

Resumo

Esta dissertação apresenta o trabalho desenvolvido na implementação de módulos para aplicação na *Web*. Esses módulos, desenvolvidos tendo em linha de conta problemas gerais, são posteriormente adaptados para solucionar problemas específicos no desenvolvimento de aplicações *Web*. Esta metodologia permite a sua reutilização em projectos futuros.

A implementação desses módulos é feita recorrendo essencialmente a linguagens de programação orientadas para a Internet como, PHP, javascript, HTML e CSS.

A dissertação descreve ainda dois projectos onde se demonstra a utilização desses mesmos módulos. O serviço “Eye-Net” consiste num sistema de visualização de câmaras IP. Este serviço pretende que um cliente devidamente autenticado possa aceder a cada uma das suas câmaras, (*que podem estar num ou vários locais*), e aí poder visualizar a imagem captada por cada câmara, ouvir o som ambiente, ligar e desligar dispositivos eléctricos remotamente, assim como captar uma imagem instantânea.

O segundo projecto “Comércio Electrónico”, consiste num sistema onde se pretende publicar e vender publicações editadas por particulares ou empresas. Este sistema pretende demonstrar a aplicação do módulo “Carrinho de Compras” e a venda *online* de produtos recorrendo ao sistema *PayPal*.

Abstract

This dissertation presents the work developed in the implementation of modules for application in the *Web*. These modules, developed to solve general problems, are later adapted to solve specific problems in the development of *Web* applications. This methodology allows it to be used in future projects.

The implementation of these modules is made essentially using programming languages for Internet as, PHP, Javascript, HTML and CSS.

This dissertation describes two more projects where it's demonstrated the use of these same modules. The service "Eye-Net" consists of a visualization system using IP cameras. This service intends that a authenticated user can accede to each one of its cameras, (*that can be in one or some places*), and to be able to visualize the image of each camera, to hear the surrounding sound, to turn off/on electric devices remotely, as well as catching an instantaneous image.

The second project, "Electronic Commerce", consists of a system that intends to disseminate and to sell publications edited for particular or companies. This system intends to demonstrate the application of the module "shopping cart" and online sells of products using the "*PayPal*" system.

Agradecimentos

Gostaria de deixar aqui registado em meu nome o agradecimento as todas as pessoas que de alguma forma contribuíram para a realização desta dissertação sem esquecer todas as pessoas que me acompanharam ao longo da minha formação académica.

Em primeiro lugar, quero agradecer ao meu orientador Professor José Gerardo Vieira da Rocha, pelos conhecimentos transmitidos, estando sempre presente, acompanhando todo o processo e orientando de forma que os objectivos proposto fossem atingidos.

Em segundo lugar, gostaria de agradecer ao Eng. José Cabral representante da entidade Netmove, pelo apoio prestado ao longo de todo o trabalho incluindo a elaboração desta dissertação.

Seguidamente, gostaria de destacar o papel importante do meu colega Pedro Rocha pelo apoio disponibilizado e nas diversas ideias que forneceu que na maior parte dos casos levaram à solução dos problemas encontrados.

Não posso deixar também de agradecer à Netmove como entidade, pela forma cordial e profissional como me acolheu no seio da sua organização e pela oportunidade que me concedeu em efectuar este trabalho.

Por último quero deixar os mais sinceros agradecimentos a todos os professores, colegas e amigos que me acompanharam ao longo do meu percurso académico.

Aos meus Pais

Índice

1	Introdução	1
1.1	Objectivos	2
1.2	Estrutura da dissertação	3
2	Aplicações e serviços orientados à Web.....	5
2.1	Sítios estáticos	6
2.2	Sítios dinâmicos.....	6
2.3	Portais	7
2.4	Instituições bancárias.....	8
2.5	Outras aplicações	9
2.5.1	Controlo remoto	9
2.5.2	Diagnóstico médico	9
2.6	Conclusão	9
3	Redes e Protocolos da Internet.....	11
3.1	Protocolos	11
3.1.1	TCP/IP.....	11
3.1.2	HTTP.....	12
3.1.3	HTTPS	12
3.1.4	NAT	12
3.1.5	DHCP	13
3.2	Serviços.....	14
3.2.1	DNS.....	14
3.2.2	DDNS.....	14

3.3	Equipamentos	14
3.3.1	Hub.....	14
3.3.2	Switch	16
3.3.3	Router.....	16
3.4	Conclusão	17
4	Tecnologias de suporte a aplicações orientadas à <i>Web</i>	19
4.1	Linguagens de programação.....	19
4.1.1	HTML	19
4.1.2	CSS	19
4.1.3	PHP	20
4.1.4	ASP.....	21
4.1.5	JAVASCRIPT.....	22
4.1.6	AJAX	22
4.2	Sistemas de Gestão de Base de Dados (SGBD)	24
4.2.1	POSTGRESQL	25
4.2.2	MYSQL.....	26
4.2.3	Microsoft Access	27
4.2.4	MSSQL	27
4.3	Aplicações de desenvolvimento	27
4.3.1	PgAdmin III	27
4.3.2	Rapid PHP.....	28
4.4	Servidores <i>Web</i>	29
4.4.1	APACHE.....	29
4.4.2	IIS.....	30
4.5	Conclusão	31

5	Módulos desenvolvidos	33
5.1	Estrutura de catálogo	33
5.2	Upload de Ficheiros	35
5.3	Actualização da BD através de ficheiros CSV	37
5.4	Criação e envio automático de emails	38
5.5	Protecção de formulários	41
5.6	Criação de acessos protegidos a utilizadores	42
5.7	Sistema de navegação	44
5.8	Carrinho de compras	45
5.9	Galeria de imagens	48
5.10	Manutenção automática da Base de Dados	49
5.11	Conclusão	50
6	Eye-NET	51
6.1	Equipamento	52
6.1.1	Câmara Axis 207W	53
6.1.2	Câmara Pixord 205P	54
6.1.3	Router DI – 524	55
6.2	BackOffice	56
6.3	FrontOffice	59
6.3.1	Colocação das imagens das câmaras nos diversos modos de visualização	64
6.3.2	Princípio de funcionamento do Eye-Net	65
6.3.3	Actuação remota de dispositivos	66
6.4	Conclusão	67
7	Comércio Electrónico	69
7.1	Arquivo de ficheiros	70

7.2	BackOffice.....	70
7.2.1	Home.....	71
7.2.2	Upload.....	71
7.2.3	Cientes	72
7.2.4	Livros	73
7.2.5	Destaques	74
7.3	FontOffice.....	75
7.4	Conclusão	78
8	Conclusões	79
8.1	Trabalho Futuro	80
	Referências.....	81

Índice de Figuras

Figura 3-1	Rede de Computadores	12
Figura 3-2	Hub.....	15
Figura 3-3	Switch	16
Figura 3-4	Router com interface Ethernet e Wi-Fi.....	17
Figura 4-1	O modelo tradicional para aplicações <i>Web</i> (esquerda) modelo Ajax (direita) ...	23
Figura 4-2	PgAdmin III	28
Figura 4-3	Rapid PHP.....	29
Figura 5-1	Modelo Conceptual.....	34
Figura 5-2	Diagrama da estrutura de catálogo.....	34
Figura 5-3	Upload de Ficheiros	35
Figura 5-4	Actualização da Base de dados através de ficheiro	37
Figura 5-5	Newsletter	40
Figura 5-6	Área de login.....	42
Figura 5-7	Recuperação de <i>password</i>	43
Figura 5-8	Botões do sistema de navegação (Netmove, Refral, 2007).	44
Figura 5-9	Modelo conceptual de suporte ao carrinho de compras.....	45
Figura 5-10	carrinho de compras	46
Figura 5-11	carrinho de compras (tipo de entrega).....	46
Figura 5-12	Carrinho de compras (método de pagamento)	47
Figura 5-13	Carrinho de compras (confirmação).....	47
Figura 5-14	Base de dados para galeria de Imagens.....	48
Figura 5-15	Galeria de Imagens (Netmove, Refral, 2007).	49
Figura 5-16	Fluxograma de manutenção automática da BD.....	50
Figura 6-1	Diagrama Estrutural Eye-Net.....	52
Figura 6-2	Câmara Axis 207W.....	53
Figura 6-3	Câmara Pixord 205P	55
Figura 6-4	Router DI - 524.....	56

Figura 6-5	Modelo Conceptual de Base de Dados	56
Figura 6-6	Página “Clientes” do BackOffice Eye-Net	57
Figura 6-7	Página “Camaras” do BackOffice Eye-Net	58
Figura 6-8	Página “Faqs” do BackOffice Eye-Net.....	59
Figura 6-9	Mapa do Sítio do sistema Eye-Net.....	60
Figura 6-10	Modo de visualização 1.....	61
Figura 6-11	Modo de visualização 2.....	62
Figura 6-12	Modo de visualização 3.....	63
Figura 6-13	Tabela câmara	65
Figura 6-14	Diagrama de fluxo de informação.....	65
Figura 6-15	Esquema de montagem para ligação de dispositivo.....	66
Figura 7-1	Diagrama de blocos Editora.....	69
Figura 7-2	Base de Dados Editora.....	70
Figura 7-3	Diagrama de Blocos BackOffice – Editora.....	71
Figura 7-4	Página de upload de ficheiros do sistema Editora	71
Figura 7-5	Página de clientes do sistema Editora.....	72
Figura 7-6	Página Livros do sistema Editora	73
Figura 7-7	Página destaques do sistema Editora	74
Figura 7-8	Diagrama de fluxo - BackOffice editora.....	74
Figura 7-9	Mapa do sítio Editora.....	75
Figura 7-10	Página de registo sistema Editora.....	76
Figura 7-11	Página de login sistema Editora	77
Figura 7-12	Página de catálogo sistema Editora.....	78

1 Introdução

A evolução tecnológica e o melhoramento das estruturas de suporte à Internet, (*e.g. computadores, redes, software de rede*), levou a que a esta obtivesse um crescimento bastante elevado desde o seu aparecimento, de tal forma que nos dias de hoje a Internet consegue estar em praticamente todo o lado com níveis de fiabilidade bastante elevados e a preços já bastante acessíveis. O desenvolvimento da Internet deve-se em grande parte ao contributo que as empresas forneceram tendo em vista os potenciais ganhos que daí podem obter.

“A Internet está a transformar a prática empresarial na sua relação com os fornecedores e os clientes, na sua gestão, no seu processo de produção, na sua cooperação com outras empresas, no seu financiamento e na valorização das acções nos mercados financeiros. O uso apropriado da Internet converteu-se numa fonte fundamental de produtividade e competitividade para todo o tipo de empresas” (Castells, 2004).

A globalização da Internet e a sua massiva utilização nas mais diversas áreas (*e.g. Governo, Escolas, Bancos, Finanças, Saúde, Desporto*), impulsionou o desenvolvimento de novas aplicações que recorrem a este meio de comunicação como forma de interagirem com o seu público-alvo. A tendência actual para o desenvolvimento de mais aplicações *web*, pressupõem o aparecimento de novas linguagens e tecnologias que permitam a implementação dessas aplicações com interfaces cada vez mais amigáveis para o utilizador.

Até finais da década de 90 as páginas de Internet não forneciam mais do que alguma informação acerca da empresa (*imagens e texto que ilustravam a actividade da empresa*) onde se recorria essencialmente à linguagem HTML (*HyperText Markup Language*). Estas páginas eram estáticas, ou seja, sempre que fosse necessária alguma alteração quer da parte gráfica (*layout*), quer de conteúdos era necessário recorrer ao programador e pedir-lhe para efectuar a respectiva alteração. Nos dias de hoje essa situação é inconcebível uma vez que as empresas tem necessidade de actualizar constantemente e em tempo real a informação disponibilizada *online*.

Como tal, tornou-se inevitável o recurso a Base de Dados (*BD*) contendo a informação necessária para uma rápida actualização dos conteúdos presentes nas páginas de Internet.

O aparecimento de linguagens como PHP (*Hypertext Preprocessor*) ou ASP (*Active Server Pages*) veio impulsionar o desenvolvimento de páginas de Internet dinâmicas (*páginas que podem ser actualizadas em tempo real através do acesso a BD*). Estas linguagens criaram ainda condições para o desenvolvimento de aplicações como “Catálogo de Produtos”, que permite a uma empresa disponibilizar *online* de uma forma estruturada todo o seu catálogo, “Comercio Electrónico”, que possibilita a venda *online* de produtos, “Sistemas de Vídeo-vigilância” recorrendo a câmaras IP entre outras.

1.1 Objectivos

As tecnologias direccionadas para o desenvolvimento de aplicações *Web* são inúmeras. Numa área onde a concorrência é bastante elevada os custos de desenvolvimento têm uma enorme importância na comercialização do produto. A selecção das tecnologias a utilizar no desenvolvimento de uma aplicação têm reflexo directo no custo final da mesma e, como tal, essa selecção adquire grande importância no projecto. Assim o trabalho desenvolvido nesta dissertação inclui um estudo das tecnologias disponíveis no momento para a execução do trabalho.

Da mesma forma que as tecnologias utilizadas têm reflexo directo no custo da aplicação, o tempo de desenvolvimento é também bastante importante. Nesse sentido este trabalho tem como finalidade o desenvolvimento de soluções e módulos aplicativos que possam resolver problemas em diferentes aplicações. Esses módulos tornam-se bastante úteis na medida em que irão reduzir de forma substancial o tempo de desenvolvimento das aplicações uma vez que ao serem desenvolvidos para casos gerais permitem futuramente ser aplicados em situações particulares com o mínimo de alterações.

Como exemplo da aplicabilidade destes módulos, são descritos de forma geral, dois projectos que recorrem a alguns dos módulos descritos para a solução de problemas encontrados.

O sistema “Eye-net”, consiste na implementação de um sistema de vídeo-vigilância recorrendo as câmaras IP, permitindo deste modo que um utilizador possa visualizar as imagens das câmaras de qualquer parte do mundo desde que possua uma ligação à Internet.

O “Comércio Electrónico” é um sistema que permite que utilizadores devidamente autenticados possam seleccionar produtos do catálogo e efectuar a sua compra, podendo para isso efectuar o pagamento quer contra reembolso quer *online* através do sistema *PayPal*.

1.2 Estrutura da dissertação

Esta dissertação apresenta o estudo e aplicação de linguagens e tecnologias direccionadas para a *Web* com o objectivo de serem aplicadas na implementação de soluções *Web*. Nesse contexto este documento está estruturado em 8 capítulos.

Após esta Introdução, segue-se o capítulo 2, que faz uma abordagem comparativa com as aplicações desenvolvidas actualmente directamente relacionadas com o trabalho desenvolvido nesta dissertação.

O capítulo 3 apresenta alguns protocolos que irão ser referidos ao longo do documento, assim como algum equipamento importante no desenvolvimento do trabalho presente no capítulo 6.

O capítulo 4 apresenta uma breve descrição das principais tecnologias e linguagens de programação disponíveis para o desenvolvimento do trabalho.

No capítulo 5 serão apresentados alguns módulos desenvolvidos com a finalidade de solucionar problemas no desenvolvimento de aplicações *Web* recorrendo às linguagens e tecnologias apresentadas no capítulo anterior.

No capítulo 6, é apresentado o estudo e desenvolvimento de um sistema de vídeo-vigilância que, recorrendo a câmaras IP, permite a utilizadores registados visualizarem as imagens captadas por cada uma das câmaras em diferentes modos de visualização.

No capítulo 7, é descrita uma aplicação *Web* que utilizando módulos desenvolvidos no capítulo 5 como “carrinho de compras” e “Criação de acessos protegidos a utilizadores”, permite que

utilizadores devidamente autenticados possam seleccionar produtos do catálogo e efectuar a sua compra podendo para isso efectuar o pagamento quer contra reembolso quer *online* através do *PayPal*.

Finalmente, no capítulo 8 apresentam-se as conclusões relativas a este trabalho e as perspectivas de utilização de novas ferramentas e o desenvolvimento e/ou melhoramento das aplicações apresentadas neste documento.

2 Aplicações e serviços orientados à Web

Este capítulo pretende demonstrar o enorme crescimento das aplicações orientadas para a Internet, indicando as áreas que mais se aproveitaram desta tecnologia assim como algumas aplicações já desenvolvidas directamente relacionadas com o âmbito deste trabalho.

Desde o seu aparecimento que a Internet tem vindo a desenvolver uma ligação muito forte com os seus utilizadores. Embora por vezes se acuse a Internet de “...incitar as pessoas a viver as suas próprias fantasias *on-line* e fugir do mundo real, ...” (Castells, 2004), a verdade é que a Internet tem contribuído de forma preponderante para o desenvolvimento das sociedades não só ao nível social mas principalmente ao nível empresarial. Apesar de tudo a Internet na sua esmagadora maioria é utilizada para fins profissionais.

“...as utilizações da Internet são fundamentalmente instrumentais e estão estreitamente relacionadas com o trabalho, a família e a vida quotidiana dos utilizadores. O correio electrónico representa mais de 85% da utilização da Internet, e a maior parte desse volume está relacionado com o trabalho...” (Castells, 2004).

De facto a Internet tem desempenhado um papel importantíssimo no dia-a-dia de cada ser humano e a sua área de aplicabilidade é enorme. A área do ensino beneficiou muito com o crescimento das aplicações *Web*. Passou a poder adquirir-se o conhecimento de uma forma mais rápida e disponível 24h por dia. As próprias aulas por vezes são dadas através de plataformas (*e-learning*), evitando assim a deslocação aos estabelecimentos de ensino.

Ao nível governamental a Internet tem sido uma ajuda preponderante nomeadamente no que respeita à facilidade de comunicação que possibilita entre os diferentes membros do governo, capacitando-os assim de um meio de resolução de problemas rápido e eficaz.

As empresas passaram a poder comunicar com clientes ou outras empresas via *Web*, permitindo assim acelerar o processo de comunicação e conseqüentemente aumentar a produtividade das mesmas.

Ao nível da Saúde a Internet também adquiriu um papel importantíssimo, sendo hoje em dia possível obter diagnósticos remotamente. Desta forma um médico pode obter a opinião de um colega acerca de um diagnóstico sem ter de se deslocar até este ou enviar toda a documentação pelo correio.

O próprio sector financeiro evoluiu enormemente à custa das aplicações *Web*, hoje em dia é possível efectuar transferências bancárias, consultas de conta, pagamentos, sem se sair do escritório.

Dependendo da aplicabilidade pretendida, as aplicações *Web* podem ser desenvolvidas mediante diferentes processos, diferentes graus de segurança, complexidade e de interactividade.

2.1 Sítios estáticos

Um “Sítio Estático”, é uma aplicação onde toda a informação disponibilizada está previamente armazenada em disco, normalmente um servidor *Web*. Essa informação é colocada manualmente pelo programador na altura de desenvolvimento do projecto e qualquer alteração necessária é efectuada pelo próprio programador. Esta aplicação é muito frequentemente usada ao nível de páginas pessoais, pessoas que querem divulgar informação a seu respeito que pode ir desde a actividade lúdica, académica, profissional, etc. Estes Sítios normalmente recorrem a linguagens de programação básicas como HTML (*Hypertext Preprocessor*), não permitindo no entanto que haja qualquer tipo de interactividade entre o utilizador e o sistema.

2.2 Sítios dinâmicos

O desenvolvimento de aplicações para a Internet que para além de disponibilizar informação online permitam também que haja interacção entre o utilizador e a aplicação, implica um desenvolvimento mais estruturado e requer mais recursos que os “Sítios Estáticos”. Os Sítios dinâmicos têm a particularidade de as suas páginas serem criadas em tempo real. Ou seja, o seu conteúdo não está armazenado em disco, ele é gerado na altura que o utilizador acede ao serviço através de um *browser* (*programa que efectua o pedido de um conteúdo disponível na Web e*

apresenta-o), recorrendo para isso a linguagens de programação próprias e caso necessário aceder a Bases de dados para recolha de informação.

Comparativamente com os “Sítios Estáticos”, estes tem como é óbvio um custo de desenvolvimento mais elevado, contudo, para além da possível interacção entre utilizador e o sistema, tem a capacidade de ser constantemente actualizado por um utilizador com permissões especiais, normalmente designado de gestor do serviço. Este pode de uma forma simples adicionar, remover ou editar informação que irá reflectir-se na página disponibilizada para o utilizador comum.

Este tipo de sítios está muito em voga, nomeadamente na divulgação de empresas que possuam um catálogo de produtos muito variado com constantes alterações de stock, de modo que justifica a actualização permanente da sua estrutura *Web*.

Um exemplo deste tipo de plataforma é o Sítio de Internet da empresa Equinotec (Netmove, Equinotec, 2007), a empresa possui um catálogo de produtos bastante diversificado que pretende disponibilizar on-line. Para isso foi implementado um “Sítio dinâmico” que suporta toda a informação. Esta estrutura para além do interface criado para os utilizadores tem ainda um outro interface com acesso restrito para um operador interno à empresa que permite que este actualize constantemente o conteúdo do sítio.

2.3 Portais

O termo “portal” é utilizado para definir “Sítios dinâmicos”, que juntamente com as características destes possibilitam a execução de serviços de administração da empresa, tendo normalmente outros sítios associados ao sítio principal. Pode-se tomar como exemplo deste tipo de sítios, o portal das Finanças (Finanças), o portal da Segurança Social (Informática, 1999) os portais dos operadores móveis (Optimus, 2007) (Tmn, 2007) (Vodafone, 2007) e o Portal da OASRN (Netmove, OASRN, 2003). Este último exemplo possui uma área, “Formação Complementar” que permite efectuar toda a gestão deste serviço, desde a inserção de alunos, inscrição em exames, divulgação de notas, calendário das aulas, etc.

2.4 Instituições bancárias

O Sector financeiro tem aproveitado o desenvolvimento das aplicações *Web* para criar plataformas *on-line* de apoio ao seu serviço. Essas plataformas permitem aos utilizadores a execução de uma série de serviços associados aos bancos, como:

- Consultar contas, créditos e cartões,
- Realizar transferências bancárias e pagamentos,
- Agendar operações,
- Aderir a cartões de débito e crédito.

Contudo essas aplicações requerem cuidados mais rigorosos ao nível da segurança do que as apresentadas até aqui. Nomeadamente no que diz respeito à conexão.

Um sistema *Web* para a actividade bancária tem obrigatoriamente de usar uma ligação segura HTTPs (*HyperText Transfer Protocol Secure*) em vez da tradicional HTTP (*HyperText Transfer Protocol*). Este tipo de ligação indica ao utilizador que o serviço possui um certificado digital, emitido por uma autoridade certificadora independente, garantindo assim a fiabilidade do serviço. Como exemplo, o sistema da Caixa Geral de Depósitos utiliza ainda outros factores para garantir a segurança dos serviços prestados (CGD, 2007).

- **Autenticação e Validação**

Os utilizadores só podem aceder ao serviço mediante a introdução de um código de acesso e respectiva password.

- **Cartão matriz**

Elemento físico que se encontra na posse do cliente e que é solicitado aquando da realização de algumas operações.

- **Cookies**

“O cliente não pode aceder ao **Caixa**directa *on-line* se o seu *browser* não aceitar *Cookies*, que são variáveis guardadas do lado do cliente. O **Caixa**directa *on-line* utiliza *cookies* de

sessão para prevenir acessos não autorizados às contas online do cliente. As *cookies* são guardadas no *browser* e expiram quando o cliente termina a sua sessão **Caixadirecta online.**” (CGD, 2007).

2.5 Outras aplicações

2.5.1 Controlo remoto

A tecnologia de fabrico de microprocessadores evoluiu de tal forma que hoje em dia pode-se encontrar microprocessadores com grandes capacidades, dimensões reduzidas e preços baixos. Dessa forma é possível integrar servidores *Web* dentro de dispositivos com dimensões reduzidas possibilitando o controlo desses dispositivos por parte das aplicações *Web* (MENDONÇA).

Por exemplo, pode-se integrar um servidor *Web* dentro de uma câmara IP de forma a que esta possa estar conectada com dispositivos eléctricos (*e.g. lâmpadas, televisões, motores*), e desta forma permitir comandar esses dispositivos remotamente.

2.5.2 Diagnóstico médico

Considere-se o caso de um médico que efectua um diagnóstico a um paciente, contudo o caso é de tal forma complicado que este necessita consultar a opinião de um especialista de um outro país. Através de uma aplicação *Web* desenvolvida para o efeito é muito fácil o médico expor o caso ao seu colega mostrando inclusive resultados de exames, e imagens de radiografias de forma que o especialista situado a milhares de quilómetros de distância consegue muito rapidamente fornecer o seu diagnóstico perante o caso apresentado.

2.6 Conclusão

As aplicações desenvolvidas para a *Web* tem um campo de aplicabilidade quase infindável sendo apenas limitadas pela imaginação do ser humano, é por isso praticamente impossível descrevê-las na sua totalidade. Este capítulo apresentou algumas dessas aplicações, nomeadamente as mais directamente relacionadas com o trabalho desenvolvido nesta dissertação.

3 Redes e Protocolos da Internet

Ao longo desta dissertação serão feitas referências a conceitos e equipamentos que para algumas pessoas podem ser pouco familiares, nesse sentido este capítulo vai abordar de uma forma sintética esses conceitos para que se consiga tirar o máximo partido da leitura da dissertação.

3.1 Protocolos

3.1.1 TCP/IP

Para que os veículos automóveis possam circular nas vias públicas foi necessário criar um conjunto de regras às quais todos os condutores obedecem, a que se chamou “Código da Estrada”. Tendo em conta a quantidade de redes de computadores existentes e o número de computadores por rede, foi, à semelhança do código da estrada necessário a criação de regras para o mundo da Internet.

Para que um computador possa trocar informações com um ou mais computadores é necessário que os computadores envolvidos na troca de informação utilizem as mesmas regras. Esse conjunto de regras é normalmente designado de TCP (*Transmission Control Protocol*). No protocolo de comunicação estão definidas todas as regras necessárias para que o computador de destino “entenda” as informações no formato que foram enviadas pelo computador de origem. Dois computadores com diferentes protocolos instalados, não serão capazes de estabelecer uma comunicação e nem serão capazes de trocar informações. O Protocolo TCP sempre que é executado utiliza o protocolo IP (*Internet Protocol*), responsável pelo transporte dos datagramas de uma rede para a Internet.

A Figura 3-1 representa uma rede local, pode verificar-se que todos os computadores ligados à rede possuem a mesma gama de IP.

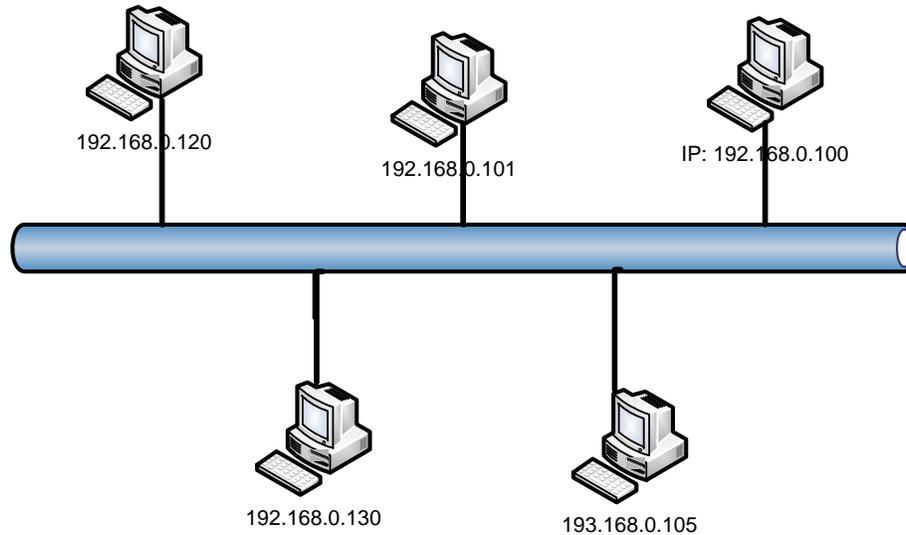


Figura 3-1 Rede de Computadores

3.1.2 HTTP

HTTP (*HyperText Transfer Protocol*) é um protocolo onde se definem as regras de comunicação na Internet. Quando se acede a um Sítio de Internet, o *browser* efectua pedidos a um servidor *Web*, este por sua vez irá responder a esses pedidos. O protocolo HTTP tem a responsabilidade de definir quais os pedidos podem ser feitos ao servidor *Web* e quais as respostas possíveis para os pedidos em questão (Torres, 2006) (Pastore, 2003).

3.1.3 HTTPS

O HTTPS (*HyperText Transfer Protocol Secure*) é a junção de dois protocolos, o HTTP descrito em cima, responsável pelos pedidos ao servidor *Web* e o SSL (*Secure Sockets Layer*), que efectua a encriptação dos dados, faz a autenticação do servidor, integridade da mensagem e autenticação do cliente (Pastore, 2003).

3.1.4 NAT

Quando se cria uma rede local é necessário ter alguma forma de efectuar a tradução dos endereços IP e portas locais para a Internet. Isso é feito através do protocolo NAT (*Network Address Translation*). Ou seja, quando um computador da rede local envia algo para a Internet, a informação passa pelo servidor onde lhe é atribuído o IP deste permitindo assim o envio da informação para a Internet, no processo inverso, quando algo é enviado da Internet para a rede local, a informação passa pelo servidor onde lhe atribuído o IP do computador.

3.1.5 DHCP

DHCP (*Dynamic Host Configuration Protocol*), é um protocolo utilizado nas redes de computadores que tem a tarefa de atribuir o IP automaticamente às máquinas ligadas numa rede (Alecrim, Protocolo DHCP, 2005).

Numa rede com poucas máquinas não existe grande problema em atribuir manualmente o IP a cada uma delas, contudo quando o número de máquinas numa rede é elevado a probabilidade de se atribuir o mesmo IP a duas máquinas distintas aumenta.

O DHCP pode ser programado de três formas distintas para a atribuição do IP às máquinas de uma rede:

- **Manual**

Neste modo de configuração existe a necessidade de associar um endereço disponível no servidor DHCP ao MAC do computador ligado à rede.

- **Automática**

Neste modo de configuração o Servidor DHCP é configurado por forma que quando uma máquina se liga pela primeira vez numa rede é-lhe atribuído um endereço IP permanente.

- **Dinâmica**

Neste modo o servidor DHCP atribui um endereço IP a um computador por tempo limitado. Ou seja, assim que o computador ao qual foi atribuído um endereço IP se desliga da rede, o IP por ele utilizado passa a estar disponível para outras ligações à rede. O servidor DHCP faz uma verificação dos IP's disponíveis na rede em intervalos de tempo pré-definidos.

3.2 Serviços

3.2.1 DNS

O DNS (*Domain Name System*) é um sistema responsável pela atribuição de nomes que identificam máquinas. O ser humano sempre teve mais facilidade em lidar com palavras do que com números, nesse sentido o DNS tem particular interesse na medida em que associa um determinado nome ao endereço IP de uma máquina. Essa tarefa é executada à custa de BDs espalhadas pelos servidores de todo o mundo onde é armazenada a informação que indica que determinado IP está associado ao nome de uma máquina (Alecrim, Básico sobre DNS, 2005).

3.2.2 DDNS

O número de IP's disponíveis na Internet tem um número limitado, como tal para se poder obter e manter um endereço IP é necessário proceder ao pagamento do aluguer desse endereço. Contudo a maior parte dos utilizadores, nomeadamente em habitações particulares, não necessita de possuir sempre o mesmo IP. O DDNS (*Dynamic DNS*) permite a criação de nomes que apontam para o IP dinâmico utilizado, permitindo assim que mesmo não possuindo um IP fixo se consiga memorizar de uma forma simples o endereço de determinada rede. Este serviço normalmente é controlado e fornecido por entidades que cobram por este serviço (Axis, 2007).

Uma implementação do DDNS é o serviço NO-IP, que permite a atribuição do DNS dinâmico (*DDNS*) sem custo associado. Este serviço permite que utilizadores que não possuam um IP fixo, possam conhecer o seu endereço IP (No-IP.com - Vitalwerks Internet Solutions, 1999).

3.3 Equipamentos

3.3.1 Hub

O Hub é essencialmente usado para redes com um número reduzido de computadores, nomeadamente redes domésticas. Isto porque é dos dispositivos de interligação de computadores de uma rede mais simples. Este dispositivo tem apenas a função de receber dados de um computador da rede e transmitir essa informação para os restantes computadores de rede, ao

contrário do *switch* que, tem a capacidade de enviar a informação apenas para o computador de destino.

Pode-se considerar vários tipos de Hubs:

- **Passivos**

O termo passivo tem a ver com o facto de o Hub não efectuar qualquer tipo de operação como por exemplo amplificação do sinal para que duas máquinas possam estar a uma distância considerável. Este tipo de Hub apenas recebe a informação de uma máquina da rede e transmiti-a para as restantes.

- **Activos**

Os Hubs activos para além da sua função de transmissão da informação de uma máquina para as restantes, tem a capacidade de amplificar os sinais recebidos. Isso permite que as máquinas possam estar as distâncias maiores do que com a utilização de Hubs passivos, normalmente 100 metros da máquina até ao Hub e mais 100 metros do Hub até à máquina que irá receber a informação.

- **Inteligentes**

Os Hubs inteligentes tem a capacidade de detectar se alguma máquina na rede está com problemas que possa de alguma forma impedir o bom funcionamento da mesma e nesse caso se for necessário desconectam da rede a máquina em questão. Permite ainda a detecção de pontos de congestionamento na rede de forma a conseguir normalizar o tráfego.



Figura 3-2 Hub

3.3.2 Switch

O *Switch* é um dispositivo que tem por função fazer o encaminhamento da informação entre máquinas de uma rede. O *Switch* tem a capacidade de criar um canal entre a máquina de origem e a de destino permitindo dessa forma que a rede não fique suspensa quando dois computadores trocam informação. Isso faz com que a rede tenha um bom desempenho excepto quando dois computadores tentam enviar informação para o mesmo destino em simultâneo.



Figura 3-3 Switch

3.3.3 Router

As redes que suportam um número elevado de máquinas utilizam quase que obrigatoriamente *Routers*. O *Router* é utilizado nestas redes porque quando comparado com o *Switch* por exemplo ele faz a mesma função deste mas para além disso tem a capacidade de seleccionar o melhor trajecto para o envio da informação de um local para outro. Numa cidade o trânsito é tanto mais complicado quanto maior a dimensão e o número de estradas da mesma. Numa rede de computadores funciona exactamente da mesma forma, nesse caso o router tem uma função importantíssima porque consegue decidir qual o percurso mais curto e menos congestionado em determinado momento.

Pode-se considerar dois tipos de *routers*:

- **Estáticos**

São *Routers* mais simples e conseqüentemente mais baratos que no processo de escolha de qual o trajecto a ser seguido pela informação apenas tem atenção a distância e não o congestionamento de cada um dos trajectos possíveis.

- **Dinâmicos**

Este *Router* torna-se mais útil quanto mais complexa for a rede, na medida em que tem a capacidade de determinar qual o caminho mais rápido entre dois pontos, mesmo que isso implique distâncias maiores.



Figura 3-4 Router com interface Ethernet e Wi-Fi

3.4 Conclusão

Este capítulo fez referência a alguns Protocolos e serviços da Internet de forma que o leitor não tenha dificuldade em assimilar a informação disponibilizada nesta dissertação. Foram ainda abordados de forma sintética alguns equipamentos onde se enumerou as principais diferenças entre esses equipamentos de forma a justificar a utilização de um em detrimento dos restantes.

4 Tecnologias de suporte a aplicações orientadas à *Web*

No desenvolvimento de uma aplicação *Web* um factor de bastante importância é a escolha das tecnologias a utilizar (*e.g. Plataforma, Linguagens de programação, base de dados*). Esta escolha depende como é óbvio de vários factores, entre os quais os conhecimentos do programador, a finalidade da aplicação, a verba disponível.

Convém por isso ter um conhecimento geral do que existe em cada momento para que na hora de selecção das tecnologias a utilizar se possa optar por aquelas que irão ter um custo mais baixo sem que isso prejudique a qualidade da aplicação.

4.1 Linguagens de programação

4.1.1 HTML

A criação de páginas de Internet pode ser extremamente simples. HTML (*HyperText Markup Language*) é uma linguagem que permite a criação de páginas de Internet com hiper-texto, conforme se pode deduzir pelo nome. Na verdade HTML não pode ser considerado uma linguagem de programação, mas sim uma linguagem de formatação, que permite a criação de páginas de Internet a partir de um ficheiro de texto formatado através de uma série de comandos. Esses comandos designados de “*tags*” são então usados para formatar um ficheiro de texto de forma a criar a página de Internet. O número de “*tags*” disponíveis é bastante elevado, na ordem das várias dezenas, contudo apenas uma pequena parte dessas “*tags*” são normalmente usadas, permitindo assim uma aprendizagem bastante rápida desta linguagem.

4.1.2 CSS

O CSS (*Cascade style Sheet*) é uma linguagem que permite personalizar uma página de Internet através da definição de regras nos elementos de marcação HTML. Deste modo consegue-se

retirar destes o controlo da formatação do documento. Com o recurso a esta linguagem de estilo consegue-se algumas vantagens como são o caso de:

- Controlo de todo o *layout* da página a partir de um único documento,
- Maior velocidade na manutenção e alteração do *layout*,
- Redução do tempo de carregamento.

4.1.3 PHP

Até finais da década de noventa uma grande percentagem de páginas de Internet eram estáticas, ou seja o seu conteúdo mantinha-se constante ao longo do tempo e não tinham a capacidade de interagir com o utilizador. Como se deve calcular essas páginas eram bastante limitadas na medida que impediam calcular operação mais complexa que implicasse comunicação entre a página e o utilizador impossibilitando assim que empresas pudessem interactuar com clientes e até com outras empresas. O PHP (*Hypertext Preprocessor*) é uma linguagem que permite a criação de páginas dinâmicas, isto é, trata-se de um código que pode ser inserido entre código HTML e efectuar certas operações que por sua vez podem criar páginas instantaneamente (PHP, 2007).

O PHP é uma tecnologia que se encontra em expansão e funciona em praticamente em todos os ambientes conhecidos. Quando se pergunta, *o que se pode fazer com o PHP?* A resposta é: qualquer coisa. O PHP é focado para ser uma linguagem de script do lado do servidor, portanto, pode-se fazer qualquer coisa que outro programa CGI (*Common Gateway Interface*) possa fazer, como: gerar páginas com conteúdo dinâmico ou enviar e receber cookies.

O código PHP é embebido dentro do código HTML e deste modo permite que o servidor a que foi acrescentado o módulo PHP consiga interpretar os comandos inseridos e assim gerar código HTML.

A ligação do PHP com HTML fazem desta linguagem uma ferramenta bastante interessante e isso torna-se pelo demais evidente nos formulários HTML. É através destes que se consegue fazer a transmissão dos dados inseridos por determinado utilizador e assim fazer o processamento desses dados, dando assim dinamismo à página.

Talvez a mais forte e mais significativa característica do PHP é seu suporte a uma ampla variedade de base de dados. Escrever uma página que consulte uma base de dados é incrivelmente simples e sendo o PHP uma linguagem de código aberto é muito fácil de adquirir o código para determinada função sem custo adicionado visto o PHP ser uma linguagem gratuita. Esta característica torna-se muitas vezes decisiva na altura de seleccionar a linguagem a utilizar no desenvolvimento da aplicação, nomeadamente quando se pensa em utilizar PHP ou a sua concorrente directa ASP (Soares, 2004). Outra característica do PHP é que é uma linguagem interpretada, não necessitando de compilador, o que pode ser um inconveniente.

4.1.4 ASP

Normalmente na implementação de sítios de Internet dinâmicos, é muito frequente colocar-se a questão: *Qual a linguagem de programação a usar?*

Á semelhança do PHP também o ASP (*Active Server Pages*), é uma linguagem de script que funciona do lado do servidor, aliás estas duas linguagens são na maior parte dos casos concorrentes uma da outra, no que se refere à implementação de sítios dinâmicos. Embora ambas as linguagens corram do lado do servidor e possam ser embutidas dentro do HTML permitindo o acesso a Base de Dados de forma a dar dinamismo à página, estas duas linguagens tem algumas diferenças que podem ser decisivas na selecção de qual delas usar. Essas diferenças, descritas a seguir, estão directamente relacionadas com o custo de utilização de cada uma das linguagens, uma vez que esse custo vai afectar o valor do projecto (Alecrim, Linguagem ASP, 2003):

- Enquanto o PHP é uma linguagem de código aberto, que pode correr sob uma plataforma Linux, conseqüentemente a sua utilização grátis, o ASP é uma linguagem propriedade da Microsoft que corre sob uma plataforma Windows e como tal tem um custo associado à sua utilização.
- O PHP é bastante flexível no que respeita à ligação com Bases de Dados, permitindo a ligação a diversas BD's, sendo que as mais frequentes são MySQL e PostgreSQL. Por sua vez o ASP apenas trabalha com ligação à Base de Dados MSSQL.

Concluindo, ambas as linguagens são boa escolha no que se refere à implementação de sítios dinâmicos. Embora o PHP seja mais rápido do que o ASP o principal factor de selecção destas

duas linguagens é o custo, e neste campo o PHP está em vantagem visto não ter custos associados ao contrário do ASP.

“The cost of running an ASP-based website implies a full Windows server platform; development costs are higher, software licenses are expensive and speed, security and flexibility are all sacrificed.” (Anstey, 2003).

4.1.5 JAVASCRIPT

O javascript é uma das linguagens mais usadas no que se refere à criação de páginas de Internet. Por ser fácil de aprender e ao mesmo tempo de utilizar juntando o facto de correr em praticamente todos os *browsers* existentes hoje em dia, torna esta linguagem bastante poderosa, permitindo dar animação às páginas de Internet. O Javascript é uma linguagem interpretada no cliente, assim que a página HTML é carregada o código javascript é executado. Isto permite que o *javascript* possa ser usado quer para introduzir animações nas páginas *Web*, quer na verificação de formulários sem ser necessário efectuar a sua submissão, tornando deste modo o processo de verificação de dados mais rápido e agradável (Alecrim, Linguagem Javascript, 2003).

4.1.6 AJAX

Todos os dias surgem aplicações desenvolvidas para utilização na *Web*, mas para que isso aconteça é necessário que os criadores dessas aplicações tenham ao seu dispor ferramentas adequadas à concepção dessas aplicações. Quando um utilizador faz uso de uma nova aplicação normalmente não tenta compreender a tecnologia que está por detrás da aplicação contudo tem senso crítico para avaliar o seu desempenho, nomeadamente ao nível do desempenho. Não é de todo agradável quando se recorre a uma aplicação estar constantemente à espera da resposta do servidor para que a página actual seja actualizada.

Aplicações recentes como Gmail, Google Suggest e Google maps, vieram demonstrar que é possível construir aplicações para uso na *Web* sem se verificar a contínua espera da resposta do servidor. A tecnologia que está por detrás destas aplicações tem o nome de AJAX (*asynchronous Javascript and Xml*). Na verdade AJAX não é uma Linguagem de programação mas sim um combinar de linguagens em que cada uma realizando a sua tarefa produz uma ferramenta bastante poderosa. O AJAX incorpora:

- HTML e CSS,
- Exibição e interação dinâmicas usando Document Object Model (*DOM*),
- Troca e manipulação de dados usando XML e XSLT,
- Retorno de dados assincronamente usando XMLHttpRequest,
- E JavaScript a fazer a interligação.

A grande maioria das aplicações *Web* do lado do cliente executa um pedido HTTP ao servidor. Este por sua vez interpreta os dados, processa-os e retorna numa página HTML. Este processo requer que a página efectue *refresh* daí os utilizadores terem a sensação que a sua página está a ser carregada novamente de cada vez que executam um operação.

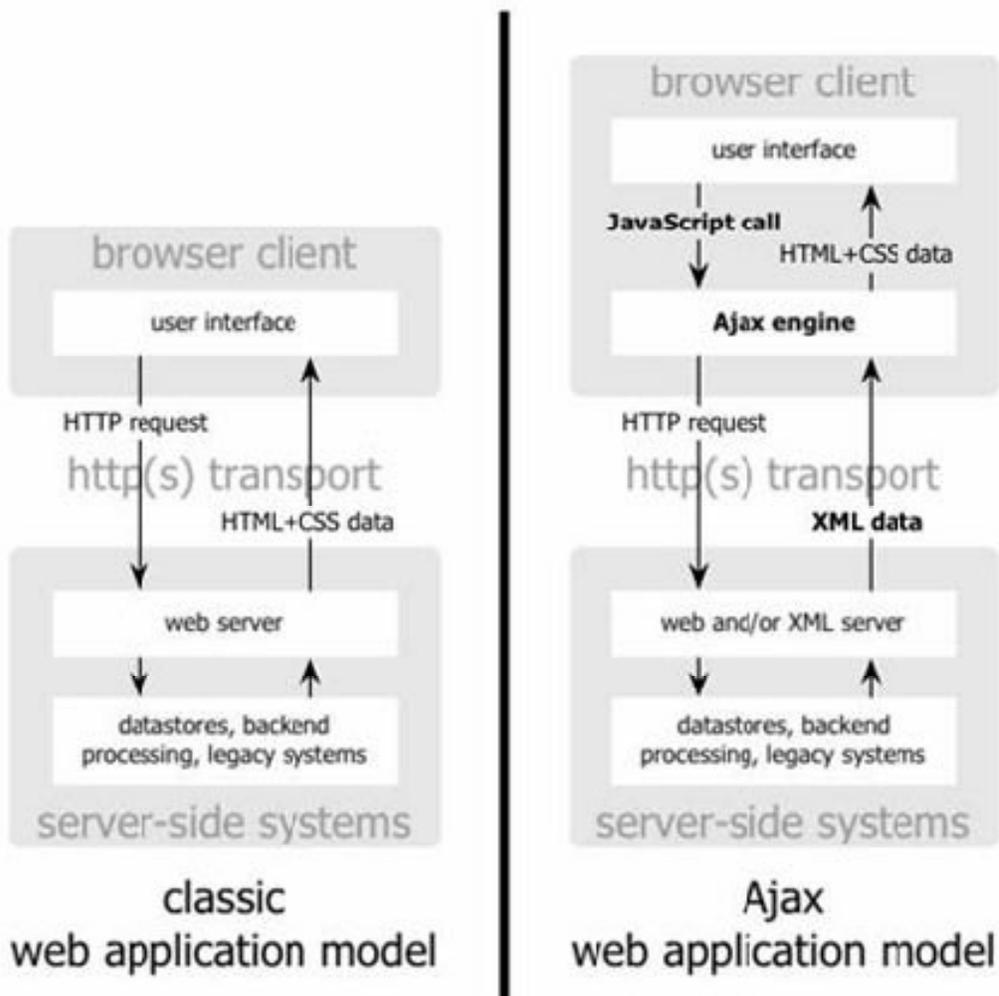


Figura 4-1 O modelo tradicional para aplicações *Web* (esquerda) modelo Ajax (direita)

As aplicações Ajax eliminam a sensação de que a página está continuamente a ser carregada. A ferramenta Ajax é introduzida entre o utilizador e o servidor, fazendo deste modo que o Browser carregue a ferramenta Ajax escrita em javascript. O Ajax permite que a interacção do utilizador com a aplicação aconteça de modo assíncrono independentemente da comunicação com o servidor.

4.2 Sistemas de Gestão de Base de Dados (SGBD)

Inicialmente a informação era guardada através de Sistema de Gestão de Ficheiros. Estes sistemas tinham a particularidade de os ficheiros de dados e as aplicações terem obrigatoriedade de possuir uma ligação física. Esse facto trazia grandes problemas nomeadamente quando várias aplicações pretendiam aceder aos mesmos dados.

O Sistema de Gestão de Base de Dados (SGBD), veio resolver os problemas apresentados pelo anterior modelo, na medida em que este sistema garante a independência dos dados relativamente às aplicações. As aplicações têm apenas uma ligação lógica e não física com o SGBD.

“O SGBD é um conjunto de software, destinado a gerir todo o armazenamento e manipulação dos dados do sistema, fazendo a interface entre o nível aplicacional e a base de dados propriamente dita.” (Pereira, 1998).

Qualquer SGBD permite a realização das seguintes tarefas:

- Operações de definição e alteração da estrutura de uma BD,
- Criação de uma BD,
- Criação de uma tabela,
- Criação dos campos de uma tabela,
- Alteração da estrutura de uma tabela,
- Eliminação de tabelas,
- Operações de manipulações de dados,
- Consulta ou pesquisa de dados,

- Inserção de dados,
- Alteração de dados,
- Eliminação de dados,
- Operações de controlo de dados,
- Atribuição ou supressão dos direitos de acesso aos dados em relação a um utilizador ou grupo de utilizadores.

4.2.1 POSTGRESQL

O PostgreSQL é um sistema de gestão de base de dados objecto-relacional (*SGBDOR*) desenvolvido no Departamento de Ciência da Computação da Universidade da Califórnia em Berkeley.

O POSTGRES foi pioneiro em muitos conceitos objecto-relacionais que estão agora a tornar-se disponíveis em algumas bases de dados comerciais. Os Sistemas de Gestão de Base de Dados Relacionais (*SGBDR*) tradicionais suportam um modelo de dados composto por uma colecção de relações com nome, contendo atributos de um tipo específico. Nos sistemas comerciais em uso, os tipos possíveis incluem número de ponto flutuante, inteiro, strings, monetário e data. É amplamente reconhecido que este modelo não é adequado para aplicações futuras de processamento de dados. O modelo relacional substituiu com sucesso os modelos anteriores em parte devido à sua "simplicidade Espartana". Entretanto, esta simplicidade tornou a implementação de certas aplicações muito difícil. O PostgreSQL oferece recursos adicionais pela incorporação dos conceitos mostrados abaixo, tornando possível aos utilizadores estenderem o sistema facilmente:

- O bloqueio no nível de tabela foi substituído por um sistema de concorrência multi-versão, permitindo a leitura de dados consistentes durante a actividade de escrita, possibilitando efectuar cópias de segurança utilizando o `pg_dump` enquanto base de dados se mantém disponível para consultas.
- A implementação de funcionalidades importantes no servidor, incluindo subconsultas, padrões, restrições e gatilhos.
- A incorporação de funcionalidades adicionais compatíveis com a linguagem SQL92, incluindo chaves primárias, identificadores entre aspas, conversão implícita de tipo de

cadeias de caracteres literais, conversão explícita de tipos e inteiros binários e hexadecimais.

- Os tipos nativos foram aperfeiçoados, incluindo vários tipos para data e hora, e suporte adicional para tipos geométricos.
- A velocidade geral do código do servidor foi melhorada em aproximadamente 20-40%, e o tempo de inicialização do servidor foi reduzido em 80% com a sua versão 6.0.

Estas funcionalidades colocam o PostgreSQL dentro da categoria de base de dados referida como *objeto-relacional*. Portanto, embora o PostgreSQL possua algumas funcionalidades de orientação a objectos, está firmemente ligado ao mundo das bases de dados relacionais. Na verdade, algumas bases de dados comerciais incorporaram recentemente funcionalidades nas quais o PostgreSQL foi pioneiro (PostgreSQL, 1996).

4.2.2 MYSQL

O MySQL é um sistema de gestão de Base de dados (SGBD) muito utilizado nomeadamente na *Web* uma vez que se encontra optimizado para essa utilização, juntamente com o facto de ser muito comum encontrar disponível serviços para alojamento de páginas *Web* com MySQL e PHP. Também importante na elevada utilização deste SGBD é o facto de correr em praticamente todos os sistemas operativos (*e.g. Windows, Linux, Mac*), e ser um sistema livre, ou seja pode ser estudado e mesmo alterado.

- Alta compatibilidade com linguagens como PHP, Java, Python, C#, Ruby e C/C++,
- Baixa exigência de processamento,
- Recursos como transactions (*transacções*), conectividade segura, indexação de campos de texto, replicação,
- Instruções em SQL, como indica o nome.

Como se pode ver nem sempre é fácil decidir quais as tecnologias a utilizar na implementação de soluções *Web*. Os dois pontos anteriores são exemplo disso, *deve-se usar MySQL ou PostgreSQL?*

Pode-se dizer que estes dois SGBD estão muito bem capacitados para as mais diversas utilizações podendo-se no entanto dizer que para utilizações que envolvam troca de grandes

volumes de dados e tratamento de informações críticas deve-se usar o PostgreSQL e para situações em que a velocidade de retorno da informação seja muito importante e não haja muita complexidade nas aplicações se deve usar MySQL (MySQL, 1995).

4.2.3 Microsoft Access

O Microsoft Access é um sistema de Gestão de Base de Dados (SGBD), propriedade da Microsoft. Esta aplicação é normalmente usada em sistemas pouco complexos com um número de utilizadores reduzido e tem obrigatoriamente de correr sobre plataformas Windows.

4.2.4 MSSQL

O MS SQL à semelhança do Access é um Sistema de gestão de Base de Dados da Microsoft, como tal só corre sob plataformas Windows. Contudo comparativamente com o Access, o MS SQL está mais habilitado para gerir BD mais complexas do que o sistema anterior, (*Microsoft Access*). Quando se utiliza BD mdb como é o caso do Access ou MS SQL e o número de visitas é grande ou a BD cresce muito, os riscos de instabilidade para o servidor são elevados. Com o MS SQL esses riscos não acontecem uma vez que o SQL SERVER está num computador diferente do servidor *Web*.

4.3 Aplicações de desenvolvimento

4.3.1 PgAdmin III

O PgAdmin III é uma poderosa plataforma de desenvolvimento para bases de dados PostgreSQL, muito utilizado devido não só a todas as suas qualidades como por se tratar de um programa freeware e ainda por correr quer em Windows 2000/XP quer em ambientes GNU/ Linux. O PgAdmin III foi desenhado de forma a responder às necessidades de todos os seus utilizadores, quer desde simples query's em SQL até ao desenvolvimento de bases de dados complexas. A interface gráfica está feita de forma a suportar todas as potencialidades do Postgres e facilitar a sua administração.

A aplicação possui ainda um construtor de query's, um editor de SQL, um editor de código server-side.

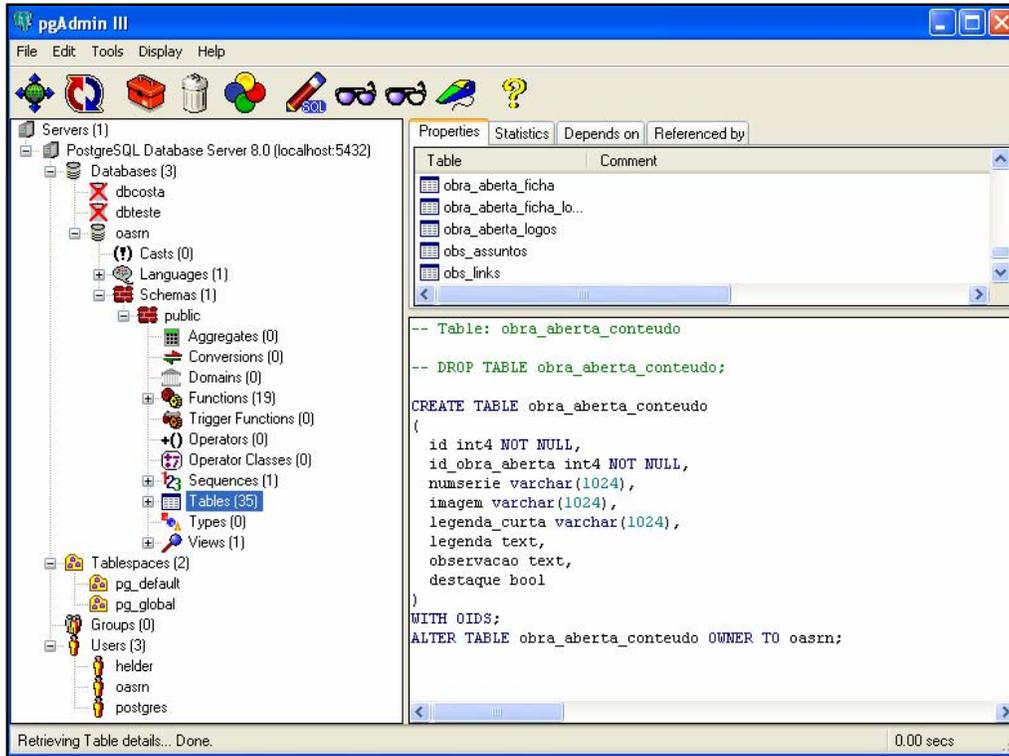


Figura 4-2 PgAdmin III

4.3.2 Rapid PHP

O Rapid PHP é uma boa escolha como editor PHP. Este software de utilização extremamente simples permite a criação de diversos tipos de ficheiros, como, PHP, HTML, XHTML, CSS, JavaScript, ASP, XML. A possibilidade de abrir documentos localmente ou através de FTP, a possibilidade de gravar esses documentos local ou remotamente através de FTP, juntamente com uma série de ferramentas integradas para validação, reutilização, navegação e formatação de códigos fazem do Rapid PHP um boa escolha quer para pequenos projectos quer para grandes projectos.

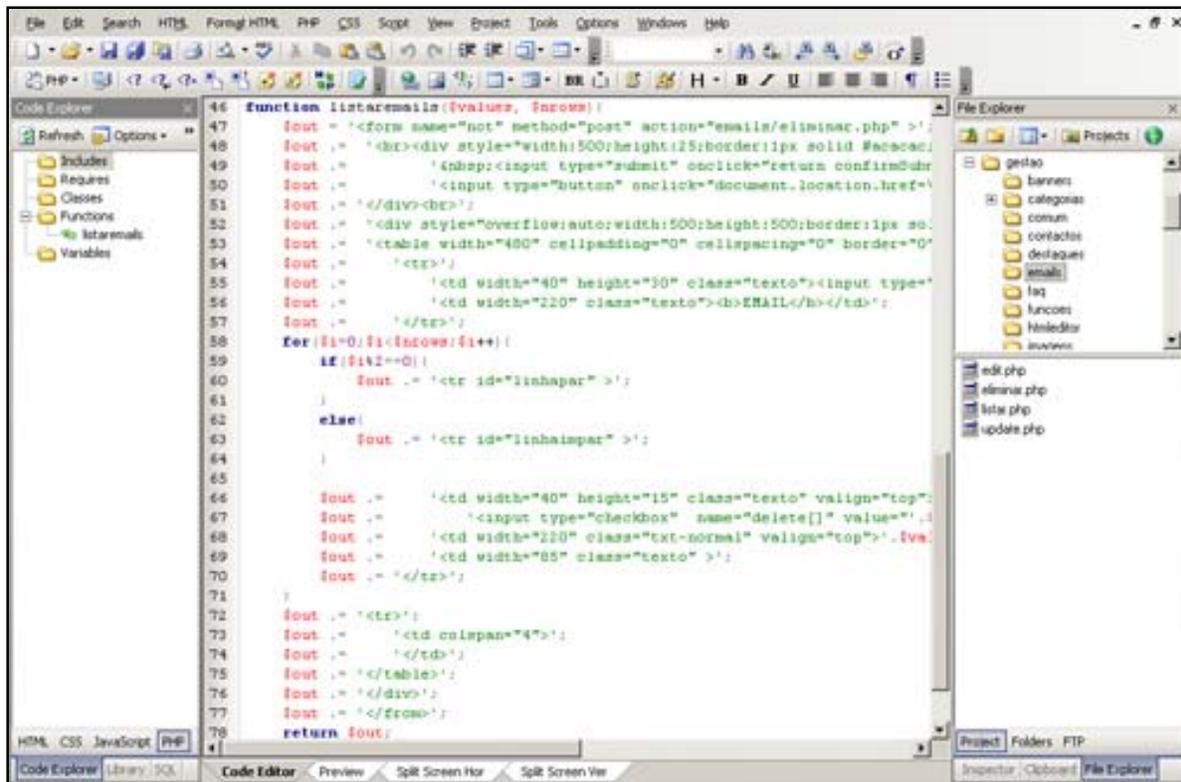


Figura 4-3 Rapid PHP

4.4 Servidores Web

O desempenho da aplicação Web está em muito dependente do servidor Web utilizado. Os dois maiores rivais são sem dúvida o servidor Apache e o Servidor da Microsoft IIS. Como qualquer tecnologia o seu uso depende em grande parte do tipo de utilização.

4.4.1 APACHE

O servidor apache é de entre todos os servidores o mais usado devidas as suas principais características, altamente configurável, pode ser executado em diferentes plataformas, é flexível, está sempre em desenvolvimento para a inclusão dos protocolos mais actualizados (*por exemplo, HTTP*), fornece o código-fonte completo e não possui licenças restritivas, pode ser configurado para diferentes funções, é composto de *módulos*, cada um implementando uma característica diferente e aumentando a funcionalidade do servidor, além de várias outras características. Além disso, o Apache é uma entidade orgânica, aqueles que beneficiam dele usando-o, por outro lado

contribuem fornecendo realces das suas características e melhorando-o. A quantidade de esforço dispendido por todos os que o usam na sua melhoria é grande o que faz com que o produto resultante seja muito forte, isto acontece somente com freeware (*grátis*) porque quando alguém paga pelo software não está geralmente dispostos a reparar erros.

Em conclusão pode-se dizer que o Apache é forte devido a ser freeware pois se fosse comercializado, como os seus principais concorrentes, ainda não estaria tão desenvolvido mesmo que todo o dinheiro da sua comercialização fosse investido no seu desenvolvimento

Abaixo estão algumas características que fazem com que este servidor *Web* seja o preferido:

- Possui suporte a scripts cgi usando linguagens como **PHP**, *Shell Script*, *ASP*,
- Suporte a autorização de acesso podendo ser especificadas restrições de acesso separadamente para cada endereço/arquivo/directório acedido no servidor,
- Autenticação requerendo um nome de utilizador e senha válidos para acesso a alguma página/sub-directório/arquivo (*suportando criptografia via Crypto e MD5*),
- Negociação de conteúdo, permitindo a exibição da página *Web* no idioma requisitado pelo Cliente Navegador,
- Suporte a virtual hosting (*é possível servir duas ou mais páginas com endereços/ portas diferentes através do mesmo processo ou usar mais de um processo para controlar mais de um endereço*),
- Suporte a IP virtual hosting,
- Suporte a name virtual hosting,
- Suporte a servidor Proxy FTP e HTTP, com limite de acesso,
- Suporte a proxy e redireccionamentos baseados em URLs para endereços Internos (Foundation, 2007).

4.4.2 IIS

O servidor da Microsoft IIS (*Internet Information Server*), é também uma boa escolha para o uso em aplicações *Web*, contudo existem alguns pontos chaves que fazem com que o servidor mais usado seja o apache.

- **Limitação de utilização**

Em primeiro lugar o IIS é um produto da Microsoft e como tal apenas corre sob plataformas desta empresa, isso faz com que o apache tenha um campo de operação mais alargado na medida que pode correr sob várias plataformas.

- **Custo**

Como se trata de um produto da Microsoft obviamente irá ter custo associados à sua utilização.

- **Segurança**

É sabido que as versões antigas do Windows não oferecem muita segurança, sendo que as mais recentes embora já mais capacitadas a esse nível continuam a não estar ao nível de outros sistemas, consequentemente o servidor IIS está mais sujeito a ataques do que o apache (Costello, 2002).

4.5 Conclusão

Este capítulo abordou as tecnologias de suporte a aplicações orientadas para a *Web*, onde se tentou justificar o porquê da utilização de umas em detrimento de outras.

A estrutura de suporte às aplicações aqui apresentadas consiste num plataforma Linux com servidor apache e base de dados PostgreSQL.

Essa estrutura teve influência decisiva na selecção das linguagens utilizadas para o desenvolvimento das aplicações descritas. Assim, o trabalho foi desenvolvido recorrendo às linguagens de programação HTML, CSS, Javascript e PHP. Se a primeiras não oferecem grande discussão na medida que são normalmente usadas na maioria das aplicações, o PHP pode provocar alguma controvérsia. No entanto a sua selecção foi definida tendo por base as razões principais apresentadas no presente capítulo como são o caso da estrutura de suporte à aplicação disponibilizada pela empresa e os custos associados à utilização das diferentes tecnologias.

5 Módulos desenvolvidos

No capítulo anterior foram apresentadas algumas tecnologias e linguagens de programação orientadas para implementação de aplicações *Web*. Foi ainda referido ao longo desta dissertação que muitas das soluções adoptadas na solução de problemas quando desenvolvidas tendo em conta casos gerais podem ser reutilizadas noutras aplicações. Neste capítulo irá descrever-se o desenvolvimento de módulos aplicativos recorrendo às linguagens de programação já apresentadas que permitem a sua aplicação em diferentes projectos.

5.1 Estrutura de catálogo

A utilização da Internet tem aumentado de tal forma que as empresas quiseram aproveitar esse facto para poderem mostrar os seus produtos a potenciais clientes através deste meio de comunicação. Mas para que isso fosse as páginas de Internet tiveram também de evoluir nesse sentido, permitindo a apresentação estruturada dos produtos de uma empresa de forma que os utilizadores possam encontrar o que pretendem de uma maneira simples e rápida. Já foi referido nesta dissertação que ainda no final da década de noventa a maior parte das páginas de Internet eram estáticas, ora isso limitava senão mesmo impossibilitava a apresentação do catálogo de produtos de uma empresa online.

Com o aparecimento de novas linguagens de programação orientadas para a *Web*, permitindo o acesso a Bases de Dados (BD), passou a ser possível criar estruturas dinâmicas que permitem actualizar o conteúdo de uma página em tempo real. Neste processo a definição da BD tem um papel fundamental no desempenho da aplicação.

Após uma primeira análise verificou-se que numa grande percentagem de casos a estrutura de catálogo de uma empresa pode ser definida da forma exemplificada na Figura 5-1.

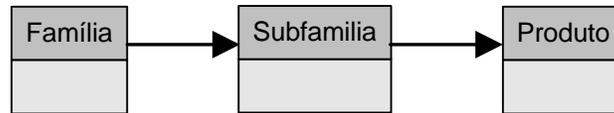


Figura 5-1 Modelo Conceptual

Contudo, como já foi referido pretende-se desenvolver módulos que possam ser reutilizados noutras aplicações. Esta estrutura sendo um pouco rígida não permite essa reutilização, na medida em que determinadas empresas não conseguem estruturar o seu catálogo de produtos segundo esta divisão.

A Figura 5-2 apresenta uma outra solução para este problema. Esta estrutura em árvore é bastante flexível o que permite que qualquer empresa consiga definir o seu catálogo de produtos segundo esta estrutura por mais complexo que este seja.

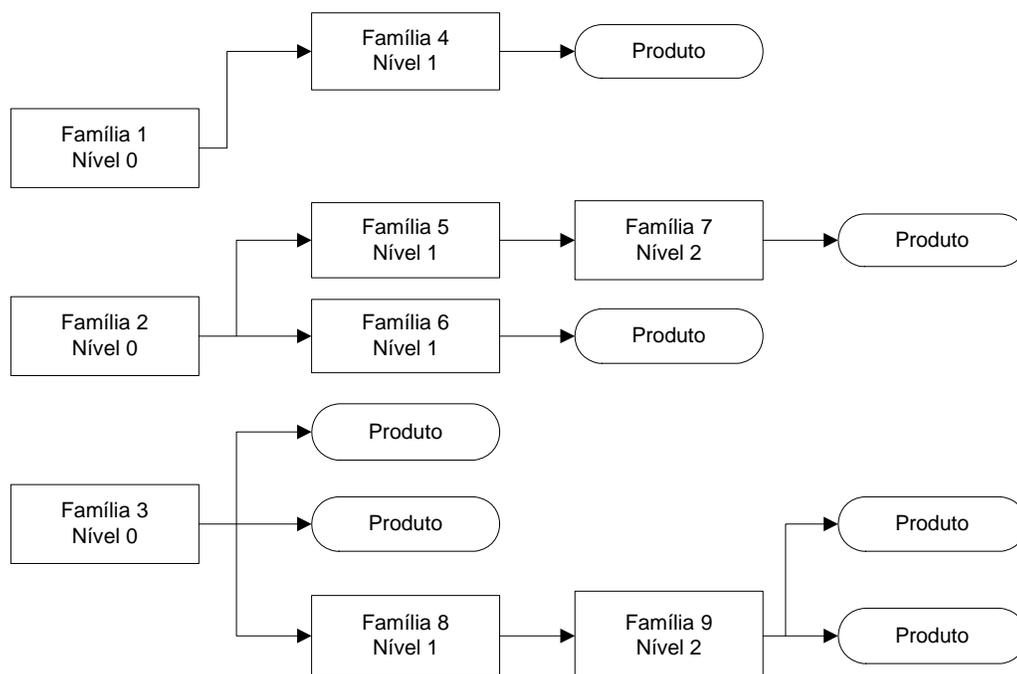


Figura 5-2 Diagrama da estrutura de catálogo

Enquanto a solução anterior obrigava que o produto estivesse associado a uma subfamília que por sua vez deveria estar associada a uma família, nesta solução o conceito de subfamília deixa de existir. Desse modo pode estar associado a uma família de nível 0 (*primeira família do ramo*), ou pode estar associado a uma família de nível 3. Nesta situação considera-se que família de nível 0 é pai da família de nível 1 e família de nível 1 é pai de família de nível 2.

Quando uma família não tiver “filhos” (*famílias associadas*), associa-se então o produto à família pretendida.

Observando a Figura 5-2, pode verificar-se que todos os ramos terminam com a inserção de um produto. Não é obrigatório que assim seja, no entanto não pode acontecer a situação de existir uma família que é pai de uma outra família e ao mesmo tempo ter um produto associado.

5.2 Upload de Ficheiros

O “Upload de ficheiros” consiste na transferência de ficheiros (*e.g. imagens, pdf's, vídeos*) presentes no computador para o servidor onde está alojada a página de Internet (Figura 5-3), com o intuito de os apresentar para os utilizadores. Estes ficheiros são normalmente associados a produtos, notícias, destaques, etc., logo faz todo o sentido possuir uma área capaz de efectuar o upload desses ficheiros para que posteriormente possam ser utilizados pela aplicação.

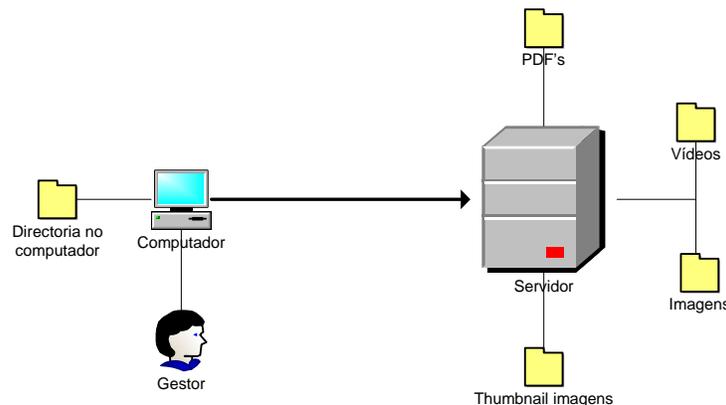


Figura 5-3 Upload de Ficheiros

O módulo de upload de ficheiros tem vários aspectos a ter em conta. O 1º dos quais é efectuar a transferência dos ficheiros para o servidor, para isso é necessário que a aplicação possua um formulário HTML onde se pode seleccionar os ficheiros pretendidos.

```
<form action="upload.php" method="post" enctype="multipart/form-data">
<input type="text" name="imagem[]" size="20" maxlength="100">
<input type="hidden" name="MAX_FILE_SIZE" value="10000">
<input name="userfile" type="file">
<input type="submit" value="Enviar">
</form>
```

O código HTML descrito em cima é responsável pela apresentação do formulário que permite a selecção dos ficheiros para posterior upload. Neste formulário o campo “action” tem a função de indicar o ficheiro que será responsável pela verificação e upload do ficheiro. O campo hidden, `MAX_FILE_SIZE` permite determinar o tamanho máximo em bytes do ficheiro, contudo é necessário ter em atenção que este valor está definido por defeito no ficheiro de configuração do PHP para um valor de 2MB (*megabytes*).

A página responsável pelo upload do ficheiro deve verificar se a directoria do servidor para a qual se pretende copiar o ficheiro já contém um ficheiro com o mesmo nome, nesse caso não permite a cópia e envia um aviso com a imagem que não foi copiada para o servidor.

Ao efectuar o upload para o servidor, o código responsável por essa acção deve colocar o ficheiro na directoria correspondente. No caso específico das imagens é necessário ter em linha de conta que o *browser*, não tem capacidade para redimensionar as imagens sem que estas percam qualidade. Para resolver essa situação pode optar-se por redimensionar as imagens num programa desenvolvido para o efeito (*e.g. Photoshop*) e depois efectuar o upload do número de imagens igual ao número de vezes que essa imagem deverá ser apresentada com dimensões diferentes. Esta solução não é aconselhada na medida em que irá obrigar a efectuar um maior número de uploads fazendo aumentar o tempo de actualização da página *Web*. Deste modo a solução passa por desenvolver uma função que permita o redimensionamento das imagens sem perda de qualidade, garantindo desta forma que as imagens não perdem qualidade melhorando também o desempenho da página *Web* uma vez que imagens mais pequenas demoram menos tempo a carregar, logo torna a página mais rápida.

O código seguinte é o exemplo de uma função responsável pelo redimensionamento de imagens.

```
function create_thumb( $filename, $wdir){
    $image2 = sscanf($filename,'%[^.].',$image);
    $image = $wdir.'/'. $image.'_thumb.jpg';
    // nova dimensão
    list($width, $height) = getimagesize($filename);
    if($width > $height)
    {
        $aux = $width / 120;
        $newwidth = "120";
        $newheight = $height/$aux;
    }
    else
```

```

{
    $aux = $height / 120;
    $newheight = "120";
    $newwidth = $width/$aux;
}
$thumb = imagecreatetruecolor($newwidth, $newheight);
$source = imagecreatefromjpeg($filename);
imagecopyresized($thumb, $source, 0, 0, 0, 0, $newwidth, $newheight, $width, $height);
imagejpeg($thumb,$image,100);
return 1;
}

```

5.3 Actualização da BD através de ficheiros CSV

A actualização do conteúdo de uma página *Web* é um processo que deve ser o mais rápido e intuitivo possível. Existem situações em que o conteúdo a inserir na página já está integralmente guardado em ficheiros. Os ficheiros CSV (comma-separated values) caracterizam-se por possuir uma estrutura organizada em colunas separadas por caracteres conhecidos, essa característica torna-se útil para a leitura dos dados contidos no ficheiro.

Este módulo consiste na criação de um formulário HTML onde deve ser inserido o nome, o utilizador e a password da BD, o nome da tabela que se pretende actualizar e o nome do ficheiro onde estão guardados os dados. Os dados do ficheiro serão integralmente copiados para a tabela em questão desde que se garanta que a estrutura do ficheiro é igual à da tabela.

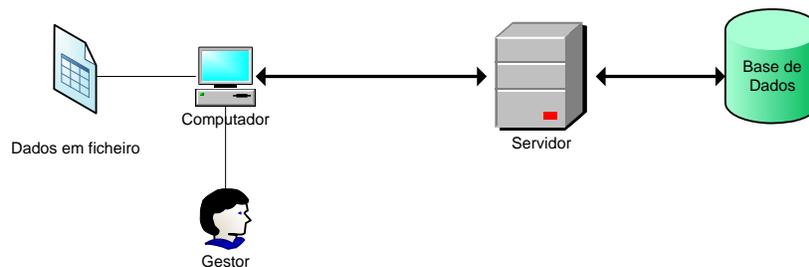


Figura 5-4 Actualização da Base de dados através de ficheiro

O processo aqui descrito é realizado recorrendo à linguagem PHP que possui funções que permitem aceder e tratar dados contidos em ficheiros.

O código seguinte exemplifica o processo de abertura, leitura e fecho do ficheiro. Após a execução do código todo o conteúdo do ficheiro ficará armazenado no array “\$data”, sendo apenas necessário efectuar a leitura do array e copiar a informação deste para a tabela pretendida.

```
$handle = fopen ($filename,"r");
while ($linha = fgetcsv ($handle, 1000, ";")) {
    $data[$row] = $linha;
    $row++;
}
fclose ($handle);
```

5.4 Criação e envio automático de emails

Um dos principais problemas ambientais com que a humanidade se debate nos dias de hoje, é com o abate excessivo de árvores para a produção de papel. Diariamente gasta-se quantidades elevadíssimas de papel em publicidade, empresas que pretende cativar potenciais clientes através de panfletos enviados para suas residências.

O módulo “criação e envio automático de emails”, (*newsletter*), é uma ferramenta que quando adicionada a uma página de Internet permite ajudar a reduzir de forma substancial o número de panfletos enviados. Esta ferramenta permite de forma rápida e fácil criar, editar e enviar uma *newsletter* electrónica para todos os utilizadores registados numa base de dados. Este conceito é um suporte de comunicação muito importante para a divulgação de assuntos relacionados com a actividade e funcionamento de uma empresa, além de ser uma ferramenta económica e ecológica. A criação da newsletter engloba as seguintes etapas:

- **Criação de emails**

A área de criação de emails disponibiliza para o utilizador um interface que lhe permite adicionar ou remover emails da Base de Dados. Este interface permite ao utilizador inserir apenas um email de cada ou vários emails de uma só vez através de um ficheiro CSV. Em qualquer dos casos os emails antes de serem inseridos na BD passam por um processo de verificação de sintaxe de forma a evitar que sejam inseridos emails incorrectos na BD. Esta área permite ainda que o utilizador possa criar um ficheiro CSV com todos os emails presentes na BD.

- **Upload de imagens**

O upload de imagens é normalmente utilizado no sistema de forma a servir as diferentes áreas do mesmo. Esta aplicação permite efectuar o upload e posterior remoção das imagens que irão ser utilizadas na criação da newsletter.

- **Construção dinâmica da *newsletter* com um *design* semelhante ao Sítio de Internet**

Depois de definido o *layout* da *newsletter*, é criado um formulário que permite ao utilizador criar a *newsletter* de uma forma rápida e simples. O utilizador terá apenas de seleccionar as imagens que pretende inserir na *newsletter* e escrever o texto associado a cada uma das imagens. Depois de inserida toda a informação, o utilizador deve clicar no botão criar que irá fazer com que a *newsletter* fique armazenada na BD.

- **Pré-visualização da *newsletter***

Esta opção permite ao utilizador ter acesso à *newsletter* com a mesma configuração com que irá chegar aos destinatários, permitindo desta forma detectar alguma irregularidade rapidamente.

- **Envio da *newsletter***

O passo seguinte à criação da *newsletter* é o envio da mesma. Para evitar que surjam problemas disponibiliza-se várias opções de envio da *newsletter*:

- **Envio de teste**

A *newsletter* é enviada para um número restrito de pessoas de forma a testar se está tudo pronto para o envio.

- **Envio para email**

A *newsletter* é enviada para um ou mais emails inseridos no formulário disponibilizado na altura do envio.

- **Envio para grupos**

Os emails estão classificados por grupo, desta forma permite enviar a *newsletter* apenas para os emails pertencentes a um determinado grupo seleccionado na altura do envio.

- **Enviar**

Esta “enviar” permite enviar a *newsletter* para todos os emails presentes na BD.

- **Remoção automática da newsletter**

As *newsletter's* criadas ficam guardadas na BD permitindo desta forma disponibilizar um histórico das *newsletter's*. Contudo é definido um número máximo de *newsletter's* para o histórico. Desta forma sempre que uma newsletter é enviada o sistema verifica se o número máximo de *newsletter's* foi atingido, nesse caso elimina a mais antiga no sistema.

A Figura 5-5 apresenta o exemplo descrito contendo uma notícia e três produtos, sendo que cada produto tem um texto e uma imagem associados.



Figura 5-5 Newsletter

5.5 Protecção de formulários

Os formulários HTML são usados de forma massiva nas páginas de Internet. Infelizmente caso não sejam correctamente protegidos estes podem tornar-se uma fonte de problemas. Quando um utilizador preenche um formulário numa página *Web* a informação inserida é processada numa outra página. Esta informação pode ser enviada por email, no caso de um formulário de contacto, mas pode também servir para actualizar dados de uma BD. O primeiro caso não apresenta grandes problemas ao nível da segurança, sendo assim o formulário pode ser protegido com uma linguagem que seja executada do lado do cliente como por exemplo javascript, uma vez que apenas será necessário verificar se determinados campos obrigatórios foram ou não preenchidos.

Tome-se como exemplo o seguinte formulário, que tem como objectivo enviar um email para determinado endereço onde nele devem constar o nome e o email inseridos no formulário. Torna-se então evidente que de alguma forma se deve obrigar o utilizador a inserir esses campos. Para efectuar essa operação pode-se recorrer à função javascript **“submitcontacto()”** que verifica, ainda do lado do cliente, se esses campos foram preenchidos. Caso não tenha sido efectuado o preenchimento desses campos deverá exibir-se um alerta indicando ao utilizador a obrigatoriedade do preenchimento desses campos.

```
<form action="recolhe_dados.php" method="post" onsubmit="javascript:submitcontacto()">
<input type="text" name="nome" size="40" maxlength="40">
<input type="text" name="email" size="40" maxlength="40">
<input type="submit" value="Enviar">
</form>
```

```
function submitcontacto() {
var text=document.contacto;
    if (text.nome.value=="") {
        alert("Nome é um campo obrigatório.");
        text.nome.focus();
        return;
    }
    if (text.email.value=="") {
        alert("Email é um campo obrigatório.");
        text.email.focus();
        return;
    }
text.submit()
}
```

Contudo nem todas as situações se resolvem de forma simples como a descrita acima.

Quando os dados enviados através de formulários HTML acedem directamente à BD, pode haver quem aproveite esse facto para através da criação de comandos SQL nesse formulário, ter acesso a dados privados ou até mesmo subscrever dados de elevada importância, chama-se a esta operação “sql injection”.

Torna-se por isso imperativo proteger todos os formulários HTML, não só do lado do cliente mas sobretudo do lado do servidor. O recurso a funções javascript como as descritas acima é um dos passos a executar. Ainda do lado do cliente deve-se sempre limitar o número máximo de caracteres a ser introduzidos num determinado campo através do comando maxlength.

Do lado do servidor deve-se garantir que as variáveis enviadas pelo formulário cumprem a restrição quanto ao número máximo de caracteres e garantir que juntamente com os dados enviados não foram inseridos caracteres especiais como plicas, aspas, asterisco, etc.

5.6 Criação de acessos protegidos a utilizadores



Área de Clientes

Utilizador

Senha OK

[Registrar](#)

[Recuperar Password](#)

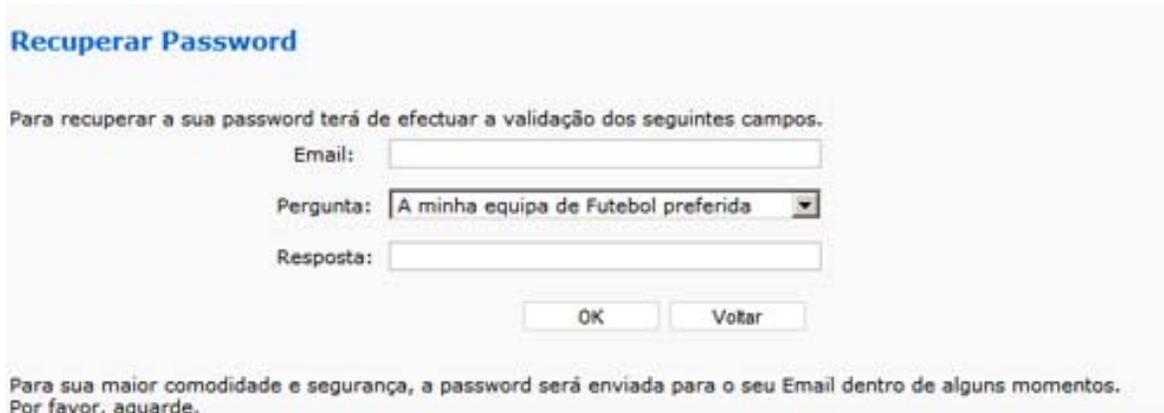
Figura 5-6 Área de login

O processo de autenticação tem vários aspectos a ter em conta. Quando um utilizador visita uma página de Internet em que é necessário passar por um processo de autenticação ele deverá preencher os campos que lhe são pedidos, isto no caso de já ter efectuado previamente o registo. Esses valores irão ser comparados com os presentes na BD e caso estejam correctos o utilizador terá permissão para aceder às áreas protegidas.

No caso do utilizador não se encontrar ainda registado, terá de efectuar o registo através do link “registar” Figura 5-6.

O processo de registo, permite que o utilizador insira a *password* pretendida, esta será encriptada e guardada na BD. Uma forma de efectuar essa operação é recorrendo ao código apresentado a seguir que em primeiro lugar divide a *password* original em caracteres individuais, converte cada um desses caracteres num número hexadecimal de 32 caracteres e por fim volta a converter novamente toda a string num número hexadecimal de 32 caracteres.

```
$pass_encrypt = preg_split('//', $password, -1, PREG_SPLIT_NO_EMPTY);  
$majorsalt = "";  
foreach ($pass_encrypt as $hashpass) {  
    $majorsalt .= md5($hashpass);  
}  
$pass_encrypt = md5($majorsalt);
```



Recuperar Password

Para recuperar a sua password terá de efectuar a validação dos seguintes campos.

Email:

Pergunta:

Resposta:

Para sua maior comodidade e segurança, a password será enviada para o seu Email dentro de alguns momentos. Por favor, aguarde.

Figura 5-7 Recuperação de *password*

Quando um Utilizador perde a sua *password* de acesso à página de Internet, deve existir uma forma de este recuperar o acesso. O exemplo demonstrado na Figura 5-7, consiste em preencher o formulário com o *email* associado à conta, e seleccionar a pergunta e respectiva resposta inseridas aquando do registo do cliente. Esta informação é recolhida numa nova página onde se verifica se os dados inseridos estão correctos, nesse caso é gerada uma nova *password* para este utilizador e enviada para o email da conta. Em seguida *password* é encriptada e guardada novamente na base de dados.

5.7 Sistema de navegação

É muito frequente em páginas de Internet o resultado de uma pesquisa ser apresentado em forma de listagem. Por vezes o número de resultados encontrados é de tal forma elevado que recorre-se a uma ferramenta a que se dá o nome de “Sistema de navegação” de forma a apresentar apenas parte desses resultados sendo que é possível de uma forma rápida apresentar os restantes.

O módulo “Sistema de navegação” consiste em associar botões de navegação (Figura 5-8), que juntamente com a função que se apresenta a seguir permitem obter apenas os resultados que poderão ser apresentados na página actual. Isso permite que a query à base de dados seja feita mais rapidamente.

```
function getdatapag($numhits, $limit, $page){
$numhits = (int) $numhits;
$limit    = max((int) $limit, 1);
$page     = (int) $page;
$numpages = ceil($numhits / $limit);
$page     = max($page, 1);
$page     = min($page, $numpages);
$offset   = ($page - 1) * $limit;
return array($offset, $limit, $page, $numpages);
}
```

O módulo de navegação funciona da seguinte forma:

Obtém-se o número total de itens, de seguida define-se o número de itens a apresentar por página e o valor da primeira página. Invoca-se a função “getdatapag”, esta função retorna os parâmetros a considerar na query à base de dados (*offset e limit*), juntamente com o número da página actual e número total de páginas. No final da página deve-se ter botões de navegação que permitem avançar ou recuar no número da página actual.

« anterior Pag. 1 de 2 próximo »

Figura 5-8 Botões do sistema de navegação (Netmove, Refral, 2007).

5.8 Carrinho de compras

A encomenda ou venda de produtos online implica que a página de Internet possua uma estrutura concebida para essa finalidade, normalmente apelidada de “carrinho de compras”. Quando um utilizador visita uma página de Internet com o intuito de efectuar a compra de um produto sabe à partida que terá de efectuar um registo onde irá fornecer uma série de dados pessoais. Normalmente o utilizador é confrontado com uma de duas situações:

- O sistema permite o utilizador seleccionar os produtos que pretende adquirir e na altura de confirmação da compra ou encomenda pede-lhe que efectue o registo ou login.
- O sistema não permite que o cliente selecione qualquer produto sem antes efectuar registo ou login.

A situação aqui descrita visa a 2ª opção. Para isso é preciso implementar a seguinte estrutura de BD Figura 5-9.

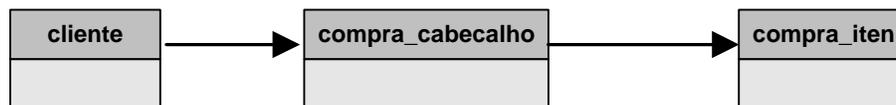


Figura 5-9 Modelo conceptual de suporte ao carrinho de compras

Quando o utilizador efectua login é automaticamente criado um cabeçalho de compra para esse cliente. A partir desse momento, sempre que o utilizador seleccionar um produto este será adicionado na tabela “compra_iten” associado ao cabeçalho criado na altura do login. No final caso o cliente efectue a compra, a BD será actualizada colocando esta compra como efectuada, caso contrário os itens são eliminados assim como o cabeçalho da compra.

O Processo de compra ou encomenda online requer 4º passos:

- **Seleção dos produtos**

O utilizador selecciona os produtos que pretende comprar, estes são adicionados ao carrinho de compras e apresentados, Figura 5-10.

1 Cesto de Compras **2** Tipo de Entrega **3** Método de Pagamento **4** Confirmação

Produto	PVP Uni.	Qtd.	Sub Total	
Princípios de Electrónica - 1	44.00 €	<input type="text" value="1"/>	44.00 €	remover
Flash 8 - Curso completo	38.30 €	<input type="text" value="1"/>	38.30 €	remover
PHP 5 Conceitos, Programação e Integração com Banco de Dados	55.00 €	<input type="text" value="1"/>	55.00 €	remover
ACTUALIZAR QUANTIDADES		Total	137.30 €	

Todos os preços de venda ao público (PVP) incluem IVA à taxa em vigor.
Caso pretenda que a sua encomenda seja enviada pelos CTT, o valor acima descrito será acrescido dos respectivos portes de envio.

Figura 5-10 carrinho de compras

- **Tipo de Entrega**

Nesta fase o utilizador deve seleccionar o tipo de entrega que pretende. Caso o utilizador pretenda que a entrega seja feita via CTT deve verificar os dados presentes no formulário “Endereço de Expedição”, e caso este pretenda que a facturação seja feita com os dados presentes nesse formulário deve responder afirmativamente à questão colocada no final da página fazendo com que os dados sejam automaticamente copiados para o formulário “Facturação”.

1 Cesto de Compras **2** Tipo de Entrega **3** Método de Pagamento **4** Confirmação

CTT Outro

Endereço de Expedição

Nome:

Telefone:

Email:

Morada:

Localidade:

Cód Postal: -

Pretende usar o mesmo endereço para a facturação?
sim Não

Endereço de Facturação

Nome:

Telefone:

Email:

Morada:

Localidade:

Cód Postal: -

Figura 5-11 carrinho de compras (tipo de entrega)

• Método de Pagamento

O método de pagamento é outra das fases do processo de compra de produtos online, as hipóteses de efectuar esta operação são inúmeras sendo que as opções que envolvam pagamento por cartões de crédito devem ter especial atenção relativamente às questões de segurança. A integração do *PayPal* é uma boa solução.

1 Cesto de Compras
 2 Tipo de Entrega
 3 Método de Pagamento
 4 Confirmação

Pagamento por Cartão de Crédito
 Pagamento à cobrança

Ao seleccionar pagamento por cartão de crédito, irá ser redireccionado para a página do paypal.

O site do PayPal permite-lhe 2 opções de pagamento:

1. Se for cliente registado, poderá usar a sua conta para efectuar o pagamento.
2. Caso não seja cliente poderá efectuar o pagamento usando o cartão de crédito.

Figura 5-12 Carrinho de compras (método de pagamento)

• Confirmação

A última etapa da compra online serve para que o utilizador possa verificar todas as opções que tomou ao longo do processo e caso seja necessário rectifica-las.

1 Cesto de Compras
 2 Tipo de Entrega
 3 Método de Pagamento
 4 Confirmação

Produto	PVP Uni.	Qtd.	Sub Total
Princípios de Electrónica - 1	44,00 €	<input type="text" value="1"/>	44,00 €
Flash 8 - Curso completo	38,30 €	<input type="text" value="1"/>	38,30 €
PHP 5 Conceitos, Programação e Integração com Banco de Dados	55,00 €	<input type="text" value="1"/>	55,00 €
Portes de Envio			6,50 €
Total			143,80 €

Todos os preços de venda ao público (PVP) incluem IVA à taxa em vigor.

Método de Envio: CTT
Método de Pagamento: Pagamento por PayPal

Figura 5-13 Carrinho de compras (confirmação)

5.9 Galeria de imagens

O módulo “Galeria de imagens” consiste numa estrutura criada para a apresentação de imagens de produtos numa página *Web* (Figura 5-15).

Sabendo-se que, “uma imagem fala mais do que mil palavras”, é importante introduzir numa página *Web* uma ferramenta capaz de efectuar a apresentação de várias imagens de um mesmo produto sem que se perca muito tempo na transição entre imagens. Para isso é necessário em primeiro lugar permitir a associação de várias imagens a um mesmo produto. Nesse sentido a BD deve ser implementada de forma a permitir essa associação (Figura 5-14).

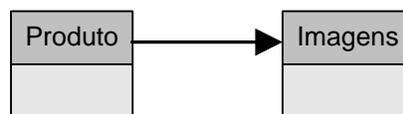


Figura 5-14 Base de dados para galeria de Imagens

Para que a visualização das imagens seja o mais intuitivo e agradável possível recorreu-se a código javascript que é executado do lado do cliente. Isso permite substituir a imagem em destaque por uma das miniaturas, bastando para isso posicionar o cursor sob uma dessas imagens.

A implementação desta tarefa é feita recorrendo ao método “onmouseover”, ele executa uma acção assim que o cursor é posicionado sobre determinado objecto, neste caso é invocada a função “troca_img” onde lhe é passado como parâmetro o nome da imagem que se pretende destacar.

```
function troca_img(nova_imagem)
{
    document.imagem.src=nova_imagem;
}
```

No caso de o produto ter mais do que três imagem associadas a ferramenta possui um sistema de navegação que permite avançar ou recuar uma unidade nas imagens apresentadas (Figura 5-15).



Figura 5-15 Galeria de Imagens (Netmove, Refral, 2007).

5.10 Manutenção automática da Base de Dados

O desempenho de uma página *Web*, (velocidade com que esta disponibiliza os dados presentes na BD), está muitas vezes directamente relacionado com o número de itens presentes na BD. Áreas como notícias, ofertas de emprego, apresentação de eventos, etc., estão constantemente a ser actualizadas, contudo na maior parte dos casos o responsável pela manutenção do serviço limita-se a inserir informação sem se preocupar em retirar os dados que estão fora de validade. Este tipo de manutenção provoca um aumento significativo da BD que por sua vez irá reflectir-se no tempo de resposta sempre que se efectua uma pesquisa na BD.

O módulo “Manutenção automática da Base de Dados” tem por objectivo eliminar o conteúdo de determinadas tabelas da BD quando o conteúdo nela presente expirou o prazo de validade definido pelo gestor do serviço.

O funcionamento deste módulo é descrito pelo fluxograma seguinte (Figura 5-16):

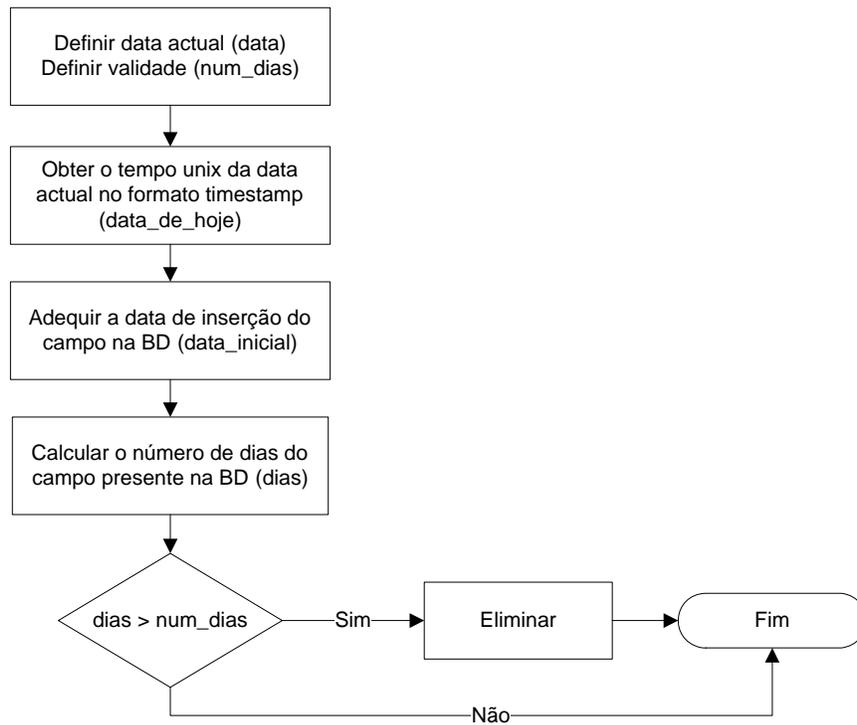


Figura 5-16 Fluxograma de manutenção automática da BD

5.11 Conclusão

Os módulos descritos neste capítulo partiram de soluções encontradas para problemas específicos no desenvolvimento das aplicações apresentadas. Essas soluções foram posteriormente reescritas tendo por base problemas gerais de forma a permitir a sua reutilização em futuras aplicações.

6 Eye-NET

Os sistemas de vídeo-vigilância têm tido um crescimento bastante elevado na última década, que se deve à enorme importância que tem sido dada às questões de segurança, de tal forma que nos dias de hoje existem instalados em bancos, transportes públicos, escritórios e até mesmo em residências particulares.

Uma grande percentagem dos sistemas instalados são sistemas simples, em que são usadas câmaras CCTV (*Close Circuit Television*) que, sendo colocadas estrategicamente são ligadas a um monitor controlado por um operador humano que em caso de emergência toma as medidas necessárias. Alguns destes sistemas possuem ainda a opção de gravação que permite registar imagens captadas num dado momento. A gravação pode ser feita recorrendo a dispositivos VCR, (*Video Cassette Recorder*), que permitem a gravação ininterrupta ao longo de várias horas ou mais recentemente usando dispositivos DVR (*Digital Video Recorder*) onde a cassete é substituída pelo disco duro, a imagem é digitalizada e comprimida permitindo, deste modo, que se efectue uma gravação contínua ao longo de vários dias. Com o aparecimento das câmaras IP, que possuem um servidor *Web* integrado, permitindo captar a imagem e distribuí-la pela rede sem precisar de equipamento auxiliar, os sistemas de vídeo-vigilância evoluíram de forma significativa. Deixou de ser necessária a presença de um operador humano no local de vigilância, uma vez que as câmaras podem ser acedidas de um qualquer local com acesso à Internet.

Tirando partido das principais vantagens das câmaras IP foi desenvolvida uma aplicação de gestão remota para visualização e gravação de imagens vídeo a que se designou por sistema “Eye-Net”.

O sistema Eye-Net pretende ser uma boa solução para quem precisa de vigiar um determinado local, independentemente do local onde se encontra. Recorrendo a câmaras IP e através de um interface construído à base das tecnologias e linguagens de programação apresentadas nesta dissertação e fazendo uso de alguns dos módulos desenvolvidos, o serviço Eye-Net permite que um determinado cliente tenha um número elevado de câmaras associadas à sua conta,

possibilitando a este a visualização de um ou mais locais de vigilância em simultâneo, como é ilustrado na Figura 6-1.

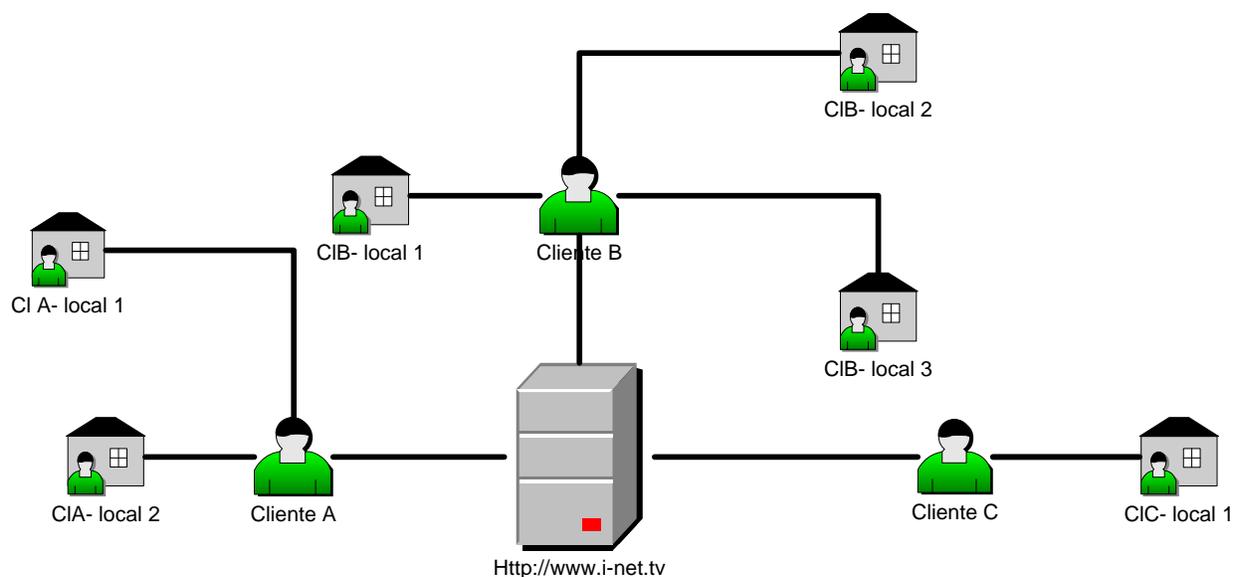


Figura 6-1 Diagrama Estrutural Eye-Net

6.1 Equipamento

As câmaras IP, devido às suas características permitem serem ligadas directamente a uma rede de Internet sem ser necessário o recurso a um computador. Este tipo de câmaras permite, para além da transmissão de vídeo, uma série de outras funcionalidades bi-direccionais tais como o controlo remoto de dispositivos eléctricos.

Dependendo das condições do local onde as câmaras se encontram, estas podem ligar-se à rede através de um cabo Ethernet. Neste caso, é possível usar câmaras com suporte a PoE (*Power Over Ethernet*), que utilizam o próprio cabo de rede como fonte de alimentação, permitindo a instalação destas câmaras em locais onde anteriormente não existia a ligação à rede eléctrica. As câmaras *wireless* são uma outra solução a aplicar no sistema com a vantagem de não necessitarem de uma ligação à rede Ethernet através de cabo.

O *Router* é um equipamento primordial na instalação do sistema, pois irá permitir que um cliente possa ter o sistema instalado e, ao mesmo tempo, poder aceder à Internet através de um outro

Equipamento. Nas secções 6.1.1 a 6.1.3 são descritas as principais características de alguns equipamentos usados.

6.1.1 Câmara Axis 207W

A câmara Axis 207W representada na Figura 6-2 (Axis, 2007) é uma câmara IP para uso interior de alta qualidade que possui uma série de características que possibilitam a sua utilização no sistema Eye-Net:

- Imagem de alta qualidade com resolução até 640x480 px,
- Possibilidade de ligação à rede sem fios ou cabo *Ethernet*,
- Microfone incorporado,
- Detecção de movimento,
- Entrada para ligação de alarme,
- Notificação de alteração de IP,
- Terminais I/O para ligação de dispositivos.

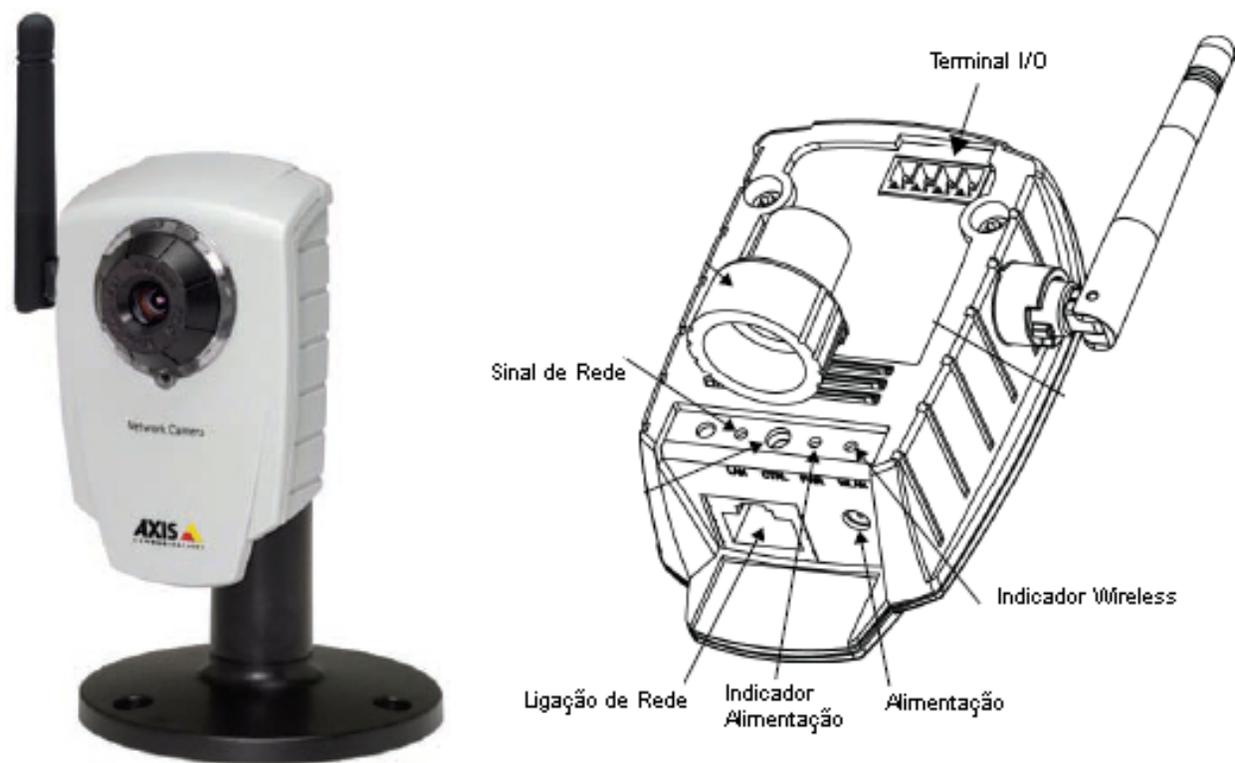


Figura 6-2 Câmara Axis 207W

Para além das características apresentadas esta câmara possibilita a transmissão de vídeo no formato MPEG-4 (*Moving Picture Experts Group-4*) (Axis, 2007) e Motion JPEG (*Joint Photographic Experts Group*) (Axis, 2007) permitindo desta forma a melhor utilização da largura de banda disponível.

- **Motion JPEG**

Este formato usa imagens JPEG que são enviadas em sequência de forma a criar a sensação de movimento. Embora a qualidade subjectiva da imagem seja bastante razoável implica um consumo de largura de banda elevado relativamente à largura de banda total disponível.

- **MPEG-4**

O formato MPEG-4 é um formato de compressão de áudio e vídeo que requer muito menos largura de banda do que o formato anterior à custa de perda da qualidade das imagens.

A Tabela 6-1, construída a partir do software disponibilizado pela câmara, apresenta a largura de banda necessária para uma câmara nas mesmas condições de utilização para os formatos de vídeo descritos anteriormente

Câmara Axis 207W					
N.º Câmaras	Resolução	Frames	Formato	Compressão	Largura de Banda
1	480x360	24	Motion JPEG	10	4256 Kbit/s
1	480x360	24	Motion MPEG 4	10	723 Kbit/s

Tabela 6-1 Cálculo da largura de banda

6.1.2 Câmara Pixord 205P

A câmara Pixord 205P representada na Figura 6-3 (PiXORD, 2007), é uma câmara IP para uso interior com as seguintes características:

- Ligação à rede por cabo Ethernet,

- Resolução de imagem até (640x480)px,
- Compressão de imagem nos formatos JPEG e Motion JPEG,
- Detecção de movimento,
- Programação de scripts de eventos,
- Protocolos TCP/IP, UDP, ARP, ICMP, HTTP, FTP; SMTP, DHCP.



Figura 6-3 Câmera Pixord 205P

Embora esta câmara não possua os requisitos da anterior poderá ser usada em ambientes onde o uso de cabos não seja um factor limitativo para a aplicação.

6.1.3 Router DI – 524

O *Router* DI-524 (D-LINK, 2004) da D-Link foi usado no sistema Eye-Net de forma a permitir a integração de várias câmaras no sistema. Este *Router* tem a particularidade de permitir a ligação de dispositivos sem fios e possuir um *Switch* incorporado que lhe permite ligar até quatro dispositivos com cabo. Esta última característica é muito importante para a instalação do sistema em residências particulares, na medida em que permite o uso simultâneo do sistema Eye-Net e da rede interna de computadores suportados pelo mesmo endereço IP.

As principais características deste dispositivo são:

- Alta velocidade sem fios,
- Firewall incorporado,
- *Switch* Ethernet integrado
- LAN sem fios com uma velocidade até 54 Mbps,
- Porta 10/100BASE-TX para ligação do modem DSL/cabo,
- Protocolo NAT com passagem por VPN,

- Bloqueio de domínios e planeamento,
- Configuração com base na Web.

A Figura 6-4 representa o interface do router DI-524.



Figura 6-4 Router DI - 524

6.2 BackOffice

O BackOffice é uma área reservada do sistema Eye-Net, que permite efectuar a gestão do serviço. Para que isso seja possível foi necessário implementar uma base de dados (BD) onde fica armazenada toda a informação referente ao sistema. A Figura 6-5 apresenta o modelo conceptual desta BD.

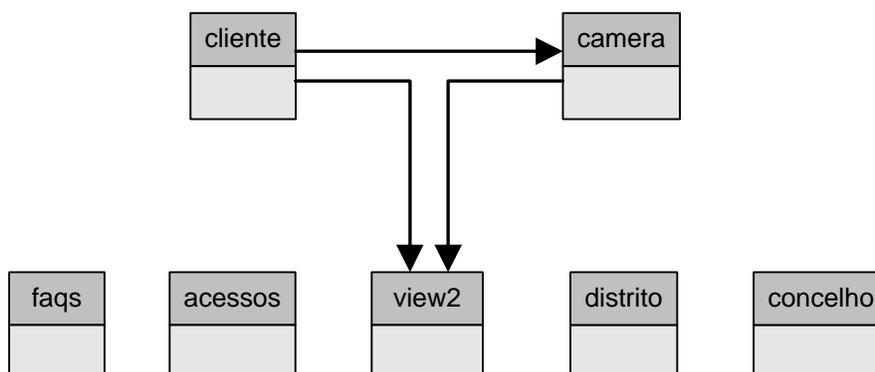


Figura 6-5 Modelo Conceptual de Base de Dados

A BD permite armazenar informações relativamente aos clientes, câmaras, acessos dos clientes ao serviço e *FAQS (Frequently Asked Questions)*.

O BackOffice está dividido em cinco áreas:

- **Home**

A página “Home” como o próprio nome indica é a página de entrada do sistema e como tal serve para informar o operador das áreas disponíveis e alguma ajuda apropriada para a correcta utilização do serviço.

- **Clientes**

A página “Clientes” é responsável pela gestão dos dados relativos a todos os clientes que subscrevam o serviço, nomeadamente dados pessoais (*nome, email, morada, telefone, NIF*), dados relativos ao sistema pretendido (*n.º de câmaras, router*) e ainda dados de segurança (*password, pergunta de recuperação de password, resposta de recuperação de password*).

A Figura 6-6 apresenta o interface da página de “clientes”.

Home	Clientes	Camaras	Impressao	Faqs
<input type="checkbox"/> NOME				
<input type="checkbox"/> António Prata	<input type="button" value="Editar"/>			
<input type="checkbox"/> Manuel Noronha	<input type="button" value="Editar"/>			
<input type="checkbox"/> Netmove	<input type="button" value="Editar"/>			
<input type="checkbox"/> Netmove Comunicação Global, Lda	<input type="button" value="Editar"/>			
<input type="button" value="Eliminar"/>				

Editar Cliente :	
Entidade	Netmove Comunicação Global, Lda
P. Contacto	Demonstração
Email	eye@netmove.pt
Morada	porto
Código Postal	4000 / 001 / porto
Distrito	Porto
Concelho	Óbidos
Telefone	111111434
Telefone SOS	32321321321
Data	2007 / 07 / 30 ex: aaaa / mm / dd
NIF	12323325
N.º Câmaras	6
Perfil	1
Router	Selecione o Router
Pergunta	A minha equipa de Futebol preferida
Resposta	benfica
Password	*****
Confirmação	*****
Estado	<input checked="" type="checkbox"/>
<input type="button" value="Enviar"/>	

Figura 6-6 Página “Clientes” do BackOffice Eye-Net

A área esquerda do ecrã é destinada a listar todos os clientes, enquanto a área direita do ecrã é utilizada para editar esses mesmos dados. De referir que um potencial cliente ao efectuar o registo na aplicação fica automaticamente inserido na BD e, portanto, irá aparecer na listagem de clientes. Contudo para o cliente ter acesso ao sistema terá de lhe ser dada autorização pelo gestor do serviço, que acontecerá depois de confirmados os dados do cliente e instalado o serviço no local pretendido.

- **Câmaras**

Esta área tem como finalidade a manutenção dos dados relativos às câmaras de cada cliente. A Figura 6-7 apresenta o *interface* da área “camaras”.

Porta	Local	Marca	Cliente		
<input type="checkbox"/>	5002	2	axis	Netmove	<input type="button" value="Editar"/>
<input type="checkbox"/>	5003	1	axis	Netmove Comunicação Global, Lda	<input type="button" value="Editar"/>
<input type="checkbox"/>	5004	1	pixord	Netmove Comunicação Global, Lda	<input type="button" value="Editar"/>
<input type="checkbox"/>	5001	2	axis	Netmove Comunicação Global, Lda	<input type="button" value="Editar"/>
<input type="checkbox"/>	5002	1	axis	Netmove Comunicação Global, Lda	<input type="button" value="Editar"/>
<input type="checkbox"/>	80	3	axis	Netmove Comunicação Global, Lda	<input type="button" value="Editar"/>
<input type="checkbox"/>	5001	1	pixord	Netmove Comunicação Global, Lda	<input type="button" value="Editar"/>

Figura 6-7 Página “Camaras” do BackOffice Eye-Net

Mais uma vez, por questões de optimização de espaço e de desempenho, o lado esquerdo do ecrã é reservado para a listagem das câmaras, enquanto o lado direito se destina à edição dos dados. Só será permitida a inserção de novas câmaras depois de seleccionado o cliente.

- **Impressão**

A instalação do serviço requer o conhecimento de uma série de informação relativa ao cliente, ao tipo de ligação à Internet que ele possui, aos locais de instalação e ainda acerca das câmaras. Torna-se útil então disponibilizar num único documento, toda a informação necessária acerca do cliente para que o instalador do sistema possua todos os dados necessários. A configuração da página é feita recorrendo a um ficheiro CSS do tipo “print” permitindo dessa forma configurar a

página de impressão de forma agradável. Esse ficheiro deve ser incluído no página HTML recorrendo à sintaxe apresentada a seguir.

```
<LINK rel="stylesheet" type="text/css" href="print.css" media="print">
```

- **FAQS**

Quando se pretende desenvolver um sistema que irá ser utilizado por diversos utilizadores, é aconselhável existir uma forma de solucionar problemas o mais rapidamente possível. A área das “FAQS” tem por objectivo disponibilizar *online* e em tempo real uma série de questões com a respectiva solução para que qualquer utilizador possa, de uma forma rápida e prática consultar e verificar se ali se encontra a resposta ao seu problema. A Figura 6-8 apresenta a área das FAQS.

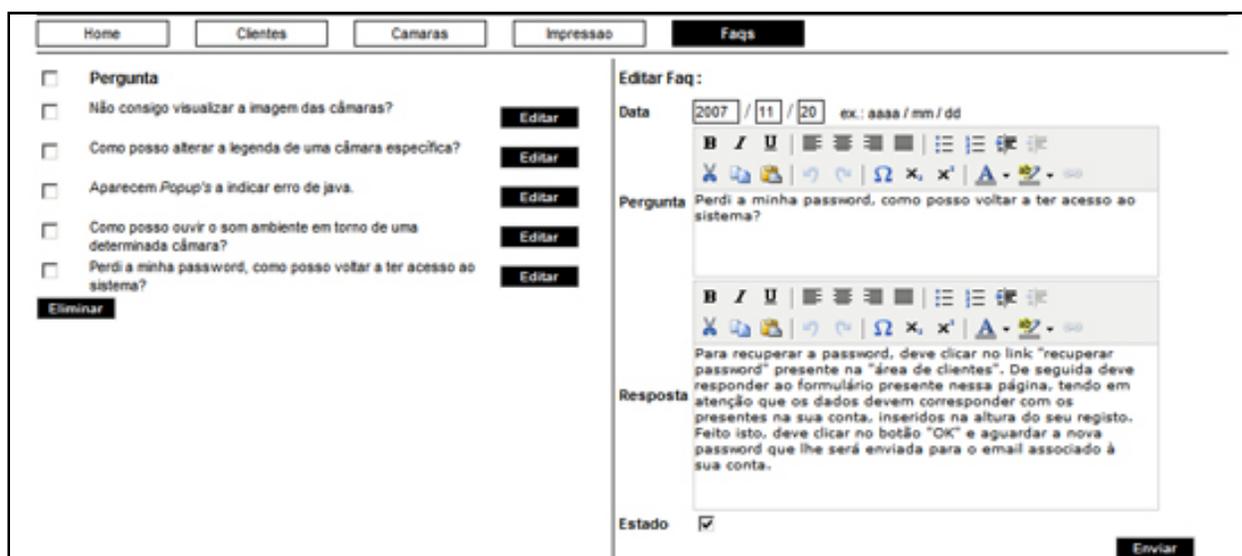


Figura 6-8 Página “Faq’s” do BackOffice Eye-Net

6.3 FrontOffice

O sistema de vídeo-vigilância Eye-Net, foi integrado numa página de Internet de forma a incorporar uma série de funcionalidades necessárias para a operação do sistema. As páginas “Empresa” e “O que é o Eye-Net?” são meramente institucionais, ou seja, não são essenciais para o funcionamento do sistema, as restantes são fundamentais para que este funcione nas devidas condições. A Figura 6-9 apresenta o mapa do sítio onde está inserido o sistema.

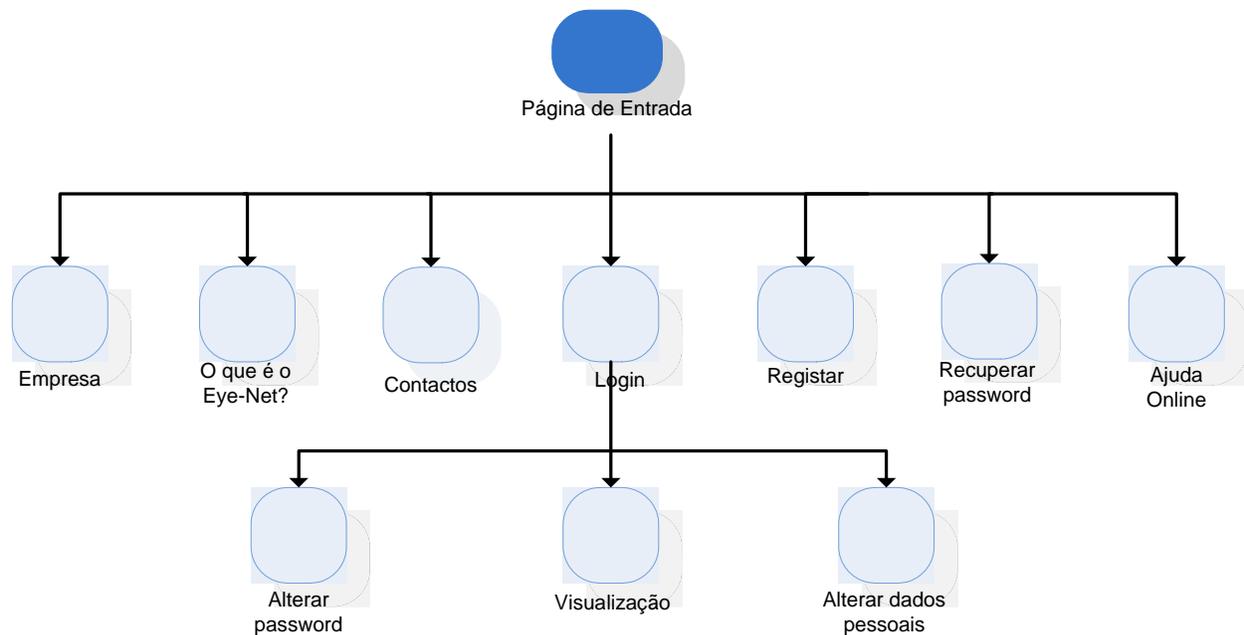


Figura 6-9 Mapa do Sítio do sistema Eye-Net

Pode-se verificar que, enquanto algumas áreas estão acessíveis logo que se entra no sistema, outras só estão disponíveis depois de se efectuar um processo de autenticação.

Para um potencial cliente poder subscrever o serviço deve entrar na página “registar” e aí preencher o formulário existente. Essa informação será inserida na BD de suporte ao sistema e o gestor deste será informado por email que recebeu um pedido de subscrição, da mesma forma o cliente receberá um email a informar da correcta recepção da informação por parte do gestor juntamente com os dados de acesso ao sistema.

Para que o cliente possa ter acesso ao serviço, terá que aguardar que a sua informação passe por um processo de verificação. Se necessário, será contactado pelo gestor e, depois de instalado o serviço no local requisitado pelo cliente, este terá então acesso ao serviço.

Ao efectuar o *login*, o cliente poderá visualizar as imagens captadas por cada uma das câmaras, em três modos de visualização diferentes. O cliente pode alternar entre cada um desses modos no painel de controlo, clicando na imagem associada a cada modo de visualização.

- **Modo de visualização 1**

O “Modo de visualização 1”, permite visualizar a imagem de uma câmara em cada instante, sendo o modo que permite captar a imagem com mais detalhe. A área central da página apresenta a imagem captada pela respectiva câmara com uma dimensão de (640x480)px. Ao entrar nesta página é seleccionada por omissão a primeira câmara do cliente, o cliente pode, no entanto, modificar a câmara visível, no canto inferior direito da imagem. A Figura 6-10 ilustra o interface deste modo de visualização.

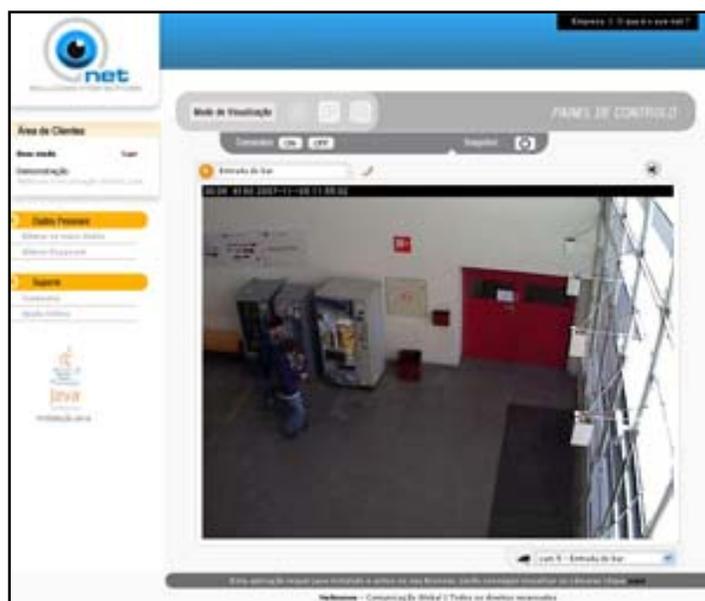


Figura 6-10 Modo de visualização 1

Cada câmara tem uma legenda associada que, caso o cliente assim o entenda, poderá servir para indicar o local onde a câmara se encontra. O cliente pode alterar a legenda através do botão de edição no painel de controlo. Ainda neste modo de visualização o cliente pode activar o áudio da câmara seleccionada (*caso esta o permita*), captar uma imagem estática de um determinado momento (*clicando no botão “Snapshot”*).

O cliente pode ainda activar dispositivos eléctricos ligados directamente a cada uma das câmaras clicando no botão “ON” presente no painel de controlo, caso a câmara possua algum objecto a ela ligado (*e.g. lâmpada, ventoinha, computador*).

- **Modo de visualização 2**

Ao seleccionar o “Modo de visualização 2”, o cliente irá ter acesso a uma área que disponibilizará a visualização de um máximo de 4 câmaras, sendo que as imagens têm uma dimensão de (320x240)px. Este painel tem a particularidade de permitir que o cliente personalize quais as câmaras visíveis num determinado momento. Isso é possível seleccionando a câmara pretendida para cada uma das 4 posições disponíveis no canto inferior direito de cada imagem. Essa informação será actualizada na BD o que irá permitir manter essa configuração na próxima vez que o cliente entrar no sistema e até que este decida alterar essa configuração. A Figura 6-11 apresenta o modo de visualização 2.

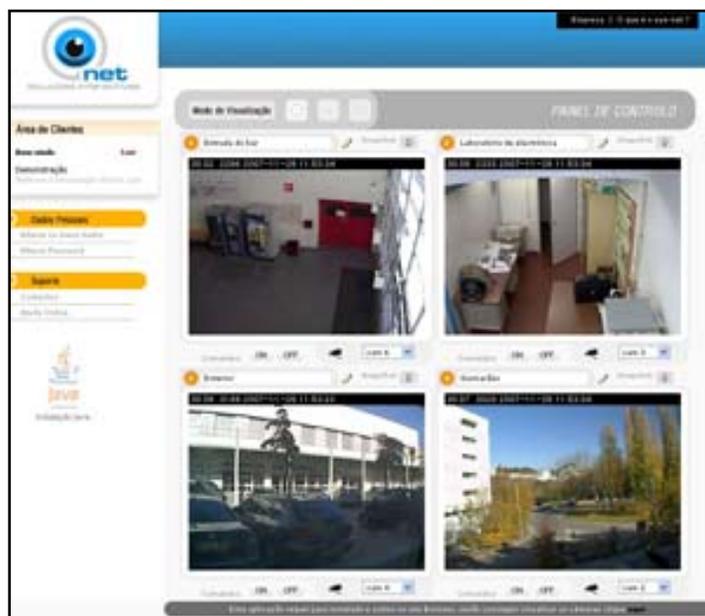


Figura 6-11 Modo de visualização 2

O Cliente pode alterar a legenda de cada uma das câmaras activas neste modo à semelhança do “Modo de visualização 1”. Pode ainda activar um dispositivo eléctrico, ligado a uma câmara, e captar uma imagem estática da cada câmara individualmente. Caso o cliente pretenda visualizar a imagem de uma determinada câmara com mais detalhe, deve clicar no número associado à posição da respectiva câmara.

- **Modo de visualização 3**

Caso o cliente pretenda visualizar numa só página todas as câmaras associadas ao sistema deve seleccionar o modo de visualização 3. Aqui poderá observar as imagens captadas por todas as câmaras com uma dimensão de (160x120)px. Este modo de visualização não tem funcionalidades associadas. Contudo permite que o cliente possa mudar directamente para o “modo de visualização 1”, clicando na identificação da respectiva imagem presente na Figura 6-12 que ilustra o interface deste modo de visualização.

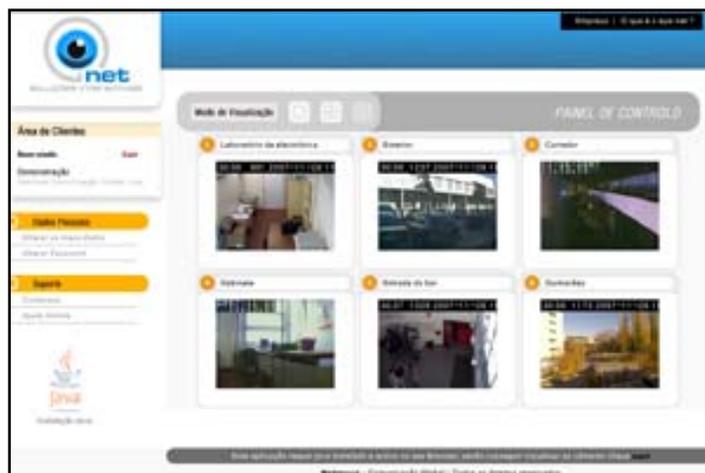


Figura 6-12 Modo de visualização 3

Sempre que o cliente alterar o modo de visualização, o sistema encarrega-se de efectuar a verificação do estado de cada uma das câmaras do cliente. Este processo é feito recorrendo à função “ping” descrita no código abaixo. Esta função faz uso da função PHP “fSOCKOPEN” que cria um socket entre a página e cada uma das câmaras. Se a câmara estiver desligada ou com problemas de comunicação a função irá retornar o valor “false” e no lugar da imagem captada pela câmara aparecerá um aviso a indicar que a câmara se encontra desligada.

```
function ping($url, $porta, $str, $int){
    @$socket = fsockopen($url, $porta, $int, $str, 5);
    if($socket){
        $state= 't';
    }
    else{
        $state = 'f';
    }
    return $state;
}
```

6.3.1 Colocação das imagens das câmaras nos diversos modos de visualização

A câmara IP capta a imagem e disponibiliza um ficheiro para que se possa exibir a imagem dentro da página HTML. Para executar esse processo recorre-se a uma *applet java (código seguinte)*, onde se indica o endereço do ficheiro de vídeo disponibilizado pela câmara e algumas variáveis de configuração.

```
<SCRIPT LANGUAGE="JavaScript">
var BaseURL = ip + porta + "/";
var Width = "largura";
var Height = "altura";
var File = "axis-cgi/mjpg/video.cgi?resolution="+Width+"x"+Height+ "&compression=40&color=1";
var output = "";
var AppletDir = BaseURL + "java/ama";
var VideoPath = "axis-cgi/mjpg/video.cgi?resolution=";
var restOfPath = "&compression=40&color=1&clock=1&date=1&text=1";
var VideoURL = BaseURL + VideoPath;
output = '<APPLET archive="ama.jar" codeBase="';
output += AppletDir + "'";
output += ' code="ama.MediaApplet" height=';
output += Height;
output += ' width=';
output += Width;
output += '>';
output += '<PARAM NAME="code" VALUE="ama.MediaApplet">';
output += '<PARAM NAME="archive" VALUE="ama.jar">';
output += '<PARAM NAME="codebase" VALUE="';
output += AppletDir + "'>';
output += '<PARAM NAME="ama_cgi-path" VALUE="axis-cgi">';
output += '<PARAM NAME="cache_option" VALUE="No">';
output += '<PARAM NAME="cache_archive" VALUE="ama.jar">';
output += '<PARAM NAME="cache_version" VALUE="0.92.1.0">';
output += '<PARAM NAME="ama_plugins" VALUE="">';
output += '<PARAM NAME="type" VALUE="application/x-java-applet;version=1.4">';
output += '<PARAM NAME="ama_url" VALUE="';
output += VideoURL;
output += Width + 'x' + Height;
output += restOfPath + "'>';
output += '</APPLET>'
document.write(output);
</SCRIPT>
```

6.3.2 Princípio de funcionamento do Eye-Net

Ao aceder ao serviço Eye-Net através da sua conta, o cliente terá acesso às imagens captadas por cada uma das câmaras associadas. Essas câmaras poderão estar num único local, ou em locais fisicamente independentes. Isso é possível porque a BD (Figura 6-13) foi construída de forma que a uma câmara é associado um cliente e o local onde esta se encontra, permitindo, deste modo, que o cliente possa visualizar vários locais em simultâneo.

camera
id_camera
id_cliente
ip
local
...

Figura 6-13 Tabela câmara

Ao digitar <http://www.i-net.tv>, o cliente está a aceder ao servidor onde o sistema está alojado (Figura 6-14, linhas azuis), devendo então efectuar o procedimento de *login*. Caso este seja bem sucedido o sistema vai percorrer a base de dados e procurar todas as câmaras associadas ao cliente em questão e vai disponibiliza-las *online*.

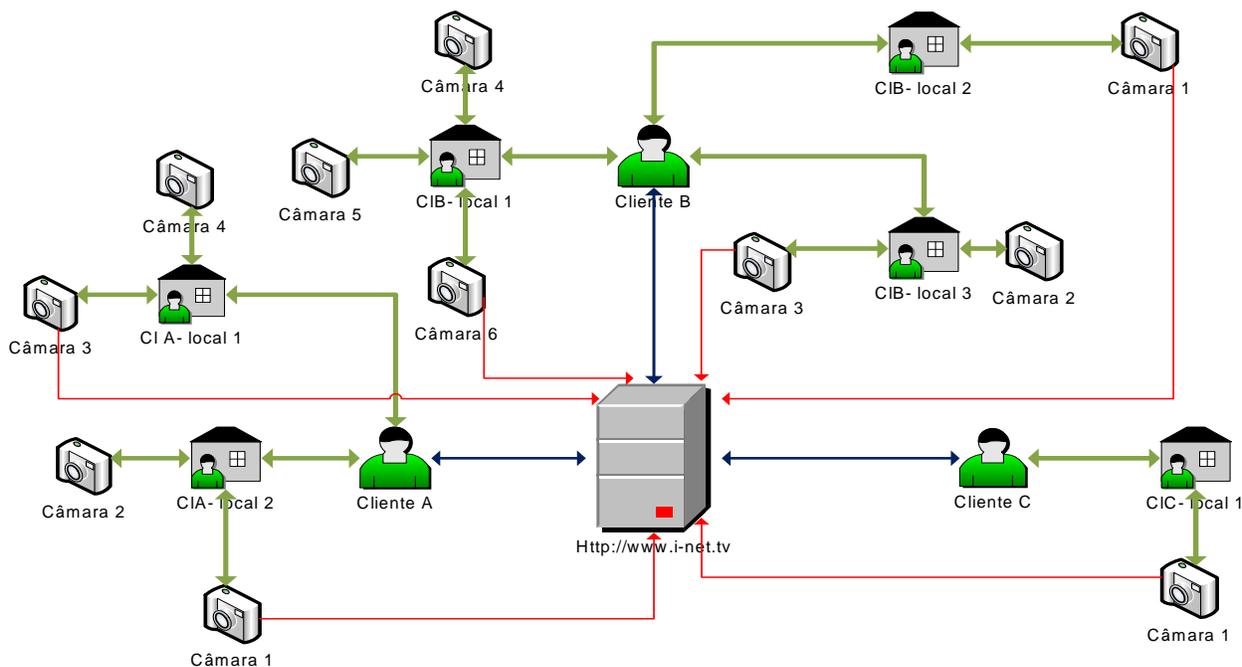


Figura 6-14 Diagrama de fluxo de informação

Na maior parte dos casos os locais onde as câmaras são instaladas não possuem um IP fixo, ou seja, o IP associado a cada uma das câmaras irá ser alterado, de tempos a tempos, sendo assim pode colocar-se a questão: *Como sabe o sistema qual o IP de cada câmara em determinado momento?*

Isso é feito recorrendo a um serviço disponibilizado pela câmara “Notificação de alteração de IP” que consiste em enviar uma notificação HTTP, a indicar que o IP da câmara em questão foi alterado. Por razões de desempenho e segurança, é suficiente que apenas uma câmara de cada local (*câmaras com o mesmo IP*) envie essa informação, embora todas o possam fazer (Figura 6-14, linhas vermelhas). A câmara envia a informação ao sistema a indicar que o seu IP foi alterado e o sistema, de forma automática, actualiza o novo IP para todas as câmaras desse local na BD e disponibiliza a imagem no PC do cliente. A partir deste momento o cliente pode visualizar todas as câmaras, sendo que o tráfego de visualização realiza-se entre câmara e cliente (Figura 6-14, linhas verdes) sem passar pelo servidor.

6.3.3 Actuação remota de dispositivos

Uma das características de algumas câmaras IP é a de permitirem a ligação remota de dispositivos eléctricos através dos terminais I/O da câmara. Esses dispositivos, (*e.g. lâmpadas, ventoinhas, computadores, sistemas de rega, alarmes*), são ligados à câmara conforme é demonstrado na Figura 6-15 e accionados a partir do sistema através de um botão que activa a saída da câmara.

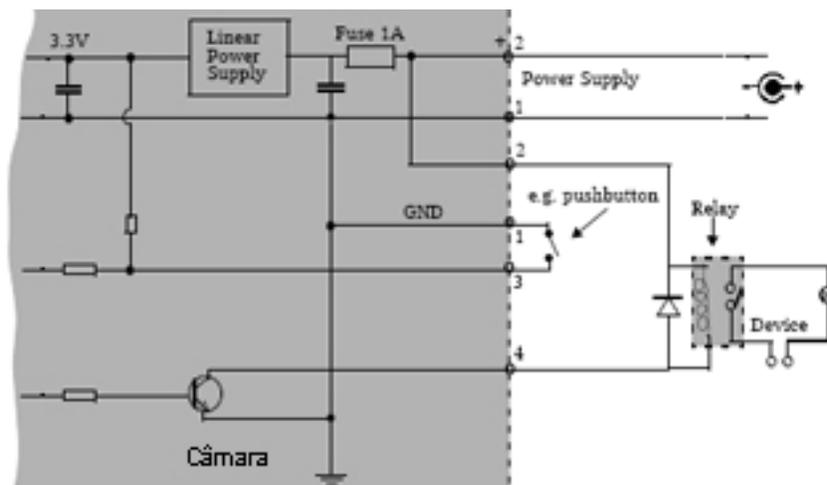


Figura 6-15 Esquema de montagem para ligação de dispositivo

6.4 Conclusão

Os objectivos propostos para a primeira fase do desenvolvimento do sistema Eye-Net foram atingidos na sua totalidade, no entanto, pretende-se, num futuro próximo tornar o sistema independente das câmaras utilizadas. Um outro objectivo para este sistema é a integração da possibilidade de gravação de vídeo e áudio quer localmente (*através de um computador ligado à rede*), quer remotamente (*accedendo ao sistema de um outro local*).

7 Comércio Electrónico

A venda de produtos através da Internet tem aumentado desde a sua criação. Os clientes verificaram que as compras através deste método são mais rápidas e cómodas uma vez que os serviços de Internet estão “abertos” 24h por dia. Os “tabus” relativamente à segurança parecem estar ultrapassados, devido ao aparecimento de serviços, como o *PayPal*, que garantem bastante fiabilidade a este serviço.

O segundo exemplo de aplicação dos módulos desenvolvidos nesta dissertação refere-se a uma editora *online*. A Editora consiste num sistema *Web* concebido para publicar e vender livros. Este sistema faz uso do módulo desenvolvido no capítulo 5, “Carrinho de Compras” e permite o pagamento *online* através da incorporação do *PayPal*.

A Figura 7-1 representa o diagrama de blocos do serviço “Editora”. Neste capítulo serão abordadas todas as áreas associadas ao serviço exemplificando o modo de funcionamento e a utilidade de cada uma delas para o sistema.

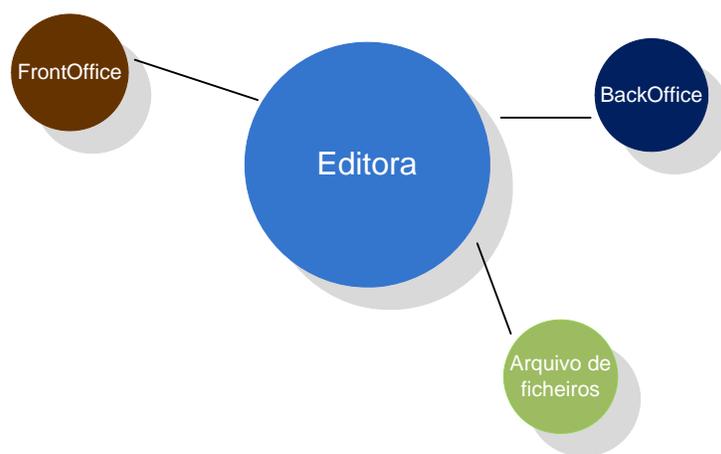


Figura 7-1 Diagrama de blocos Editora

7.1 Arquivo de ficheiros

A área “Arquivo de ficheiros” tem a responsabilidade de armazenar todo o tipo de ficheiros necessários para o correcto funcionamento do sistema (*e.g. imagens, pdf’s, vídeos*). Na verdade, esta área funciona como que um repositório de ficheiros do sistema, que se encontra dividido em directorias de modo a que os ficheiros armazenados sejam agrupados por tipo. Esses ficheiros são copiados através de uma ferramenta específica desenvolvida para essa finalidade e implementada no BackOffice.

7.2 BackOffice

Esta área permite gerir o conteúdo das áreas dinâmicas do serviço. Nesta área, um utilizador de *BackOffice* pode actualizar a informação do serviço que está sob a sua responsabilidade. Essa informação irá permitir actualizar em tempo real a área acessível pelos utilizadores. A Figura 7-2 ilustra o modelo conceptual da base de dados (BD) de suporte ao *BackOffice*.

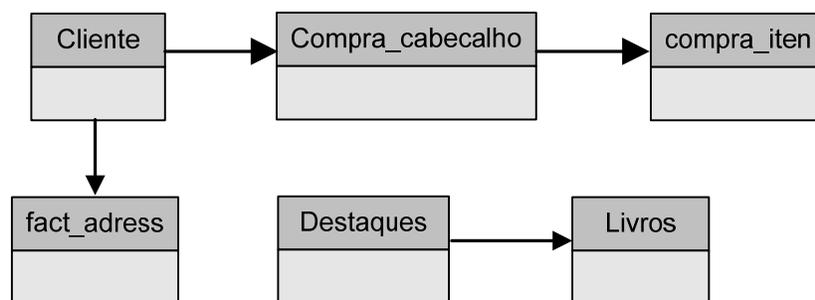


Figura 7-2 Base de Dados Editora

Para armazenar a informação foi necessário implementar a BD da Figura 7-2. Esta BD permite guardar informação sobre os clientes que subscrevem o serviço e as respectivas compras efectuadas. Permite também guardar informação sobre os livros que deverão aparecer disponíveis para os clientes.

A BD é então “alimentada” pelo *BackOffice* que está dividido em cinco áreas independentes, conforme se pode verificar pela Figura 7-3.

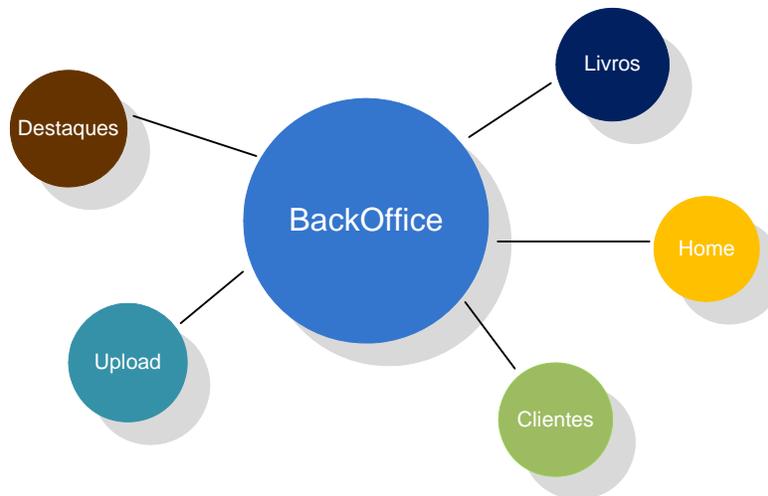


Figura 7-3 Diagrama de Blocos BackOffice – Editora

7.2.1 Home

A página “Home” é a página de entrada da área de *BackOffice*. Esta página tem como objectivo apresentar todas as funcionalidades disponíveis e efectuar uma apresentação do sistema. O menu existente nesta página estará presente em todas as páginas do *BackOffice* de forma a facilitar a navegação entre as diferentes opções.

7.2.2 Upload

A Figura 7-4 apresenta a página de “Upload”.



Figura 7-4 Página de upload de ficheiros do sistema Editora

A secção “Upload” permite ao gestor do serviço copiar ficheiros guardados num computador para o servidor onde está alojado o sistema. Desta forma, é possível associar esses ficheiros com os produtos que se pretendem disponibilizar *online*. Na aplicação aqui descrita foi apenas necessário permitir o *upload* de imagens.

O formulário HTML, presente no lado esquerdo do ecrã, permite ao gestor seleccionar três imagens guardadas localmente e efectuar o upload para o servidor. Caso não seja detectado nenhum problema, como por exemplo uma das imagens seleccionadas já estar presente na directoria de destino, irá aparecer do lado direito do ecrã uma miniatura da imagem copiada para o servidor.

No lado direito do ecrã o gestor pode então, em qualquer altura, ter acesso a uma listagem com as imagens associadas ao sistema e, caso este assim o pretenda, eliminar as imagens que não deseja utilizar no sistema.

7.2.3 Clientes

A secção “Clientes” (Figura 7-5) é responsável pelo tratamento da informação referente a todos os utilizadores que subscrevam o serviço.

Nome	Editar
<input type="checkbox"/> NOME	
<input type="checkbox"/> Helder Feliciano Almeida da Costa	Editar
<input type="checkbox"/> José Cabral	Editar
<input type="checkbox"/>	Eliminar

Editar Utilizador:	
Nome	Helder Feliciano Almeida da Costa
Empresa	Teste
Email	helder@teste.pt
Morada	Avenida 24 de Junho, n.º 2003 1º MR
Código Postal	4800 206 Azurém
Telefone	963182354
cel	122406789
Password	****
Confirmação	****
Data	2007 / 08 / 21 em: 00:00 / 00 / 00
Estado	<input checked="" type="checkbox"/>
Enviar	

Figura 7-5 Página de clientes do sistema Editora

Ao subscrever o serviço o utilizador é automaticamente inserido na base de dados de suporte ao sistema e o seu nome irá aparecer na listagem de clientes do lado esquerdo do ecrã. O gestor do sistema será então informado por email que um novo utilizador efectuou o registo. Ao receber o email, o gestor dirige-se a esta secção do BackOffice e na listagem de clientes deve efectuar uma de duas opções:

- **Editar**

Os dados do utilizador irão aparecer no formulário do lado direito do ecrã, permitindo deste modo ao gestor efectuar a confirmação dos dados do utilizador. Caso os dados estejam correctos o gestor poderá validar a subscrição deste utilizador.

- **Eliminar**

Caso o gestor não tenha intenção de validar o registo deste utilizador, deve seleccionar a “checkbox” referente a este utilizador e clicar no botão “Eliminar”. Desse modo, o utilizador será eliminado da BD.

7.2.4 Livros

Os livros disponíveis para compra são inseridos através da secção “Livros”, Figura 7-6. Esta página segue a estrutura das restantes páginas do *BackOffice* para que a ferramenta seja fácil de utilizar. Assim, do lado esquerdo do ecrã está presente uma listagem de todos os livros presentes na BD e que poderão aparecer *online*. O lado direito do ecrã é reservado para o formulário que permite adicionar novos livros ou editar toda a informação referente aos livros já inseridos na BD.



Figura 7-6 Página Livros do sistema Editora

7.2.5 Destaques

A secção do *BackOffice* “Destaques”, Figura 7-7 permite que o gestor controle quais os livros que devem aparecer nessa área de destaque. Do lado esquerdo do ecrã está disponível uma listagem com todos os livros do sistema. Para destacar um desses livros o gestor deverá seleccionar a “checkbox” referente ao livro pretendido. O lado direito do ecrã é reservado para a listagem dos livros que estão actualmente em destaque. Para remover esses livros deve-se seleccionar a “checkbox” do respectivo livro e clicar no botão “eliminar”. Esta opção não irá eliminar o livro da BD, apenas irá retirá-lo de destaque.



Figura 7-7 Página destaques do sistema Editora

A Figura 7-8 apresenta o fluxo de informação quando se utiliza o BackOffice do sistema.

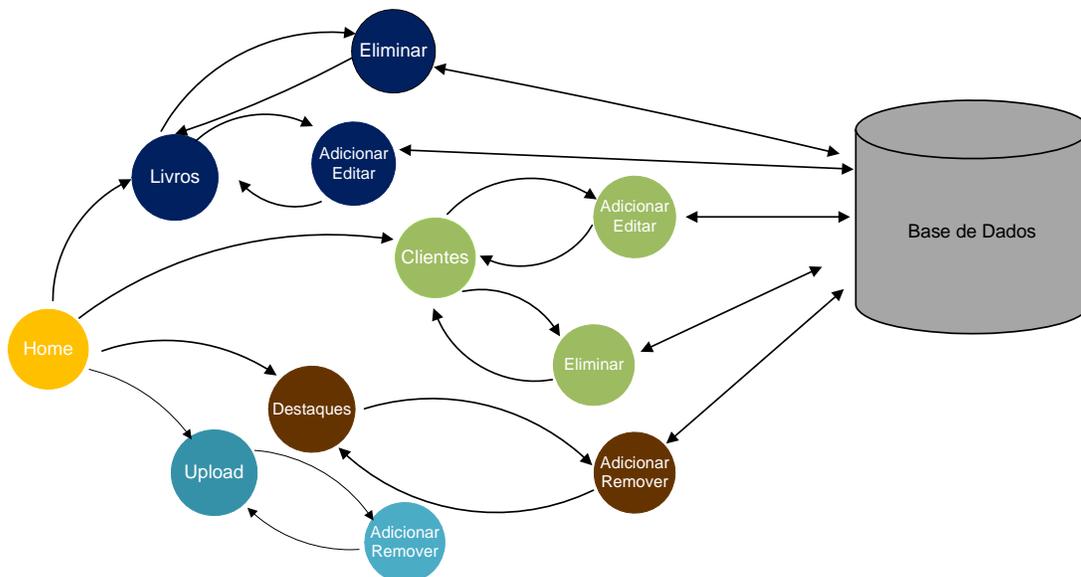


Figura 7-8 Diagrama de fluxo - BackOffice editora

Através deste diagrama pode verificar-se a forma como cada umas das diferentes secções do BackOffice trabalha para actualizar a Base de Dados do sistema.

7.3 FontOffice

O *BackOffice* é uma área com acesso restrito, onde apenas o gestor do serviço pode ter acesso. Por sua vez, o *FrontOffice* é a área do sistema que vai estar disponível para todos os utilizadores. O modelo da Figura 7-9 apresenta, de uma forma geral, a estrutura actual do sistema Editora, em que cada bloco identifica cada uma das suas áreas principais. A seguir são descritos, com mais detalhe, cada um dos blocos apresentados.

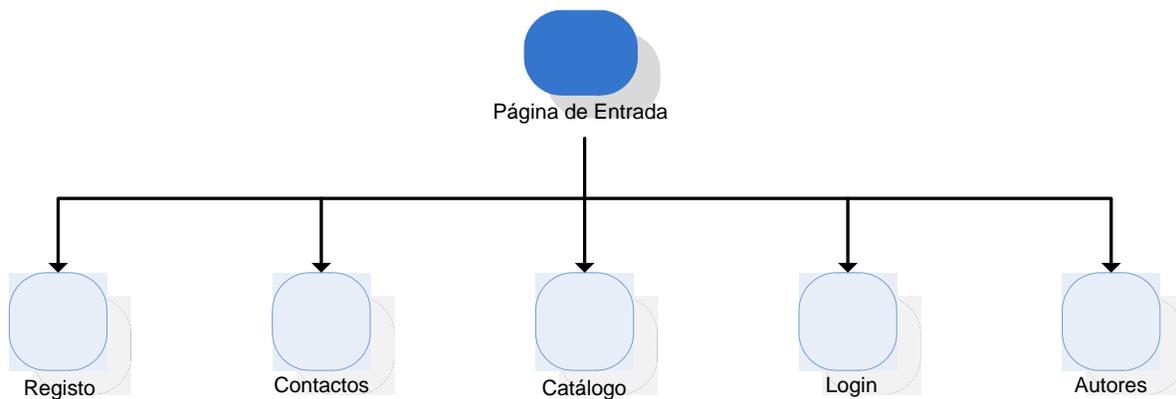


Figura 7-9 Mapa do sítio Editora

- **Página de entrada**

A página de entrada permite o acesso directo a todas as áreas que compõem o sistema. Esta página é composta por um texto descritivo do sistema e permite ainda o destaque de produtos. Esses produtos em destaque podem ser alterados através do BackOffice.

- **Contactos**

A secção de contactos tem como finalidade fornecer aos utilizadores os dados necessários para que estes possam contactar a empresa responsável pelo serviço. No entanto, os utilizadores têm a possibilidade de contactar directamente com a empresa através do formulário HTML presente nesta página.

- **Autores**

A página “Autores” não permite qualquer interacção entre os utilizadores e o sistema, trata-se de uma página estática, ou seja, o conteúdo nela presente não vai ser alterado com muita frequência. Esta página destina-se a informar os utilizadores de como proceder caso pretendam editar alguma obra através do serviço.

- **Registo**

O sistema “Editora” é de acesso livre, ou seja, qualquer utilizador consegue aceder a este sistema sem precisar autenticar-se. Contudo caso o utilizador deseje efectuar alguma compra neste serviço terá de efectuar o registo no mesmo. A Figura 7-10 ilustra a página de “Registo”.

Este formulário permite-lhe efectuar o registo na Editora Netmove e efectuar a compra de artigos Online.

Nome (*)	Helder Feliciano Almeida da Costa
Empresa	Netmove
Morada (*)	Avenida 24 de Junho, n.º 2203 1º MB
Cód Postal (*)	4800 - 4800
Localidade (*)	Azúem
Telefone (*)	963108354
E-mail (*)	helder@netmove.pt
Password (*)	****
Confirmar (*)	****
Contribuinte (*)	123456789

[*) Campo Obrigatório

Enviar Limpar

Figura 7-10 Página de registo sistema Editora

A secção “Registo” disponibiliza um formulário que permite que um utilizador efectue o registo ou altere os seus dados no caso de já se encontrar registado. Ao preencher o formulário os dados do utilizador serão automaticamente inseridos na BD e o gestor será informado, por email, que um novo utilizador efectuou registo no serviço.

- **Login**

A página de *login* permite que um utilizador que se encontre registado e devidamente validado pelo gestor possa efectuar compras no sistema “Editora”. Para efectuar o *login* o utilizador deve inserir o seu email e *password* no formulário presente nesta página (Figura 7-11); caso os dados estejam correctos irá aparecer no ângulo superior direito da página a identificação do cliente e o estado do carrinho de compras.



Figura 7-11 Página de login sistema Editora

- **Catálogo**

A página de “Catálogo” tem a responsabilidade de apresentar aos utilizadores todos os livros presentes na BD. Cada livro é apresentado pelo nome, uma imagem em miniatura da sua capa (*caso exista*), o nome do autor e o preço do livro. O sistema permite que estejam presentes na BD livros que ainda não foram editados. Neste caso, o livro irá aparecer nesta página com o estado de “disponível brevemente”. Esta página permite a apresentação de 9 livros em simultâneo (3x3). Foi implementado nesta página o módulo “Sistema de navegação”, descrito na secção 5 de forma a permitir a visualização de todos os livros no caso de o número de livros presentes na BD ser superior a 9.

Nesta página (Figura 7-12) o utilizador pode optar por adicionar o livro pretendido ao carrinho de compras caso o pretenda adquirir, ou pode ver a informação mais detalhada acerca de cada um dos livros clicando sobre a imagem do livro respectivo.



Figura 7-12 Página de catálogo sistema Editora

7.4 Conclusão

A aplicação apresentada neste capítulo visa demonstrar a possível utilização de alguns dos módulos desenvolvidos no capítulo 5, nomeadamente dos módulos “Criação de acessos protegidos a utilizadores”, “Carrinho de compras”, “Protecção de formulários” e “Galeria de imagens”. A utilização desses módulos permitiu a implementação do serviço no espaço de tempo mais reduzido.

8 Conclusões

Quando se desenvolvem aplicações para uso na *Web* deve-se ter sempre em mente que quando surge um problema provavelmente já houve alguém que se deparou com um semelhante. A primeira fase para a solução de um problema deve ser a pesquisa de eventuais soluções já implementadas e testadas que possam ajudar na sua resolução. Por outro lado, na fase de análise para o desenvolvimento de uma aplicação *Web* é importante que se tome como referência casos gerais para que a solução encontrada possa ser aplicada em diferentes situações. Um exemplo concreto que demonstra esta situação é o módulo descrito nesta dissertação “catálogo de produtos”. Após uma primeira análise chegou-se à conclusão que o problema poderia ser resolvido por dois modelos distintos. O primeiro baseava-se na ideia de que o catálogo de produtos de uma empresa era organizado recorrendo a uma divisão de família, subfamília e produto. No entanto conclui-se que este modelo sendo bastante rígido não permitia a sua aplicabilidade em todas as situações. Apresentou-se então um novo modelo baseado numa estrutura em árvore que possuía total flexibilidade permitindo deste modo estruturar o catálogo de produtos de qualquer empresa.

Os módulos apresentados nesta dissertação foram desenvolvidos com o objectivo de solucionar problemas que surgiram no desenvolvimento das duas aplicações apresentadas, mas que pudessem ao mesmo tempo ser usados em aplicações futuras. Para tal, adoptaram-se soluções modulares para que certos desenvolvimentos pudessem se reutilizados noutros projectos.

O trabalho desenvolvido foi suportado à base das linguagens HTML, PHP, Javascript e CSS com suporte a base de dados PostgreSQL, sendo que o número de tecnologias e linguagens disponíveis para o desenvolvimento de aplicações *Web* é bastante grande conforme foi referido ao longo da dissertação. A escolha dessas tecnologias deve ficar ao critério de cada um, tendo em conta a aplicação desenvolvida.

Os módulos desenvolvidos foram utilizados nas duas aplicações apresentadas sendo que alguns foram usados apenas numa delas como por exemplo o “carrinho de compras” que apenas foi usado no sistema de “comércio electrónico”, enquanto outros foram usados nas duas aplicações,

como por exemplo o módulo “Criação de acessos protegidos a utilizadores” ou “protecção de formulários”. Os objectivos propostos foram cumpridos na medida em que o desenvolvimento desses módulos proporcionou a resolução dos problemas encontrados nas duas aplicações desenvolvidas.

Os diferentes *browsers* (*Internet Explorer, Mozilla Firefox ou Opera*) têm certas incompatibilidades entre si, nomeadamente no que se refere à formatação das páginas HTML. Quando se recorre ao uso de CSS para efectuar a formatação das páginas é necessário testar as aplicações nos diferentes browsers de forma a garantir que o código desenvolvido se comporta da forma pretendida. Nesse ponto os objectivos foram também conseguidos na medida em que as soluções implementadas foram testadas nos três *browsers* enumerados acima.

As duas aplicações desenvolvidas e descritas nesta dissertação foram devidamente testadas e encontram-se em funcionamento, o que comprova a eficácia das soluções adoptadas, contribuindo dessa forma para o cumprimento dos objectivos propostos nesta dissertação.

8.1 Trabalho Futuro

O trabalho futuro passa por continuar a desenvolver novos módulos para aplicação *Web* que permitam a sua utilização nas mais diversas aplicações. Esse desenvolvimento deve ser sempre precedido de um estudo das linguagens e tecnologias disponíveis no momento, para que, de entre as possíveis se seleccione as que permitam desenvolver a aplicação com custos mais reduzidos sem que isso signifique perda de qualidade da aplicação.

Relativamente ao sistema “Eye-Net” apresentado nesta dissertação, convém salientar que se encontra optimizado para duas marcas de câmaras IP. O que se pretende atingir com este sistema é que seja o mais universal possível, estando preparado para qualquer tipo de câmara de vigilância, incluindo a capacidade de aproveitar soluções já implementadas com base em câmaras CCTV. Neste momento o sistema “Eye-Net” tem a capacidade de captar uma imagem estática de um dado momento (*Snapshot*). Pretende-se implementar no sistema a capacidade de efectuar gravação de vídeo e áudio localmente e remotamente, através de um computador, tornando este sistema mais atractivo em termos de mercado.

Referências

Alecrim, E. (2005). Básico sobre DNS.

Alecrim, E. (2003). Linguagem ASP.

Alecrim, E. (2003). Linguagem Javascript.

Alecrim, E. (2005). Protocolo DHCP.

Anstey, M. (2003). *PHP vs ASP*. Obtido em 2007, de <http://marty.anstey.ca/programming/php/articles/>

Axis. (2007). *Technical guide to network video*. Técnico.

Castells, M. (2004). *A Galáxia Internet Reflexões sobre Internet, Negócios e Sociedade*. Gulbenkian.

CGD. (2007). *CGD*. Obtido em 2007, de <http://www.cgd.pt>

Coelho, P. (1997). *Animação de páginas na Worl Wide Web com Javascript*. FCA.

COMER, D. E. (1994). *Internetworking With TCP/IP*. Prentice Hall.

Costello, S. (2002). RSA: Security in 2002 worse than 2001, exec says.

D-LINK. (2004). *D-LINK*. Obtido em 2007, de D-LINK : <http://www.dlink.pt/>

Finanças, G. d. (s.d.). *Portal do Ministério das Finanças e da Administração Pública*. Obtido em 2007, de <http://www.min-financas.pt/>

Foundation, T. A. (2007). *The Apache Software Foundation*. Obtido em 2007, de <http://www.apache.org/>

Informática, I. d. (1999). *Segurança Social*. Obtido em 2007, de <http://www.seg-social.pt/>

MENDONÇA, I. T. (s.d.). *CONTROLE DE DISPOSITIVOS VIA WEB*. p. 15.

- MySQL. (1995). *MySQL*. Obtido em 2007, de <http://www.mysql.com>
- Netmove. (2007). *Equinotec*. Obtido em 2007, de <http://www.equinotec.com>
- Netmove. (2003). *OASRN*. Obtido em 2007, de <http://www.oasrn.org>
- Netmove. (2007). *Refral*. Obtido em 2007, de Refral: <http://www.refral.pt>
- No-IP.com - Vitalwerks Internet Solutions, L. (1999). *No-IP*. Obtido em 2007, de No-IP: <http://www.no-ip.com/>
- Optimus*. (2007). Obtido em 2007, de Optimus: <http://www.optimus.pt/Site+Optimus/MassMarket>
- Pastore, P. (2003). *Breve introdução aos protocolos HTTP, HTTPS e TLS*. Obtido de http://www.gta.ufrj.br/grad/03_1/http/
- Pereira, J. L. (1998). *Tecnologia de Base de dados*. FCA.
- PHP*. (2007). Obtido em 2007, de PHP: <http://www.php.net>
- PiXORD. (2007). *PiXORD Corporation*. Obtido em 2007, de PiXORD Corporation: <http://www.pixord.com/>
- PostgreSQL. (1996). *PostgreSQL*. Obtido em 2007, de <http://www.postgresql.org/>
- Puryear, D. (2007). Apache vs. IIS.
- Soares, W. (2004). *PHP: Conceitos, programação e integração com bancos de dados*.
- Tmn. (2007). *Tmn*. Obtido em 2007, de <http://www.tmn.pt>
- Torres, B. (2006). *Introdução ao protocolo HTTP*. Obtido em 2007, de <http://www.obasicodaweb.com/introducao-ao-http>
- Vodafone. (2007). *Vodafone*. Obtido em 2007, de <http://www.vodafone.pt/main/particulares>

