratuitos, tais como Oracle, SQL Server, MySQL, PostgreSQL, Firebird, dBase, SQLite, etc. A SGBD escolhida para o desenvolvimento foi o MySQL, pelos seguintes factores:

- Ser de acesso gratuito e *open-source*.
- Facilidade de integração com a linguagem dinâmica processada do lado do servidor escolhida para o projecto (PHP).
- Fiabilidade e performance.

- Popularidade, permitindo a facilidade de acesso a documentação técnica, bem como possibilitar factor de confiança/qualidade na sua utilização.
- Disponibilidade do servidor junto das empresas de *hosting*.

Um estudo publicado na eWeek por Timothy Dyck, em 2002, demonstrou que o MySQL se encontrava a par com o conceituado SGBD Oracle (ligeiramente com melhor performance) na devolução de páginas *Web* por segundo, conforme se pode observar no gráfico da Fig. 4-5 (DYCK, 2002).

"De acordo com os resultados obtidos pode-se concluir que o MySQL é um SGBD veloz ideal para aplicações onde velocidade seja algo mais crucial do que segurança" (DELFINO, et al., 2006). Um outro estudo relevou que, comparativamente ao PostgreSQL o MySQL apresentou melhores resultados de desempenho (o *benchmark* realizado avaliou os módulos estrutura e carga, mono-usuário e multi-usuário), onde o PostgreSQL revelou-se mais forte apenas no campo de carga e estrutura (PIRES, et al., 2005).

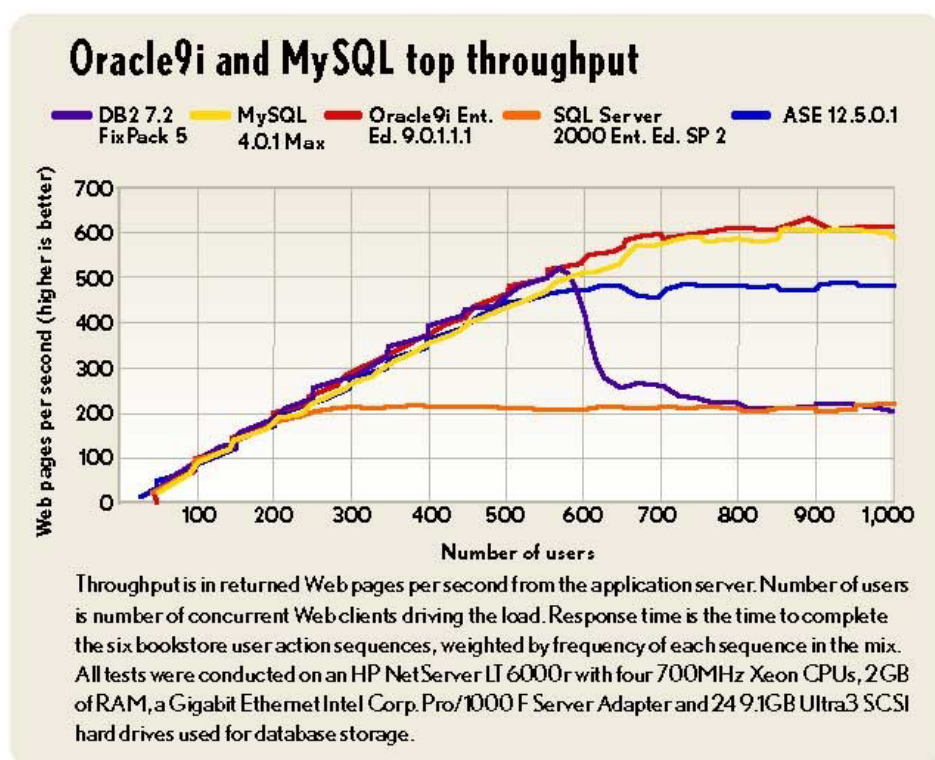


Fig. 4-5 Comparação de performance entre SGBD na devolução de páginas *Web* por segundo

4.1.4 O serviço de envio de SMS

O último elemento necessário para o sistema para atingir os objectivos propostos, consiste no serviço que possibilita o envio de SMS. Como foi descrito no Capítulo 2 desta tese, é possível enviar SMS através de uma aplicação *Web* através de uma das seguintes formas:

- Conectar ao SMSC da operadora em questão e requisitar o envio de SMS.
- Conectar a uma gateway de SMS, usando a API disponibilizada pelo fornecedor, que posteriormente contacta os SMSC das diversas operadoras para solicitar o envio da SMS.
- Usar um terminal (ou mais) acoplado a um PC para enviar SMS.

A solução utilizada na resolução do problema, foi o recurso de uma gateway de uma operadora de voIP, a voipcheap. A escolha desta gateway, em detrimento de outras gateways ou formas anteriormente especificadas ficou a dever-se à avaliação de custo/benefício nesta fase de desenvolvimento de um protótipo, em função das seguintes variáveis:

- O custo por SMS / custo de carregamento mínimo de *plafond* (uma vez que os custos são todos suportados pelo autor). As gateways profissionais exigem a compra de um número mínimo de SMS (por exemplo 50.000).
- À simplicidade de configuração/envio do pedido de envio de SMS, uma vez que para o desenvolvimento do protótipo não existe uma grande necessidade de performance no envio de (grandes quantidades de) SMS. Pretende-se testar e comprovar o funcionamento / viabilidade na resolução do problema proposto nesta tese junto de uma pequeno universos de utilizadores.

A gateway do Voipcheap funciona através de um pedido do tipo GET através do URL:

```
https://www.voipcheap.com/myaccount/sendsms.php?username=variavel1&password=variavel2&from=variavel3&to=variavel4&text=variavel5
```

No exemplo anterior as variáveis 1,2,3,4 e 5 são substituídas pela mesma ordem por:

- Nome do utilizador com contrato na operadora.
- Palavra passe do utilizador.
- Número de telefone do emissor.

- Número do telefone do destinatário.
- Texto da SMS.

O pedido do tipo GET é despoletado através de instruções PHP, que envia os dados finais após manipulação da base de dados.

4.2 Descrição do sistema

O sistema foi desenvolvido durante o período de 8 meses, tendo sido escritas durante esse período aproximadamente 3000 linhas de código PHP, HTML, CSS e Javascript proprietário (desenvolvido inteiramente pelo autor). Na totalidade o sistema possui cerca de 6300 linhas, uma vez que foram adoptadas algumas bibliotecas *open-source* com o objectivo de acelerar o desenvolvimento.

Podemos segmentar (embora existam tarefas desenvolvidas paralelamente) o processo de desenvolvimento nas seguintes fases:

- preparação do *Hardware e Software* base de suporte ao sistema;
- edição do *layout* usando HTML e CSS; preparação da base de dados;
- desenvolvimento da aplicação *Web* usando linguagens dinâmicas no lado do cliente e no lado do servidor.

4.2.1 A arquitectura da aplicação *Web*

A aplicação *Web*, que proporciona um interface do sistema ao utilizador, é constituída através da agregação de várias tecnologias sobre um único "tecto" (mistura).

Esta agregação é conseguida através de um ficheiro central, o "index.php", desenvolvido com recurso a PHP, que tem como função agregar todas as tecnologias envolvidas, através da inclusão de ficheiros externos ou através de etiquetas (*tags*) que especificam a linguagem a ser utilizada. No exemplo seguinte podemos observar um extracto do ficheiro "index.php", mostrando a junção de *tags* HTML com *tags* PHP.

```
<div class="central">
  <div class="esq">
    <?php
      include("inc_menu.php"); //incluir módulo do menu
    ?>
  </div>
  <?php
    include("inc_autenticacao.php"); //incluir módulo de autenticação
  ?>
</div>
```

Todo o processamento da aplicação *Web* é conseguido através de envio de pedidos do *browser* ao ficheiro "index.php". Este ficheiro agrega as seguintes tecnologias:

- HTML - *Hyper text markup language*
- CSS - *Cascade style sheets*
- Javascript
- PHP - *Hypertext preprocessor*

Na Fig. 4-6 podemos observar a forma de como todas estas tecnologias são articuladas, através do ficheiro "index.php", através da inclusão de outros ficheiros, numa lógica modular ou de camadas, com tecnologias e funções específicas na aplicação.

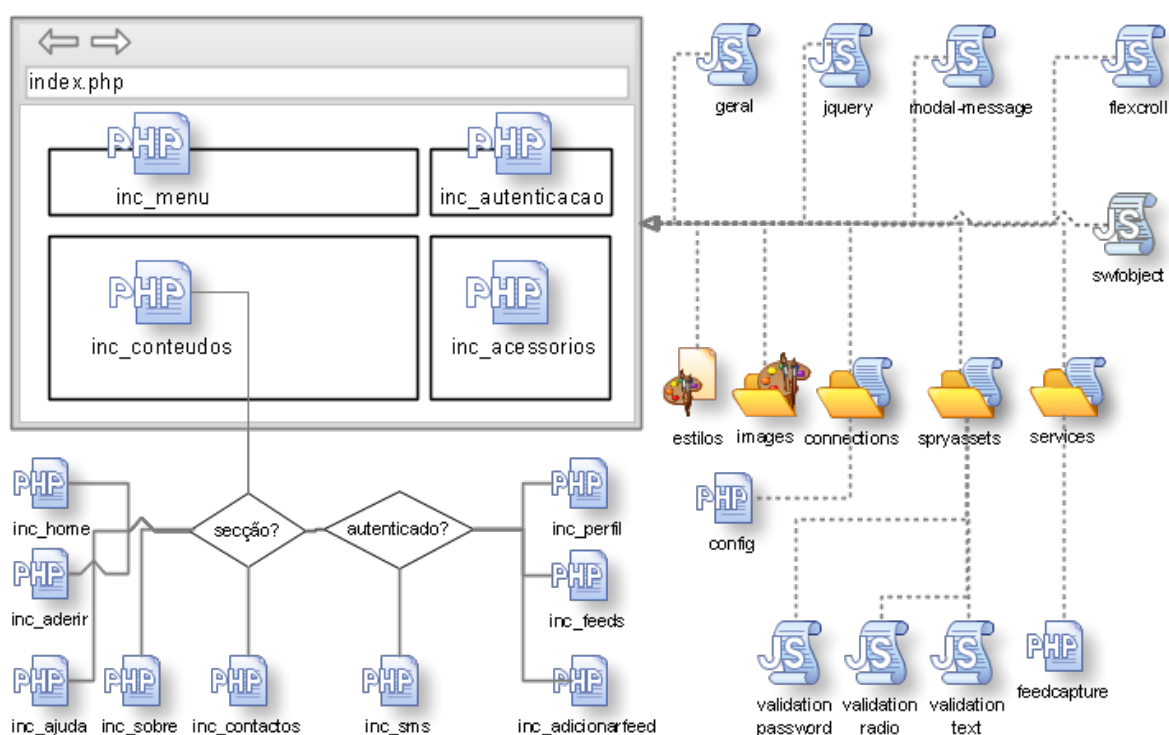


Fig. 4-6 Arquitectura da aplicação *Web*, de interface ao utilizador

Nos pontos seguintes iremos focar, com maior detalhe, as diversas camadas da aplicação *Web* desenvolvida.

4.2.2 Configuração e instalação do *Hardware* e *Software* base

Para a disponibilização do sistema, para que possa ser devidamente testado, foi necessário alojar a solução final numa máquina dedicada. Neste subcapítulo iremos descrever as características do *Hardware* e *Software* de base utilizados, bem como os pormenores de configuração adoptados.

4.2.2.1 Hardware

A lista de *Hardware* consistiu nos seguintes equipamentos, normalmente para fins domésticos, disponíveis no universo do autor:

- O PC utilizado para alojar o protótipo consiste numa máquina com processador *Intel Pentium* 1.60 GHz com 591 MHz, com 512 MB de RAM.
- *Router* ASUS WL500W.
- *Modem* digital Thomson, modelo THG540.
- UPS (*uninterruptible power supply*) da Mustek, modelo *PowerMust Office* 650.

Sob o propósito de aferir a *performance* da solução do *hardware* utilizado, foram efectuados testes de desempenho em termos de velocidade de processamento e devolução da informação. Os testes consistiram em efectuar pedidos simultâneos (100 e 1000 pedidos simultâneos) ao servidor *web* (do sistema desenvolvido), registando-se o tempo decorrido entre os pedidos ao servidor e recepção da resposta por parte deste.

Adicionalmente, para poder-mos ter dados que nos permitam comparar a *performance* do sistema com outros sistemas *web*, efectuamos os mesmos testes ao *site* do ISCTE e ao motor de busca Google.pt.

A Tabela 4-2 e a Fig. 4-7 sintetizam os resultados obtidos durante os testes.

Tabela 4-2 Resultados dos testes de pedidos e devolução de páginas *web*

	RSS2SMS		Google.pt		ISCTE	
Pedidos Simultâneos	100	1000	100	1000	100	1000
Tempo decorrido (em segundos)	6.5	88.7	5.7	159.6	3.9	32
Pedidos Falhados	0	5	0	0	0	0
Tempo por Pedido (milisegundos)	65.051	88.77	57.72	159.6	39.88	24.78
Taxa de Transferência (Kbytes/seg)	81.19	59.4	119.84	43.2	12.63	20.3

Através dos resultados obtidos, consideramos que a solução de equipamentos utilizados (ainda que de uso doméstico) são suficientes para proceder a um pequeno teste de utilização do sistema junto de uma pequena comunidade de utilizadores (até 100 utilizadores simultâneos), mas insuficientes para serem utilizados num ambiente de produção, uma vez que não podemos prever quantos outros pedidos, para além dos nossos, por exemplo o google.pt estaria a processar no mesmo momento em que os testes foram executados.

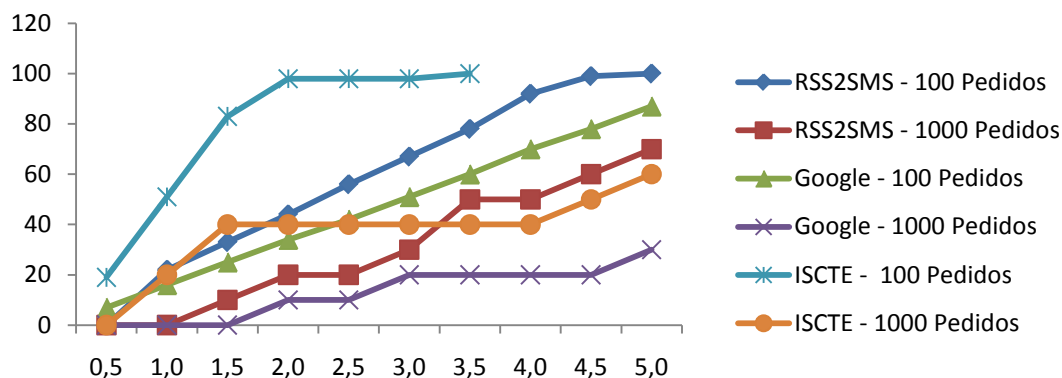


Fig. 4-7 Devolução de páginas *web* por segundo

O sistema, com a solução de *hardware* descrita anteriormente, conseguiu devolver 100 pedidos em 6.5 segundos, um resultado muito próximo de um tempo de resultado óptimo de 5 segundos numa transacção *web* (MONTEIRO, et al., 2000).

A conectividade à Internet baseou-se numa solução doméstica proporcionada *pelo Internet Service Provider Zon - Tv Cabo*, com 1 Mb de largura de banda em *upload*. A fim de confirmar a veracidade da velocidade, foram efectuadas medições. Segundo medição efectuada pelo Speedtest.net, realizadas em 19-12-2009, as velocidades de comunicação registadas nesta ligação foram:

- 0.92 *Mega bit* por segundo, de velocidade de *upload*.
- 20.70 *Mega bit* por segundo, de velocidade de *download*.
- 20 milissegundos de *ping* (comando que recorre à analogia do jogo *ping-pong* para testar a conectividade entre equipamentos).

Foi necessário configurar o router a fim de poder-se reencaminhar os pedidos ao *port* 80 (porta convencionada para a comunicação TCP/UDP para o serviço HTTP), e ao *port* 21, para o PC onde esta alojado o servidor HTTP e FTP (*File Transfer Protocol*). Para atingir estes objectivos foram efectuadas as seguintes configurações no router:

- Associação no router Thomson de um endereço IP (*Internet Protocol*) ao MAC (*Media Access Control*) Address da placa de rede do PC. O objectivo consiste em que a máquina obtenha sempre o mesmo IP na rede interna.
- Foram configuradas no router alguns mapeamentos estáticos aos *port* 80 e 21, para permitir o acesso externo de indivíduos conectados à Internet aos serviços disponibilizados (HTTP e FTP) no PC ligado à rede de interna. Cada entrada consiste na especificação do *port* disponível ao público e na especificação do

endereço IP (*Internet protocol*) da rede privada e o respectivo *port*, para onde se pretende reencaminhar os pedidos vindos do exterior.

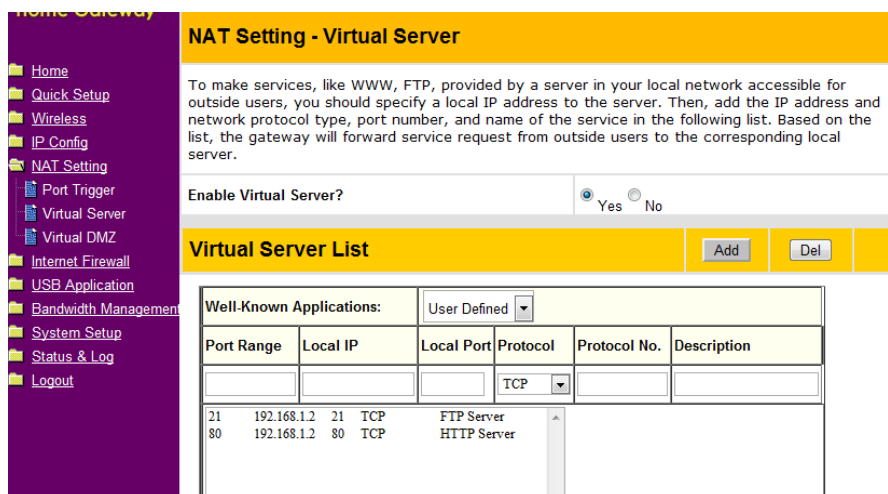


Fig. 4-8 Configuração de mapeamentos estáticos, para o reencaminhamento de tráfego nas portas 80 e 21 do exterior para o interior

O passo seguinte consistiu em obter um URL (*Uniform Resource Location*) com o objectivo de ser associado ao endereço IP atribuído pelo ISP (visível e acessível a partir do exterior), a fim de tornar a experiência mais amigável para o utilizador.

Uma vez que o serviço de ligação à Internet utilizada atribui endereços IP dinâmicos, isto é, o endereço IP modifica-se automaticamente a cada nova ligação, ou ao fim de algum tempo, recorreremos a um serviço gratuito de DNS dinâmico, o DynDNS, que permite usar ligações típicas de "*dialup*" como se de um recurso IP próprio se tratasse. Este tipo de serviço oferece mecanismos para actualizar automaticamente o IP e de o associar a um URL. A actualização do IP pode ser conseguida através de uma das seguintes formas:

- Instalação de um *Software* cliente (gratuito) do DynDNS na máquina, que comunica à DynDNS cada vez que o IP altera.
- Configuração de um serviço existente no router (opção existente em alguns *routers*).
- Actualizar manualmente o IP no site do DynDNS.

Neste caso concreto, o router utilizado tem integrado um serviço de actualização para o DynDNS e como tal foi esta a opção utilizada. A configuração consiste em introduzir as credenciais fornecidas pela DynDNS.

O URL escolhido foi r2s.ath.cx. Esta (aparentemente estranha) escolha consistiu apenas na criação de uma sigla da expressão "RSS 2 SMS", associado ao domínio, fornecido pela DynDNS, que tivesse menos caracteres de extensão. A finalidade era poupar o máximo de caracteres no endereço URL final, com o objectivo de este ser poder ser incluído nas futuras SMS.

Em termos práticos, a partir de agora, o utilizador final apenas decora o URL e não o (longo, dinâmico e difícil de decorar) endereço IP.

4.2.2.2 *Software*

O *Software* de base utilizado para o alojamento do sistema consiste na seguinte lista:

- servidor de HTTP, para disponibilização do serviço *Web*;
- servidor SGBD MySQL.
- um servidor de FTP, para acesso remoto com a finalidade de publicação e actualização de ficheiros da *source* do sistema de informação desenvolvido.

O sistema operativo utilizado para o teste e alojamento do protótipo consiste no *Windows XP Professional*, com *Service Pack 3*.

Para a instalação destes três componentes foi usada uma distribuição pré-configurada e gratuita, o XAMPP. O XAMPP, na versão *Windows*, é constituído pelo seguinte *Software* (entre outros):

- Apache, o servidor de HTTP escolhido.
- MySQL, a SGBD escolhida para o sistema.
- FileZilla FTP Server, para poder-mos efectuar envios e descargas de ficheiros para/do servidor.
- PHP, a linguagem de programação escolhida para o desenvolvimento *server side*.
- phpMyAdmin, aplicação *Web* de gestão do SGBD MySQL.

A instalação é efectuada mediante um processo gráfico muito simplificado. Na configuração geral, visível na Fig. 4-9, indicamos que o Apache, MySQL e FileZilla devem correr como um processo do sistema, que deve iniciar-se automaticamente no arranque do sistema. Adicionalmente foram efectuadas algumas afinações às configurações do PHP, através da edição do seu ficheiro de configuração "php.ini", nomeadamente na activação das seguintes extensões (*plugins*):

- Extensão cURL, que permite comunicação através de diversos protocolos como FTP, FTPS, HTTP, HTTPS, Telnet, etc.
- Extensão OpenSSL, para permitir comunicações seguras sobre SSL (*Secure Sockets Layer*), necessária para a comunicação HTTPS com a gateway.

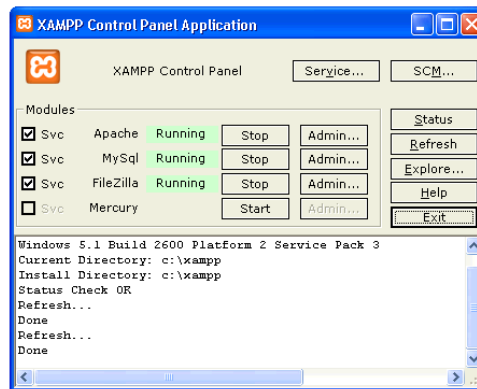


Fig. 4-9 Painel de controle do XAMPP

Após o arranque de todos os serviços, configurados os utilizadores e respectivas permissões para os servidores de SGBD e FTP, foi confirmado se todos os serviços estavam disponíveis/visíveis a partir do exterior, através do *ping* ao URL configurado e também da consulta do endereço http numa janela de um *browser*.

```
C:\Users\Helder>ping r2s.ath.cx
Disparando r2s.ath.cx [81.84.200.23] com 32 bytes de dados:
Resposta de 81.84.200.23: bytes=32 tempo=75ms TTL=64
Resposta de 81.84.200.23: bytes=32 tempo=2ms TTL=64
Resposta de 81.84.200.23: bytes=32 tempo=1ms TTL=64
Resposta de 81.84.200.23: bytes=32 tempo=9ms TTL=64

Estatísticas do Ping para 81.84.200.23:
    Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de perda),
Aproximar um número redondo de vezes em milissegundos:
    Mínimo = 1ms, Máximo = 75ms, Média = 21ms
```

Fig. 4-10 Ping efectuado ao domínio r2s.ath.cx

Após estas fases de configuração, o servidor (máquina física) está pronto para receber o sistema.

4.2.3 Edição da estrutura base da interface *Web*, através de HTML e CSS

A aplicação *Web*, do ponto de vista da aplicação cliente *Web* (o *browser*), não é mais do que de uma página *Web* construída com recurso à especificação HTML, auxiliada da especificação CSS (para a formatação gráfica). A interacção mais profunda com o utilizador, é conseguida através de instruções que são processadas no lado no servidor e não no lado do cliente. De facto, do lado do cliente ou até para o utilizador final, a aplicação *Web* não se distingue de outro vulgar site: é constituída, tal como qualquer

outro site, de texto, inclusão de imagens, formulários, hiperligações, etc. Constitui apenas a interface, a parte visível para a aplicação *Web*. Mas este facto não poderá ser uma causa de descrédito, ou de menor importância, uma vez que depende directamente da interface, e da sua boa concepção, a forma com que o utilizador interage com maior ou menor facilidade com a aplicação (grau de usabilidade).

A componente HTML, de interface da aplicação *Web*, foi desenvolvido com recurso à técnica *tableless* (ausência de tabelas como elemento primordial na construção de um *layout* de uma página *Web* - técnica tradicional), que delega todas as funções da especificação gráfica do *layout* na especificação de regras CSS avançadas, separando assim o conteúdo da formatação. Na prática consiste na criação de regras CSS que alteram o comportamento tradicional de uma *tag*, como por exemplo um parágrafo, definindo a sua posição no ecrã, largura, altura, entre outros atributos.

No exemplo seguinte podemos visualizar um cabeçalho normal, com o título da aplicação, de nível (importância) 1 especificado através de *tags* HTML (neste caso o "`<h1>`"). Este possui uma hiperligação para o ficheiro "index.php".

```
<h1><a href="index.php">RSS2SMS</a></h1>
```

O referido código HTML produziria, sem recurso a formatações CSS, o seguinte resultado no *browser* visível na Fig. 4-11.

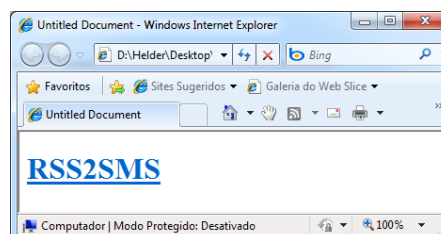


Fig. 4-11 Exemplo de cabeçalho HTML de nível 1, sem recurso a CSS

Este resultado, seria facilmente interpretado por um serviço de indexação de um motor de busca (conhecido por *crawler*) ou por *browsers* com fins específicos, como *browsers* para invisuais ou de dispositivos móveis (normalmente mais limitados), uma vez que o conteúdo é totalmente especificado através de texto e consequentemente totalmente acessível.

No entanto, o aspecto gráfico do resultado é pobre e pouco apelativo ao utilizador. Se quisermos melhorar, usando uma logomarca, tradicionalmente colocaríamos uma imagem em detrimento do texto. O resultado seria mais apelativo, mas mais inacessível.

vel para *browsers* com fins específicos, como os que existem em dispositivos móveis (cujo ecrã tem por vezes sérias limitações em termos de largura e altura) e outros *outputs*, uma vez que o texto foi substituído por uma *tag* que associa uma imagem (que não consegue ser descrita, sem recurso a legendas).

```
<h1><a href="index.php"></a></h1>
```

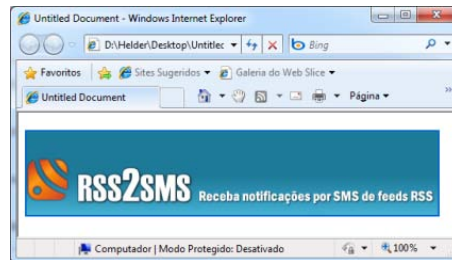


Fig. 4-12 Exemplo de cabeçalho com uma imagem, usando a técnica tradicional

Tradicionalmente, a solução passaria pela concepção de versões das páginas formatadas e orientadas aos *outputs* específicos, resultando numa redundância de dados (e de trabalho). Através de CSS conseguimos produzir regras que permitem alterar o funcionamento *standard* da *tag* para outro indicado, através da modificação dos seus atributos. No exemplo seguinte podemos visualizar três regras CSS, especificadas apenas para o tipo de média *screen*, que altera o funcionamento normal do cabeçalho de nível 1.

```
<style type="text/css" media="screen">
h1 {
    background-image: url(images/logo.jpg);    background-repeat: no-repeat;
    display: block; width: 205px; height: 100px; /*altera a caixa e atribui um fundo*/
}
h1 a {
    display: block; height: 100px; width: 205px; /* altera a caixa do link*/
}
h1 a span {
    display: none; /*oculta tudo o que estiver envolvido na Span, neste caso o texto */
}
</style>
(...)
<h1><a href="index.php"><span>RSS2SMS</span></a></h1>
```

Em termos práticos, o resultado no ecrã seria o mesmo que o apresentado na Fig. 4-12, mas de uma forma totalmente acessível para os indexadores dos motores de busca e outros tipos de *output* (tipos de *media*) tais como: impressão em papel, sintetizador de voz, dispositivo de *braille*, dispositivo móvel, entre outros.

A especificação da estrutura do *layout* para a aplicação *Web* proposta foi conseguida, como vimos, através de técnicas avançadas de CSS. Para atingir o objectivo final, a estrutura geral do *layout* foi dividida em partes, com recurso a uma *tag* HTML com essa função: a *tag* DIV (*division*).

```
<div id="frame">
  <div class="logo">Aqui fica a logomarca</div>
  <div class="separador"></div>
  <div class="central">
    <div class="esq">Aqui fica o menu</div>
    <div class="dir">Aqui fica a zona de autenticação</div>
  </div>
  <div class="separador2"></div>
  <div class="inferior">
    <div id="conteudos">Aqui ficam os conteúdos</div>
    <div class="acessorios">Aqui ficam os acessórios</div>
  </div>
  <div class="rodape">Aqui fica o conteúdo do rodapé</div>
</div>
```

Cada um destes divisores foi formatado com regras CSS que indicavam o seu posicionamento no ecrã, entre outras formatações.

O código HTML e CSS foi editado manualmente, linha a linha, com o objectivo de obter um total controle sobre o código produzido. A estrutura base do *layout*, ou seja o template geral do *layout* da interface *Web*, gerou cerca de 200 linhas de formatação CSS e cerca de 200 linhas de HTML.

4.2.4 Utilização de dinamismo no lado do cliente com Javascript

O Javascript, quando disponível no *browser* do utilizador final, permite produzir comportamentos dinâmicos directamente accionados e executados através do próprio *browser*, mediante a interacção com o utilizador. Entre inúmeras possibilidades, pode-se evitar, por exemplo, pedidos ou envios desnecessários de informação ao servidor, resultando assim numa poupança de recursos.

Normalmente a utilização de Javascript decorre do controle de eventos (normalmente situações em que o utilizador interage com elementos da própria página). Por outro lado, não nos podemos fiar totalmente com o Javascript, uma vez que a sua execução pode estar bloqueada no *browser* ou até nem ser suportada. Ainda assim, devemos recorrer sempre que possível ao Javascript para oferecer um maior dinamismo nas páginas, mas não descurando a produção de um plano alternativo, caso este não exista ou não esteja activo.

A utilização de Javascript no protótipo decorreu nas seguintes funcionalidades, por ordem de importância / pertinência.

- Validação de formulários.
- Usar a técnica Ajax (*Asynchronous Javascript And XML*) a evitar o carregamento desnecessário de informação redundante, quando necessário.
- Carregamento de filmes em formato *Shockwave Flash*.
- Mostrar avisos ao utilizador com grafismo mais apelativo.
- Formatar o elevador de deslocamento vertical, existente na zona central, para um grafismo mais apelativo e enquadrado na identidade gráfica produzida.

Para atingir este objectivo foram desenvolvidas rotinas em Javascript, bem como foram adoptadas algumas livrarias, classes e *scripts* de origem *open-source*, a fim de acelerar o processo de desenvolvimento. Foram adoptadas no desenvolvimento:

- jQuery, uma livraria de javascript que através da sua utilização permite escrever menos código (JQUERY, 2009).
- flexcroll, um *script* que permite modificar o aspecto gráfico dos elevadores (HESIDO, 2009).
- *Modal message*, uma classe em javascript que permite mostrar uma mensagem no ecrã sobreposta aos elementos da página (DHTMLGOODIES, 2009).
- swfobject, uma classe em javascript que permite embeber o *flash player* na página (SWFOBJECT, 2009).
- *spry framework*, uma livraria em javascript, usada neste caso para a validação de formulários (ADOBE LABS, 2009).

A utilização dos exemplos anteriores consiste na maior parte das vezes, na inclusão dos respectivos ficheiros fonte através da *tag HTML Script*, como podemos verificar na linha seguinte:

```
<script text="text/javascript" src="swfobject.js"></script>
```

Para além das rotinas anteriores, foi desenvolvida uma rotina com 85 linhas em Javascript com a função de enviar um pedido Ajax, com o objectivo de recarregar zonas específicas na página, sem que seja necessário um recarregamento total da mesma.

4.2.5 Criação da base de dados

Após no processo de análise e transposição para o modelo relacional, estudado na secção 3.2.3, procedeu-se à criação do *script* em SQL (*Structured Query Language*), recorrendo a instruções DDL (*Data Definition Language*) para a estrutura da base de dados. O exemplo seguinte mostra a instrução de criação da base de dados:

```
create database if not exists rss2sms;
```

As bases de dados em MySQL são implementadas através de um directório, contendo arquivos que correspondem as tabelas. Estes directórios são criadas dentro do directório "/xampp/mysql/data/". Neste caso passa a existir um subdirectório chamado "rss2sms". No exemplo seguinte podemos verificar um exemplo de criação de uma das tabelas que foi criada na base de dados. Para o desenvolvimento do protótipo foi adoptado o *storage engine* (mecanismo de armazenamento) por defeito do MySQL, o MyIsam, que está vocacionado para o alto desempenho (MYSQL, 2009).

```
create table status (  
  kStatusId char(1) not null,  
  aStatus varchar(20) default null,  
  primary key (kStatusId)  
) engine=MyISAM default charset=latin1 collate=latin1_general_ci;
```

Esta instrução gerou no directório da base de dados 3 arquivos, cada um com a sua função:

- "status.frm", contendo a definição da tabela.
- "status.MYD", que tem a função de guardar os dados.
- "status.MYI", onde se encontram registados os índices da tabela.

Após a criação da base de dados, e de todas as tabelas previstas, podemos avançar para o desenvolvimento da aplicação.

4.2.6 Dinamismo no lado do servidor

O desenvolvimento no lado do servidor, conforme anunciado anteriormente, foi efectuado com recurso à linguagem PHP. Este desenvolvimento foi segmentado em duas partes:

- a aplicação *Web* visível pelo utilizador final (secção 4.2.6.1);
- os processos e serviços que automatizam o processo de captura da informação contida nas *feeds* e o envio de SMS ao utilizador (secção 0).

4.2.6.1 A aplicação *Web* de interface ao utilizador

A aplicação *Web* desenvolvida, em termos físicos, é constituída por um conjunto de ficheiros interligados. Nesta secção descrevem-se os ficheiros que são interpretados do lado do servidor, neste caso os ficheiros PHP.

Dentro de cada ficheiro PHP é possível encontrar uma mistura de código HTML, CSS, Javascript e também PHP. Cada um destes códigos será processado no seu respectivo âmbito: o código PHP é processado primariamente pelo servidor; o resultado gerado após este processamento é enviado ao *browser* que interpreta e processa o resultado (as restantes linguagens).

Em termos gerais, podemos resumir as rotinas e funções desenvolvidas na aplicação *Web* do protótipo através dos seguintes pontos:

- Tratamento de variáveis do tipo *GET*, submetidas ou não através de um formulário, preenchido pelo utilizador.
- Tratamento de variáveis do tipo *POST*, enviadas pelo *browser* através da submissão de um formulário.
- Ligação e manipulação do SGBD MySQL (leitura, consulta ou escrita de dados) usando por vezes os dados recolhidos nos pontos anteriores.
- Controle de sessões, por forma a restringir o acesso a alguns módulos a utilizadores com sessão activa.

No desenvolvimento da aplicação *Web*, cuja finalidade é ser manipulada directamente pelo utilizador final, a estrutura de ficheiros foi architectada numa lógica modular: todo os pedidos passam por um ficheiro central, o ficheiro "index.php". Este ficheiro inclui, conforme o pedido, os módulos solicitados (distribuídos em diversos ficheiros PHP).

Todo o código PHP poderia ter sido criado dentro de apenas um único ficheiro, mas por uma questão de organização foi desenvolvido desta forma. Esta architectura é descrita visualmente na Fig. 4-6 na página 71.

A inclusão de cada módulo é efectuado através da instrução *include*, dentro de uma etiqueta PHP (obrigatória, para delimitar todas as instruções PHP).

```
<?php
    include("inc_conteudos.php"); //incluir o ficheiro externo
?>
```

A solicitação de cada módulo é efectuada através de uma variável do tipo *GET*, a variável "s" (de secção), colocada na *query string* (interrogação) do URL. É através do valor que for atribuído a essa variável que a aplicação *Web* controla qual o módulo a ser incluído.

O exemplo seguinte analisa as diferentes partes de um hipotético URL:

```
http://r2s.ath.cx:80/tese/index.php?s=sobre&variavel2=123&variavel3=34#marcador
```

1 2 3 4 5 6 7 8 7 8 7 9

No exemplo anterior, as cores / números correspondem às seguintes partes:

1. A parte azul escura do URL corresponde ao protocolo utilizado na comunicação, neste caso o de hipertexto.
2. A parte a verde corresponde ao domínio, ou seja o nome do servidor ao qual estamos a efectuar o pedido.
3. A verde seco é indicada qual a porta de comunicação utilizada (quando diferente da porta 80 - a porta convencionada).
4. A parte azul clara indica qual o(s) eventual(ais) subdirectório(s) abertos.
5. A laranja está indicado qual o ficheiro que está a ser requisitado.
6. O ponto de interrogação faz a separação no URL entre o endereço físico e os dados que são passados para o servidor.
7. A vermelho encontramos as diversas variáveis, do tipo *GET*, e respectivos valores assignados, que são enviados conjuntamente com o pedido, ao servidor. Podemos verificar a existência de 3 variáveis: "s", "variavel2" e "variavel3". As variáveis são separadas entre si com o caractere "&".
8. A castanho, o marcador interno, correspondendo a uma zona dentro do documento a ser aberto. O *browser* automaticamente irá fazer os elevadores actualizarem de forma a mostrar a zona do documento solicitada.

Através do PHP, e do controle efectuado à variável "s" da *query*, que será incluído condicionalmente cada módulo.

O seguinte exemplo é o extracto do ficheiro "inc_conteudos.php", que corresponde à zona onde surge cada módulo que desenvolve o conteúdo de cada secção da aplicação.

```

<?php
    $s=$_GET["s"]; //extrair dados da hipotética variável "s" da query string
    if (empty($s)) { //confirmar se existe valor atribuído
        $ficheiro = "inc_home"; //ficheiro a ser incluído por defeito
    } else {
        $ficheiro = "inc_{$s}"; //construção do nome do ficheiro
    }

    if (file_exists("$ficheiro.php")) { // se o ficheiro do módulo existe fisicamente
        include "$ficheiro.php"; //inclui o módulo
    } else {
        $ficheiro = "inc_home"; //se não existe, inclui o módulo do home
    }
?>

```

As ligações ao SGBD MySQL são produzidas através de instruções PHP que permitem ligar e manipular informação do MySQL. O exemplo seguinte documenta um exemplo de ligação simples, com retorno registos de uma dada tabela:

```

<?php
$config = mysql_connect("localhost", "username", "password"); // conectar ao SGBD
mysql_select_db($database_config, $config); //seleccionar uma base de dados

$query_feeds = ("select * from feed"); //definir a instrução SQL a executar
$feeds = mysql_query($query_feeds, $config); // executar a instrução SQL no SGBD

while ($row_feeds = mysql_fetch_assoc($feeds) { //ciclo - por cada linha do recordset
    echo $row_feeds["atitle"] . "<br/>"; //escrever uma linha com o título da feed
}

mysql_close($config); //fechar a ligação para libertar recursos
?>

```

A interacção com o utilizador foi produzida através das seguintes formas:

- Hiperligações tradicionais, construídas de acordo com o modelo de variáveis *query string* explicadas anteriormente.
- Manipulação de formulários.

Os formulários, que são constituídos por um conjunto de *tags* HTML, são preenchidos e submetidos pelo utilizador. Foram produzidos os seguintes formulários:

- Formulário de adesão, para permitir a que o visitante possa registar-se no sistema.
- Formulário de autenticação, permitindo a que o utilizador possa autenticar-se/ iniciar uma nova sessão no sistema.
- Formulário de perfil, oferecendo a possibilidade ao utilizador de actualizar os seus dados pessoais.

- Formulário de adicionar uma nova *feed* ao perfil, para que o utilizador possa adicionar uma nova *feed* através do seu URL.
- Formulário de filtro de *feed*, por forma a possibilitar a que o utilizador possa indicar palavras-chave para a filtragem de conteúdos.

A informação preenchida no formulário e submetida pelo utilizador é enviada ao servidor para que possa ser recepcionada e tratada pelas linguagens *server side*. O PHP possui um conjunto de variáveis de sistema (através de matrizes criadas e disponibilizadas automaticamente), entre as quais algumas delas direccionadas para a recolha de informação submetida ao servidor. O exemplo seguinte demonstra a integração de um pequeno formulário HTML com a linguagem PHP.

```
<?php
if ($_POST["button"]) { //verifica se foi submetido via POST a variável button.
    echo "O seu nome é: " . $_POST["nome"] . "<br/>"; //escreve nome
    echo "O seu nome tem " . strlen($_POST["nome"]) . " letras."; //indica número de letras
}
?>

<form id="form1" name="form1" method="POST">
<p><label>Nome:<input type="text" name="nome" id="nome" /></label></p>
<p><input type="submit" name="button" id="button" value="Submeter" /></p>
</form>
```

O controle de sessões, para permitir a autenticação dos utilizadores, é conseguido através das variáveis de sessão, também disponibilizadas pelo PHP. Ao contrário das restantes variáveis, que são descartadas no final do processamento, as variáveis de sessão permanecem enquanto a sessão decorrer. As sessões finalizam por uma das seguintes razões:

- Existir alguma instrução PHP que indique o final da sessão ou o início de uma nova sessão.
- O utilizador desligar a janela do *browser*.
- O utilizador não efectuar mais nenhum pedido ao servidor durante algum tempo (finalizar por *timeout*).

O seguinte exemplo demonstra a forma de como as variáveis de sessão em PHP funcionam e como foram implementadas no sistema.

```
<?php
session_start(); /* instrução necessária para criar ou continuar uma sessão. É necessária em todos os ficheiros onde se recorra ao uso de sessões */
```

```
$_SESSION["userid"]=1; //atribuir valor 1 à variável de sessão"id"
$_SESSION["nome"]="Hélder Pestana"; //idêntico ao anterior
?>
```

No exemplo anterior estamos a especificar atribuir valor directamente, mas estes valores poderiam vir dinamicamente de uma base de dados.

Com base neste exemplo podemos usar as variáveis de sessão para restringir o acesso indevido do utilizador a uma das páginas ou módulos da aplicação *Web*, da seguinte forma:

```
<?php
session_start();/* instrução necessária para criar ou continuar uma sessão. É necessária em todos os ficheiros onde se recorra ao uso de sessões */

if (!$_SESSION["userid"]) { //se a variável não existe ou não tem valor atribuído
    header("Location: index.php"); //reencaminha para outra página
}
?>
```

Caso queiramos destruir a sessão e iniciar outra, podemos usar as seguintes instruções

```
<?php
session_start();/* instrução necessária para criar ou continuar uma sessão. É necessária em todos os ficheiros onde se recorra ao uso de sessões */

if ($_GET["act"]=="logout") { //caso exista um pedido de logout
    unset($_SESSION["userid"]); //destruir uma variável de sessão individualmente
    session_destroy(); //destrói a sessão actual (toda a informação)
    session_start(); //iniciar uma nova sessão
}
?>
```

O quadro seguinte resume todos os ficheiros PHP desenvolvidos, indicando a sua finalidade, número de linhas escritas (de um total de aproximadamente 1800 linhas) e a indicação se o módulo corresponde a uma área privada (cujo acesso é restrito apenas a utilizadores autenticados/ com sessão activa).

Tabela 4-3 Quadro resumo dos ficheiros PHP desenvolvidos no sistema

Ficheiro PHP	Priv.	Lin.	Sumário
index		150	Definir a estrutura geral do modelo do <i>layout</i> em HTML, CSS. Em cada zona respectiva é efectuado o <i>include</i> aos ficheiros seguintes. Inclui também as funções de login e logout do sistema.
config		323	Contém as configurações gerais da aplicação, bem como as funções gerais.
inc_autenticacao		30	Mostra o formulário para o utilizador permitir iniciar a sua sessão, autenticando-se no sistema. Quando autenticado faz o inverso, mostra o botão que permite o utilizador desligar a sua ses-

			são.
inc_menu		55	Contém o menu da aplicação <i>Web</i> , que é disponibilizado quando o utilizador está autenticado no sistema. Quando não está autenticado a zona é utilizada para mostrar uma animação a incentivar o visitante a registar-se.
inc_conteudos		16	Gere a zona central de conteúdos, incluído condicionalmente o módulo/secção solicitada.
inc_acessorios		38	Zona reservada aos acessórios e ferramentas no sistema. Neste caso está a ser usada para indicar quais as <i>feeds</i> mais populares.
inc_home		7	Mostra o texto introdutório à aplicação, solicitando o utilizador a registar-se.
inc_aderir	Sim	150	Permite o utilizador criar um novo registo. Apenas utilizadores não autenticados podem aceder a este módulo
inc_perfil	Sim	150	Permite ao utilizador autenticado no sistema modificar os seus dados pessoais.
inc_adicionarfeed	Sim	81	Mostra o formulário que permite ao utilizador autenticado poder adicionar uma <i>feed</i> ao seu perfil. Após a submissão chama (via Ajax ou em alternativa via POST, caso o Javascript esteja indisponível) o serviço " <i>feedcapture.php</i> ".
inc_myfeeds	Sim	234	Permite ao utilizador autenticado gerir as suas <i>feeds</i> (eliminar, definir filtros).
inc_sms	Sim	70	Possibilita o utilizador autenticado consultar o histórico de SMS enviados.
feedcapture	Sim	384	Verifica se a <i>feed</i> indicada já existe no sistema e associa-a ao utilizador. Caso não exista, tenta extrair e efectuar o <i>parsing</i> à <i>feed</i> . Regista todas as entradas (antigas) da <i>feed</i> como enviadas.
inc_ajuda		21	Mostra os tópicos de ajuda.
inc_sobre		3	Mostra a autoria do sistema.
inc_contactos		31	Mostra o formulário de contacto.
inc_sendmail		45	Envia por email os dados recebidos de um formulário.
inc_termos		0	Futuro texto de termos e condições de uso.
inc_privacidade		0	Futuro texto de termos de privacidade.

4.2.6.2 Os processos automatizados

Para que o sistema de informação proposto possa fazer sentido, existe a necessidade que alguns processos tenham que ser despoletados automaticamente, num intervalo regular. Foram desenvolvidos os seguintes processos, considerados fulcrais para o bom funcionamento do sistema:

- Procedimento de captura de *feeds*.

- Procedimento de envio de SMS.

Cada um destes procedimentos foi desenvolvido através de ficheiros PHP, sem qualquer interface gráfica, processados no lado do servidor.

Tabela 4-4 Ficheiros PHP relativos aos processos automatizados do sistema

Ficheiro PHP	Linhas	Sumário
capture	284	Verifica quais as <i>feeds</i> subscritas pelos utilizadores, tenta ler e extrair informação de todas as entradas da <i>feed</i> que ainda não estejam registadas na base de dados do sistema.
sendsms	160	Verifica quais as entradas que ainda não foram enviadas aos subscritores (novas) e procede ao seu envio.
isonline	3	Apenas devolve a mensagem "Yes". Serve para que os ficheiros anteriores possam verificar se existe ligação à Internet, mediante um pedido HTTP a este ficheiro, antes de efectuar qualquer procedimento de captura ou envio de informação.

O processo de captura consulta a base de dados procurando obter a lista de *feeds* subscritas pelos utilizados. Por cada uma destas, tenta efectuar capturar a informação da *feed* via HTTP.

```
<?php
//a instrução seguinte guarda a informação devolvida ao URL indicado na variável page
$page=@file_get_contents("http://Webmanager.ipt.pt/site/rss.asp?id=noticias");
?>
```

Após a leitura com sucesso, podemos extrair dados através do *parsing* do XML devolvido.

```
<?php
if (!$page) { // se não devolveu nada...
    echo "Erro, não devolveu nada..."; // ...devolve erro
} else { //se devolveu informação...
    $myxml = new SimpleXMLElement($page); // ...criar o objecto XML
    $feedtitle=utf8_decode($myxml->channel->title); //extrai o título da feed
    echo "Leu correctamente a feed " . $feedtitle;
}
?>
```

O processo de envio de consiste num processo idêntico, mas o URL ao qual é feito um pedido de comunicações via HTTP é o serviço de envio de SMS proporcionado pela operadora Voipcheap. A única grande diferença reside na resposta devolvida: no caso

de uma *feed*, o resultado surge em XML usando a especificação RSS ou Atom; no caso da resposta da Voipcheap, o resultado é uma estrutura XML proprietária.

O exemplo seguinte mostra a resposta em XML de pedido simples à gateway do Voipcheap sem qualquer parâmetro (devolvendo um erro):

```
<SmsResponse>
  <version>1</version>
  <result>0</result>
  <resultstring>failure</resultstring>
  <description>The parameter to (destination number) is missing</description>
  <endcause></endcause>
</SmsResponse>
```

Através do resultado obtido é possível saber se o existiu ou não um envio (indicado quando o *node* (nó) *result* possui um resultado superior a zero), permitindo a que seja possível o controle interno da informação que foi enviada efectivamente por SMS ou não.

A automatização destes dois processos, de leitura e de envio, foi conseguida através da instalação no servidor de um *Software* gratuito de *cron job* (automatização de tarefas através de um intervalo regular de tempo) denominado por pycron.

Através deste *Software* é possível especificar uma nova tarefa indicando: o comando a ser executado (e respectivos parâmetros); o intervalo de tempo (ou data específica) em que a tarefa deve ser executada.

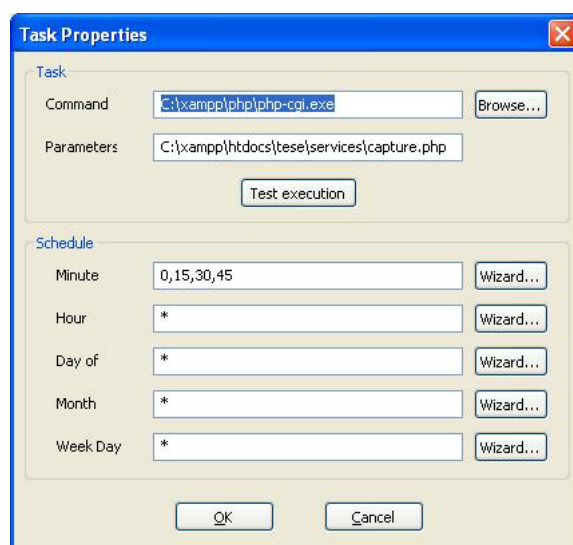
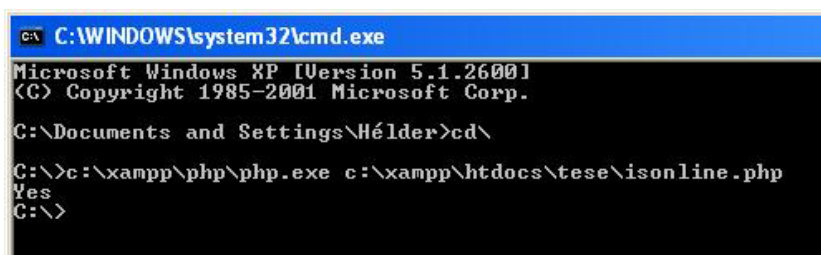


Fig. 4-13 Criar uma nova tarefa automatizada no pycron

Neste caso, foram especificadas duas tarefas, com vista a automatizar os dois processos previstos: a leitura de informação das *feeds*; o envio de SMS.

A execução dos ficheiros PHP é concretizada através da sua execução via linha de comandos, através da execução do interpretador "php.exe".



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Helder>cd\

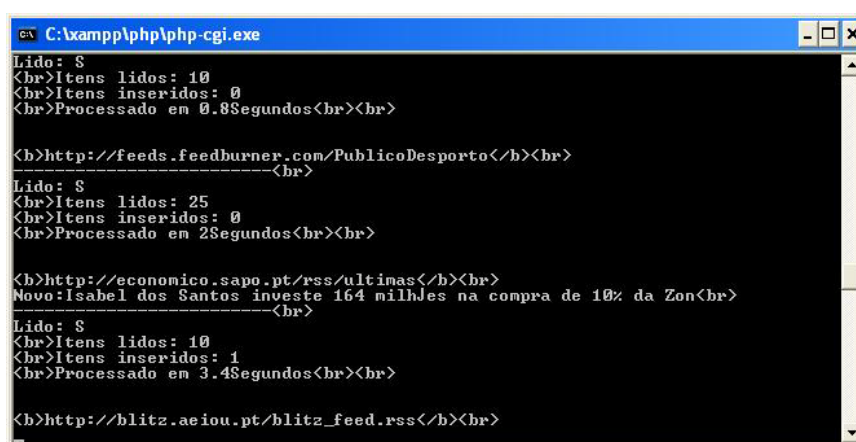
C:\>c:\xampp\php\php.exe c:\xampp\htdocs\tese\isonline.php
Yes
C:\>
```

Fig. 4-14 Executar ficheiros PHP via linha de comandos

Especificamos que em ambos os casos, em termos gerais, as tarefas deviam ser executadas todos os dias em qualquer mês do ano, a todas as horas. Seguidamente e isoladamente para cada uma das tarefas especificamos execuções em minutos diferentes, a fim de distribuir a utilização dos recursos da máquina:

- para a leitura das *feeds* definimos um intervalo regular de 15 em 15 minutos
- para o processo de envio definimos um envio de 4 em 4 minutos.

Após definir as tarefas no pycron, o serviço executa as tarefas nos intervalos fixados.



```
C:\xampp\php\php-cgi.exe
Lido: S
<br>Itens lidos: 10
<br>Itens inseridos: 0
<br>Processado em 0.88segundos<br><br>
<b>http://feeds.feedburner.com/PublicoDesporto</b><br>
-----<br>
Lido: S
<br>Itens lidos: 25
<br>Itens inseridos: 0
<br>Processado em 2segundos<br><br>
<b>http://economico.sapo.pt/rss/ultimas</b><br>
Novo: Isabel dos Santos investe 164 milhjes na compra de 10% da Zon<br>
-----<br>
Lido: S
<br>Itens lidos: 10
<br>Itens inseridos: 1
<br>Processado em 3.4segundos<br><br>
<b>http://blitz.aeiou.pt/blitz_feed.rss</b><br>
```

Fig. 4-15 Leitura das *feeds* subscritas através de uma tarefa automatizada num intervalo regular

A figura anterior mostra a execução automatizada da tarefa de captura de informação das diversas *feeds* subscritas pelos utilizadores.

Neste capítulo apresentou-se em detalhe a estrutura do sistema, composto pelas tecnologias PHP, HTML, CSS, MySQL e Javascript, num total de aproximadamente 6000 linhas de código.

No capítulo seguinte vamos abordar a apresentação e discussão de resultados.

5 Apresentação e discussão dos resultados

O desenvolvimento do Sistema de Notificações Automáticas de Informações Provenientes de Fontes *Web*, através de Mensagens Escritas para Telemóveis, culminou com a concepção e desenvolvimento de um protótipo funcional.

No presente capítulo mostram-se os resultados obtidos na sua avaliação, quer através de testes efectuados sobre o protótipo, quer através de dados obtidos através da realização de um questionário/teste de utilização do protótipo construído.

5.1 Teste de avaliação quanto à acessibilidade do protótipo

Os testes de acessibilidade visam aferir se o protótipo é acessível, isto é, aferir o grau de conformidade do código fonte (e conteúdo) do protótipo com as orientações da W3C para a acessibilidade das páginas *Web* (W3C, 2008).

O motor de validação utilizado para este objectivo foi o validador eXaminator da UMIC - Agência para a Sociedade do Conhecimento, IP. Este motor tem características únicas que o distinguem dos demais 130 motores de validação existentes, que é o facto de funcionar totalmente em linha, não requerendo qualquer tipo de instalação. Agrega, para além dos resultados da sua análise, resultados de outros motores de validação, nomeadamente:

- O validador de XHTML da W3C, que valida a sintaxe de um documento HTML.
- O validador de CSS da W3C, que valida a sintaxe de um documento CSS ou regras CSS contidas em documentos HTML.

Importa referir que nenhum motor de validação de acessibilidade consegue validar totalmente os três níveis de conformidade da W3C para a acessibilidade ("A", "AA" e "AAA"), existindo pontos que requerem sempre uma validação humana.

A análise do eXaminator indicou que o protótipo se encontra em conformidade "AAA" (nível máximo), devolvendo um índice de satisfação 10 (de 0 a 10). Foram produzidos 18 testes, todos com resultado positivo. Esta análise está disponível na íntegra no apêndice A2.

Tabela 5-1 Resultados da avaliação do protótipo quanto à acessibilidade

Conformidade	Testes Positivos / Testes aplicados
Prioridade 1 - "A"	2 / 2
Prioridade 2 - "AA"	13 / 13
Prioridade 3 - "AAA"	3 / 3

Os testes efectuados à prioridade 1 procederam às seguintes verificações:

- Colocação de texto alternativo nos botões gráficos
- Existência de código alternativo aos *scripts* (javascript).

Os testes efectuados à prioridade 2 consistiram em:

- Associação de etiquetas de rótulos (*label*) a campos de formulário
- Não utilização de *tags* ou atributos obsoletos ou não aconselhados.
- Declaração do tipo de documento.
- Verificação das hiperligações e do texto associado às mesmas.
- Utilização de cabeçalhos na estrutura do documento.
- Validação da sintaxe do código HTML.
- Utilização de CSS na formatação de estilo da página.
- Validação da sintaxe do código CSS.
- Uso de medidas relativas na especificação do tamanho da letra.
- Não utilização de tabelas na construção do *layout*.

Os testes efectuados à prioridade 3 verificaram:

- Identificação do idioma principal da página.
- Colocação de texto por omissão nos formulários.
- Separação com um ou mais caracteres de hiperligações adjacentes.

O relatório solicitava a verificação manual de 1 ocorrência (aviso) na prioridade 3: inexistência de informação sobre documentos relacionados. Existe, no entanto, uma

inconsistência neste aviso uma vez que, adoptando o exemplo fornecido para a resolução do aviso, leva-nos a um erro de má formação de HTML.

Quanto à velocidade de processamento, por cada pedido, as páginas oscilaram entre 0.02 e 0.21 segundos (tempo mínimo e máximo obtido respectivamente).

5.2 Teste de desempenho do sistema

Com vista a avaliar o desempenho funcional do protótipo organizou-se um teste de utilização do protótipo desenvolvido. O teste de avaliação consistiu em três fases:

1. Resposta a um questionário inicial, antes da visualização / utilização do protótipo, com a finalidade de aferir os conhecimentos e opiniões do inquirido.
2. Utilização do protótipo livremente durante um período mínimo de 48H.
3. Resposta a um questionário final, após a utilização do protótipo, para avaliar a opinião do inquirido sobre a sua experiência com a utilização do protótipo.

A população escolhida para avaliar o desempenho do sistema é constituída por indivíduos com conhecimentos básicos de informática na óptica de utilizador, e que tenham telemóvel. Dada a dificuldade em realizar uma amostragem probabilística, optou-se por uma amostragem não aleatória, por conveniência, recorrendo aos elementos voluntários da população que estavam disponíveis (MEXIA, et al., 2005). A amostra é constituída por 25 voluntários, 20 destes caracterizados por serem do sexo masculino e 5 do sexo feminino, com idades compreendidas entre os 13 e os 62 anos, com uma formação académica diversificada. Durante este subcapítulo iremos analisar os dados obtidos sobre o questionário inicial, da utilização do protótipo e questionário final.

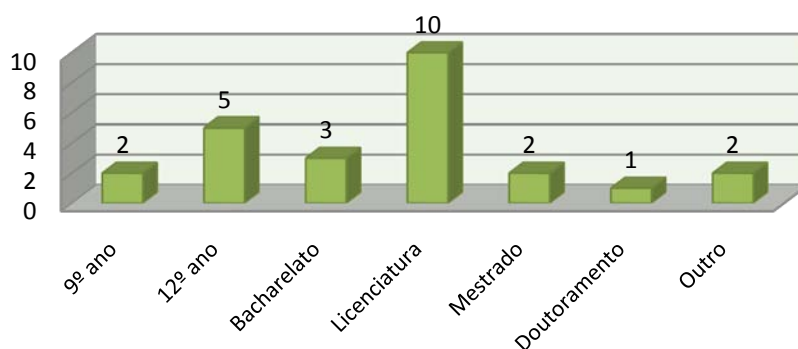


Fig. 5-1 Formação académica dos inquiridos

5.2.1 Apresentação e análise de resultados do questionário inicial

Pretende-se neste ponto apresentar os dados recolhidos no questionário inicial. Começamos por apresentar dados sobre: caracterização das formas de acesso, locais e dispositivos que o inquirido usa para aceder à Internet, bem como os seus hábitos de utilização; identificação dos dispositivos electrónicos que costumam levar consigo para qualquer lugar; aferir os hábitos de utilização de SMS; conhecer se o inquirido utiliza alguma forma de sindicância/agregação ou notificação automática de conteúdos provenientes da *Web*; aferir se teria interesse ou não em receber essas notificações por SMS.

5.2.1.1 Apresentação de resultados

Todos os inquiridos, assinalaram possuir pelo menos uma forma de acesso à Internet. Quase metade destes (11) assinalaram possuir mais do que uma forma de acesso à Internet. Todos os inquiridos indicaram possuir ligação à Internet a partir de casa, através de ligação física (cabo/ADSL), e no local de trabalho (Fig. 5-2). É importante assinalar o baixo nível de respostas para os acessos móveis, revelando que os inquiridos não têm hábito de aceder à Internet usando formas de acesso móvel, como os acessos 3G ou GPRS.

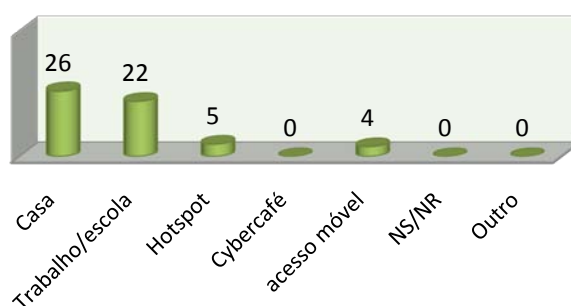


Fig. 5-2 Locais em que os inquiridos têm acesso à Internet

Quando questionados sobre o seu grau de conhecimentos informáticos (auto-avaliação), cerca de 67% apontou possuir conhecimentos médios e os restantes 33% assinalaram ter bons conhecimentos. Ninguém assinalou ter fracos conhecimentos informáticos.

Sobre os dispositivos que utilizam frequentemente para aceder à Internet (Fig. 5-3), foi com pouca surpresa que o computador pessoal assumiu o principal e único dispositivo a que todos os inquiridos assinalaram, a larga distância da segunda opção, o tele-

móvel. Estes dados vem corroborar a hipótese que os inquiridos não tem hábitos de acesso móvel à Internet.

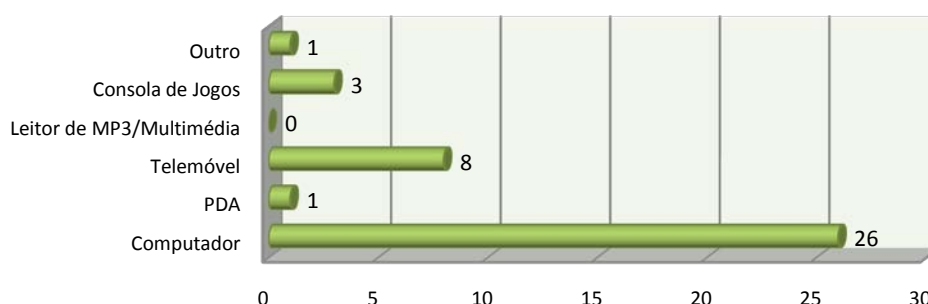


Fig. 5-3 Dispositivos que utiliza para aceder à Internet

Inquiridos sobre que dispositivos costumam levar para qualquer local (trabalho, viagem, praia, cinema, etc.), o dispositivo mais indicado foi o telemóvel (Fig. 5-4), mas seguido de muito perto, para grande surpresa, pelo computador pessoal (depreendemos que se trate de um computador portátil).

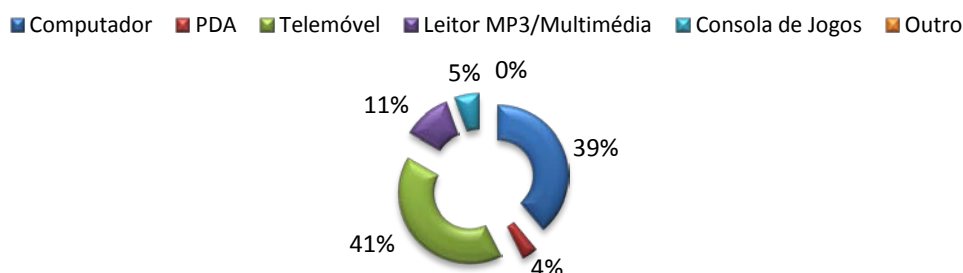


Fig. 5-4 Dispositivos que os inquiridos levam para qualquer lugar

Sobre o tempo diário médio que despendem a aceder à Internet, o resultado é muito equilibrado até às 4 horas diárias. Inquiridos sobre a quantidade de sites que visitam regularmente, todos os dias, a resposta mais seleccionada assinalou mais que 10 sites, muito embora mais de metade dos inquiridos tenham respondido um número inferior a 10. Quanto ao processo de pesquisa de informação (pertinente para o inquirido) nos sites indicados, apesar de 54% dos inquiridos indicarem que o processo de pesquisa de informação é fácil, 38% indicaram que necessitam que necessitam de pesquisar informação e filtrar resultados para obterem a informação que procuram, enquanto que só apenas 8% indicaram ser um processo difícil.

Questionados sobre se costumam subscrever notificações automáticas de conteúdos de algum site, a maioria respondeu que não (13 inquiridos) muito embora os resultados sejam muito equilibrados.

Aos inquiridos que responderam positivamente à questão anterior (12 inquiridos), foi-lhes colocado 3 perguntas adicionais no seguimento da questão:

1. Quanto ao *Software*/formas que os inquiridos recorrem para recepcionar essas notificações (Fig. 5-5), a resposta mais indicada foi o email, com 9 respostas, seguida da resposta leitura por *Software* apropriado, com 6 respostas.
2. Inquiridos sobre o tempo médio (em horas) entre o momento de envio da notificação e o momento de leitura do inquirido (Fig. 5-6), a resposta mais apontada foi até 4h, onde a média das respostas rondou as 5h. A resposta mínima foi 0h e a resposta máxima foi 24h.
3. Quando questionados sobre a relação de satisfação no que diz respeito à velocidade de envio/recepção da questão anterior, a maioria dos inquiridos (50%) indicou que o seu grau de satisfação era médio (Fig. 5-7).
4. Questionados sobre se gostariam de poder receber notificações em qualquer lugar, 64% dos inquiridos respondeu positivamente contra 36% (Fig. 5-8).

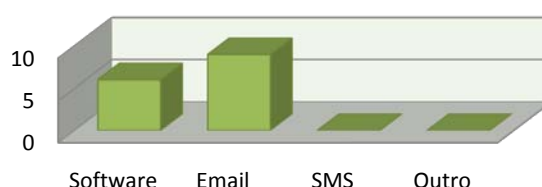


Fig. 5-5 Forma de recepção / leitura das notificações

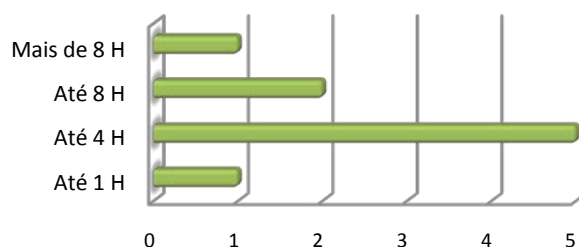


Fig. 5-6 Tempo médio decorrido entre o envio e a leitura da notificação

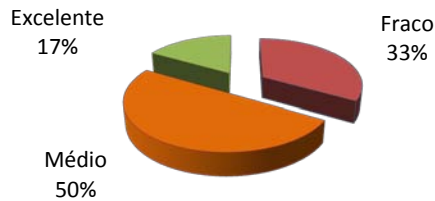


Fig. 5-7 Grau de satisfação da relação entre o tempo de envio e leitura da notificação

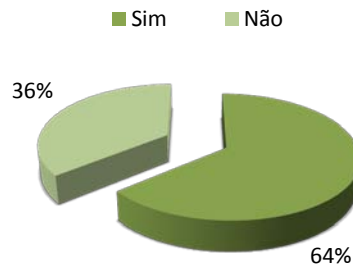


Fig. 5-8 Aferição sobre a vontade de receber notificações em qualquer lugar

Quanto ao conhecimento dos serviços/formatos de sindicacão RSS/Atom/RDF ou Podcast, 72% dos inquiridos (18) afirmaram ter algum tipo de conhecimento sobre os mesmos. A maioria (14 respostas) afirmou ainda que entre os sites que visitam diariamente, alguns destes, possuem um dos formatos de sindicacão indicados anteriormente (Fig. 5-9).

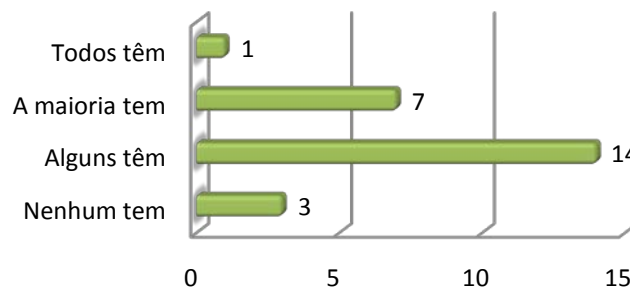


Fig. 5-9 Existência de sindicacão de feeds nos sites habituais dos inquiridos

Apesar de 72% conhecerem os serviços/formatos de sindicacão, apenas 64% (16 inquiridos) afirmaram utilizar ou saber para que servem. Aos inquiridos que responderam positivamente foi-lhes solicitado que indicassem quais os dispositivos físicos que usaram para aceder aos referidos serviços (Fig. 5-10). A opção computador pessoal foi largamente mais indicada, com 15 indicações, contra apenas 2 indicações para o

telemóvel. Ninguém afirmou usar a consola de jogos, PDA ou leitor de MP3/multimédia para o fazer.

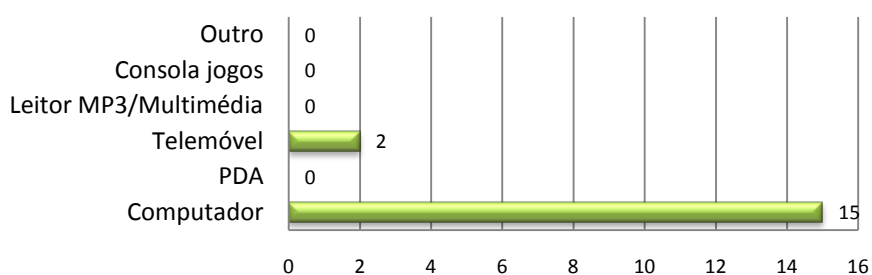


Fig. 5-10 Dispositivos utilizados na leitura de feeds

Finalmente, foi aferida a utilização dos inquiridos de SMS do telemóvel e principalmente saber a sua opinião sobre a possibilidade de conjugação das SMS com a leitura de feeds. A larga maioria, 84%, indicou utilizar regularmente as SMS no telemóvel, indicando que dominavam todos os aspectos de manipulação das SMS de uma forma média ou excelente. A maioria dos inquiridos manifestou vontade de receber notificações por SMS, com 93% (Fig. 5-11).

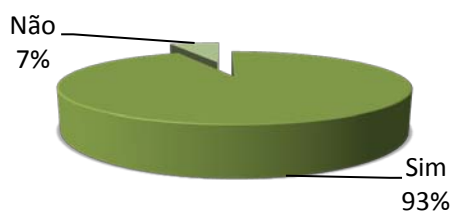


Fig. 5-11 Disponibilidade dos inquiridos em receber notificações por SMS

Questionados sobre se a utilização de SMS para notificar actualizações melhoraria o tempo entre o envio e a leitura (Fig. 5-12), 68% não soube ou não quis responder, 32% indicaram positivamente e ninguém respondeu negativamente.

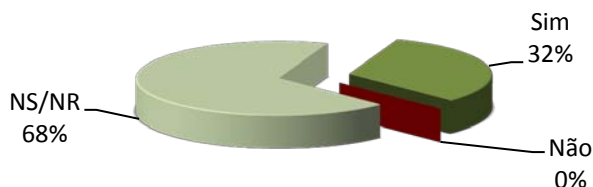


Fig. 5-12 Opinião dos inquiridos sobre se a utilização das SMS melhoraria o tempo dispendido entre o envio e a leitura da informação.

5.2.2 Estatísticas sobre a utilização do protótipo

A utilização do protótipo foi realizada entre os dias 05 de Dezembro de 2009 e o dia 13 de Dezembro de 2009. Os voluntários registaram-se em dias diferentes, durante o período referido, tendo cada um destes utilizado o sistema durante o período mínimo de 48H.

Os dados que apresentamos agora foram registados durante todo o período. Durante todo o período foram processados 21 *feeds* diferentes, normalmente provenientes de serviços noticiosos ou de canais temáticos.

Tabela 5-2 Feeds subscritas pelos utilizadores

Feeds	
BLITZ	Peopleware
b-on:Biblioteca do Conhecimento Online	Pitchfork: Latest News
Ciência - Google Notícias	Publico.pt - Desporto
Clubeinvest.com - Análises Publicadas	Publico.pt - Geral
Correio da Manhã OnLine	RCAAP Rss Feeder
Diário Económico	Record OnLine - Benfica
Elektor.pt: Notícias	Record OnLine - Porto
Infovini Notícias	Record OnLine - Sporting
jn.pt - Mundo	tek
Ministério da Administração Interna	TopRural - Opiniones sobre Casa WladiVal
Notícias - Instituto Politécnico de Tomar	

A *feed* que mais produziu entradas foi o Público, com 635 entradas. De forma geral, as *feeds* que geraram mais entradas consistiam em *feeds* de órgãos de comunicação.

Cada utilizador subscreveu a notificação em média de 1.7 *feeds*, gerando cada uma destas uma média 124.7 novos itens (entradas) durante todo o período.

Tabela 5-3 Estatísticas sobre as *feeds* subscritas pelos utilizadores

Total de <i>feeds</i> subscritas	21
Média de <i>feeds</i> subscritas pelo utilizador	1.7
Total de <i>feeds</i> subscritas que produziram o envio de SMS	15
Total de itens de <i>feeds</i> processados durante todo o período	2745
Média de itens processados por <i>feed</i>	124.7
Número máximo de itens processados numa <i>feed</i>	635

O sistema de informação gerou durante todo o período 1102 SMS para os 25 utilizadores. Cada utilizador recebeu em média 44.1 SMS durante todo o período, 29.8 por cada *feed*. Por dia cada utilizador recebeu em média 17.1 SMS.

Tabela 5-4 Estatísticas de envio de SMS para os utilizadores

Total de SMS enviadas	1102,0
Média de SMS enviadas por dia	122.4
Média de SMS enviadas por utilizador durante todo o período	44.1
Média de SMS enviadas por utilizador, por dia	17.1
Máximo de SMS enviadas por utilizador (ocorrência mais alta)	131
Média de SMS enviadas por cada <i>feed</i>	73.5
Média de SMS enviadas por cada <i>feed</i> subscrita pelo utilizador	29.8

5.2.3 Apresentação e análise de resultados do inquérito final

O inquérito final teve por objectivo aferir a experiência de utilização do sistema, em termos de usabilidade, acessibilidade, e também do serviço obtido.

Todas as questões tinham uma escala de respostas possíveis entre 1 a 5, de muito mau a muito bom respectivamente. Passamos a apresentar os resultados obtidos.

5.2.3.1 Apresentação de resultados

Questionados sobre a usabilidade (facilidade de utilização / interacção) do protótipo, os inquiridos atribuíram sua maioria o valor de muito bom (63%), e os restantes 37% com o valor bom. Nenhum dos inquiridos respondeu suficiente, mau ou muito mau.

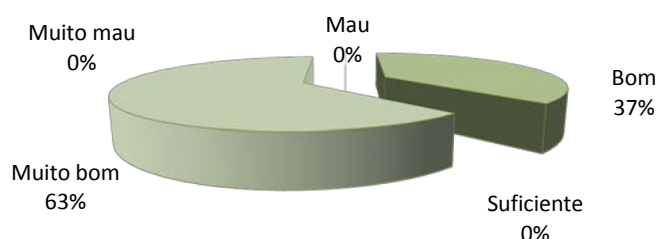


Fig. 5-13 Apreciação do protótipo quanto à sua usabilidade

No que diz respeito à apreciação da arquitectura de informação do protótipo, ou seja a sua estrutura de informação, as opiniões dos inquiridos divergiram entre muito bom e bom, por apenas um valor.

Relativamente ao aspecto visual do site os valores assinalados distribuíram-se entre bom, com 54% dos votos, muito bom e suficiente.

O processo de criar um registo usando o protótipo foi classificado pela maioria dos inquiridos como sendo muito bom com 71%, 25% bom, 4% suficiente.

O processo de gestão de *feeds* obteve votação idêntica, muito embora com valores inferiores: 58%, 38%, 4% para muito bom, bom e suficiente respectivamente. O processo de filtragem, da informação provenientes de *feeds*, mostrou também um acréscimo nos valores médios, aumentando o valor do suficiente. Ainda assim os resultados são positivos com 54% para muito bom, 33% para bom e 13% para suficiente.

Seguidamente, aos inquiridos foi questionada a sua avaliação sobre a qualidade do conteúdo das mensagens SMS recebidas, face ao limite de 160 caracteres, para obter entendimento sobre o assunto retratado. Esta será provavelmente a questão mais pertinente em todo o inquérito, uma vez que é analisado, por parte dos utilizadores, um aspecto que dita a viabilidade do sistema testado.

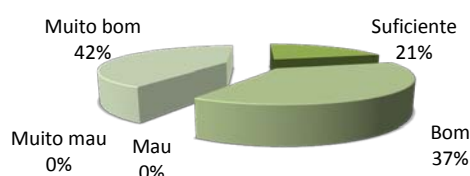


Fig. 5-14 Qualidade do conteúdo dos SMS enviados, face ao seu limite de 160 caracteres



Fig. 5-15 Resultados do questionário final sobre a utilização do protótipo

Estes indicadores, revelados através destes dados, permitem-nos concluir que o protótipo atingiu positivamente os seus objectivos.

6 Conclusões

A Internet assume-se, actualmente, como uma das principais formas de comunicação e de partilha de conhecimento, onde os indivíduos trocam cada vez mais informação entre si e, conseqüentemente, encontram-se conectados durante cada vez mais tempo.

Neste processo, na maior parte das vezes, o utilizador assume o papel activo na procura por informação actualizada. A presente tese procurou estudar a possibilidade de inversão dos papeis: o desenvolvimento de um serviço automático que possibilite que a própria informação, assim que disponível, fosse canalizada automaticamente para o mais comum dos cidadãos (que assume um papel passivo), em qualquer lugar a qualquer hora.

A solução consistiu em desenvolver um sistema de informação que utiliza um dos serviços do mais comum dos dispositivos que qualquer cidadão transporta para qualquer lugar: o telemóvel e uma das suas formas mais básicas de comunicação, as SMS.

O projecto consistiu no desenvolvimento de um sistema que permitisse a notificação da actualização de conteúdos, provenientes da leitura de informação contida em *feeds*, recorrendo ao envio de SMS para o telemóvel de cada subscritor.

Começamos por examinar os requisitos do sistema de informação com maior profundidade e identificar quais os actores intervenientes e a forma de como estes interagem com o sistema. Foram identificados três actores: o utilizador final, que pode adicionar e gerir a sua subscrição de *feeds* RSS ou Atom que são relevantes para si; o leitor de *feeds*, um processo automatizado que procede à leitura de todas as *feeds* subscritas, registando em base de dados todas as novas informações; o estafeta de SMS, um processo automatizado que envia as notificações por SMS para o subscritor todas as novas informações verificadas nas referidas *feeds*.

No estudo e análise do problema, numa fase prévia ao desenvolvimento do protótipo do sistema, foram adoptadas técnicas de modelação com recurso a UML.

O desenvolvimento do protótipo do sistema proposto foi concretizado sob a forma de uma aplicação *Web*. A concepção da aplicação *Web* foi concretizada através da adopção de tecnologias de código aberto ou de acesso gratuito, entre as quais destacamos a linguagem de programação PHP, o sistema de gestão de bases de dados MySQL, para além da (já esperada) utilização das especificações HTML e CSS e da utilização linguagem dinâmica do lado do cliente Javascript.

Foram escritas, aproximadamente, 3000 linhas de código (PHP, HTML, CSS, JavaScript) e integradas 3400 linhas de código provenientes de bibliotecas *open-source*, para um total de 6400 linhas de código incluídas no projecto.

Após o desenvolvimento do protótipo, foi aferida a sua qualidade através da realização dos seguintes testes: testes efectuados por motores de validação de sintaxe HTML e CSS, para aferir a sua consistência em termos de construção; testes de acessibilidade, para verificar se não existem no protótipo obstáculos à acessibilidade; teste de utilização por parte de 27 voluntários, para aferir a sua viabilidade.

Os testes de validação de sintaxe e de acessibilidade confirmaram que o protótipo se encontra 100% de acordo com as especificações e orientações da W3C - *World Wide Web Consortium*: quanto à boa sintaxe das linguagens utilizadas no desenvolvimento do protótipo, bem como quanto a sua conformidade em relação às orientações da W3C para a acessibilidade, sendo-lhe atribuída o nível máximo de conformidade "AAA".

Os testes de utilização do sistema, por parte dos utilizadores (voluntários), consistiu em 3 fases sequenciadas: o preenchimento de um questionário inicial, de aferição de conhecimentos; a utilização do protótipo durante o período mínimo de 48 horas; o preenchimento de um questionário final de avaliação do desempenho do protótipo.

Os resultados do inquérito inicial permitiram conhecer a realidade dos voluntários. Todos os indivíduos indicaram possuir pelo menos uma forma de acesso, à Internet, em casa ou no trabalho mas, no entanto, muitos poucos afirmaram possuir ou utilizar formas de acesso móvel. A maioria indicou não usar notificações automáticas de conteúdos, onde os que indicaram positivamente indicaram recorrer ao email e a *Software* de leitura próprio para recepcionar as notificações. A maioria indicou querer

receber notificações de forma mais rápida. Questionados sobre a hipótese de receberem notificações de actualizações de sites *Web* através de SMS a esmagadora maioria (93%) indicou positivamente, um índice importante para este estudo, uma vez que revela claramente um grande interesse por parte dos potenciais utilizadores.

Os resultados globais obtidos com a utilização do protótipo, permitem afirmar com alguma segurança que o protótipo cumpriu todos os objectivos propostos, com excelentes resultados: todos os pontos avaliados decorrentes do uso do protótipo, como a apreciação visual, facilidade de interacção com o protótipo, qualidade do conteúdo do SMS, organização, etc. receberam avaliações entre 4 a 5 valores (de 1 a 5). O factor mais importante talvez resida no facto de nenhum dos voluntários que utilizou o protótipo atribuir uma pontuação negativa a qualquer um dos pontos avaliados no questionário.

Durante o período de utilização, o sistema processou e enviou na totalidade para os 25 utilizadores cerca de 1102 SMS, com notificações de actualizações nas *feeds* subscritas por estes. Cada utilizador recebeu em média 17.1 SMS por dia provenientes de uma média de 1.7 *feeds* subscritas por utilizador.

O entusiasmo decorrente do processo de desenvolvimento e utilização do protótipo criado levantou, conforme esperado, algumas novas questões pertinentes. Algumas solicitam melhorias em versões futuras protótipo; outras apontam questões que poderão culminar na realização de futuros estudos. As melhorias que sugerimos para o futuro:

- Possibilidade de o utilizador limitar o número de SMS recebidas, num período a fixar por este (limitar SMS enviadas por hora, dia, mês).
- Possibilidade de o utilizador definir qual o período de validade para que a informação lhe seja entregue via SMS.
- Possibilidade de activar ou desactivar as hiperligações no final das SMS.
- Possibilidade de permitir o envio de informação em múltiplas SMS.
- Melhorar a qualidade das abreviaturas para permitir enviar mais informação numa só SMS.
- Detectar duplicações nas entradas das *feeds* a fim de evitar envio de SMS redundantes.

- Utilizar os dados registados no histórico para aferir o grau de periodicidade de uma *feed*, com o objectivo de poupar recursos do sistema em leituras desnecessárias.
- Utilizar os dados registos no histórico para calcular e notificar automaticamente o administrador do sistema, o momento em que o sistema necessitará de novos recursos.
- Possibilidade do sistema ser operado através de SMS, através de envio de comandos por SMS.
- Melhorar o sistema de filtragem de conteúdos das *feeds*.
- Estudar a infra-estrutura de *hardware* ideal para o sistema, de modo a permitir a que este opere nas melhores condições, por forma a garantir a melhor qualidade possível no serviço proporcionado.
- Enviar SMS directamente pelo próprio sistema, sem recurso a terceiros, melhorando o desempenho do sistema.

7 Bibliografia e Referências

- ADKISSON e Heidi** Examining the Role of De Facto Standards on the Web [Online] // Boxes and Arrows. - 13 de 10 de 2003. - 13 de 12 de 2009. - http://www.boxesandarrows.com/view/examining_the_role_of_de_facto_standards_on_the_web.
- ADKISSON Heidi** Web Design Practices [Online]. - 2005. - 13 de 12 de 2009. - <http://www.webdesignpractices.com>.
- ADOBE LABS** Spry - Javascript-based framework [Online]. - 2009. - 2009. - <http://labs.adobe.com/technologies/spry/home.html>.
- AGUADO Juan Miguel e INMACULADA J. Martínez** La comunicación móvil en el ecosistemainformativo:de las alertas SMSal Mobile 2.0 [Online] // Trípodos, número 23. - 2008. - 29 de 05 de 2009. - <http://www.raco.cat/index.php/Tripodos/article/view/118915/154119>.
- ALENCAR Anderson** Bibliotecas Digitais: uma nova aproximação [Online] // Universidade Federal de Paraíba. - 2004. - 11 de Julho de 2007. - <http://www.informacaoesociedade.ufpb.br/pdf/IS1410409.pdf>.
- ALMEIDA Robson L.** Disseminação de Conteúdos na Web: A tecnologia RSS como proposta para a comunicação científica [Online]. - 2008. - 28 de Abril de 2009. - http://eprints.rclis.org/13971/1/dissertação_final_rss.pdf.
- ALMEIDA Robson L.** Notificação Automática de conteúdo em Ciência da Informação: Uma experiência na utilização de feeds RSS em Periódicos Eletrônicos [Online] // Renote. - 2007. - 28 de Abril de 2009. - http://www.cinted.ufrgs.br/renote/nov2004/artigos/a6_distribuicao_conteudos.pdf.
- AMARAL Sueli** Web Sites: uso de tecnologias no cumprimento das funções da biblioteca [Online] // Universidade Federal de Paraíba, Informação e Sociedade: estudos. - 2006. - 7 de Julho de 2006. - <http://www.informacaoesociedade.ufpb.br/pdf/IS1520501.pdf>.
- ARANI Jafar Asgari** M-Learning from a Cell Phone: Improving Students' EMP Learning Experience [Online] // LNC 2007 Conference. - 2007. - 29 de 05 de 2009. - <http://linc.mit.edu/events/2007/2007LINCBook.pdf>.
- AtomEnabled** What is Atom [Online]. - 2004. - 07 de 01 de 2010. - <http://www.atomenabled.org/>.
- AYERS Danny e WATT Andrew** Beginning RSS and Atom programming [Online]. - John Wiley and Sons, 2005. - 18 de 05 de 2009. - [http://books.google.pt/books?id=9WILhQJcF9sC&printsec=frontcover&dq="An+info](http://books.google.pt/books?id=9WILhQJcF9sC&printsec=frontcover&dq=)

mation+feed+is+a+regularly+updated+XML+document+that+contains+metadata+about+a+news+source+"&source=gbs_summary_r&cad=0.

BAIRRÃO Margarida Documentos digitais e novas formas de cooperação entre Bibliotecas Escolares e a Comunidade Global [Livro]. - [s.l.] : DEGAI, Universidade de Aveiro, 2003.

BATES M.J. Towards an integrated model of information seeking and searching [Artigo] // New Review of Information Behaviour Research. - 2002.

BAUMGARTNER Robert, GOTTLOB Georg e HERZOG Marcus Visual Programming of Web Data Aggregation Applications [Online]. - Institut für Informationssysteme Technische Universität Wien, 1999. - 26 de 05 de 2009. - <http://www.isi.edu/info-agents/workshops/ijcai03/papers/Herzog-ijcai03-herzog.pdf>.

BAX Marcelo Peixoto As bibliotecas na Web e vice-versa [Secção do Livro] // Perspectivas em Ciência da Informação. - 1998. - Vol. 3.

BERNERS LEE Tim Presentation to CDA challenge by CDT et al [Online] // W3. - 28 de 02 de 1996. - 23 de 06 de 2009. - <http://www.w3.org/People/Berners-Lee/9602affi.html>.

BERNERS-LEE Tim The WorldWideWeb browser [Online] // Tim Berners-Lee. - 2009. - 07 de 12 de 2009. - <http://www.w3.org/People/Berners-Lee/WorldWideWeb>.

BERNERS-LEE Tim, HENDLER J. e LASSILA O. The semantic web: a new form of web content that is meaningful to computers will unleash a revolution of new possibilities [Online] // Scientific American. - Maio de 2001. - 28 de Abril de 2009. - <http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21>.

BERUBE Linda Digital Reference Overview [Online] // Ukoln. - 2003. - 11 de Julho de 2006. - <http://www.ukoln.ac.uk/public/nsptg/virtual/>.

BOOCH Grady, RUMBAUGH James e JACOBSON Ivar The Unified Modeling Language User Guide [Online]. - Addison Wesley, 20 de 10 de 1998. - 25 de 06 de 2009. - <http://www.dcc.fc.up.pt/~zp/aulas/0506/es/geral/bibliografia/Addison%20Wesley%20-%20The%20UML%20User%20Guide.pdf>.

BRANDT D. Scott Evaluating Information on the Internet [Online] // Information Today, Inc. - 1996. - 11 de Julho de 2006. - <http://www.lib.purdue.edu/itd/techman/evaluate.html>.

BROOKS Terrence A. Web Search: how the Web has changed information retrieval [Online] // Information Research. - 2003. - 11 de Julho de 2006. - <http://InformationR.net/ir/8-3/paper154.html>.

BRUNO Ricardo E. Content Management and XML: the needed interaction [Online]. - 2005. - 05 de 01 de 2009. - <http://eprints.rclis.org/archive/00006243/01/pag1012.htm>. - 1695-5498.

CARDOSO Gustavo Internet [Livro]. - [s.l.] : Quimera, 2003.

CCE eEurope 2005: Uma sociedade da informação para todos [Online] // União Europeia. - 28 de 05 de 2002. - 24 de 04 de 2009. - http://www.eu.int/information_society/eeurope/2002/news_library/documents/eeurope2005/eeurope2005_pt.pdf.

- CODD Edgar Frank** <http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf> [Online]. - 06 de 1970. - 10 de 12 de 2009. - <http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf>.
- COELHO J. Dias** A Sociedade da Informação e do Conhecimento - Um Desafio Epistemológico nos Sistemas de Informação [Livro]. - Lisboa : Faculdade de Economia - Universidade Nova de Lisboa, 2000.
- CORREIA Ana Maria Ramalho** Moving Towards An Information Society In Portugal [Livro]. - [s.l.] : INETI, 1999.
- COUNTER The** JavaScript Stats (entre 2008 e 2009) [Online] // thecounter.com. - 03 de 2009. - 13 de 12 de 2008. - <http://www.thecounter.com/stats/2009/March/javas.php>.
- DAMAS Luís** SQL [Livro]. - [s.l.] : FCA, 2002. - 5ª Edição.
- DELFINO Sérgio, NUNES Fátima e SPOTO Edmundo** Avaliação de Desempenho de SGBDs de Código Aberto [Online] // II Jornada do Conhecimento e da Tecnologia, UNIVEM, Marília, SP. - 2006. - 08 de 12 de 2009. - http://galileu.fundamet.br/jornada/artigos/computacao/Sergio_Delfino.pdf.
- DEVHOME** Short Message Service / SMS Tutorial [Online] // Developers ' Home. - 24 de Abril de 2009. - <http://www.developershome.com/sms/>.
- DHTMLGOODIES** modal message script [Online] // dhtmlgoodies.com. - 2009. - 2009. - <http://www.dhtmlgoodies.com/index.html?whichScript=modal-message>.
- DR** Decreto-Lei n.º 16/2005, Diário da República, I SÉRIE-A, N.º12 de 18 de Janeiro de 2005 [Online] // UMIC. - 18 de Janeiro de 2005. - Julho de 2006. - http://www.unic.pt/NR/rdonlyres/A14DD6B2-4C5A-4AFF-B31A-E473B7D97DEC/3615/DL_16_2005.pdf.
- DR** Resolução do Conselho de Ministros nº 115/98, DR nº 201, I Série B, de 1 de Setembro de 1998 [Online] // MCTES. - 1 de Setembro de 1998. - http://www.mctes.pt/index.php?id_categoria=15&id_item=164&action=2.
- DR** Resolução do Conselho de Ministros nº 16/96, DR nº 69, II Série, de 21 de Março de 1996 [Online] // MCTES. - 21 de Março de 1996. - 07 de 2006. - http://www.mctes.pt/index.php?id_categoria=15&id_item=119&action=2.
- DYCK Timothy** Server Databases Clash [Online] // eWeek. - 02 de 2002. - 08 de 12 de 2009. - <http://www.eweek.com/c/a/Database/Server-Databases-Clash/>.
- DZIEKANIAK Gisele e KIRINUS Josiane** Web Semântica [Online] // Scribd. - 2008. - 28 de Abril de 2009. - <http://www.scribd.com/doc/3462693/LIDO-2-Web-Semantica>.
- ELLERMAN Castedo** Channel Definition Format [Online] // W3C. - Microsoft, 03 de 10 de 1997. - 26 de 05 de 2009. - <http://www.w3.org/TR/NOTE-CDFsubmit>.
- FARIA João Pascal** MEEC - Programação Orientada a Objectos [Online] // Faculdade de Engenharia da Universidade do Porto. - 2001. - 07 de 01 de 2010. - <http://paginas.fe.up.pt/~jpf/teach/POO/>.
- FIGUEIREDO Bruno** Web Design - Estrutura, Concepção e Produção de sites Web [Livro]. - 2007. - ISBN 972-722-433-4.

GARLAND K.J., ANDERSON S.J. e NOYES J.M. The Intranet as a learning tool: a preliminary study [Online] // Information Research. - 1998. - 7 de Julho de 2006. - <http://informationr.net/ir/4-1/paper51.html>.

GASTALDO Denise e MIDORIKAWA Edson Processo de Engenharia de Requisitos Aplicado a Requisitos Não-Funcionais de Desempenho – Um Estudo de Caso [Online]. - 2003. - 24 de 06 de 2009. - http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER03/denise_gastaldo.pdf.

GCELPT Plano Tecnológico [Online] // Gabinete do Coordenador da Estratégia de Lisboa e do Plano Tecnológico. - 2005. - 24 de 04 de 2009. - <http://www.planotecnologico.pt>.

GOVERNO Resolução do Conselho de Ministros n.º 155/2007 [Online]. - 2 de 10 de 2007. - 13 de 12 de 2009. - http://www.acesso.unic.pt/legis/rcm_155_07.htm.

GUHA Ramanathan Curriculum Vitae [Online]. - 2005. - 26 de 05 de 2009. - <http://www.guha.com/cv.html>.

GUHA Ramanathan Meta Content Framework [Online]. - 2005. - 23 de 06 de 2009. - http://www.guha.com/mcf/mcf_spec.html.

GUTIERREZ Suzana Distribuição de Conteúdos e Aprendizagem Online [Online] // RENOTE - Revista Novas Tecnologias na Educação. - Novembro de 2004. - 28 de Abril de 2009. - http://www.cinted.ufrgs.br/renote/nov2004/artigos/a6_distribuicao_conteudos.pdf.

HARVARD RSS 2.0 at Harvard Law [Online] // Harvard Law. - 6 de 04 de 2004. - 26 de 05 de 2009. - <http://cyber.law.harvard.edu/rss/rssVersionHistory.html>.

HAZAËL-MASSIEUX Dominique Construindo um layout sem tabelas [Online] // W3C. - 20 de 12 de 2005. - 13 de 12 de 2009. - <http://www.maujor.com/w3ctuto/layout.html>.

HEDSTROM Margaret Digital Preservation: A Time Bomb for Digital Libraries, Computers and the Humanities [Livro]. - [s.l.] : Springer Netherlands, 1997.

HENDERSON John R. ICYouSee: T is for Thinking - A Guide to Critical Thinking About What You See on the Web [Online] // Library Webmaster, Ithaca College. - 2006. - 11 de Julho de 2006. - <http://www.ithaca.edu/library/training/think.html>.

HESIDO FleXcroll, Flexible and Accessible Custom Scroll Bars [Online]. - 2009. - 2009. - <http://www.hesido.com/web.php?page=customscrollbar>.

HORSTMANSHOF Louise How SMS makes a difference to Communication [Online] // Faculty of Arts, University of Sydney. - 2003. - 25 de 05 de 2009. - <http://conferences.arts.usyd.edu.au/viewabstract.php?id=53&cf=3>.

IWS Internet Usage in Europe [Online] // Internet World Stats. - 31 de 03 de 2009. - 2009 de 05 de 26. - <http://www.internetworldstats.com/stats4.htm#europe>.

IWS Top 47 Countries With The Highest Internet Penetration Rate [Online] // Internet World Stats. - 30 de 06 de 2008. - 26 de 05 de 2009. - <http://www.internetworldstats.com/top25.htm>.

JACOBSON I, CHRISTERSON M. e JONSSON P., ÖVERGAARD, G. Object Oriented Software Engineering - A Use Case Driven Approach” [Livro]. - [s.l.] : Addison Wesley, 1992.

JACOBSON Ivar, BOOCH Grady e RUMBAUGH James The Unified Software Development Process [Livro]. - [s.l.] : Addison-Wesley, 1999.

JQUERY jQuery - The write less, do more, Javascript Library [Online]. - 2009. - 2009. - <http://jquery.com/>.

KAPSALIS Vassilis, KOUBIAS Stavros e PAPAPOPOULOS OPC-SMS: a wireless gateway to OPC-based data sources [Online] // Computer Standards & Interfaces. - 2002. - 22 de 05 de 2009. - <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.1511>.

KEOGH Jim e DAVIDSON Ken XML Demystified [Livro]. - [s.l.] : MC Graw Hill, 2005.

KOEHLER Wallace Digital libraries and World Wide Web sites and page persistence [Online] // Information Research. - 1999. - 11 de Julho de 2006. - <http://informationr.net/ir/4-4/paper60.html>.

KRUG Steve Don't make me think - A Common Sense Approach to Web Usability [Livro]. - [s.l.] : Editora New Riders, 2000.

KUBLY Kristin Guiding Students in Using the World Wide Web for Research [Online] // Instructional Conference Technology, Middle Tennessee State University. - 1997. - 11 de Julho de 2006. - <http://www.mtsu.edu/~itconf/proceed97/kubly.html> .

LE BODIC Gwenaël Mobile messaging technologies and services: SMS, EMS and MMS [Online] / ed. Sons John Wiley and. - 2005. - 19 de 05 de 2009. - <http://books.google.pt/books?id=AWQgapsAIvoC&printsec=frontcover>.

MARKTEST 3,9 milhões de utilizadores de Internet [Online]. - 2008. - 28 de Abril de 2009. - <http://www.marktest.pt/internet/default.asp?c=1294&n=1860>.

MARKTEST 61% dos portugueses informam-se pela Internet [Online]. - 2008. - 28 de Abril de 2009. - <http://www.marktest.pt/novadir/default.asp?c=1682&n=1906>.

MARKTEST Internet na maioria dos lares nacionais [Online]. - 2008. - 28 de Abril de 2009. - <http://www.marktest.pt/internet/default.asp?c=1294&n=1942>.

MARKTEST Portugueses enviam 12 SMS por dia [Online]. - MARKTEST, 31 de Dezembro de 2008. - <http://www.marktest.com/wap/a/n/id~123e.aspx>.

MARKTEST Portugueses enviam 5 SMS por dia [Online]. - 30 de Maio de 2006. - 24 de Abril de 2009. - <http://www.marktest.com/wap/a/n/id~aec.aspx>.

MARQUES Célio Os Hipermedia no Ensino Superior. Aplicação da Teoria da Flexibilidade Cognitiva [Livro]. - Cartaxo : T Media, 2005. - ISBN 972-99480-1-1..

MARTZOUKOU K. A review of Web information seeking research: considerations of method and foci of interest [Online] // Information Research. - 2004. - 11 de Julho de 2006. - <http://InformationR.net/ir/10-2/paper215.html>.

MCGOVERN Gerry, NORTON Rob e O'DOWD Catherine Como Escrever para a Web [Livro]. - [s.l.] : Editora Centro Atlântico, 2002.

MCT Livro Verde para a Sociedade Da Informação em Portugal [Online] // Missão para a Sociedade da Informação / Ministério da Ciência e da Tecnologia. - 2000. - 24 de 4 de 2009. - <http://www.posc.mctes.pt/documentos/pdf/LivroVerde.pdf>.

- MCT Portugal** na Sociedade de Informação [Online] // Site da MCT. - 1999. - 24 de Abril de 2009. - http://www.inst-informatica.pt/servicos/informacao-e-documentacao/biblioteca-digital/sociedade-da-informacao-1/2005-e-anos-anteriores/Portugal_na_SI.pdf/at_download/file.
- MEDEIROS Maysa Regina e CLARO Daniela Barreiro** Conhecendo ASP, PHP e JSP [Online]. - 2002. - 7 de 12 de 2009. - http://nuclinfo.famato.org.br/down/poster_AI_02.pdf.
- MEMÓRIA Filipe** Design para a Internet [Livro]. - São Paulo : Editora Campus, 2006.
- MEXIA João Tiago e RODRIGUES Paulo Canas** Estatística Matemática - Apontamentos teóricos [Online]. - Universidade Nova de Lisboa, 2005. - 14 de 12 de 2009. - <http://ferrari.dmat.fct.unl.pt/services/EstatMat/>.
- MOBCOMTECH SMS** (Short Message System) Mobile Technology, International [Online] // mobilecomms-technology.com. - 24 de Abril de 2009. - <http://www.mobilecomms-technology.com/projects/sms/>.
- MOBILECOMMS SMS** (Short Message System) Mobile Technology, International [Online] // mobilecomms-technology.com. - 1992. - 25 de 01 de 2010. - <http://www.mobilecomms-technology.com/projects/sms/>.
- MONTEIRO Edmundo e BOAVIDA Fernando** Engenharia de Redes Informáticas [Livro]. - [s.l.] : FCA, 2000.
- MOODY Daniel e Walsh Peter** Measuring The Value Of Information: An Asset Valuation Approach [Online]. - 1999. - 25 de 05 de 2009. - <http://www.info.deis.unical.it/~zumpano/2004-2005/PSI/lezione2/ValueOfInformation.pdf>.
- MYSQL** Chapter 13. Storage Engines [Online] // dev.mysql.com. - 2009. - 20 de 12 de 2009. - <http://dev.mysql.com/doc/refman/5.1/en/storage-engines.html>.
- NAHUZ Fernanda** World Wide Web: Aspectos Teóricos dos Mecanismos de Busca [Online] // Informação e Sociedade: estudos. - 2000. - 11 de Julho de 2006. - em <http://www.informacaoesociedade.ufpb.br/pdf/IS929901.pdf>.
- NETCRAFT** November 2009 Web Server Survey [Online] // NetCraft. - 10 de 11 de 2009. - 07 de 12 de 2009. - http://news.netcraft.com/archives/2009/11/10/november_2009_web_server_survey.html.
- NIEDERAUER Juliano** Desenvolvendo Websites com PHP [Online]. - 2007. - 7 de 12 de 2009. - <http://www.novateceditora.com.br/livros/php/capitulo8575220500.pdf>.
- NIELSEN J.** Top Ten Web Design Mistakes of 2005 [Online] // Useit. - 2005. - 11 de Julho de 2006. - <http://www.useit.com/alertbox/designmistakes.html>.
- NOTTINGHAM Mark** RSS Tutorial [Online]. - 28 de Abril de 2009. - <http://www.mnot.net/rss/tutorial/>.
- NUNES Mauro e O'NEILL Henrique** Fundamental de UML [Livro]. - Lisboa : FCA, 2004. - 5ª Edição.
- O'DOCHERTY Mike** Object-Oriented Analysis & Design - Understanding system development with UML 2.0 [Livro]. - [s.l.] : John Wiley & Sons, Ltd, 2005.

OLA Andersson e LARSSON Niclas RSS – The future of internal [Online]. - Vaxjo University, 2005. - 24 de 06 de 2009. - <http://vxu.diva-portal.org/smash/record.jsf?pid=diva2:206676>.

OLIVEIRA Érica e NORONHA Daisy A Internet Como Canal de Comunicação Científica [Online] // Informação e Sociedade. - 2001. - 11 de Julho de 2006. - <http://www.informacaoesociedade.ufpb.br/pdf/IS1210212.pdf>.

OLIVER Kevin M., WILKINSON Gene L. e BENNETT Lisa T. Evaluating the Quality of Internet Information Sources [Online] // Education Resources Information Center. - 1997. - 11 de Julho de 2006. - http://eric.ed.gov/ERICDocs/data/ericdocs2/content_storage_01/0000000b/80/22/df/29.pdf.

ORGERA Scott What Is A Web Browser? [Online] // About.com. - 29 de 10 de 2009. - 08 de 12 de 2009. - <http://browsers.about.com/od/howbrowserswork/a/whatisabrowser.htm>.

PETERS Oscar [et al.] Motives for SMS Use [Online] // 53rd Annual Conference of the International Communication Association. - 04 de 2003. - 25 de 05 de 2009. - <http://doc.utwente.nl/59780/>.

PHP Introduction - Manual [Online] // PHP.NET. - 4 de 12 de 2009. - 7 de 12 de 2009. - <http://www.php.net/manual/en/introduction.php>.

PHP-Stats PHP Usage Stats for April 2007 [Online] // PHP.NET. - 2007. - 07 de 12 de 2009. - <http://www.php.net/usage.php>.

PINTO Sandra Marques, CALIXTO José António e LOPES Pedro Faria Gestores de Recursos Electrónicos: Que formação na era digital? [Online] // BAD - Associação Portuguesa de Bibliotecários, Arquivistas e Documentalistas. - 2004. - 24 de 04 de 2009. - <http://badinfo.apbad.pt/congresso8/com31.pdf>.

PIRES Carlos, NASCIMENTO Rilson e SALGADO Ana Comparativo de Desempenho entre Bancos de Dados de [Online]. - Universidade Federal de Pernambuco, 2005. - 08 de 12 de 2009. - http://www.itautec.com.br/iFileExplorer/Arquivo/Empresa/Documentos/Comparativo_%20Bancos%20de%20dados_%20SW%20Livre.pdf.

POLLOCK Annabel e HOCKLEY, Andrew What's Wrong with Internet Searching [Online] // D-Lib Magazine. - 1997. - 11 de Julho de 2007. - <http://www.dlib.org/dlib/march97/bt/03pollock.html>.

PORTER John Douglas Camel Case [Online]. - Cunningham & Cunningham, Inc., 22 de 05 de 2008. - 12 de 12 de 2009. - <http://c2.com/cgi/wiki?CamelCase>.

PROTALINSKI Emil October 2009 browser stats: Firefox finally passes IE6 [Online] // ARS Technica. - 10 de 2009. - 08 de 12 de 2009. - <http://arstechnica.com/microsoft/news/2009/11/october-2009-browser-stats-firefox-finally-passes-ie6.ars>.

RAMALHO José C. e HENRIQUES Pedro XML & XSL - da teoria à prática [Livro]. - [s.l.] : FCA, 2002.

RAY Kathryn e Day Joan Student attitudes towards electronic information resources [Online] // Information Research. - 1998. - 11 de Julho de 2006. - <http://informationr.net/ir/4-2/paper54.html>.

- REIS António Ribas** A Informação e as Instituições de Ensino Superior [Online] // Encontro Latino de Economia Política da Informação, Comunicação e Cultura. - 2005. - 25 de 05 de 2009. - <http://www.rp-bahia.com.br/biblioteca/pdf/AntonioRibasReis.pdf>.
- RETFORD Brian e SCHWARTZ Jordan** How to Build an SMS Service [Online]. - O'Reilly, 2007. - 19 de 05 de 2009. - http://books.google.pt/books?id=9qI_nY52tDQC&printsec=frontcover.
- SEGUY Damien** PHP stats evolution for October 2008 [Online]. - 10 de 2008. - 7 de 12 de 2009. - http://www.nexen.net/chiffres_cles/phpversion/18821-php_stats_evolution_for_october_2008.php.
- SERRÃO João e MARQUES Joaquim** Programação com PHP 4 [Livro]. - [s.l.] : Editora FCA, 2001.
- SILVA Armando Malheiro e RIBEIRO Fernanda** Formação, perfil e competências do profissional da Informação [Online] // VIII Congresso da BAD. - 2004. - 11 de Julho de 2006. - <http://sapp.telepac.pt/apbad/congresso8/com16.pdf>.
- SILVA Edna Lúcia da, MENEZES Estera Muszkat e BISSANI Márcia** A internet como canal de comunicação científica [Online] // Informação & Sociedade: Estudos. - 2002. - 25 de 05 de 2009. - <http://www.ies.ufpb.br/ojs2/index.php/ies/article/view/163/157>.
- SWFOBJECT** [Online]. - 2009. - 2009. - <http://code.google.com/p/swfobject/>.
- SYNDIC8** [Online]. - 26 de 05 de 2009. - 26 de 05 de 2009. - <http://www.syndic8.com/stats.php?Section=feeds#tabtable>.
- TILLMAN Hope** Evaluating Quality on the Net [Online]. - Harvard University, 1995. - 11 de 07 de 2006. - <http://www.hopetillman.com/findqual.html>.
- TIOBE** TIOBE Programming Community Index for December 2009 [Online] // TIOBE Software. - 12 de 2009. - 12 de 2009. - <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>.
- TOMAÉL M. I. [et al.]** Avaliação de fontes de informação na Internet: critérios de qualidade [Online] // Informação & Sociedade. - 2001. - 11 de 07 de 2006. - <http://www.informacaosociedade.ufpb.br/pdf/IS1120101.pdf>.
- UE** eEurope 2005, SCADPlus: eEurope2005 [Online] // União Europeia. - 2005. - 24 de 04 de 2009. - <http://www.eu.int/scadplus/leg/pt/lvb/l24226.htm>.
- UE** Internet access and use in the EU27 in 2008 [Online] // Europa.eu. - 02 de 12 de 2008. - 29 de 05 de 2009. - <http://europa.eu/rapid/pressReleasesAction.do?reference=STAT/08/169&format=>.
- UGHETTO Vico** Criação Inovadora de Sites [Livro]. - [s.l.] : Editora FCA, 2006.
- UMIC** Ligar Portugal [Online] // Site de Ligar Portugal. - 2005. - Julho de 2006. - <http://www.ligarportugal.pt>.
- UMIC** Plano de Acção para a Sociedade da Informação [Online]. - 2003. - 24 de 4 de 2009. - http://www.inst-informatica.pt/servicos/informacao-e-documentacao/biblioteca-digital/sociedade-da-informacao-1/2005-e-anos-anteriores/I_Plano_Accao_SI.pdf/at_download/file.

- VIEIRA João** Programação com Active Server Pages [Livro]. - [s.l.] : FCA, 2001. - 2ª Edição.
- VIEIRA João** Programação com ASP.NET [Livro]. - [s.l.] : FCA, 2004. - 2ª Edição : Vol. Volume I e II.
- W3C** Cascading Style Sheets, level 2 CSS2 Specification [Online] // w3.org. - W3C, 12 de 5 de 1998. - 08 de 12 de 2009. - <http://www.w3.org/TR/2008/REC-CSS2-20080411/>.
- W3C** HTML 4.01 Specification [Online] // w3.org. - 24 de 12 de 2004. - 08 de 12 de 2009. - <http://www.w3.org/TR/1999/REC-html401-19991224/>.
- W3C** SCRIPTING AND AJAX [Online] // w3.org. - 12 de 2009. - 08 de 12 de 2009. - <http://www.w3.org/standards/webdesign/script>.
- W3C** Tim Berners-Lee [Online]. - 2009. - 23 de 06 de 2009. - <http://www.w3.org/People/Berners-Lee/>.
- W3C** Web Content Accessibility Guidelines 2.0 [Online]. - 11 de 12 de 2008. - 13 de 12 de 2009. - <http://www.w3.org/TR/WCAG20/>.
- W3Counter** W3Counter [Online]. - 08 de 12 de 2009. - 11 de 2009. - <http://www.w3counter.com/globalstats.php>.
- WILDEROM Bastiaan Pieter Marinus e SANTOS José Cícero dos** Aplicações Web Dinâmicas com ASP.NET [Livro]. - [s.l.] : Editora Érica, 2005.
- WILLIAMS Hugh e LANE David** PHP and MySQL [Livro]. - [s.l.] : O'Reilly, 2004.
- WINER DAve** [Online] // Scripting News. - 28 de 07 de 1999. - 26 de 05 de 2009. - <http://archive.scripting.com/1999/07/28>.
- Winer Dave** scriptingNews format [Online] // Scripting. - 15 de 12 de 1997. - 26 de 05 de 2009. - <http://davenet.scripting.com/1997/12/15/scriptingNewsInXML>.
- YAMAOKA Eloi** Recuperação da informação na Web [Online]. - 2003. - 29 de Abril de 2009. - <http://www.scribd.com/doc/379383/RECUPERACAO-DA-INFORMACAO-NA-WEB> .
- YU Holly** Library Web Content Management: Needs and Challenges [Online]. - California State University, 2005. - 05 de 01 de 2009. - <http://www.idea-group.com/downloads/excerpts/1591405335.pdf>.

Apêndices

A1. Diferentes Especificações de Feeds

Neste apêndice procuramos demonstrar as diferenças entre algumas das versões / especificações da syndicação de conteúdos.

Especificação RSS 0.91 - Netscape/UserLand

```
<?xml version="1.0"?>
<rss version="0.91">
  <channel>
    <title>WriteTheWeb</title>
    <link>http://writetheweb.com</link>
    <description>News for web users that write back</description>
    <language>en-us</language>
    <copyright>Copyright 2000, WriteTheWeb team.</copyright>
    <managingEditor>editor@writetheweb.com</managingEditor>
    <webMaster>webmaster@writetheweb.com</webMaster>
    <item>
      <title>Giving the world a pluggable Gnutella</title>
      <link>http://writetheweb.com/read.php?item=24</link>
      <description>WorldOS is a framework on which to build programs that work like
Freenet or Gnutella -allowing distributed applications using peer-to-peer routing.</description>
    </item>
    <item>
      <title>Syndication discussions hot up</title>
      <link>http://writetheweb.com/read.php?item=23</link>
      <description>After a period of dormancy, the Syndication mailing list has become
active again, with contributions from leaders in traditional media and Web syndication.</description>
    </item>
  </channel>
</rss>
```

Especificação RSS 1.0 - RSS-DEV Working Group

```
<?xml version="1.0"?>

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns="http://purl.org/rss/1.0/">
  <channel rdf:about="http://www.xml.com/xml/news.rss">
    <title>XML.com</title>
    <link>http://xml.com/pub</link>
    <description>
      XML.com features a rich mix of information and services
      for the XML community.
    </description>
    <image rdf:resource="http://xml.com/universal/images/xml_tiny.gif" />
    <items>
      <rdf:Seq>
        <rdf:li resource="http://xml.com/pub/2000/08/09/xslt/xslt.html" />
        <rdf:li resource="http://xml.com/pub/2000/08/09/rdfdb/index.html" />
      </rdf:Seq>
    </items>
  </channel>

  <image rdf:about="http://xml.com/universal/images/xml_tiny.gif">
    <title>XML.com</title>
    <link>http://www.xml.com</link>
    <url>http://xml.com/universal/images/xml_tiny.gif</url>
  </image>

  <item rdf:about="http://xml.com/pub/2000/08/09/xslt/xslt.html">
    <title>Processing Inclusions with XSLT</title>
    <link>http://xml.com/pub/2000/08/09/xslt/xslt.html</link>
    <description>
      Processing document inclusions with general XML tools can be
      problematic. This article proposes a way of preserving inclusion
      information through SAX-based processing.
    </description>
  </item>

  <item rdf:about="http://xml.com/pub/2000/08/09/rdfdb/index.html">
    <title>Putting RDF to Work</title>
    <link>http://xml.com/pub/2000/08/09/rdfdb/index.html</link>
    <description>
      Tool and API support for the Resource Description Framework
      is slowly coming of age. Edd Dumbill takes a look at RDFDB,
      one of the most exciting new RDF toolkits.
    </description>
  </item>
</rdf:RDF>
```

Especificação RSS 2.0 - Userland / Harvard

```
<?xml version="1.0"?>
<rss version="2.0">
  <channel>
    <title>Liftoff News</title>
    <link>http://liftoff.msfc.nasa.gov/</link>
    <description>Liftoff to Space Exploration.</description>
    <language>en-us</language>
    <pubDate>Tue, 10 Jun 2003 04:00:00 GMT</pubDate>
    <lastBuildDate>Tue, 10 Jun 2003 09:41:01 GMT</lastBuildDate>
    <docs>http://blogs.law.harvard.edu/tech/rss</docs>
    <generator>Weblog Editor 2.0</generator>
    <managingEditor>editor@example.com</managingEditor>
    <webMaster>webmaster@example.com</webMaster>
    <item>
      <title>The Engine That Does More</title>
      <link>http://liftoff.msfc.nasa.gov/news/2003/news-VASIMR.asp</link>
      <description>Before man travels to Mars, NASA hopes to design new engines that will let us fly
through the Solar System more quickly. The proposed VASIMR engine would do that.</description>
      <pubDate>Tue, 27 May 2003 08:37:32 GMT</pubDate>
      <guid>http://liftoff.msfc.nasa.gov/2003/05/27.html#item571</guid>
    </item>
    <item>
      <title>Astronauts' Dirty Laundry</title>
      <link>http://liftoff.msfc.nasa.gov/news/2003/news-laundry.asp</link>
      <description>Compared to earlier spacecraft, the International Space Station has many luxuries,
but laundry facilities are not one of them. Instead, astronauts have other options.</description>
      <pubDate>Tue, 20 May 2003 08:56:02 GMT</pubDate>
      <guid>http://liftoff.msfc.nasa.gov/2003/05/20.html#item570</guid>
    </item>
  </channel>
</rss>
```

(HARVARD, 2004)

Especificação Atom

```
<?xml version="1.0"?>
<feed xmlns="http://www.w3.org/2005/Atom">

  <title>Example Feed</title>
  <link href="http://example.org"/>
  <updated>2003-12-13T18:30:02Z</updated>
  <author>
    <name>John Doe</name>
  </author>
  <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>

  <entry>
    <title>Atom-Powered Robots Run Amok</title>
    <link href="http://example.org/2003/12/13/atom03"/>
    <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
    <updated>2003-12-13T18:30:02Z</updated>
    <summary>Some text.</summary>
  </entry>

  <entry>
    <title>Atom-Powered Robots Run Amok</title>
    <link href="http://example.org/2003/12/13/atom03"/>
    <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
    <updated>2003-12-13T18:30:02Z</updated>
    <summary>Some text.</summary>
  </entry>

</feed>
```

(AtomEnabled, 2004)

A2. Questionário inicial e final

Questionário

Primeira Parte – Pré-utilização do protótipo

1. Sexo:
 Masculino Feminino
2. Idade: _____
3. Como considera o seu nível de conhecimentos informáticos:
 Fraco Médio Bom
4. Formação escolar:
 4ª Classe 7º Ano 9º ano 12º ano
 Bacharelato Licenciatura Mestrado Doutoramento
 Outro:

5. Indique quais as formas de acesso à Internet que utiliza (resposta múltipla):
 Cabo/ADSL Placa 3G WI-FI/Hotspot 3G/GPRS no
Telemóvel
 NS/NR (Não sabe/não responde) Outro:

6. Locais em que acede à Internet (resposta múltipla):
 Casa Trabalho/Escola Hotspot
Cybercafé
 em qualquer lugar (acesso móvel) NS/NR Outro:

7. Que dispositivos costuma utilizar para aceder à Internet (resposta múltipla):
 Computador PDA Telemóvel Leitor
Mp3/Multimédia
 Consola Jogos Outro:

8. Que dispositivos costuma levar consigo para qualquer local (trabalho, viagem ,
praia, cinema, etc.) (resposta múltipla):
 Computador PDA Telemóvel Leitor
MP3/Multimédia

Consola Jogos Outro:

9. Indique o **número** de horas diárias que dispense na consulta de sites:

10. Indique o **número** de sites que visita com regularidade diariamente:

11. Como considera o processo de pesquisa de informação (pertinente para si) nestes sites: **Fácil** de encontrar **Médio** – tenho que pesquisar e filtrar **Difícil** – tenho que pesquisar e filtrar muito para encontrar o que procuro

12. Costuma subscrever notificações automáticas de conteúdos de algum site?

Sim Não NS/NR

NOTA: Se respondeu Não ou NS/NR prossiga para ao ponto 17.

13. Se respondeu “Sim” à questão 12 que forma recebe as notificações?

(se respondeu Não ou NS/NR prossiga para o ponto 17)

Através de *Software* para o efeito (Outlook, leitor de *feeds*, *widgets*, *plugins*, etc.)

Email SMS Outro, especificar

14. Qual o **tempo** de leitura médio (**em horas**) entre o envio da notificação até a sua leitura/conhecimento? _____

15. Como considera o tempo decorrido entre o envio da notificação e a sua leitura?

Fraco Médio Excelente

16. Gostaria de receber notificações de forma mais rápida?

Sim Não

17. Gostaria de poder receber notificações automáticas de conteúdos de algum site, em qualquer lugar?

Sim Não

18. Conhece algum dos seguintes serviços de sindiciação de conteúdos (*feed*):

RSS/ATOM/RDF/PODCAST

Sim Não

19. Dos sites que visita regularmente algum possui um ou mais serviços descritos no ponto anterior?
- Nenhum tem Alguns têm A maioria tem Todos têm
20. Utiliza (ou já utilizou) os serviços descritos no ponto 18?
- Sim Não
21. Se respondeu “Sim” na ponto anterior, em que dispositivos costuma usar:
- Computador PDA Telemóvel Leitor
MP3/Multimédia
- Consola Jogos Outro:
-
22. Costuma utilizar as SMS (serviço de mensagens escritas) com regularidade no seu telemóvel?
- Sim Não NS/NR
23. Como considera o seu grau de conhecimento sobre o envio e recepção de SMS:
- Fraco Médio Excelente NS/NR
24. Gostaria de receber notificações por SMS de actualizações de conteúdos de um determinado site do seu interesse?
- Sim Não
25. Se respondeu sim ao ponto 16, considera que a notificação por SMS melhoraria o tempo entre o envio e o seu conhecimento / leitura?
- Sim Não

Parte 2 – Após utilização do protótipo

Responda às seguintes questões indicando de 0 (muito mau) a 5 (muito bom).

26. Como considera a usabilidade (facilidade de utilização / interacção) do site?
 1 2 3 4 5
27. Como considera a estrutura de informação (organização) do site?
 1 2 3 4 5
28. Como considera o aspecto visual do site?
 1 2 3 4 5
29. Como classifica a facilidade de criar o seu perfil?
 1 2 3 4 5
30. Como classifica o processo de adicionar um novo *feed* ao seu perfil?
 1 2 3 4 5
31. Como classifica o processo de gestão de *feeds* no seu perfil?
 1 2 3 4 5
32. Como classifica o processo de filtragem de conteúdos?
 1 2 3 4 5
33. Como considera o conteúdo recebido em cada SMS, em termos de suficiência para o seu entendimento / leitura, face ao limite de 160 caracteres?
 1 2 3 4 5
34. Como classifica as abreviaturas usadas no SMS, em termos de entendimento/leitura?
 1 2 3 4 5
35. Como considera o seu grau de satisfação global relativamente ao serviço testado?
 1 2 3 4 5

O seu inquérito termina aqui, obrigado.

A3. Resultados da validação de acessibilidade

acesso > **Examinator**

Resultados do Validador eXaminator da [UMIC – Agência para a Sociedade do Conhecimento, IP.](#)

notas técnicas: [validador eXaminator](#) | [índice web@x](#) | [logótipo dinâmico "Certified Accessibility"](#).

Examinator

77 elementos em <http://r2s.ath.cx/tese/index.php>

A calcular os resultados. (cálculos efectuados com sucesso)

A efectuar ligação a <http://validator.w3.org> para verificar o código (X)HTML. (o eXaminator conseguiu estabelecer ligação ao validador de (X)HTML do W3C)

A efectuar ligação a <http://jigsaw.w3.org/css-validator> para verificar as folhas de estilo CSS. (o eXaminator conseguiu estabelecer ligação ao validador de CSS do W3C)

Esta página passa a bateria de testes do eXaminator para as prioridades 1, 2 e 3 (conformidade "aaa")

Índice web@X (0-10): 10.0
(o índice deve ser usado apenas como referência dos responsáveis pelos conteúdos do sítio web)

Ver relatório qualitativo

O índice web@X resulta da utilização do validador de Acessibilidade Web eXaminator.

Última actualização: 20 Setembro de 2009.
Programa ACESSO da UMIC - Agência para a Sociedade do Conhecimento, I.P.

Relatório qualitativo:

RSS2SMS - Receba notificações por SMS de feeds RSS
<http://r2s.ath.cx/tese/index.php>

Resultado da análise de 77 elementos








Índice Web@X (0-10): 10.0 (o índice deve ser usado apenas como referência dos responsáveis pelos conteúdos do sítio Web)








Esta página passa a bateria de testes do eXaminator para as prioridades 1, 2 e 3 (conformidade "aaa")

Conformidade	Testes ok / Testes aplicados
Testes de prioridade 1	2 / 2
Testes de prioridade 2	13 / 13
Testes de prioridade 3	3 / 3

Não deixe de rever manualmente as 7 ocorrências localizadas na página.

Testes de prioridade 1

<p>Texto alternativo nos botões gráficos.</p>	 <p>Correcto! Há 1 botão(ões) gráfico(s) e todos têm o atributo "alt". Deve verificar se os textos alternativos afixados se encontram correctos.</p> <p>Todos os botões gráficos dos formulários (<input type="image">) devem ter o atributo "alt" com uma legenda que explicita a função que cumpre. Na generalidade dos casos a legenda deve ser igual ao lettering gráfico afixado no botão.</p> <p>Forneça um equivalente textual para todo o elemento não textual. Pode ser feito através do atributo "alt", ou "longdesc" ou no conteúdo do elemento. (Prioridade 1) (Exemplo)</p>
<p>Conteúdo alternativo para scripts.</p>	 <p>Há 1 elemento(s) <script> no corpo da página (<body>) e todos eles se encontram acompanhados pelo seu correspondente <noscript>. Verifique o conteúdo alternativo.</p> <p>Os <script> incluídos no corpo do documento devem ter conteúdos alternativos em <noscript> que descrevam a sua acção ou constituam mesmo um substituto da função desenvolvida em <script>. Se a função do <script> depende da posição que ocupa no <body> então é seguro que esse <script> necessita de um equivalente alternativo em <noscript> e que substitua a função <script> exactamente na mesma posição da página. Se a localização do <script> é indiferente, verifique se o mesmo não poderá ser colocado no <head> da página em vez de no <body>.</p> <p>Forneça um equivalente textual para todo o elemento não textual. Pode ser feito através do atributo "alt", ou "longdesc" ou no conteúdo do elemento. (Prioridade 1) (Exemplo)</p>
<p>Testes de prioridade 2</p>	
<p>Declaração do tipo de documento/Sintaxe.</p>	 <p>Correcto! Há uma declaração do tipo de documento (DTD).</p> <p>Uma DTD indica a sintaxe/gramática utilizada no código fonte da página e permite verificar a correcta aplicação desse código - validar a assertividade de escrita do código.</p> <p>Crie documentos validando a notação com a gramática formal publicada. (Prioridade 2) (Exemplo)</p>
<p>Elementos obsoletos.</p>	 <p>Correcto! Não se usam elementos obsoletos em HTML 4.01 (<applet>, <center> ou , por exemplo).</p> <p>Os elementos obsoletos são construções que foram sendo substituídas por outras mais apropriadas e devem evitar-se porque podem provocar problemas de acessibilidade. Por outro lado, a sua utilização evidencia uma falta de actualização na área do desenho <i>Web</i> por parte do profissional que o realizou.</p> <p>Evite o uso de notação obsoleta das tecnologias do W3C. (Prioridade 2) (Exemplo)</p>
<p>Atributos obsoletos.</p>	 <p>Correcto! Não se usam atributos obsoletos em HTML 4.01 ("hspace", "bgcolor" ou "nowrap", por exemplo).</p> <p>Os atributos obsoletos são construções que foram descontinuadas e progressivamente substituídas por outras mais apropriadas, nomeadamente através do uso de CSS, pelo que devem evitar-se, senão mesmo eliminar-se, por poderem provocar problemas de acessibilidade. Por outro lado, a sua utilização revela falta de actualização por parte do profissional <i>Web</i> que a realizou.</p> <p>Evite o uso de notação obsoleta das tecnologias do W3C. (Prioridade 2) (Exemplo)</p>
<p>Destino dos links.</p>	 <p>O objectivo de cada link parece estar claramente identificado.</p> <p>Os links devem indicar de maneira muito clara e sucinta para onde apontam, de forma a que os utilizadores possam saber o que encontrarão se decidirem escolher esse link.</p> <p>Identifique claramente o destino de cada <i>link</i>. (Prioridade 2) (Exemplo)</p>
<p>Associação de etiquetas com controlos.</p>	 <p>Correcto! Há 2 etiqueta(s) (<label>) associada(s) correctamente com o(s) respectivo(s) controlo(s) de formulário.</p>

	<p>O atributo "for" de cada etiqueta deve coincidir com o atributo "id" do controlo de formulário que identifica. Isto é especialmente útil para os utilizadores que utilizam leitores de ecrã para navegar. Associe explicitamente os rótulos aos respectivos controlos. (Prioridade 2)(Exemplo)</p>
Utilização de cabeçalhos.	<p> Correcto! Usam-se cabeçalhos (<h1>~<h6>). Devem-se marcar correctamente os cabeçalhos da página e não mesclar os seus níveis aleatoriamente apenas para fazer uso dos estilos (tamanhos de fonte) que estão, por defeito, associados aos cabeçalhos. Use CSS para formatar a cor, o tamanho e o tipo de letra dos cabeçalhos. Actualmente eles constituem um importante elemento facilitador da navegação por todos aqueles que fazem uso de tecnologias de apoio. Use os elementos cabeçalho (<H1>...<H6>) para transmitir a estrutura dos documentos e utilize-os de acordo com as especificações. (Prioridade 2)(Exemplo)</p>
Validação do código (X)HTML.	<p> Correcto! O código da página está correcto. É importante evitar os erros de sintaxe no código fonte para que a página seja correctamente interpretada pelos agentes de utilizador, nos quais se incluem as ajudas técnicas. Validator de (X)HTML do W3C Crie documentos validando a notação com a gramática formal publicada. (Prioridade 2) (Exemplo)</p>
Validação do código das folhas de estilo CSS.	<p> Correcto! O código das CSS está correcto. É importante evitar os erros de sintaxe das folhas de estilo para permitir a sua correcta interpretação. Validator de CSS do W3C Crie documentos validando a notação com a gramática formal publicada. (Prioridade 2) (Exemplo)</p>
Elementos HTML de apresentação.	<p> Correcto! Não se usam elementos HTML de apresentação como , <center> ou para controlar a apresentação da página. É incorrecto utilizar elementos da linguagem de programação (X)HTML para controlar a forma/estilo de apresentação da página. Use (X)HTML para estruturar/suportar o conteúdo e as folhas de estilo CSS para controlar o estilo da página. Use folhas de estilo para controlar a disposição dos elementos na página e a forma de os apresentar. (Prioridade 2) (Exemplo)</p>
Atributos de apresentação.	<p> Correcto! Não se usam atributos como "size", "color" ou "face" para controlar a apresentação da página. É incorrecto utilizar atributos da linguagem de programação (X)HTML para controlar a forma/estilo de apresentação da página. Use (X)HTML para estruturar/suportar o conteúdo e as folhas de estilo CSS para controlar o estilo da página. Use folhas de estilo para controlar a disposição dos elementos na página e a forma de os apresentar. (Prioridade 2) (Exemplo)</p>
Utilização de medidas absolutas.	<p> Correcto! Não se usam medidas absolutas. Se se utilizam unidades de medida relativas, e não absolutas, os utilizadores poderão ampliar ou reduzir o tamanho dos elementos da página de acordo com as suas necessidades. Por outro lado, as páginas adequam-se a ecrãs com diferentes resoluções, redimensionando automaticamente os seus elementos. Use unidades relativas em vez de absolutas nos valores dos atributos da linguagem de notação e valores das propriedades das folhas de estilo. (Prioridade 2)(Exemplo)</p>
Utilização de CSS.	<p> Correcto! Usam-se estilos para controlar a apresentação da página.</p>

As folhas de estilo representam um grande passo em frente para a *Web* uma vez que a sua utilização permite separar o conteúdo da apresentação das páginas.
Use folhas de estilo para controlar a disposição dos elementos na página e a forma de os apresentar. **(Prioridade 2)** (Exemplo)

Utilização de tabelas para maquetar a página (tabelalayout).



Correcto! Não se usam tabelas para maquetar a página.

As tabelas são elementos para apresentar conteúdo tabular e não devem utilizar-se para apresentar outro tipo de conteúdo. Isto é especialmente importante para quem utiliza navegadores de base texto ou leitores que lêem linha a linha os dados existentes no ecrã.
Não deve usar tabelas para formatar páginas a não ser que a tabela faça sentido quando em formato linear. Caso contrário, se a tabela não fizer sentido, forneça um equivalente alternativo (o qual poderá ser uma versão linear). **(Prioridade 2)**(Exemplo)

Testes de prioridade 3

Caracteres por defeito.



Correcto! Há 1 controlo(s) vazio(s) com caracteres por omissão.

Todos os campos de edição e áreas de texto devem conter algum carácter porque existem ajudas técnicas que não podem colocar o foco nos controlos vazios.
Até que os agentes do utilizador consigam manipular controlos vazios correctamente, inclua caracteres predefinidos de preenchimento nas caixas de edição e nas áreas de texto. **(Prioridade 3)** (Exemplo)

Identificação do idioma principal da página.



Correcto! É especificado "PT" como idioma principal. Verifique se o código se encontra correcto.

Em todas as páginas deve-se indicar o idioma principal do documento. Se o documento estiver em XHTML, deve-se-á verificar que, para além do atributo "lang", se utiliza também "xml:lang".
Identifique o idioma principal do documento. **(Prioridade 3)** (Exemplo)

Links adjacentes.



Correcto! Há 8 link(s) e todos estão separados por caracteres imprimíveis.

Alguns agentes de utilizador não conseguem diferenciar os links se estes se encontrarem colados uns aos outros, sem qualquer marca ou carácter imprimível não "linkado" a separá-los. Algo do tipo: link1 | link2 | link3, em que | é um carácter sem ligação.
Até que os agentes do utilizador consigam distinguir *links* adjacentes, inclua caracteres não-linkados, circundados por espaços, entre os links adjacentes. **(Prioridade 3)**(Exemplo)

Verificações manuais a empreender

Obstáculos à acessibilidade (Testes de prioridade 2)

Título da página.



A página tem um título que deve verificar manualmente. É o texto "RSS2SMS - Receba notificações por SMS de feeds RSS" adequado como título para esta página?

Deve existir um título para a página, através do elemento <title>, e o mesmo deve ser claro, descritivo e conciso.
Forneça metadados para acrescentar informações semânticas às páginas e aos sítios *Web*. **(Prioridade 2)** (Exemplo)

Blocos de informação.



Há uma média de 6 palavras em cada bloco de informação.

Estruture e segmente correctamente os textos utilizando títulos, subtítulos, parágrafos e listas. Lembre-se

que os blocos de informação demasiado grandes dificultam a sua compreensão.
Divida grandes blocos de informação em grupos mais geríveis e apropriados. **(Prioridade 2)**(Exemplo)

**Versão
(X)HTML.**



Versão XHTML.

É importante usar as tecnologias do W3C quando as mesmas já se encontram disponíveis (alvo de recomendação final) e sejam suportadas, nomeadamente por um vasto leque de agentes de utilizador (navegadores *Web*, tecnologias de apoio, etc). Recomenda-se o uso das tecnologias do W3C, pela sua aceitação generalizada em todo o mundo e também pelo facto de as mesmas incorporarem as questões de acessibilidade desde a concepção.

Use tecnologias W3C quando a mesma esteja disponível e seja apropriada para uma tarefa. Utilize as versões mais recentes, desde suportadas. **(Prioridad 2)** (Exemplo)

**Etiquetas
para contro-
los de formu-
lário.**



Há 2 etiqueta(s) (<label>) posicionadas adequadamente.

Uma vez que alguns agentes de utilizador/aplicações não suportam associações explícitas entre as etiquetas e os controlos de formulário, a etiqueta deve rodear o controlo que identifica.

Até que os agentes do utilizador suportem associações explícitas entre os rótulos e os controlos de formulário, para todos os controlos com rótulos implicitamente associados, certifique-se que os rótulos se encontram apropriadamente posicionados. **(Prioridade 2)**(Exemplo)

**Contraste
das cores.**



As definições de cores de fundo e de primeiro plano (caracteres) foram localizadas nas CSS. Deverá verificar o contraste das cores existente nas imagens).

É importante que os utilizadores com baixa visão possam desactivar as cores presentes na página para evitar as combinações de cor com contraste inadequados para si. A fixação das cores no código HTML dificulta esta personalização. Todas as definições de cor devem ser feitas com CSS.

Certifique-se que as combinações das cores de fundo e do texto fornecem um contraste suficiente quando visualizados por alguém que tenha défices de percepção de cor ou quando a mesma é visualizada num ecrã a preto e branco. **(Prioridade 2 para imagens, 3 para textos)** (Exemplo)

Ruídos à acessibilidade (Testes de prioridade 3)

**Informação
sobre docu-
mentos rela-
cionados.**



Correcto! Utilizam-se 1 atributo(s) "accesskey".

Os atalhos de teclado permitem, a quem utiliza o teclado para navegar, aceder rapidamente aos elementos mais importantes da página. **É particularmente importante colocar uma tecla de atalho para acesso ao campo de edição do motor de busca.**

Defina teclas de atalho para links importantes (incluindo os que se encontram nos mapas de imagem client-side), controlos de formulário, e grupos de controlos de formulários.**(Prioridade 3)** (Exemplo)

**Utilização de
accesskey.**



Não se utiliza o atributo "accesskey".

Os atalhos de teclado permitem, a quem utiliza o teclado para navegar, aceder rapidamente aos elementos mais importantes da página. **É particularmente importante colocar uma tecla de atalho para acesso ao campo de edição do motor de busca.**

Defina teclas de atalho para *links* importantes (incluindo os que se encontram nos mapas de imagem client-side), controlos de formulário, e grupos de controlos de formulários.**(Prioridade 3)** (Exemplo)



Markup Validation Service

Check the markup (HTML, XHTML, ...) of Web documents

Jump To: [Congratulations - Icons](#)

This document was successfully checked as XHTML 1.0 Transitional!

Result:	Passed	
Address:	<input type="text" value="http://r2s.ath.cx/tese/"/>	
Encoding:	iso-8859-1	<input type="button" value="(detect automatically)"/>
Doctype:	XHTML 1.0 Transitional	<input type="button" value="(detect automatically)"/>
Root Element:	html	
Root Namespace:	http://www.w3.org/1999/xhtml	



The W3C validators rely on community support for hosting and development.
[Donate](#) and help us build better tools for a better web.

Options

- Show Source Show Outline List Messages Sequentially Group Error Messages by Type
- Validate error pages Verbose Output Clean up Markup with HTML Tidy

[Help](#) on the options is available.



Serviço de validação de CSS do W3C

Resultados da validação CSS do W3C para <http://r2s.ath.cx/tese/> (CSS nível 2.1)

Ir para: [Alertas \(15\)](#) [CSS validada](#)

Resultados da validação CSS do W3C para <http://r2s.ath.cx/tese/> (CSS nível 2.1)

Parabéns! Não foram encontrados erros na sua folha de estilo.

Este documento é válido para as [CSS nível 2.1](#) !

Para mostrar aos seus leitores que você teve o cuidado de criar uma página web interoperável, você pode inserir um selo nas páginas válidas. Veja a seguir o código XHTML a ser usado para mostrar na sua página o citado selo:



```
<p>  
<a href="http://jigsaw.w3.org/css-validator/check/referer">  
  
</a>  
</p>
```



```
<p>  
<a href="http://jigsaw.w3.org/css-validator/check/referer">  
  
</a>  
</p>
```