



Universidade do Porto
Faculdade de Engenharia

FEUP



José Jorge Magalhães Dias Gomes

Projecto Loja WEB

004(047.3) LEIC
EIC5202 2004/GOMj

Outubro, 2004

**Faculdade de Engenharia da Universidade do Porto
Licenciatura em Engenharia Informática e Computação**



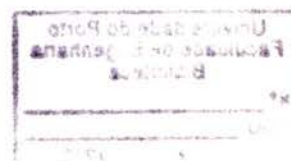
**Projecto Loja Web na
Global Notícias Publicações, S.A.**

Relatório do Estágio Curricular da LEIC 2003/2004

José Jorge Magalhães Dias Gomes

Orientador na FEUP: Prof. João José Pinto Ferreira
Orientador na Global Notícias Publicações, S.A.: Eng^o. Carlos Matos

Outubro de 2004



004(047.3) LECT EIC5202 2004/60Mg

Universidade do Porto	
Faculdade de Engenharia	
Biblioteca	
Nº	81498
CDU	004.41(047.3)
Data	20/03/2006

Estágio apoiado pelo Programa de Desenvolvimento Educativo para Portugal



Dedicado à minha família

Resumo

No âmbito da disciplina “Estágio” do segundo semestre do quinto ano da Licenciatura em Engenharia Informática da Faculdade de Engenharia da Universidade do Porto, foi contactado o Jornal de Notícias (empresa Global Notícias Publicações, S.A.) tendo em vista possibilitar um projecto que aproximasse os finalistas à realidade prática do mundo do trabalho.

O estágio proposto pela empresa teve o título de “Projecto Loja Web” e tinha por objectivos iniciais o desenvolvimento de uma loja *online* (a “Loja Web”) de venda de produtos do grupo Global Notícias e a criação de ferramentas de gestão (“Backoffice”) para manipular a área comercial subjacente à loja *online* (criação de novos produtos, alteração dos existentes, entre outras funcionalidades). Posteriormente considerou-se proveitoso adicionar ao projecto a componente “Cartão Virtual” que consistia na criação de um *site* inspirado no Cartão JN, um cartão virtual associado à anterior loja *online* do Jornal de Notícias que, numa estratégia de fidelização, pretendia levar os utilizadores a registarem-se e a acumular pontos para receber descontos na loja.

Foram então implementados três *sites*. O *site* da Loja Web possui funcionalidades avançadas como possibilidade de visualização do estado das encomendas ou o cancelamento das mesmas. O *site* do Cartão Virtual permite aos utilizadores registados a participação em passatempos e inclui as funcionalidades de registo e manutenção dos dados fornecidos, bem como a consulta de históricos de movimentos do cartão virtual, encomendas efectuadas e passatempos participados. A Loja Web e o Cartão Virtual são geridos pela parte de Backoffice que possibilita inserir ou alterar produtos e passatempos, indicar os vencedores dos passatempos, efectuar a gestão do processamento das encomendas e visualizar estatísticas referentes a vendas, registos no cartão virtual e participações em passatempos.

Para o desenvolvimento do projecto recorreu-se a tecnologia .NET da Microsoft, nomeadamente a páginas *web* em ASP.NET e à linguagem C#. Como repositório de dados foi usada a base de dados SQL Server 2000, também da Microsoft.

Agradecimentos

Gostaria de agradecer a simpatia e apoio dados pelas pessoas pertencentes ao Departamento de Produção e Informática e ao Departamento de Marketing presentes no Jornal de Notícias. Não só aos envolvidos mais directamente com o projecto como a Sofia, a Maria João e a Carla, o Eng^o. Matos e o Jorge, mas também aos restantes elementos que me ajudaram tecnicamente ou simplesmente ao proporcionar momentos de boa disposição durante o dia-a-dia de trabalho.

Gostaria igualmente de agradecer ao meu orientador de estágio da faculdade, o Prof. João José Pinto Ferreira, pela sua disponibilidade para me orientar e apoio dado durante o estágio.

Numa altura em que a minha vida académica termina, aproveito para agradecer aos meus pais e restantes familiares o apoio que sempre fizeram questão em dar.

Por último, é justo mencionar o apoio monetário dado pelo PRODEP (Programa de Desenvolvimento Educativo para Portugal).

Índice de Conteúdos

1	Introdução.....	1
1.1	Apresentação da Global Notícias Publicações, S.A.	1
1.2	O “Projecto Loja Web” na Global Notícias Publicações, S.A.	2
1.3	Organização do Relatório	3
2	Análise do problema.....	4
2.1	Loja Web.....	4
2.2	Cartão Virtual.....	5
2.3	Backoffice.....	6
2.4	Actores.....	8
2.4.1	Utilizador não-autenticado	8
2.4.2	Utilizador autenticado	8
2.4.3	Operador de Backoffice	8
3	Revisão Tecnológica.....	9
3.1	Soluções existentes no mercado.....	9
3.2	Apresentação das tecnologias envolvidas.....	10
3.2.1	XML	10
3.2.2	Internet Information Services 6.0	11
3.2.3	SQL Server 2000	12
3.2.3.1	<i>Database Server</i>	12
3.2.3.2	<i>Analysis Services</i>	13
3.2.3.3	<i>English Query</i>	14
3.2.4	.NET Framework.....	14
3.2.4.1	Arquitectura.....	15
3.2.4.2	Funcionamento.....	16
3.2.4.3	ASP.NET	17
3.2.4.3.1	ASP.NET Web Applications	17
3.2.4.3.2	XML Web Services.....	18
3.2.4.4	ADO.NET	19
3.2.5	<i>Secure Sockets Layer</i> , chaves públicas e privadas e certificados digitais.....	20
4	Especificação.....	22
4.1	Arquitectura lógica.....	22
4.1.1	Divisão em pacotes.....	22
4.1.1.1	Base de dados	22
4.1.1.2	Lógica de negócio	23
4.1.1.3	Interface	23
4.1.2	Modelo de classes do domínio.....	23
4.2	Arquitectura física	25
4.2.1	Distribuição (<i>Deployment</i>).....	25
4.2.2	Descrição dos componentes.....	25
4.2.2.1	Loja Web	26
4.2.2.1.1	<i>Web User Controls</i>	27
4.2.2.1.2	Página inicial.....	28

4.2.2.1.3	Registo no Cartão Virtual	29
4.2.2.1.4	Visualizar produtos e colocar uma encomenda.....	29
4.2.2.1.5	Secção da Conta Pessoal.....	30
4.2.2.1.6	Secção de Ajuda	32
4.2.2.2	Cartão Virtual	32
4.2.2.2.1	<i>Web User Controls</i>	32
4.2.2.2.2	Página inicial.....	33
4.2.2.2.3	Registo no Cartão Virtual e secção de ajuda	33
4.2.2.2.4	Secção da Conta Pessoal.....	34
4.2.2.2.5	Secção dos Passatempos.....	35
4.2.2.3	Backoffice.....	36
4.2.2.3.1	<i>Web User Controls</i>	36
4.2.2.3.2	Página inicial.....	36
4.2.2.3.3	Secção das Estatísticas.....	37
4.2.2.3.4	Secção de Gestão de Produtos	39
4.2.2.3.5	Secção de Gestão das Encomendas	40
4.2.2.3.6	Secção de gestão dos Passatempos	40
5	Implementação	42
5.1	Camada de base de dados.....	42
5.1.1	Tabelas.....	43
5.1.2	<i>Triggers</i>	47
5.1.3	<i>Stored Procedures</i>	48
5.2	Camada de lógica de negócio	49
5.2.1	Publicacao.....	49
5.2.2	Categoria.....	50
5.2.3	Pontos.....	50
5.2.4	Preco.....	50
5.2.5	Regiao.....	51
5.2.6	Iva	51
5.2.7	Produto.....	52
5.2.8	Carrinho	54
5.2.9	Pais.....	55
5.2.10	DadosEnvio.....	56
5.2.11	HabLiterarias.....	56
5.2.12	Utilizador	57
5.2.13	Estado.....	58
5.2.14	TipoPagamento.....	58
5.2.15	Encomenda	59
5.2.16	Estatistica.....	61
5.2.17	MailingList	62
5.2.18	Passatempos	62
5.2.19	Pergunta.....	64
5.2.20	Premio.....	65
5.2.21	Resposta	65
5.2.22	TipoPassatempo	65
5.2.23	Misc.GUI	66
5.2.24	Misc.MailingList.....	66

5.2.25	Misc.Picture.....	67
5.2.26	Misc.Security.....	67
5.3	Camada de interface gráfico.....	68
5.4	Windows Service	70
5.5	Web Service	71
6	Conclusão.....	72
	Referências e Bibliografia.....	75
ANEXO A:	Imagens da aplicação	77
	Cartão Virtual	77
	Loja Web	85
	Backoffice.....	90

1 Introdução

1.1 Apresentação da Global Notícias Publicações, S.A.

O grupo Portugal Telecom é um operador global de telecomunicações com um portfólio de negócios diversificado que abarca todos os segmentos do sector: negócio fixo, móvel, multimédia, dados e soluções empresariais.

Além de Portugal, a empresa encontra-se também no Brasil, Marrocos, Guiné-Bissau, Cabo Verde, Moçambique, Timor, Angola, Quénia, China e São Tomé e Príncipe.

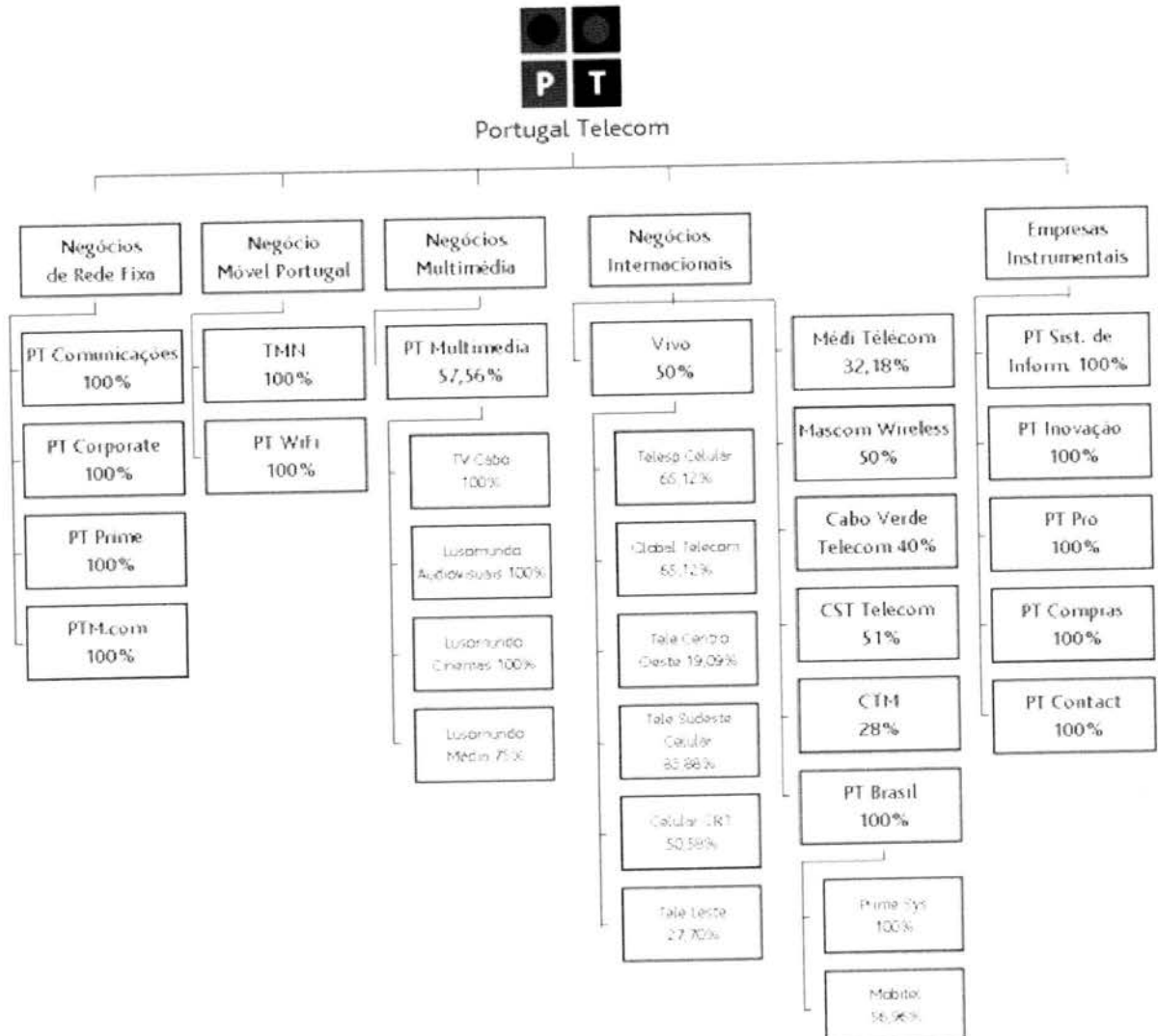


Ilustração 1 - Organograma do grupo Portugal Telecom

Uma das partes do grupo Portugal Telecom, a PT Multimédia posiciona-se em várias áreas de negócios como o cabo, conteúdos, cinemas e media.

Os media do grupo estão integrados na *holding* Lusomundo Media que detém participações na imprensa, rádio, novos media e editoras livreiras. Entre as várias empresas participadas contam-se:

- Diário de Notícias, SA;

- Jornal de Notícias, SA (Jornal de Notícias e Naveprinter SA);
- Someios E.P. Lda (Notícias Magazine e Publicultura, SA);
- NP - Notícias de Portugal SRL (Lusa);
- Jornal do Fundão, Lda;
- Radio Notícias, SA (TSF - Rádio Jornal de Lisboa, Lda; RadioPress, Lda; Rádio Jornal do Norte, Lda);
- Prodiário, SA (24 Horas, Inc e Tal & Qual);
- Açormédia, Lda (Açoreano Oriental; Rádio C. Açores, Lda; Rádio C. Aberto, Lda);
- Empresa do D.N, Lda (Diário de Notícias Madeira; TSF Madeira; SDIM, Lda);
- Pressmundo, SA (Volta ao Mundo; Evasões; Viver com Saúde; National Geographic);
- Grande Reportagem, Lda;
- Editorial Notícias, Lda (Oficina do Livro Lda).

Como parceiros de negócios, a Lusomundo Media conta também com:

- Naveprinter, Indústria Gráfica do Norte, SA;
- Empresa Gráfica Funchalense, SA;
- VASP, Distribuidora de Publicações

A 1 Janeiro de 2004, o grupo PT decidiu efectuar a fusão das publicações escritas da Lusomundo Media numa só empresa, a Global Notícias Publicações, SA. Essa nova empresa tornou-se responsável pelos jornais Jornal de Notícias, Diário de Notícias, 24 Horas e Tal & Qual, e pelas revistas Evasões, National Geographic Portugal, Grande Reportagem, Notícias Magazine, Playstation2 e Volta ao Mundo.

Actualmente a Global Notícias é presidida pelo Dr. Henrique Granadeiro e o seu conselho de administração, apontado pela Lusomundo Media, conta com 5 elementos: Dr. Lemos Ribeiro, Eng.º José Marquitos, Dr. Miguel Moreno, Dra. Josete Berrones e Eng.º Manuel Soares.

1.2 O “Projecto Loja Web” na Global Notícias Publicações, S.A.

O Jornal de Notícias, a quem foi efectuado o contacto para submeter propostas de estágio curricular, possuía um *site* de venda dos produtos das promoções que regularmente oferece com o jornal mediante o pagamento de uma quantia adicional sobre o preço normal da publicação. Este *site* era a “face” *online* de uma pequena loja física existente na sede do Jornal de Notícias e responsável pelo despacho das encomendas efectuadas na Internet.

Associado à loja *online* estava um “Cartão Virtual”. Este cartão estava inserido numa estratégia dupla de construção de uma base de dados de e-mails para acções de marketing e de fidelização ao *site* da edição *online* do Jornal de Notícias, pois o utilizador acumulava pontos no seu cartão através da visualização de certas páginas em www.jn.pt e era indispensável o registo no cartão para adquirir produtos na loja *online*. Os utilizadores registados no cartão podiam ter também acesso a passatempos (e aos respectivos prémios) realizados no *site*, e podiam subscrever um serviço que lhes permitiria receber as notícias principais ou de certas secções da edição *online* do Jornal de Notícias diariamente por e-mail.

Com a aglutinação das várias publicações numa só empresa (a Global Notícias) e a criação de nova loja físicas que agora deveriam vender todos os produtos de todas as publicações, tornou-se necessário criar uma nova loja *online* nos mesmos moldes, tendo como base a loja *online* existente para o JN. Era também conveniente modificar o *site* do anterior “Cartão Virtual” do JN, globalizando-o para um “Cartão Virtual Global Notícias”.

Tendo em consideração a experiência passada da loja *online* e Cartão Virtual do JN (*sites* relativamente desactualizados parcialmente devido à necessidade do departamento de Marketing ter de passar informações para departamento de Informática inserir ou actualizar, sobrecarregando-o com trabalho), era proveitoso aproveitar a oportunidade de dotar o departamento de Marketing no Jornal de Notícias (responsável pelas lojas *online* e físicas e *site* do Cartão Virtual) com ferramentas de *backoffice* que possibilitassem a sua total autonomia.

É, pois, numa óptica de resolução de problemas e aproveitamento de oportunidades que surge o “Projecto Loja Web”.

1.3 Organização do Relatório

O presente relatório encontra-se dividido em sete partes principais, cada uma delas abordando um tema diferente.

Numa primeira secção (a actual) é apresentada a instituição onde decorreu o estágio, e é descrito o projecto, o contexto em que este surgiu e o seu enquadramento na organização.

Seguidamente é efectuada a análise do problema colocado pelo projecto, sendo abordados os principais requisitos aos quais a aplicação deverá obedecer.

Na terceira secção, “Revisão Tecnológica”, são descritas as tecnologias a utilizar no projecto. São igualmente referidas as possíveis soluções para o problema que existam do mercado e que melhor se enquadrariam no pretendido pela organização.

Na secção seguinte é apresentada e descrita detalhadamente a solução proposta. É feito um esforço de especificação em termos de arquitecturas físicas e lógicas que sustentariam a aplicação.

Na quinta secção é explicada como foi efectuada a implementação do protótipo. A aplicação é apresentada a um nível mais baixo e são descritos os problemas e respectivas soluções encontradas.

Na última secção é efectuado um balanço do trabalho desenvolvido durante o estágio, possíveis melhoramentos que a aplicação desenvolvida poderá sofrer e abordado o potencial impacto que o projecto terá na organização.

2 Análise do problema

Partindo dos objectivos do estágio e da sua descrição verifica-se que o problema apresentado pelo projecto poderá ser decomposto em três partes com funções distintas, correspondentes aos três *sites* a desenvolver: Loja Web, Cartão Virtual e Backoffice. Através da definição de requisitos aos quais estas partes deverão obedecer e das funcionalidades que o sistema pretendido deverá contemplar, as fases de especificação e implementação serão facilitadas.

2.1 Loja Web

O contexto da Loja Web inclui algumas palavras-chave como produtos, encomendas ou utilizadores.

Será preciso ter em conta que a Loja Web permitirá a compra a todos os internautas mas que os utilizadores registados no Cartão Virtual poderão usar os seus pontos para comprar os produtos a preços mais reduzidos. Isto obrigará a monitorizar os pontos que os utilizadores têm e que podem gastar, e os produtos a terem vários preços com diferentes pontos associados. Também é necessário compreender que os preços já incluem o IVA e os portes, variando por região de envio, pelo que a visualização dos preços no *site* está dependente da escolha de um país (que pertencerá a uma região de envio). Outro aspecto a ter em conta é o facto das encomendas terem estados (podem estar nos estados pendente ou paga, por exemplo) e desse estado dependerem certas opções disponibilizadas aos utilizadores. Por último, algumas situações especiais deverão ser contempladas. Primeiro, deverão existir colecções de produtos de maneira que seja possível comprar toda uma colecção ou apenas um dos itens que a constituem. Outra situação é o facto de, ao longo do tempo, um produto poder pertencer a mais que uma publicação. Finalmente, os produtos existentes no *site* do Centro de Documentação e Informação do Jornal de Notícias deverão ser comprados através da Loja Web. Estes produtos não serão listados na Loja Web, devendo o *site* ser inteligente ao ponto de processar a informação fornecida pelo *site* do CDI no redireccionamento para a Loja Web.

Outros dos principais requisitos aos quais a Loja Web deverá obedecer são:

- permitir a visualização dos produtos pela marca do grupo associada;
- permitir a visualização dos produtos pela categoria do produto;
- permitir a pesquisa de produtos;
- mostrar na sua página inicial alguns produtos em destaque;
- facilitar a ligação à página do Cartão Virtual;
- disponibilizar ao utilizador um sistema de ajuda sobre os vários aspectos relacionados com a utilização do *site*;
- disponibilizar um serviço de registo no Cartão Virtual para novos utilizadores que lhes envie um e-mail de confirmação no final do processo;
- ter uma zona exclusiva para os utilizadores registados, onde estes possam visualizar e modificar informação pessoal, requisitar o envio de uma nova *password* para o seu e-mail

(em caso de esquecimento), ver o histórico das encomendas efectuadas e cancelar as encomendas pendentes;

- enviar um e-mail de confirmação quando um utilizador registado altera os seus dados pessoais;
- permitir a visualização do estado das encomendas e o cancelamento das ainda não processadas a todos os utilizadores, registados ou não;
- exigir a autenticação ao utilizador (pedindo o número de cartão ou e-mail e a *password*) quando este deseja aceder à zona de utilizadores registados (excepto no caso de envio de nova *password*);
- permitir, a qualquer momento, o *login* aos utilizadores registados;
- mostrar, a todo o momento, aos utilizadores autenticados, o seu saldo de pontos;
- permitir, a qualquer momento, a modificação da zona de envio;
- disponibilizar, no detalhe de cada produto, a descrição, os preços (com e sem descontos de pontos), uma imagem alusiva e a possibilidade de o adicionar ao carrinho;
- permitir a escolha de preço com descontos de pontos apenas aos utilizadores registados que se tenham autenticado e possuam pontos suficientes para essa opção;
- indicar se um determinado produto não existe em stock, não permitindo que seja adicionado ao carrinho;
- possuir um carrinho de compras que permita modificar quantidades, eliminar itens e avançar para a finalização da encomenda;
- exigir os dados de envio e de pagamento e mostrar uma página final de confirmação dos dados fornecidos, durante o processo de finalização da encomenda;
- disponibilizar aos utilizadores, aquando do fornecimento dos dados pessoais, a possibilidade de aceitarem receber informações adicionais da Global Notícias;
- enviar um e-mail de confirmação quando um utilizador coloca ou cancela uma encomenda.

2.2 Cartão Virtual

O *site* do Cartão Virtual deverá oferecer funcionalidades de gestão semelhantes às disponibilizadas aos utilizadores registados na Loja Web, além de outras mais relacionadas com este *site*. A possibilidade da participação pelos utilizadores registados em passatempos no *site* do Cartão Virtual leva à definição de requisitos adicionais. Assim, será necessário criar uma infra-estrutura que suporte passatempos de vários tipos. Deverão existir passatempos cuja participação consiste na escrita de uma frase, outros consistirão em responder a perguntas de escolha múltipla e outros deverão consistir apenas no pressionar de um botão. Deverá ainda existir um tipo de passatempo que conjugue a resposta a perguntas de escolha múltipla com a criação de uma frase. Será preciso ter em conta a existência de prazos durante os quais os passatempos decorrem, bem como a possibilidade de o decorrer dos passatempos estar restrito a x participações correctas (esta possibilidade só poderá existir para os passatempos de perguntas e respostas ou em que basta pressionar um botão – nos outros tipos de passatempos

a noção de participação correcta não é linear). Ao conceito de passatempo (e de participação) estão associados os de prémio e de vencedores. Os prémios que cada passatempo tem serão, finalizado o passatempo, atribuídos às participações, definindo os vencedores (um passo que será dado pelo Backoffice).

Outras das principais linhas a que o Cartão Virtual deverá obedecer são:

- facilitar a ligação às páginas do grupo Global Notícias;
- facilitar a ligação à página da loja *online*;
- disponibilizar ao utilizador um sistema de ajuda sobre os vários aspectos relacionados com a utilização do *site*;
- disponibilizar um serviço de registo no Cartão Virtual para novos utilizadores que lhes envie um e-mail de confirmação no final do processo;
- disponibilizar aos utilizadores, aquando do fornecimento dos dados pessoais, a possibilidade de aceitarem receber informações adicionais da Global Notícias;
- ter uma zona exclusiva para os utilizadores registados, onde estes possam visualizar e modificar informação pessoal, requisitar o envio de uma nova *password* para o seu e-mail (em caso de esquecimento), ver o histórico das encomendas efectuadas e cancelar as encomendas pendentes, ver o histórico dos movimentos do seu cartão virtual e ver o histórico dos passatempos participados (e respectivo detalhe das participações) ;
- enviar um e-mail de confirmação quando um utilizador registado altera os seus dados pessoais;
- enviar um e-mail de confirmação quando um utilizador registado cancela uma encomenda;
- exigir a autenticação ao utilizador (pedindo o número de cartão ou e-mail e a *password*) quando este deseja aceder à zona de utilizadores registados (excepto no caso de envio de nova *password*) ou quando quiser participar num passatempo;
- permitir, a qualquer momento, o *login* aos utilizadores registados;
- mostrar, a todo o momento, aos utilizadores autenticados, o seu saldo de pontos;
- permitir a visualização os passatempos actuais e anteriores e, para estes, os vencedores e, dependente do tipo de passatempo, as respostas correctas ou frases vencedoras;
- permitir apenas uma participação por utilizador, por passatempo.

2.3 Backoffice

Por servir para gerir a Loja Web e o Cartão Virtual, o Backoffice vai ter de ter em conta todos os conceitos afectos a estes *sites*. O acesso ao Backoffice deverá implicar uma autenticação e, com a finalidade de simplificar a navegação, deverão ser criadas secções temáticas que agreguem as diversas funcionalidades disponíveis.

Assim, uma secção referente a dados estatísticos deverá:

- disponibilizar estatísticas relativas à Loja Web, nomeadamente os totais mensais das vendas por publicação, por produtos, por clientes, por localidade ou país, por sexos, por faixas etárias ou por habilitações literárias;

- disponibilizar estatísticas relativas aos registos no Cartão Virtual, nomeadamente os registos mensais por localidade ou país, por sexos, por faixas etárias ou por habilitações literárias;
- disponibilizar estatísticas relativas às participações nos passatempos do Cartão Virtual, nomeadamente os totais mensais das participações por utilizadores, por passatempos, por localidade ou país, por sexos, por faixas etárias ou por habilitações literárias;
- disponibilizar estatísticas relativas aos utilizadores que aceitaram receber informações adicionais da Global Notícias, nomeadamente os totais mensais desses pedidos por localidade ou país, por sexos, por faixas etárias ou por habilitações literárias;
- adicionalmente, disponibilizar os endereços electrónicos dos utilizadores que aceitaram receber informações adicionais da Global Notícias.

Uma secção referente à gestão de produtos deverá:

- permitir a introdução de novos produtos e consulta/alteração dos existentes, sendo que a imagem a ser associada ao produto deverá ser automaticamente redimensionada para um tamanho específico e uma pré-visualização do produto, tal como apareceria na página da Loja Web, deverá ser mostrada;
- disponibilizar uma listagem dos produtos cujo stock se encontre deficitário;
- permitir a escolha dos produtos que estarão em destaque na página principal da Loja Web.

A secção de gestão de encomendas deverá:

- disponibilizar a listagem das encomendas efectuadas, permitindo a filtragem destas por data, estado, número de cartão ou número de encomenda;
- disponibilizar o detalhe de cada encomenda (incluindo uma versão especial para impressão) e opções de alteração de estado destas (cancelamento, dar como pagas as pendentes ou como expedidas as pagas).

A última secção temática, referente aos passatempos, deverá:

- permitir a introdução de novos passatempos e consulta/alteração dos existentes (que não estejam a decorrer ou que já tenham terminado), sendo que a imagem a ser associada ao passatempo deverá ser automaticamente redimensionada para um tamanho específico e uma pré-visualização do passatempo, tal como apareceria na página do Cartão Virtual, deverá ser mostrada;
- disponibilizar uma listagem dos passatempos já terminados e sem vencedores atribuídos;
- permitir, para os passatempos terminados e sem vencedores atribuídos, a associação entre as participações válidas e os prémios disponíveis (inserção de vencedores).

2.4 Actores

O desenho da aplicação a desenvolver deverá distinguir os vários actores ou intervenientes envolvidos, bem como as funcionalidades distintas a que terão acesso.

2.4.1 Utilizador não-autenticado

O *Utilizador não-autenticado* representa o utilizador que navega pela Loja Web ou Cartão Virtual sem estar autenticado no sistema. Este actor pode, por exemplo, visionar os diversos itens da Loja Web, adicioná-los ao carrinho de compras e colocar uma encomenda (podendo monitorizar posteriormente o seu estado) ou visualizar a informação de suporte disponível. Pode opcionalmente registar-se no Cartão Virtual ou requisitar o envio por e-mail de uma *password* temporária (se for um utilizador registado). No *site* do Cartão Virtual pode também ver os passatempos actuais e os vencedores e respostas correctas (ou frases vencedoras) dos últimos passatempos decorridos.

2.4.2 Utilizador autenticado

O actor *Utilizador autenticado* pode efectuar o mesmo que o *Utilizador não-autenticado*. Adicionalmente pode usar os seus pontos para obter descontos nas encomendas da Loja Web e participar nos passatempos activos do Cartão Virtual. Este utilizador também pode gerir os seus dados pessoais e ver os históricos dos pontos que acumulou/gastou, das encomendas que efectuou e dos passatempos nos quais participou.

2.4.3 Operador de Backoffice

O *Operador de Backoffice* tem poderes sobre o sistema que lhe permitem, através de uma ferramenta de *backoffice*, gerir vários aspectos da Loja Web e do Cartão Virtual como sejam a inserção, modificação ou eliminação de produtos da Loja Web, o processamento das encomendas efectuadas, o lançamento, modificação ou eliminação de novos passatempos para o *site* do Cartão Virtual e a atribuição dos respectivos vencedores. A ferramenta de *backoffice* à qual o *Operador de Backoffice* tem acesso também lhe permite visualizar dados estatísticos sobre produtos, encomendas, passatempos e utilizadores, e obter os endereços de e-mail dos utilizadores que desejam receber informações adicionais da Global Notícias.

3 Revisão Tecnológica

3.1 Soluções existentes no mercado

Relativamente ao que foi proposto, não existe nenhuma solução que consiga “cobrir” na totalidade a especificidade do projecto. Existe um grande número de soluções para *sites de e-commerce*, alguns deles com funcionalidades de *backoffice* integradas, mas a componente “Cartão Virtual” não é contemplada (não existem soluções que permitam passatempos, nem acumulação de pontos, nem descontos através do uso destes numa loja *online*). A falta dessa funcionalidade que é parte integrante das estratégias de marketing da organização, assim como a possibilidade de se atingir um maior grau de personalização, acaba por justificar o desenvolvimento do projecto.

De entre as soluções existentes no mercado que mereceram análise mais cuidada, os produtos da SAP foram os que mais se aproximaram do pretendido, sendo que por a empresa já possuir software desta empresa a adaptação e integração das mesmas poderia ser facilitada. No entanto, por não fazerem exactamente o pretendido, e pela sua complexidade e custo, a aquisição destas ferramentas revelar-se-ia uma opção desequilibrada, apresentando mais que o necessário em alguns campos e sendo insuficiente noutros.

O SAP Internet Sales permite, aos clientes SAP já com SAP R/3, criar rapidamente uma presença na *web* sem necessitar de implementar uma solução de *Customer Relationship Management*. A solução oferece a personalização de *templates* para lojas B2C ou B2B que disponibilizam um catálogo *online* dos produtos da empresa, com possibilidade de pesquisa. A integração com o SAP R/3 leva a que informações sobre preços, características, disponibilidades de produtos ou tempos de entrega estejam sempre actualizados e que os clientes possam monitorizar o estado em que se encontram as encomendas efectuadas.

O mySAP CRM é uma solução global de *Customer Relationship Management*, pelo que não trata exclusivamente da venda de produtos *online*, mas sim de todos os processos envolvidos com os clientes desde o planeamento de acções de marketing, passando pela comercialização de produtos, até ao serviço pós-venda. Relativamente ao *e-commerce*, o mySAP CRM permite as vendas na Internet, possibilitando aos clientes funcionalidades semelhantes às do SAP Internet Sales mas com nível de interacção maior. Existe suporte para processos relacionados com a lealdade e a procura por parte dos utilizadores e para a realização de análises aos hábitos dos utilizadores do *site*. As outras vertentes do mySAP CRM estão relacionadas com o Marketing (planeamento e gestão de campanhas, análise de dados para personalização de produtos, segmentação por clientes, suporte para tele-marketing) e Vendas (oportunidades e previsões de vendas, planeamento de actividades, gestão da assistência pós-venda, análise e comunicação com vendedores, revendedores e distribuidores).

De referir que a PHC, cujo software a empresa também possui, disponibiliza uma solução modular direccionada para a Internet chamada PHC Digital, cujo módulo Digital dFront permite a venda de produtos e a gestão de encomendas efectuadas. Infelizmente a solução foi desenvolvida primariamente numa óptica B2B, revelando-se inadequada e insuficiente para o pretendido.

Na mesma óptica modular, a Sage Infologia tem uma solução chamada Sage e-commerce (funcionando interligada com o seu software Gestão Comercial 100), cujo módulo Loja

permitiria vendas ao grande público. Através de um assistente a apresentação gráfica da loja *online* seria facilitada e os catálogos de produtos seriam obtidos através da Gestão Comercial 100, o qual permitiria o processamento das encomendas e monitorização destas por parte dos compradores.

3.2 Apresentação das tecnologias envolvidas

Apesar de ter sido dada liberdade de escolha ao estagiário relativamente às tecnologias a utilizar para desenvolver o projecto, a escolha de produtos da Microsoft foi acordada quase de imediato, pelo facto da organização possuir o software em causa e por o utilizar em várias áreas com possível envolvimento no projecto.

Inicialmente será apresentado o XML e o *web server* IIS, ambos fortemente interligados com as restantes tecnologias, seguidamente a base de dados SQL Server 2000 e a .NET Framework. São também abordadas questões relativas à segurança, nomeadamente relativas ao uso de certificados digitais.

3.2.1 XML

XML significa *eXtensible Markup Language* e é uma linguagem descritiva que usa *tags* e se encontra orientada a dados, um pouco à imagem do HTML (este orientado para a apresentação gráfica). Ao envolver os dados em *tags*, numa estrutura predefinida, simples ou complexa, mas claramente perceptível para todos, o XML transformou-se num standard não só para a Internet, mas também levou à fácil partilha de informação entre diferentes sistemas e entre diferentes organizações. No fundo, trata-se de uma metalinguagem (baseada em SGML) que permite descrever diferentes realidades em diferentes tipos de documentos.

A estrutura de um ficheiro XML pode ser definida num DTD (*Document Type Description*) ou *schema*. Um DTD fornece uma descrição formal (em *XML Declaration Syntax*) onde se definem os nomes dos *tags*, o seu tipo de conteúdo e a sua localização. Um *schema* (ficheiros de extensão .XSD) é mais avançado que um DTD. A sintaxe de um *schema* é similar à do XML mas permite um maior número de tipos de dados (e até criar tipos de dados) e definir mais restrições.

Os dados de ficheiros XML são actualmente visualizáveis em quase todos os *browsers web* e para formatar os documentos podem-se utilizar *stylesheets* CSS, no entanto, para dados organizados de forma pouco linear torna-se preferível recorrer à linguagem XSL (*eXtensible Stylesheet Language*) que inclui suporte para: XSLT (*XSL Transformations*), XSL-FO (*XSL Formatting Objects*) e XPath (*XML Path Language*). O XSLT permite complexas operações como a geração de tabelas ou listas a partir do conteúdo de ficheiros XML, misturando elementos de HTML com *tags* específicas chamadas “elementos” para obter as transformações. Os *formatting objects* permitem definir com grande detalhe o *layout* e estilo (um pouco à imagem das *stylesheets* CSS), mas também são mais complexos e requerem ferramentas adicionais para obter o resultado final (actualmente os XSL-FO estão mais direccionados para a impressão e são normalmente transformados em ficheiros PDF). O XPath é usado, por exemplo, pelo XSLT para a navegação no XML (de uma forma análoga à usada pelo endereçamento do sistema operativo).

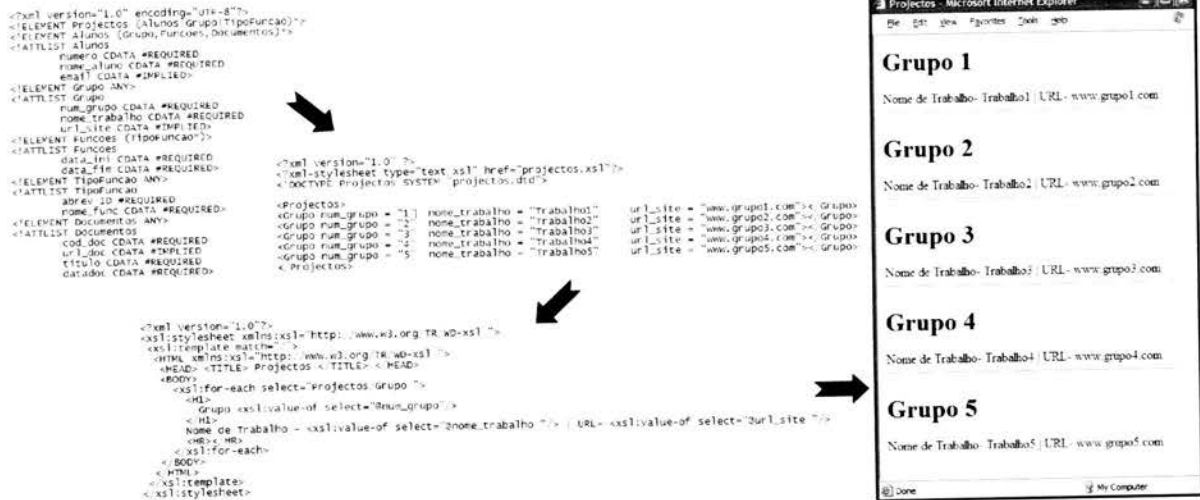


Ilustração 2 - Uso de XML, DTD, XSLT e XPath

3.2.2 Internet Information Services 6.0

O IIS 6.0 é o *web server* da Microsoft incluído em todas as versões do Windows Server 2003, proporcionando uma infra-estrutura escalonável e segura para aplicações *web*. O IIS 6.0 permite às organizações o rápido *deployment* de *sites* e suporta IPv6.0, QoS, SSL, FTP e o uso de processadores a 32 e 64 bits.

Entre as suas principais características encontram-se:

- elevada disponibilidade, graças a uma arquitectura tolerante a falhas que diferencia as aplicações alojadas no servidor (em unidades chamadas *application pools*) e monitoriza a saúde destas, efectuando *resets* ou reciclando processos consoante necessário;
- fácil manutenção, usando as ferramentas de gestão, ficheiros de configuração em XML ou *scripts* numa consola;
- melhor escalabilidade, com o uso de um novo *driver* que efectua o *parsing* e cache HTTP, e através do suporte para *web gardens*, é possível ter mais aplicações *web* num único servidor sem impacto adicional na performance;
- integração com a plataforma .NET, nomeadamente nas vertentes ASP.NET e XML Web Services, facilitando e encurtando o tempo necessário para o desenvolvimento de aplicações que utilizam esta tecnologia;
- maior segurança, ao se encontrar, por defeito, protegido pelos valores máximos de segurança (assim essa segurança só será reduzida em determinados campos conforme a necessidade das aplicações), por recorrer a uma nova *framework* de autenticação (incluindo suporte para o .NET Passport da Microsoft), por possuir melhorias ao nível do SSL e por correr os processos (por defeito) numa conta de reduzidos privilégios.

3.2.3 SQL Server 2000

O SQL Server 2000 é a base de dados da Microsoft para organizações de pequena a grande dimensão. Está limitado a correr sobre os sistemas operativos da Microsoft, integrando-se bem com os mesmos.

O SQL Server 2000 é composto por três componentes: *Database Server*, a base de dados e servidor de acesso à mesma; *Analysis Services*, ferramentas relacionadas com *data mining* e OLAP (*online analytical processing*); e *English Query*, que permite aos utilizadores finais colocar questões em inglês que são transformadas pelo sistema em instruções SQL.

3.2.3.1 Database Server

O componente de base de dados do SQL Server é uma base de dados relacional escalonável, baseada em SQL, e com suporte integrado para XML.

Existe suporte para componentes lógicos como tabelas, índices, *triggers*, vistas, utilizadores e procedimentos chamados *stored procedures*. Estes procedimentos consistem num conjunto de instruções SQL que são pré-compiladas e guardadas na base de dados, evitando que a cada nova consulta ocorra uma compilação com os consequentes ganhos em termos de performance. Adicionalmente, gera-se menos tráfego cliente/servidor pois substitui-se transmitir toda a instrução SQL para uma consulta por uma linha de código que invoca a *stored procedure*.

Os principais APIs suportados pelo SQL Server incluem ActiveX Data Objects (ADO), OLE DB, Open Database Connectivity (ODBC) e baseados em ODBC como Remote Data Objects (RDO) e Data Access Objects (DAO) e Embedded SQL for C (ESQL).

O dialecto SQL suportado pelo SQL Server 2000 é o Transact-SQL (T-SQL), uma linguagem baseada no standard SQL-92. Comparativamente a outros dialectos como o PL/SQL, o T-SQL apresenta-se menos poderoso, só aceitando índices do tipo B-Tree e *triggers* do tipo AFTER ou INSTED OF, e não suportando *arrays*, por exemplo.

O acesso a dados também é possível graças ao XPath. O XPath é uma linguagem que permite a navegação (como um grafo) e escolha de partes de documentos XML. Após definir uma *schema* sobre dados existentes numa tabela ou vista, basta efectuar uma consulta XPath para devolver a informação relativa à *schema* invocada.

Também suporta replicação (manter cópias das bases de dados em vários locais) segundo um modelo que comporta um servidor que publica os dados, outro que subscreve os dados e, em alguns casos, um servidor que distribui os dados. A replicação existente pode ser de três tipos: *snapshot*, que copia todos os dados e objectos como existem num determinado momento (normalmente usado periodicamente em casos onde os dados são poucas vezes alterados e não é grave os subscretores possuírem cópias ligeiramente); *transactional*, no qual a cada transacção ocorrida a informação modificada é copiada os subscretores; *merge*, em que as várias cópias podem trabalhar autonomamente relativamente ao existente no servidor original, ocorrendo numa dada altura a fusão é sincronização entre as várias cópias (para garantir a integridade são mantidos *logs* das modificações e definidas regras em caso de conflitos).

Uma das grandes vantagens do SQL Server 2000 são as ferramentas que contém que, além de intuitivas, permitem o uso de *wizards* para uma fácil gestão e manutenção da base de dados. A

principal dessas ferramentas é o *Enterprise Manager* que possibilita criar novas bases de dados, definir ou alterar tabelas, inserir ou editar dados, definir novos *triggers*, vistas ou *stored procedures*. É possível também executar tarefas administrativas tais como *backups*, políticas de segurança (definindo utilizadores e as suas permissões relativamente a bases de dados ou tabelas) ou importação/exportação de dados. De referir, relativamente à importação/exportação de dados, que esta pode ocorrer a partir de, ou para, vários formatos de bases de dados usando os *Data Transformation Services (DTS)*, um conjunto de ferramentas que permitem extrair, transformar ou consolidar informação de várias origens para um ou múltiplos destinos. É também possível gerar *scripts SQL* sobre todos os objectos existentes da base de dados.

Outras ferramentas importantes são o *Query Analyser* e o *SQL Profiler*. Com o *Query Analyser* é possível desenvolver consultas, executar múltiplas consultas simultaneamente, ver os seus resultados e analisar o plano de execução escolhido para cada consulta. Adicionalmente o *Query Analyser* sugere modificações à consulta executada, ou tabelas e índices envolvidos, que permitiriam melhorar a performance.

O *SQL Profiler* permite monitorizar os eventos ocorridos numa instância do SQL Server, fornecendo uma análise cuidada sobre quais os componentes (*stored procedures*, por exemplo) que estão a ter um impacto mais negativo sobre a performance do sistema.

O SQL Server define para si próprio quatro bases de dados de sistema (*master*, *model*, *tempdb* e *msdb*). Dentro destas bases de dados existem vários *stored procedures* predefinidos e que são úteis para operações diversas como, por exemplo, o *backup* da base de dados ou a criação de uma página *web* com os resultados de uma consulta.

3.2.3.2 Analysis Services

Trata-se de um servidor *middle-tier* que suporta *online analytical processing (OLAP)* e *data mining*. A informação existente numa *data warehouse* é analisada e organizada em “cubos” multidimensionais que contêm os dados agregados e pré-calculados, proporcionando aos utilizadores o acesso rápido a informações de apoio à decisão provenientes de complexas consultas.

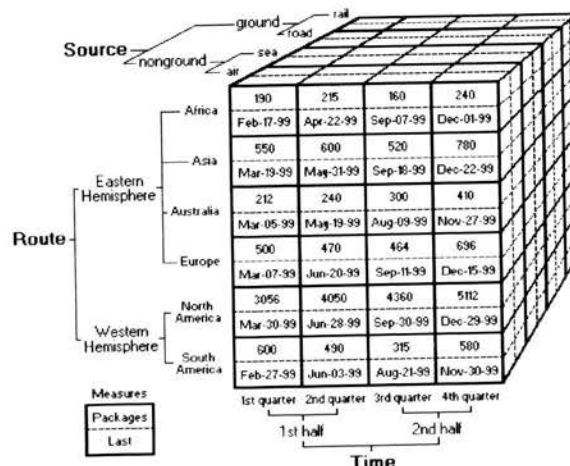


Ilustração 3 - Exemplo de um cubo multidimensional com a informação agregada

Modelos de *data mining* podem ser aplicados a dados provenientes do OLAP ou de bases de dados relacionais. A exportação dos dados pode ser facilitada pelo componente *PivotTable* usado, por exemplo, pelo Excel para recolher informações no servidor e apresentá-las ao utilizador.

3.2.3.3 *English Query*

Usando a funcionalidade *English Query* é possível criar aplicações que efectuem consultas, sobre a base de dados ou informações provenientes do OLAP, utilizando a língua inglesa em substituição de instruções SQL.

O funcionamento do *English Query* implica definir elementos da semântica. Primeiramente as entidades (objectos do mundo real com significado para o utilizador) e associá-las a tabelas ou campos. Posteriormente é necessário definir relações entre as entidades como, por exemplo, “*clientes compram produtos*”. O dicionário de inglês incluído encarrega-se de interpretar as restantes estruturas da frase. Após ser compilado, o *deployment* do modelo obtido pode ocorrer recorrendo às linguagens Visual Basic ou C++, ou pela Internet através de páginas ASP.

3.2.4 .NET Framework

O conceito .NET da Microsoft é aplicado a várias tecnologias, entre elas a .NET Framework e o Visual Studio.NET.

Estas duas tecnologias estão interligadas, uma vez que a .NET Framework é a base de toda a plataforma .NET, sendo necessária para o desenvolvimento nas linguagens de programação existentes no Visual Studio.NET. Sendo uma plataforma recente, tudo foi pensado para a integração na Internet e é também grande a preocupação com a interoperabilidade, podendo, graças à comum plataforma fornecida pelas classes da .NET Framework, qualquer aplicação em C# ser, por exemplo, facilmente transposta para Visual Basic.NET sem dificuldades.

3.2.4.1 Arquitectura

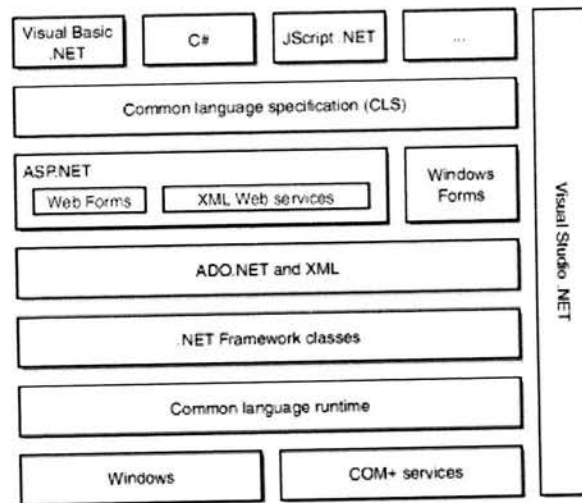


Ilustração 4 - A plataforma .NET

Como se pode observar pela ilustração, a .NET Framework trabalha através de chamadas à Win32 API e serviços COM+.

Sobre esses serviços do sistema posiciona-se a *common language runtime* (CLR). O CLR carrega e corre código (*managed code*) baseado no que os programadores escreveram em linguagens suportadas pela *framework*.

As classes da .NET Framework são bibliotecas comuns às linguagens suportadas pela *framework*. Estas permitem a total liberdade de escolha por parte dos programadores pois as funcionalidades destas classes existem ao nível da *framework*, não se encontrando limitadas a apenas uma das linguagens.

O ADO.NET e XML constituem um conjunto adicional de classes que recorrem às bibliotecas da .NET Framework. O ADO.NET substitui o anterior ADO na função de proporcionar acesso a bases de dados e fontes de informação, fornecendo agora suporte para XML.

As singularidades existentes em aplicações vocacionadas para a Internet relativamente às tradicionais, justificam a divisão em Windows Forms e ASP.NET. Os Windows Forms são utilizados para a apresentação de aplicações-cliente do Windows, um pouco à imagem do que a *Microsoft Foundation Class Library* (MFC) disponibilizava para o Visual Basic. Direccionado para a Internet está o ASP.NET, nova versão do anterior Active Server Pages (ASP), que possui classes específicas para os Web Forms e os XML Web Services.

Para poder aceder à .NET Framework, as linguagens precisam de cumprir um conjunto de regras e procedimentos que garantem a viabilidade das funcionalidades comuns da *framework*. Essas regras estão definidas na *common language specification* (CLS).

3.2.4.2 Funcionamento

O CLR inicialmente compila o código para uma linguagem intermédia, a *Microsoft Intermediate Language* (MSIL), da primeira vez que a aplicação corre (todas as subsequentes chamadas à aplicação recorrerem já ao MSIL directamente). Após o código estar em MSIL, um compilador *Just-In-Time* (JIT) processa a conversão para código nativo.

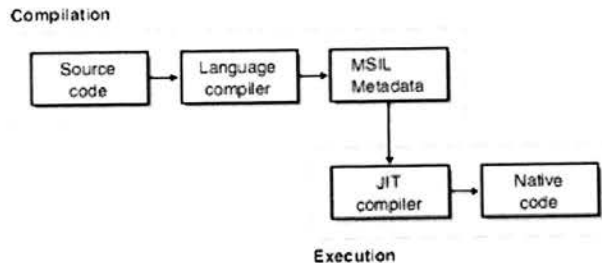


Ilustração 5 - Compilação e execução de *managed code*

No final do processo de compilação é obtida uma *assembly*, um conjunto de ficheiros que podem ser logicamente agrupados e distribuídos. Pode conter variados recursos, mas normalmente incorpora apenas código em MSIL de um executável ou DLL's e o respectivo *assembly manifest* que contém meta-informação. A utilidade desta meta-informação é a de fornecer uma descrição do código (incluindo métodos e argumentos ou referências a bibliotecas) que não depende da linguagem utilizada durante o desenvolvimento.

O uso de uma linguagem intermédia permite ganhos importantes em termos de performance relativamente, por exemplo, à arquitectura da Java Virtual Machine (JVM). Nesta, um compilador Java cria *byte code* que é passado à JVM quando a aplicação é chamada. Não existe linguagem intermédia e, por essa razão, o código está sempre a ser interpretado quando é chamado (em .NET o código já foi interpretado e é simplesmente passado a nativo).

A arquitectura do compilador JIT também permite ganhos ao funcionar de uma forma inteligente. Como nem todas as linhas de código de um programa são executadas (linhas referentes ao processamento de erros, por exemplo), o compilador não transforma todo o código MSIL em código nativo. Ao ser carregada uma classe, um *stub* é associado a cada um dos seus métodos. Quando um método é chamado pela primeira vez o *stub* passa o controlo ao compilador JIT para transformar o código em nativo. Nas próximas chamadas a esse método o *stub* apontará directamente para o código nativo anteriormente criado. Isto permite também maior segurança, pois os programadores podem definir no código dos programas políticas de segurança (com verificações a ocorrerem durante a execução do programa, à medida que métodos são chamados).

Durante a compilação e execução, a CLR trabalha sobre código e recursos chamados de *managed code* e *managed data*.

Managed code é o código que proporciona suficiente informação à CLR para que esta:

- Dado um endereço dentro do código, localize a meta-informação que descreve o método;
- Consiga percorrer a *stack*;
- Consiga processar excepções;
- Possa guardar e retornar informações de segurança.

Antes da CLR poder executar estas tarefas, um processo de verificação ocorre sobre o código. O compilador JIT verifica se o código MSIL e a meta-informação são *type-safe*, ou seja, código que acede a endereços de memória que lhes estão atribuídos.

A *managed data* são os recursos alocados e libertados automaticamente pela CLR durante a execução de um programa. Isto ocorre graças à *garbage collection* que, à medida que objectos deixam de ser utilizados, liberta o espaço alocado em memória para estes.

3.2.4.3 ASP.NET

A componente ASP.NET da .NET Framework é composta por duas vertentes: uma direccionada para a construção de aplicações compostas por um interface *web* (ASP.NET Web Applications), e outra vocacionada para os emergentes XML Web Services. Em ambos os casos existe uma dependência do servidor *web* da Microsoft (IIS) para o desenvolvimento e implementação dos projectos.

3.2.4.3.1 ASP.NET Web Applications

Uma aplicação em ASP.NET é normalmente constituída por um conjunto de páginas *web* chamadas Web Forms (extensão .aspx). Essas páginas consistem em HTML normal com algumas *tags* específicas ao ASP.NET. A interacção com o utilizador é garantida através de conjunto de controlos HTML que correm no cliente e podem desencadear eventos no servidor (botões, por exemplo). O número de controlos HTML existentes é reduzido pelo que houve necessidade de desenvolver outros controlos específicos chamados Web Controls. Alguns Web Controls são similares a HTML Controls e outros são tão diferentes como complexos como, por exemplo, o *DataGrid* e *DataList* (que facilmente permitem a visualização de dados) ou controlos que permitem a validação de dados inseridos por utilizadores.

Apesar de também ser possível inserir código directamente no ficheiro .aspx, tal não se revela positivo para a pretendida divisão entre *design* gráfico e implementação lógica. Por isso, a cada página está associado um ficheiro de código, designado de *codebehind*, onde são definidas as respostas a vários eventos, desde o que deverá ocorrer quando a página é inicialmente carregada (ir à base de dados e preencher caixas de texto com esses dados, por exemplo) até as acções a tomar quando um botão é pressionado (introduzir os dados existentes numa caixa de texto na base de dados). O ficheiro de *codebehind* pode ser escrito recorrendo a qualquer uma das linguagens .NET sendo, no entanto, necessário especificar, na página correspondente, qual a linguagem utilizada num *tag* especial chamado *Page*. A directiva *Page* é, na realidade, obrigatória e através dela são definidos outras informações importantes como, por exemplo, qual a classe existente no ficheiro de código a que se deseja associar a página, se o estado das variáveis deve ser mantido entre viagens ao servidor (o protocolo HTTP não suporta estados e é esse o comportamento habitual), ou se existirá suporte para objectos de sessão guardados no servidor. Outra *tag* especial é *Register*, e é graças a ela que são inseridos *User Controls* numa página.

Um *User Control* é um componente reutilizável, parte de uma página que se irá repetir por várias páginas (um cabeçalho, por exemplo) e que é possível definir uma única vez num ficheiro de extensão .ascx e, com recurso a um *tag*, inserir num local específico nas páginas

de destino. É também possível criar outro tipo de componentes reutilizáveis chamados *Custom Controls*. Os *Custom Controls* são, normalmente, usados para criarem versões personalizadas de *Web Controls* que, compilados num DLL, podem ser distribuídos e integrados no Visual Studio.NET (um exemplo seria aglutinar uma caixa de texto com o texto que normalmente a descreve num único objecto). Uma mistura de *User Controls* com *Custom Controls*, os *Composite Custom Controls* funcionam como *User Controls*, mas são compilados num DLL e prontos a distribuir (sem necessitar de um ficheiro .ascx).

Cada *ASP.NET Web Application* possui um ficheiro de configuração chamado *Web.config*. Através deste ficheiro de configuração é possível definir, para toda a aplicação: variáveis globais (*keys*); o tipo de autenticação (nenhuma, através do IIS, através do sistema *Passport* da Microsoft ou através do preenchimento de um formulário numa página – neste caso é necessário criar código que autentique os utilizadores relativamente a uma base de dados ou servidor LDAP da *ActiveDirectory*) e que páginas deverão necessitar de autorização para serem vistas; páginas personalizadas que deverão aparecer em caso de erro do sistema; ou opções relativas as informações de sessão (onde, como e durante quanto tempo manter activa uma sessão).

Outro ficheiro importante, criado automaticamente pelo sistema, é *Global.asax*. Através deste ficheiro é possível escrever código a ser executado quando a aplicação ou a sessão começa ou acaba.

Em termos de melhoria de performance, o ASP.NET também nos proporciona o uso de caches. Recorrendo à cache da aplicação é possível, por exemplo, retornar informação da base de dados uma vez, guardá-la na cache e aceder directamente à cache nas próximas vezes. O conteúdo da cache da aplicação poderia manter-se indefinidamente, ser refrescado passado algum tempo, ou estar associado a um ficheiro (monitorizado pelo ASP.NET) que, quando modificado, levaria ao refrescar da informação. É também possível determinar ao nível da página se desejamos que o seu conteúdo seja guardado na cache ou, num caso mais real, apenas partes das páginas sejam memorizadas (partindo uma página em *user controls* e definido a esse nível quais deveriam ir para a cache).

3.2.4.3.2 XML Web Services

Os XML Web Services são interfaces que se baseiam num conjunto de standards e protocolos para proporcionar a fácil partilha de informação e potenciar a comunicação entre organizações e pessoas através da Internet.

Um Web Service é descrito (*service description*) por de uma notação XML chamada WSDL (*Web Services Description Language*). Através de vários campos (como *Types*, *Message*, *Operation*, *Port Type*, *Binding*, *Port* ou *Service*), o WSDL descreve o que o Web Service consegue fazer, onde reside e como o invocar,

Para que os potenciais clientes possam ter acesso aos Web Services, as suas descrições são disponibilizadas numa espécie de directório, o qual segue a especificação UDDI (*Universal Description, Discovery and Integration*). Numa perspectiva de negócio, este repositório permite às empresas a publicação de um Web Service (e os respectivos interfaces), e aos clientes a pesquisa por serviços de um certo tipo que, posteriormente, poderão ser interligados com os seus próprios programas.

A comunicação na rede entre os Web Services e seus clientes ocorre recorrendo ao SOAP (*Simple Object Access Protocol*). Baseado em XML, o SOAP usa documentos-mensagem que descrevem as chamadas a funções e os seus parâmetros. Cada mensagem contém um *Envelope* (que define o conteúdo da mensagem), um *Header* (que, opcionalmente, contém informação sobre o próprio cabeçalho) e um *Body* (que contém informação de chamada e de resposta). As mensagens enviadas correm, regra geral, sobre HTTP, mas é possível a aplicação sobre outros protocolos como, por exemplo, SMTP. Tipicamente, o cliente do Web Service formata um pedido em SOAP e envia-o por via de uma instrução HTTP POST. O Web Service recebe a mensagem, cria um objecto e chama um método com os parâmetros fornecidos. Por último, o Web Service cria uma mensagem SOAP de resposta com o conteúdo devolvido.

3.2.4.4 ADO.NET

Tal como o ASP.NET, também o ADO.NET (componente para acesso a dados) é uma evolução natural para a nova plataforma .NET do anterior ADO (*ActiveX Data Objects*).

Os dados transmitidos com recurso ao ADO.NET estão encapsulados em XML, o que permite uma maior transparência e funcionalidades acrescidas.

Para obter os dados é necessário distinguir três fases: a conexão à base de dados, a execução do comando que se pretende ver cumprido e a navegação nos dados possivelmente devolvidos. Estas fases são tratadas com recurso a classes específicas para os interfaces ODBC, OleDb e a base de dados SQL Server.

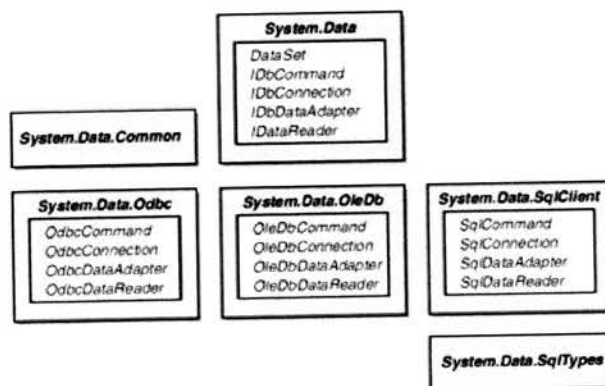


Ilustração 6 - Hierarquia de classes relacionadas com ADO.NET

Destaque para a diferenciação da forma como os dados podem ser percorridos após serem retornados pela base de dados. Tanto se poderá usar um *DataSet*, uma cópia em memória dos dados com total liberdade de navegação, ou usando um *DataReader*, um objecto mais optimizado mas que permite apenas a leitura e de uma forma sequencial. O objecto *DataSet* terá de ser usado conjuntamente com um *DataAdapter* que “liga” o *DataSet* à origem dos dados, permitindo o retorno e gravação de informação. Tanto *DataReaders* como *DataSets* podem ser associados a um Web Control de uma página ASP.NET, permitindo o seu preenchimento de uma forma simples.

3.2.5 *Secure Sockets Layer*, chaves públicas e privadas e certificados digitais

A tecnologia *Secure Sockets Layer* foi desenvolvida pela Netscape e é considerado o standard em termos de protocolo de segurança na *web*. O protocolo proporciona cifragem, autenticação de servidor ou cliente através de TCP/IP e verificação da integridade das mensagens.

O protocolo SSL recorre a pares de chaves assimétricas públicas e privadas (tecnologia RSA, *Rivest-Shamir-Adleman*) para cifrar e decifrar mensagens. A chave pública é distribuída, pelo contrário a chave privada é mantida em segredo. Assim, dados codificados pela chave pública apenas podem ser decifrados pela chave privada e vice-versa.

Usando este mecanismo é possível garantir a identidade dos intervenientes na comunicação. Por exemplo, o utilizador X envia uma mensagem ao utilizador Y, este cifra-a com a sua chave privada e devolve-a ao utilizador X que, usando a chave pública de Y, a decifra e compara com a enviada inicialmente (se forem iguais o utilizador X sabe que está a comunicar com o utilizador Y). Uma abordagem mais correcta consiste no uso de assinaturas digitais. Recorrendo ao exemplo dado anteriormente, o utilizador Y, ao receber a primeira mensagem não a codifica com a sua chave privada, mas cria uma *message digest* (uma mensagem codificada unidireccionalmente – usando, por exemplo, o algoritmo MD5 ou SHA – a partir da mensagem de X) e esta sim é codificada e enviada. O utilizador X descodifica a mensagem e compara o *message digest* com um computado a partir da mensagem que enviou inicialmente.

Se dois utilizadores distribuírem chaves públicas e ambos disserem que são “X” é difícil saber que estará a falar verdade. Assim, surgiu o conceito de certificado digital. Um certificado é constituído pelo nome da entidade idónea que o atribui, o nome do sujeito a que pertence o certificado, a sua chave pública e a validade do certificado. Para garantir a autenticidade do certificado, este é codificado pela chave privada da entidade emissora (cuja chave pública é conhecida).

Para a comunicação após a autenticação, é efectuada a partilha da chave de um algoritmo criptográfico simétrico (como o DES, *Data Encryption Standard*) que é enviado usando o mecanismo de chaves públicas e privadas. Assim, tendo ambas as partes a chave do algoritmo, as mensagens subsequentes podem ser codificadas recorrendo a esta (o que permite a confidencialidade).

Apesar de se saber de quem se trata e das mensagens serem confidenciais (graças à codificação), estas ainda podem ser desviadas. Por essa razão, é adicionado à mensagem um MAC (*Message Authentication Code*), um código que garante a autenticidade. O MAC utilizado é gerado por um código unidireccional (semelhante ao utilizado para a construção de *message digests*) a partir da chave simétrica usada para a comunicação e de parte da mensagem.

Assim, para o caso específico de um *site* que necessite de usar SSL, este terá de instalar um certificado digital num servidor. Como foi dito, este permite:

- autenticar o *site*, pois o certificado digital indica ao *browser web* do utilizador que a comunicação está a ocorrer com a organização indicada no certificado;
- manter a integridade e confidencialidade, graças à cifragem das mensagens trocadas entre servidor e cliente usando a tecnologia SSL que impede a intercepção e modificação dos dados;

- identificar visitantes, caso estes possuam, eles próprios, certificados digitais pessoais, o que levaria uma autenticação automática destes.

Explicando passo a passo os eventos, tudo se inicia quando o utilizador contacta um endereço protegido do *site* (identificável pelo endereço se iniciar por https) e o servidor responde enviando automaticamente o certificado digital. O *browser web* gera uma chave de sessão (de 40 ou 128-bit, conforme as potencialidades do *browser*) para codificar as mensagens enviadas. A chave de sessão também é codificada usando a chave pública que o *site* forneceu no certificado, pelo que apenas o *site* a poderá descodificar e, assim, descodificar a mensagem enviada. A partir daí as comunicações serão seguras (usando os mecanismos descritos anteriormente) e o utilizador será informado pelo seu *browser* (normalmente através de um ícone de uma chave ou um cadeado).

4 Especificação

A solução proposta, tendo em conta os requisitos apresentados e os recursos existentes na organização assentaria em termos tecnológicos no uso de tecnologia Microsoft .NET Framework 1.1, nomeadamente na sua vertente ASP.NET e C#, sendo o servidor *web* e a base de dados também Microsoft, Windows 2003 Server (IIS 6.0) e SQL Server 2000, respectivamente.

Em traços gerais, remetendo os aspectos mais pormenorizados para os diagramas seguintes e respectiva descrição, a solução teria nas suas vertentes Loja Web, a loja *online*, Cartão Virtual, o Cartão Virtual Global Notícias e Backoffice, de ser composta por um conjunto de páginas *web* (.aspx) diferentes e específicas mas que partilham os mesmos ficheiros de lógica de negócio, podendo também através destes aceder à base de dados onde se encontra a informação de utilizadores, produtos, encomendas ou passatempos.

4.1 Arquitectura lógica

4.1.1 Divisão em pacotes

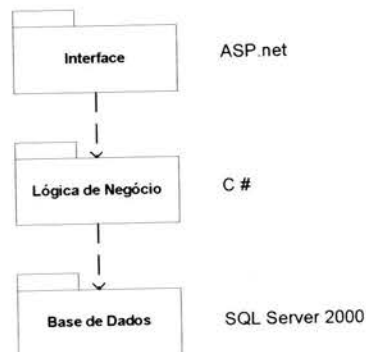


Diagrama 1 - Diagrama de pacotes

O projecto assenta em três camadas ou módulos lógicos: base de dados, lógica de negócio e apresentação ao utilizador através de um interface gráfico.

4.1.1.1 Base de dados

Representa a base de dados (em SQL Server 2000) onde irá ser guardada a informação persistente relevante para o correcto funcionamento do sistema. Neste pacote também se incluem as *stored procedures* através das quais se efectua a comunicação entre a camada de base de dados e a camada de lógica de negócio.

4.1.1.2 Lógica de negócio

A lógica de negócio do sistema é composta por um conjunto de ficheiros em C#, cujas classes representam os objectos relevantes ao contexto da aplicação, e por métodos que permitam a ligação com a base de dados.

4.1.1.3 Interface

Composta por páginas em ASP.NET (.aspx) responsáveis pela interacção entre o utilizador e o sistema, e pelos correspondentes ficheiros de *codebehind* onde são efectuadas chamadas a métodos presentes na lógica de negócio que efectivam a comunicação com a base de dados.

4.1.2 Modelo de classes do domínio

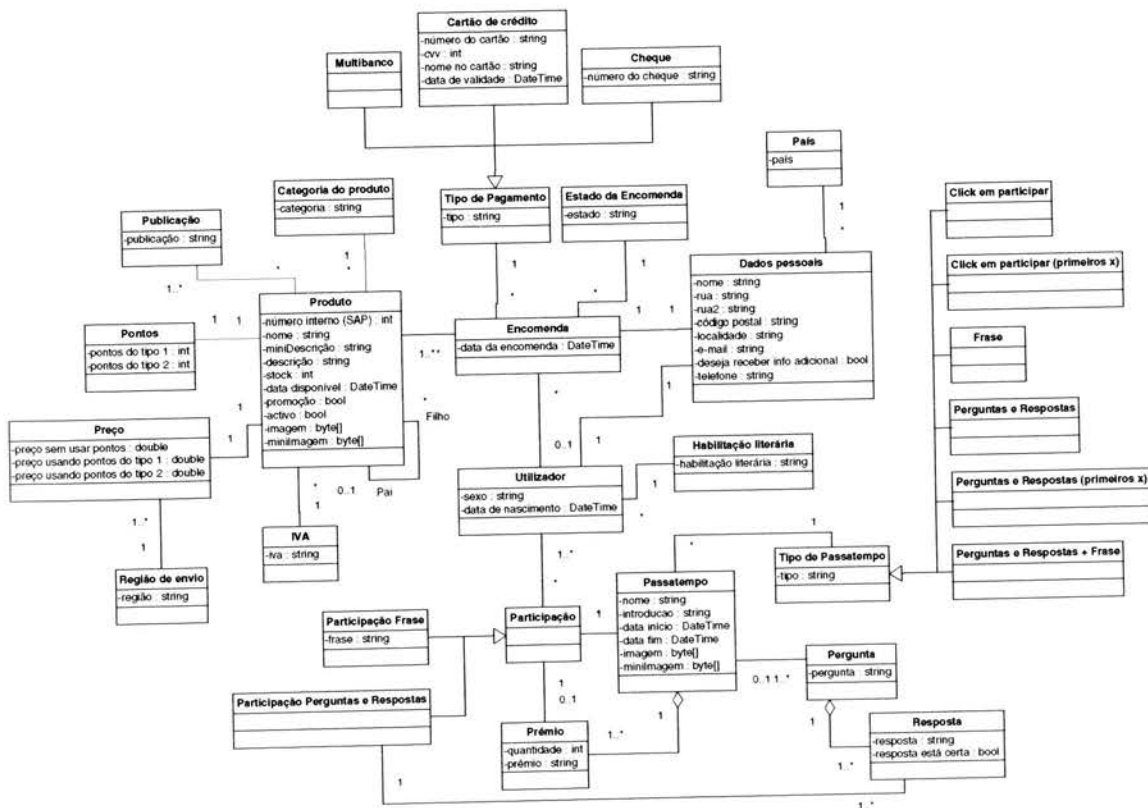


Diagrama 2- Modelo de classes do domínio

Uma *Publicação* possui vários *Produtos* e um *Produto* pode pertencer a vários *Publicações*.
 Uma *Produto* pertence a uma *Categoria* enquanto que uma *Categoria* pode ter vários *Produtos*.
 Um *Produto* é constituído por um código adicional (número interno) para mais fácil identificação na solução SAP existente na organização, um nome, as duas descrições e duas imagens alusivas que aparecem em diferentes locais dos *sites*, o stock existente, a data em que

fica disponível (aparece) no *site* da Loja Web, se aparece na parte “em destaque” da página principal, se está activo (é visualizável na Loja Web).

Um *Produto* possui *Pontos* (do tipo 1 e do tipo 2) e está associado uma categoria de *IVA*.

Um *Produto* também tem *Preços* (diferenciáveis consoante não se usa pontos, ou se usa pontos do tipo 1 ou do tipo 2), sendo que para cada *Região de envio* corresponde um *Preço* diferente.

Um *Produto* pode opcionalmente pertencer a um outro *Produto*, fazendo assim parte de uma colecção representada na sua totalidade pelo *Produto* aglutinador.

Para um *Utilizador* se registar no Cartão Virtual tem de fornecer um conjunto de dados (o nome, sexo, data de nascimento, morada, telefone, habilitações literárias, o endereço electrónico e se deseja receber informação adicional da Global Notícias). Parte desses dados (*Dados pessoais*) são igualmente exigidos quando um indivíduo coloca uma *Encomenda* (são os dados para envio e contacto).

Uma *Encomenda* é constituída por um ou mais *Produtos* e tem associada a data da colocação da encomenda que pode ser efectuada tanto por *Utilizadores* registados no Cartão Virtual como por indivíduos que escolheram não se registar.

Uma *Encomenda* precisa de ter um *Tipo de Pagamento*, que tanto pode ser do tipo *Multibanco*, *Cheque* (possuindo um número de cheque) ou *Cartão de Crédito* (possuindo um número de cartão, o nome impresso no cartão, uma data de validade e um CVV¹).

Um *Passatempo* é de um certo *Tipo de Passatempo*, que tanto pode ser:

- *Click em Participar* ou *Click em Participar (primeiros x)*, passatempos em que basta pressionar um botão para participar e que diferem no número de participações permitidas (no segundo caso, após as primeiras x participações correctas o passatempo termina);
- *Frase*, passatempos em é pedida uma frase para poder participar;
- *Perguntas e Respostas* ou *Perguntas e Respostas (primeiros x)*, passatempos cuja participação compreende em conjunto de perguntas de escolha múltipla e que diferem no número de participações permitidas (no segundo caso, após as primeiras x participações correctas o passatempo termina);
- *Perguntas e Respostas + Frase*, passatempos nos quais é pedido ao utilizador que responda correctamente a perguntas de escolha múltipla e escreva uma frase alusiva ao passatempo.

Um *Passatempo* pode possuir, consoante o tipo de passatempo, um conjunto de *Perguntas* às quais estão associadas um conjunto de *Respostas* (certas e erradas).

Cada *Utilizador* pode ter uma *Participação* num *Passatempo*. Uma *Participação* pode, consoante o tipo de passatempo participado, incluir uma frase ou o conjunto de respostas dadas.

Um *Passatempo* possui um ou mais *Prémios* e cada *Prémio* é atribuível a uma *Participação*.

¹ “Card Verification Value” (CVV2 ou CVC2) - código existente no verso dos cartões de crédito aceites (VISA ou MasterCard / EuroCard) que, fornecido à entidade bancária juntamente com os restantes dados de pagamento permite verificar a autenticidade e liquidez do cartão

4.2 Arquitectura física

4.2.1 Distribuição (*Deployment*)

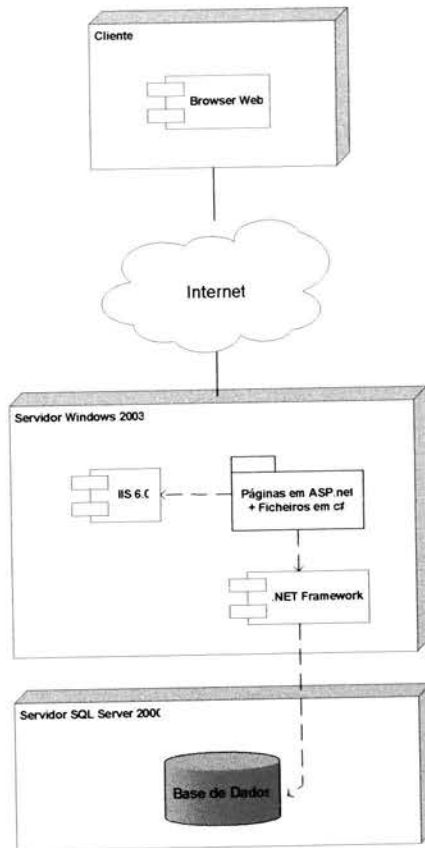


Diagrama 3 - Diagrama de distribuição

O sistema assenta numa arquitectura *3-tier* na qual o cliente acede à aplicação via Internet (usando um *browser web*). As páginas dinâmicas acedidas no servidor *web* (IIS) são em ASP.net e a lógica de negócio é expressa em ficheiros em C# pelo que, em ambos os casos, é necessário o recurso à .NET Framework. Por último, os dados residentes numa base de dados em SQL Server 2000 são objecto das consultas e modificações via a .NET Framework.

4.2.2 Descrição dos componentes

A aplicação tem, maioritariamente, como componentes páginas em ASP.NET (ficheiros .aspx associados a ficheiros .aspx.cs de *codebehind*) que acedem a um DLL para processamento lógico e acesso à base de dados.

Podemos dividir os componentes em quatro grandes grupos:

- a Loja Web, composta por páginas reveladas aos utilizadores quando visionam a Loja Web e o DLL gerado para essas páginas;
- o Cartão Virtual, composto pelas páginas do *site* do Cartão Virtual Global Notícias e o DLL gerado para essas páginas;

- o Backoffice composto pelas páginas correspondentes à parte de *backoffice* do sistema e o DLL correspondente;
- um DLL, comum às três partes anteriores, composto pela parte da lógica de negócio.

Adicionalmente, também existe um Windows Service e um Web Service cujo funcionamento se encontra também dependente do DLL gerado a partir dos ficheiros de código da lógica de negócio.

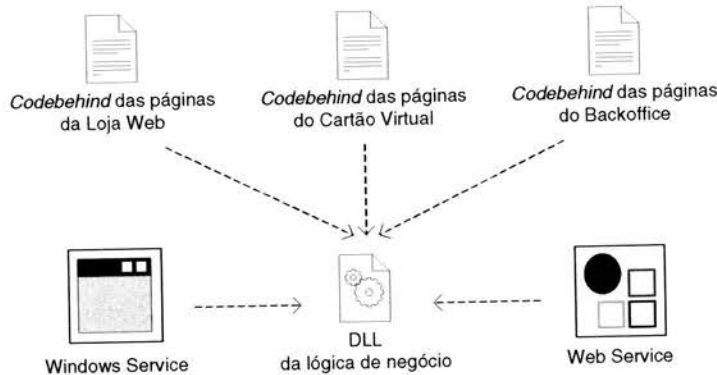


Diagrama 4 – Relação dos componentes com a lógica de negócio

Nas próximas páginas podemos aferir da relação entre páginas (componentes) e verificar a sua semelhança com os casos de uso que a aplicação oferece.

4.2.2.1 Loja Web

A Loja Web possui páginas que não contém código HTML relevante, sendo usadas apenas por outras páginas com o intuito de invocar o seu *codebehind*.



Diagrama 5 - Páginas usadas exclusivamente para processamento de dados

As páginas `adicionarCarrinho.aspx` e `adicionarCarrinhoCDI.aspx` são chamadas quando se adiciona um produto a partir da página da Loja Web (`detalhe.aspx`) ou a partir da página do CDI, respectivamente. A página `eliminaCarrinho.aspx` é chamada quando se elimina um item no carrinho de compras (`carrinho.aspx`). A página `logout.aspx` pode ser invocada a partir de qualquer página no *site*, bastando que este se encontre autenticado. A página `picReader.aspx` devolve a imagem associada a um produto (versão maior ou menor), sendo invocada onde essas imagens necessitam de ser reveladas.

As seguintes páginas são páginas de erro personalizadas que aparecem ao utilizador quando erros inesperados acontecem (`erro.aspx`), ou quando se tenta aceder a páginas proibidas ou inexistentes.



Diagrama 6 - Páginas de erro da Loja Web

As páginas de erro e a página `picReader.aspx` têm correspondentes nos *sites* do Cartão Virtual e Backoffice e, por necessitar de autenticação, a página `logout.aspx` também existe no *site* do Cartão Virtual.

4.2.2.1.1 *Web User Controls*

Grande parte do acesso às páginas existentes no *site* da Loja Web acontece graças à presença de *Web User Controls* embebidos em todas as páginas.

O controlo `menuCtl.aspx` é incluído na parte esquerda de cada página e apresenta um menu onde aparecem as publicações da Global Notícias que vendem produtos na Loja Web, as várias categorias de produtos existentes e um *link* de ajuda (`ajuda.aspx`). Ao seleccionar uma das publicações ou categorias, o internauta é remetido para uma listagem dos produtos da Loja Web afectos à publicação ou categoria (`produtos.aspx`).

O controlo `headerCtl.aspx` encontra-se do cabeçalho das páginas e contém *links* para:

- a página de registo no Cartão Virtual Global Notícias (`registo.aspx`);
- a página de autenticação (`login.aspx`) ou de saída de sessão (`logout.aspx`), caso o utilizador já se encontre autenticado (em ambos os casos, em caso de sucesso o utilizador é enviado para a página principal – `default.aspx`);
- a página de conta pessoal dos clientes da Loja Web (`conta.aspx`);
- a página onde é revelado o conteúdo do carrinho de compras do cliente (`carrinho.aspx`);
- a página onde é definida a região de envio, afectando a visualização dos preço dos produtos (`região.aspx`);
- a página do Cartão Virtual Global Notícias.

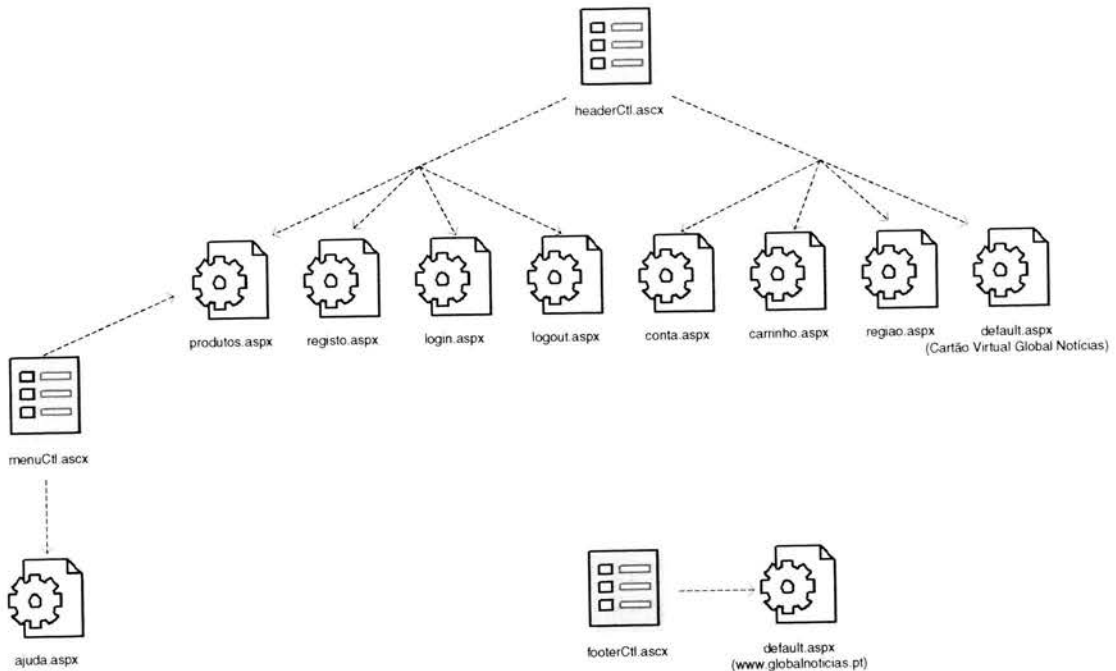


Diagrama 7 - Web User Controls usados no site da Loja Web

Adicionalmente através da funcionalidade de pesquisa de produtos (sem parameterização, por publicação ou por categoria) existente em headerCtl.aspx, este controlo também está relacionado com produtos.aspx, onde são mostrados os resultados da pesquisa.

É também em headerCtl.aspx que um utilizador registado pode verificar que está autenticado (é mostrada uma mensagem de boas vindas e os pontos disponíveis e gastos na sessão).

O controlo footerCtl.aspx posiciona-se no rodapé das páginas e permite, através de um *link*, o acesso à página principal da Global Notícias.

4.2.2.1.2 Página inicial

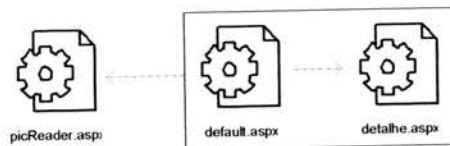


Diagrama 8 - Visão principal a partir da página inicial

A página principal está ligada a todas as páginas referenciadas nos controlos da Loja Web. Adicionalmente tem ligação à página picReader.aspx devido à necessidade de ver as imagens associadas aos produtos em destaque (mostrados na página inicial) e à página detalhe.aspx, o local para onde são redireccionados os utilizadores que desejem ver mais pormenores sobre o artigo em promoção seleccionado.

4.2.2.1.3 Registo no Cartão Virtual



Diagrama 9 - Secção de registo no Cartão Virtual

Na página `registo.aspx`, o utilizador pode inserir todos os dados necessários ao registo no cartão, sendo posteriormente redireccionado para a página `registoOK.aspx` que confirma ao utilizador que a operação de registo foi concluída com sucesso.

4.2.2.1.4 Visualizar produtos e colocar uma encomenda

Um utilizador que deseje ver ou escolher os produtos existentes na Loja Web deverá ir parar à página `produtos.aspx`. Esta é a página onde é mostrada uma listagem dos produtos pertencentes a uma publicação ou categoria (caso se escolham essas opções existentes em `menuCtl.ascx`) ou que correspondam ao critério definido numa pesquisa (usando a caixa de texto em `headerCtl.ascx`). Escolhendo ver o detalhe de um dos produtos, o utilizador é levado a `detalhe.aspx` onde são dados mais detalhes sobre o produto e é dada a opção de adicionar o produto ao carrinho (com ou sem uso de pontos, caso o utilizador esteja autenticado e possua pontos suficientes para esse produto). Tanto `detalhe.aspx` como `produtos.aspx` têm em comum uma ligação a `picReader.aspx` para ver imagens alusivas ao produto e, quando os produtos visualizados são colecções, *links* para ver os sub-produtos de uma colecção (listagem mostrada mais uma vez em `produtos.aspx`).

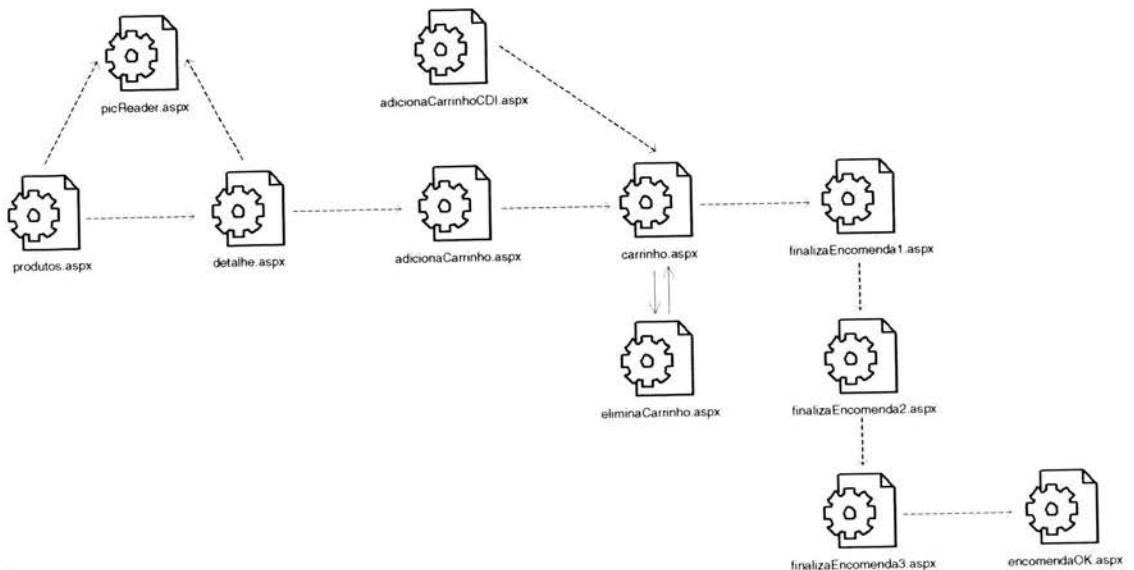


Diagrama 10 - Passos para a colocação de uma encomenda

Ao adicionar um produto ao carrinho o utilizador é levado a `carrinho.aspx` após uma série de verificações efectuadas em `adicionaCarrinho.aspx` (ou em `adicionaCarrinhoCDI.aspx`, caso o utilizador venha redireccionado do *site* do Centro de Documentação e Informação do Jornal

de Notícias). Em carrinho.aspx o utilizador vê a listagem de produtos existentes no seu carrinho de compras, podendo eliminar itens (chamada a eliminaCarrinho.aspx para processar o pedido e reenvio para carrinho.aspx) ou modificar quantidades. Quando estiver satisfeito com o seu conteúdo o utilizador pode escolher colocar a encomenda, sendo enviado para finalizaEncomenda1.aspx para preencher os dados de envio. Se estiver autenticado, os campos existentes nesta página são automaticamente preenchidos com os dados fornecidos aquando do registo no Cartão Virtual. O próximo passo encontra-se em finalizaEncomenda2.aspx e constitui a escolha da forma de pagamento e preenchimento de dados adicionais relativos à mesma. A última hipótese de recuar na colocação da encomenda é em finalizaEncomenda3.aspx. Aqui é mostrado o resumo da encomenda a efectuar e os dados fornecidos nas páginas anteriores. Se concordar com os dados revelados o utilizador é enviado para encomendaOK.aspx, a página que confirma que a encomenda foi colocada, onde são dadas instruções relativamente a como efectuar o pagamento e fornecidos códigos para monitorizar o estado da encomenda agora efectuada.

4.2.2.1.5 Secção da Conta Pessoal

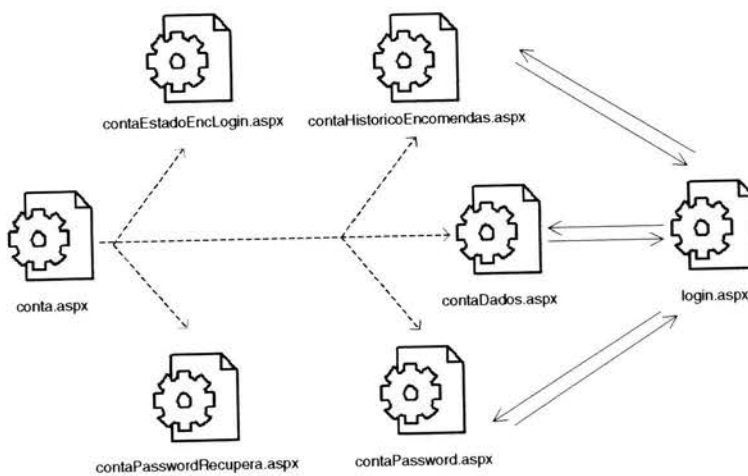


Diagrama 11 - Visão global da secção da Conta Pessoal

Acedendo à página principal da secção da Conta Pessoal os utilizadores têm acesso às várias funcionalidades que compõem essa secção.



Diagrama 12 - Ver estado de uma encomenda

Os utilizadores não registados podem ver o estado das encomendas efectuadas usando o número da encomenda e um código adicional, ambos disponibilizados na página de confirmação de encomenda efectuada. Assim, o utilizador não registado introduz os códigos da encomenda na página `contaEstadoEncLogin.aspx` e é enviado para a página `contaEstadoEnc.aspx`, caso os dados estejam correctos. Nesta página o estado e resumo da

encomenda são mostrados, podendo o utilizador cancelar a encomenda (se ainda estiver no estado de “nova” ou “pendente”). Se a encomenda for cancelada o utilizador é reencaminhado para uma página que confirma que a operação foi efectuada (contaEncCancelada.aspx).



Diagrama 13 - Recuperação de *password*

Os utilizadores registados que se tenham esquecido da sua *password* podem recorrer à página `contaPasswordRecupera.aspx`, que lhe pede a sua data de nascimento e o e-mail com que se registou no Cartão Virtual. Caso estes dados sejam confirmados na base de dados, o utilizador é enviado para a página `contaPasswordRecuperaOK.aspx` que lhe confirma que foi enviada uma nova *password* por e-mail.



Diagrama 14 - Modificar os dados pessoais e a *password*

Os utilizadores registados que desejem modificar os dados pessoais fornecidos aquando do registo no Cartão Virtual podem fazê-lo acedendo às páginas `contaDados.aspx` e `contaPassword.aspx` respectivamente. Se não estiverem autenticados ao acederem a estas páginas os utilizadores são enviados para a página `login.aspx` sendo, após autenticação correcta, redireccionados para a página inicialmente pedida. Na página `contaDados.aspx` o utilizador é confrontado com os mesmos campos existentes em `registo.aspx` (preenchidos com os dados anteriormente fornecidos) e pode modificar os dados pretendidos. É mostrada a página `contaDadosOK.aspx`, semelhante a `registoOK.aspx`, para confirmar o sucesso da operação. Na página `contaPassword.aspx`, o utilizador insere a nova *password* em duas caixas de texto (uma delas de confirmação) e, se o preenchimento for correcto, é-lhe mostrada a página `contaPasswordOK.aspx` que demonstra o sucesso da operação.

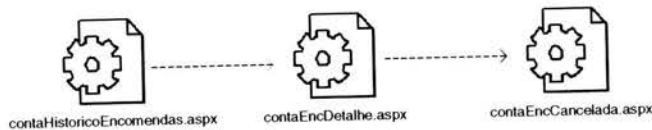


Diagrama 15 - Visualizar o histórico das encomendas efectuadas

A última opção da secção Conta Pessoal contempla os utilizadores registados que desejem ver uma relação das encomendas por si efectuadas. Tal como no caso da modificação de dados pessoais e de *password* é necessário estar autenticado para aceder a esta subsecção. Na página `contaHistoricoEncomendas.aspx` é mostrada a listagem de encomendas efectuadas (com alguns dados como a data, o estado ou o custo) e, seleccionando uma delas, poderá ver mais alguns detalhes (como os itens da encomenda, por exemplo) na página `contaEncDetalhe.aspx`. Nesta página, caso o estado da encomenda o permita, o utilizador registado pode cancelar a encomenda, sendo redireccionado para `contaEncCancelada.aspx` em caso de sucesso da operação de cancelamento.

4.2.2.1.6 Secção de Ajuda

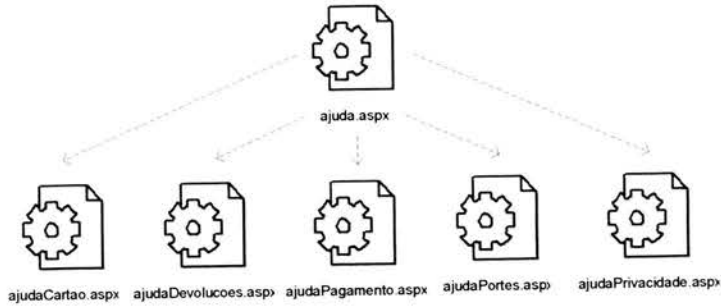


Diagrama 16 - Visão da secção de ajuda

Os utilizadores da Loja Web têm acesso a uma página de ajuda com *links* para várias outras páginas que explicam/clarificam certas dúvidas que possam surgir a quem navega o *site* e não conhece o seu funcionamento.

4.2.2.2 Cartão Virtual

4.2.2.2.1 Web User Controls

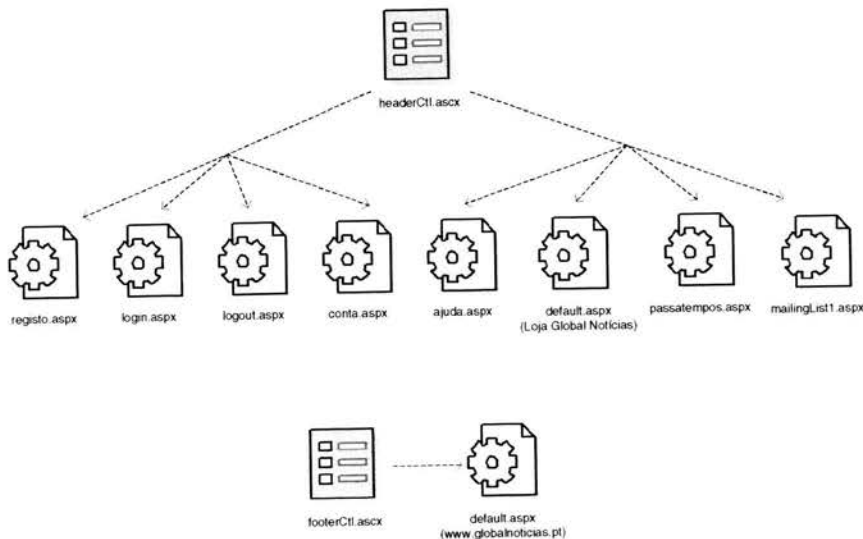


Diagrama 17 - Web User Controls usados no site do Cartão Virtual

O *site* do Cartão Virtual possui, tal como o *site* da Loja Web, um conjunto de *Web User Controls* embebidos em todas as páginas e que possibilitam o acesso rápido a várias secções.

O controlo `headerCtl.ascx`, colocado no cabeçalho, disponibiliza acesso às seguintes páginas:

- `registo.aspx`, que permite o registo de utilizadores no Cartão Virtual;
- `login.aspx` ou `logout.aspx`, que permitem ao utilizador registado autenticar-se ou terminar a sua sessão caso esteja autenticado (em ambos os casos, em caso de sucesso o utilizador é enviado para a página principal – `default.aspx`);
- `conta.aspx`, a secção de Conta Pessoal dos utilizadores registados;

- `passatempos.aspx`, a secção dos passatempos do Cartão Virtual;
- `mailingList1.aspx`, a primeira página da secção que permite a escolha de secções das publicações da Global Notícias para receber por e-mail;
- `ajuda.aspx`, a secção de ajuda;
- a página principal da Loja Web.

O controlo `footerCtl.ascx`, colocado no rodapé, permite o acesso à página principal do grupo Global Notícias.

4.2.2.2.2 Página inicial

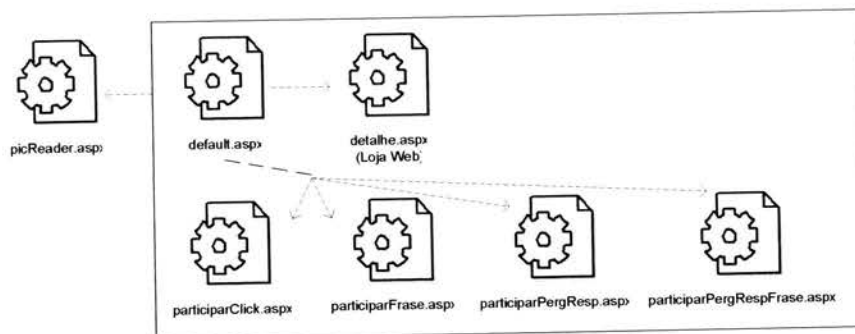


Diagrama 18 - Visão principal a partir da página inicial

Na página inicial do *site* do Cartão Virtual (`default.aspx`), ao utilizador é mostrado um produto dos que estão em promoção na Loja Web e um *link* para o seu detalhe (`detalhe.aspx`), e um dos passatempos actualmente a decorrer no *site* e um *link* para participar no referido passatempo (`participarClick.aspx`, `participarFrase.aspx`, `participarPergResp.aspx` ou `participarPergRespFrase.aspx` consoante o passatempo seja do tipo “Click para participar”, “Frase”, “Perguntas e Respostas” ou “Perguntas e Respostas + Frase” respectivamente). Como tanto o produto como o passatempo (assim como as páginas de detalhe de produto e de participações em passatempos) possuem imagens alusivas, a página `picReader.aspx` é chamada.

4.2.2.2.3 Registo no Cartão Virtual e secção de ajuda

O registo no Cartão Virtual e a secção de ajuda são em tudo semelhantes às mesmas secções existentes no *site* da Loja Web pelo que se remetem pormenores destas secções para o anteriormente explicado.

4.2.2.2.4 Secção da Conta Pessoal

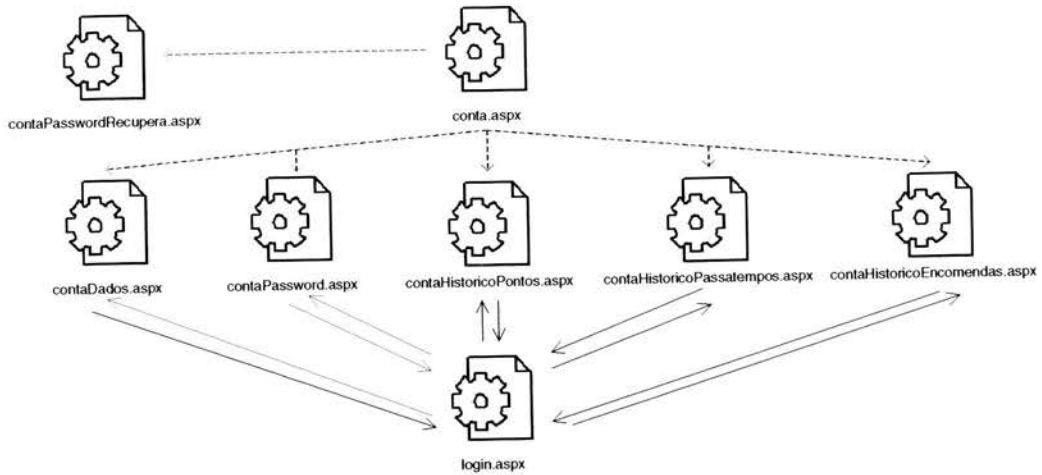


Diagrama 19 - Secção da Conta Pessoal

A secção da Conta Pessoal existente no *site* do Cartão Virtual é semelhante à existente no *site* da Loja Web. As únicas diferenças são a inexistência da funcionalidade de verificação do estado de encomendas (esta secção da Conta Pessoal é exclusiva para utilizadores registados no Cartão Virtual e todas as páginas – excepto `contaPasswordRecupera.aspx` – exigem a autenticação) e a existência de um histórico de pontos e de um histórico de passatempos participados. As páginas `contaPasswordRecupera.aspx`, `contaDados.aspx`, `contaPassword.aspx` e `contaHistoricoEncomendas.aspx` são iguais em termos de funcionamento às existentes na Loja Web pelo que se remete para o explicado anteriormente relativamente a estas subsecções.

Sobre as novas funcionalidades, a página `contaHistoricoPontos.aspx` mostra uma listagem dos pontos ganhos ou gastos, as datas em que tal ocorreu e o motivo para o sucedido.

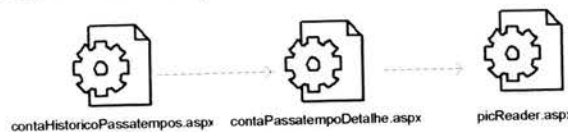


Diagrama 20 - Histórico de passatempos

A página de histórico de passatempos (`contaHistoricoPassatemplos.aspx`) mostra uma listagem dos passatempos participados pelo utilizador e permite, para cada um dos passatempos, a visualização (em `contaPassatempoDetalhe.aspx`) de quais as respostas ou frases dadas na participação, as respostas correctas ou frases vencedoras e os vencedores do passatempo (dependendo do tipo de passatempo participado). Por mostrar imagens alusivas aos passatempos também existe uma relação entre `contaPassatempoDetalhe.aspx` e `picReader.aspx`.

4.2.2.2.5 Secção dos Passatempos

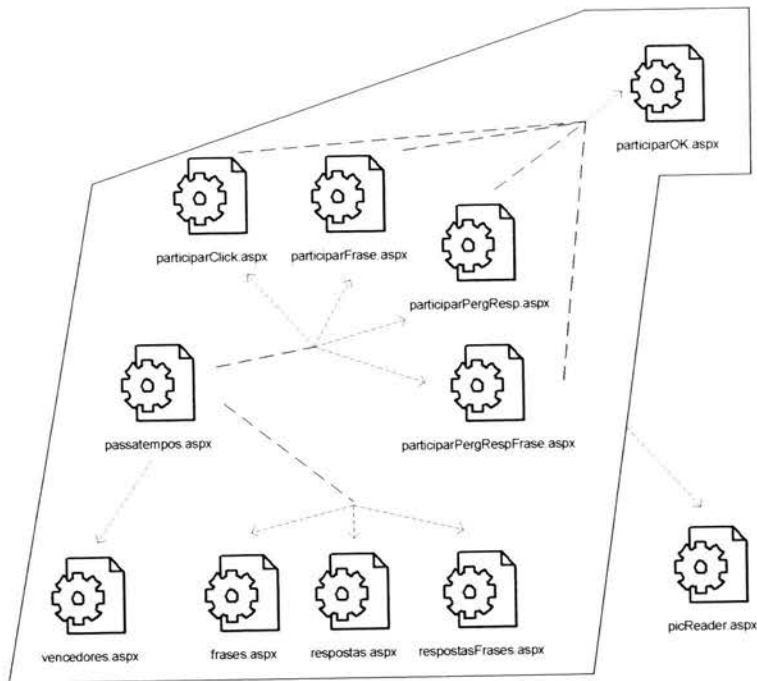


Diagrama 21 - Secção dos Passatempos

Na página principal da secção passatempos o utilizador pode ver os passatempos activos e participar neles (`participarClick.aspx`, `participarFrase.aspx`, `participarPergResp.aspx` ou `participarPergRespFrase.aspx` consoante o passatempo seja do tipo “Click para participar”, “Frase”, “Perguntas e Respostas” ou “Perguntas e Respostas + Frase” respectivamente). Após a participação no passatempo o utilizador é redireccionado para `participarOK.aspx` que confirma que a sua participação foi contabilizada.

Igualmente em `passatempos.aspx` o utilizador pode ver os últimos passatempos terminados, assim como os seus vencedores (em `vencedores.aspx`). Pode verificar quais as frases que ganharam prémios (se o passatempo for do tipo “Frase”), quais as respostas correctas (se o passatempo for do tipo “Perguntas e Respostas”) ou tanto as frases como as respostas (se o passatempo for do tipo “Perguntas e Respostas + Frase”).

Em todas as páginas desta secção são mostradas imagens relativas aos passatempos pelo que existe uma relação com `picReader.aspx`.

4.2.2.3 Backoffice

4.2.2.3.1 Web User Controls

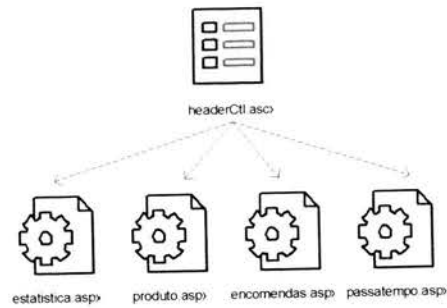


Diagrama 22 - Web User Control usados no Backoffice

A parte de Backoffice do sistema usa apenas um *Web User Control* embebido no cabeçalho de todas as páginas e que permite o rápido acesso às secções existentes:

- as estatísticas (estatística.aspx);
- a gestão de produtos (produto.aspx);
- o processamento de encomendas (encomendas.aspx);
- a gestão de passatempos (passatempo.aspx).

4.2.2.3.2 Página inicial

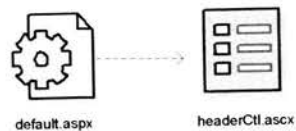


Diagrama 23 - Visão da página principal

Como se pode ver pelo esquema, a página inicial não tem nenhum conteúdo especialmente relevante instruindo apenas o utilizador a utilizar os *links* existentes em headerCtl.ascx para aceder à secção pretendida.

4.2.2.3.3 Secção das Estatísticas

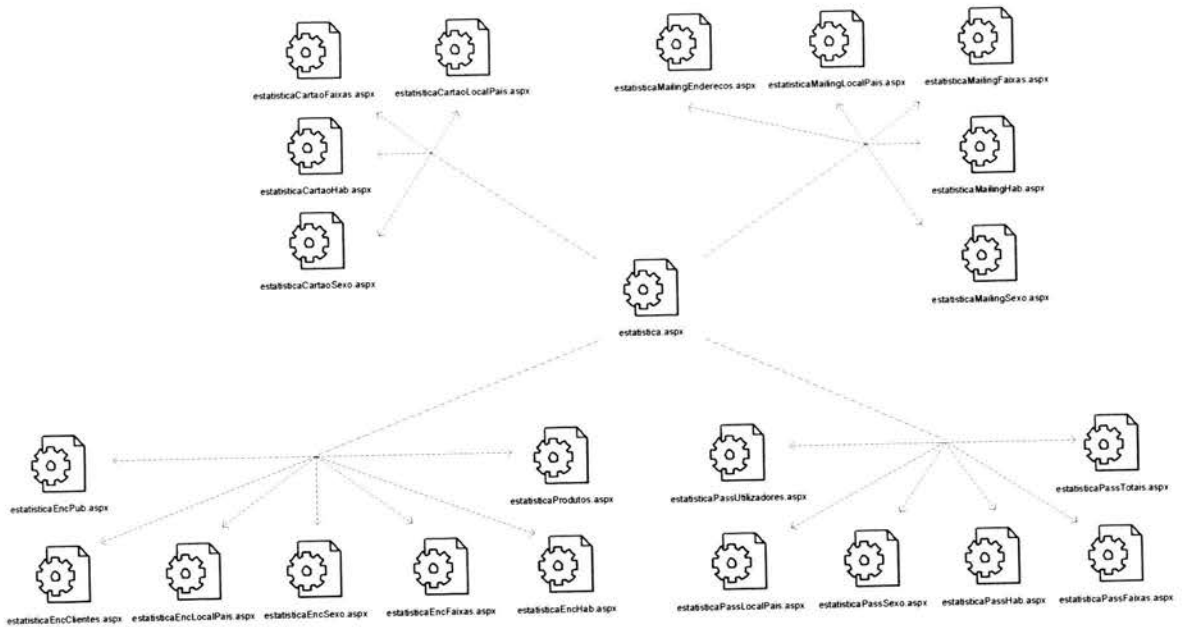


Diagrama 24 - Visão da secção das Estatísticas

A partir da página estatística.aspx o utilizador tem acesso a vários dados estatísticos divididos por subsecções:

- relativos às vendas na Loja Web:
 - número e percentagens de vendas por localidades portuguesas ou por países (estatísticaEncLocalPais.aspx), em todos os meses ou num mês específico;
 - número e percentagens de vendas por sexos – dos utilizadores registados – (estatísticaEncSexo.aspx), em todos os meses ou num mês específico;
 - número e percentagens de vendas por faixas etárias – dos utilizadores registados – (estatísticaEncFaixas.aspx), em todos os meses ou num mês específico;
 - número e percentagens de vendas por habilitações literárias – dos utilizadores registados – (estatísticaEncHab.aspx), em todos os meses ou num mês específico;
 - número e valor dos produtos vendidos por publicações (estatísticaEncPub.aspx), em todos os meses ou num mês específico (podendo diferenciar por encomendas realizadas por todos ou só por utilizadores registados ou não-registados);
 - a identidade dos melhores clientes da Loja Web, a quantidade e valor das encomendas realizadas (estatísticaEncClientes.aspx), em todos os meses ou num mês específico (podendo diferenciar por encomendas realizadas por todos ou só por utilizadores registados ou não-registados);
 - os produtos (quantidade e valor) mais vendidos (estatísticaProdutos.aspx), em todos os meses ou num mês específico (podendo diferenciar por produtos de uma publicação ou categoria).

- relativos aos registos no Cartão Virtual:
 - número e percentagens de registos por localidades portuguesas ou por países (estatisticaCartaoLocalPais.aspx), em todos os meses ou num mês específico;
 - número e percentagens de registos por sexos (estatisticaCartaoSexo.aspx), em todos os meses ou num mês específico;
 - número e percentagens de registos por faixas etárias (estatisticaCartaoFaixas.aspx), em todos os meses ou num mês específico;
 - número e percentagens de registos por habilitações literárias (estatisticaCartaoHab.aspx), em todos os meses ou num mês específico.
- relativos às participações nos passatempos:
 - número e percentagens de participações por localidades portuguesas ou por países (estatisticaPassLocalPais.aspx), em todos os meses ou num mês específico;
 - número e percentagens de participações por sexos (estatisticaPassSexo.aspx), em todos os meses ou num mês específico;
 - número e percentagens de participações por faixas etárias (estatisticaPassFaixas.aspx), em todos os meses ou num mês específico;
 - número e percentagens de participações por habilitações literárias (estatisticaPassHab.aspx), em todos os meses ou num mês específico;
 - a identidade dos utilizadores que mais participam e número de participações (estatisticaPassUtilizadores.aspx), em todos os meses ou num mês específico;
 - número e percentagens dos passatempos mais participados (estatisticaPassTotais.aspx), em todos os meses ou num mês específico;
- relativos aos utilizadores que desejam receber informações adicionais da Global Notícias:
 - número e percentagens de pedidos por localidades portuguesas ou por países (estatisticaMailingLocalPais.aspx), em todos os meses ou num mês específico;
 - número e percentagens de pedidos por sexos – dos utilizadores registados – (estatisticaMailingSexo.aspx), em todos os meses ou num mês específico;
 - número e percentagens de pedidos por faixas etárias – dos utilizadores registados – (estatisticaMailingFaixas.aspx), em todos os meses ou num mês específico;
 - número e percentagens de pedidos por habilitações literárias – dos utilizadores registados – (estatisticaMailingHab.aspx), em todos os meses ou num mês específico.

Adicionalmente foi incluído nesta secção de estatísticas, na parte relativa aos utilizadores que desejam receber informações adicionais, um *link* para uma página (estatisticaMailingEnderecos.aspx) onde se pode ver todos os endereços de e-mail dos utilizadores que pretendem receber essas informações (ou apenas os endereços dos registados ou não-registados).

4.2.2.3.4 Secção de Gestão de Produtos

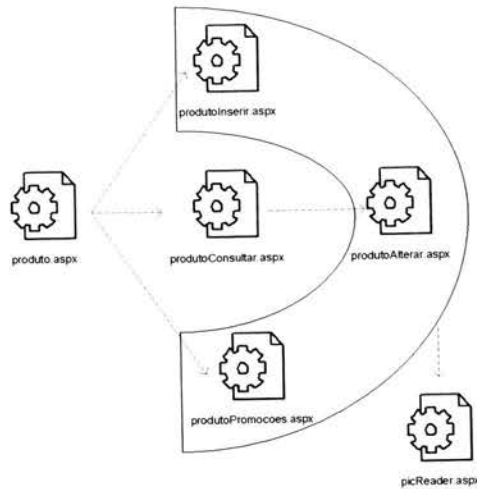


Diagrama 25 – Secção de Gestão de Produtos

Na página principal da secção de gestão de produtos o utilizador vê uma listagem dos produtos cujo stock está em estado crítico, podendo escolher um desses produtos e editar o seu stock. Existem também três *links* que permitem acesso a subsecções desta categoria.

Um dos *links* leva à página de introdução de novos produtos (`produtoInserir.aspx`) onde são pedidos todos os dados necessários (publicação, categoria, nome, imagem, preços e pontos, entre outros), mostrado um *preview* de como o produto apareceria na Loja Web e, se corresponder ao desejado, inserido o produto.

O segundo *link* envia o utilizador para a página `produtoConsultar.aspx` onde, fornecendo um texto, são procurados produtos que correspondam aos critérios e os resultados mostrados sob forma de listagem. Escolhendo um desses resultados o utilizador é reencaminhado para `produtoAlterar.aspx` onde pode alterar o produto (uma página em tudo semelhante a `produtoInserir.aspx` com a diferença de os campos estarem inicialmente preenchidos com os dados actuais do produto e o *preview* do produto já se encontrar visível).

Por último, a página `produtoPromocoes.aspx` também se encontra acessível através de `produto.aspx` e permite o seleccionar dos produtos em destaque que aparecem na Loja Web. Na página são mostrados *previews* dos produtos actualmente em destaque (como aparecem na Loja Web) e também campos de pesquisa semelhantes aos que se encontram em `produtoConsultar.aspx` para permitir o aparecimento de uma listagem com produtos e acrescentar/retirar um deles ao lote dos artigos em destaque.

A existência de *previews* em `produtoInserir.aspx`, `produtoAlterar.aspx` e `produtoPromocoes.aspx` leva a que estas páginas necessitem de aceder a `picReader.aspx`.

4.2.2.3.5 Secção de Gestão das Encomendas

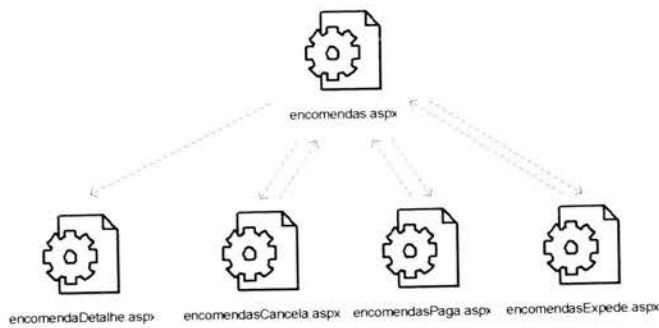


Diagrama 26 - Visão da secção de gestão de Encomendas

Na página principal da secção de gestão de encomendas o utilizador é confrontado, na sua parte superior, com campos para inserir critérios de parameterização (delimitar por estado, datas, clientes ou número de encomendas) a aplicar sobre a listagem de encomendas que aparece na parte inferior da página. Escolhendo uma das encomendas existentes na listagem é mostrada uma descrição mais detalhada, no fundo da página, e o utilizador passa a ter acesso às opções (dependendo do estado no qual se encontra a encomenda) de:

- abrir uma página (*encomendaDetalhe.aspx*) na qual é mostrado o detalhe da encomenda num formato mais conveniente para impressão;
- dar a encomenda como cancelada, chamando a página *encomendasCancela.aspx* que processa o pedido (no seu *codebehind*) e redirecciona para *encomendas.aspx*, mostrando as encomendas canceladas;
- dar a encomenda como paga, chamando *encomendasPaga.aspx* que processa o pedido e redirecciona para *encomendas.aspx*, mostrando as encomendas pagas;
- dar a encomenda como expedida, chamando *encomendasExpede.aspx* que processa o pedido e redirecciona para *encomendas.aspx*, mostrando as encomendas expedidas.

4.2.2.3.6 Secção de gestão dos Passatempos

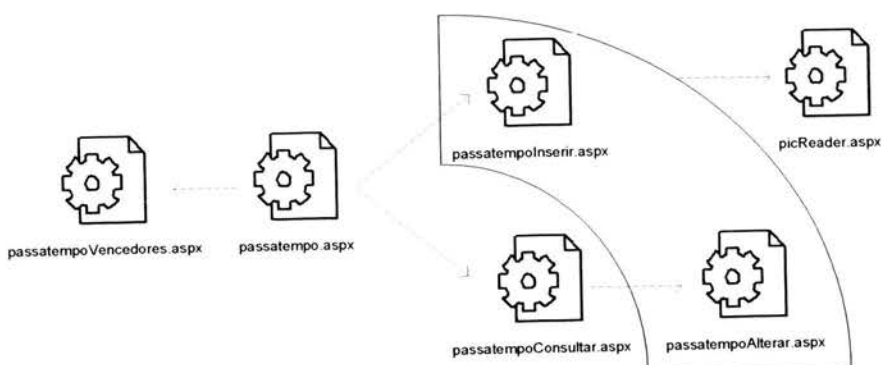


Diagrama 27 - Secção de gestão dos Passatempos

A secção de gestão de passatempos possui mecanismos similares aos existentes na secção de gestão de produtos.

O utilizador também tem acesso a uma página (`passatempoInserir.aspx`) onde deve inserir os dados para um novo passatempo (tipo, nome, imagem, prémios, entre outros), e onde é mostrado um *preview* de como o passatempo apareceria no *site* do Cartão Virtual, antes da inserção na base de dados.

Tal como na secção de gestão de produtos também existe uma página de pesquisa (`passatempoConsultar.aspx`) que permite a escolha do passatempo a alterar (em `passatempoAlterar.aspx`).

Na página principal, onde na secção de gestão de produtos apareciam os produtos com stock crítico, existe agora uma listagem dos passatempos recentemente terminados e aos quais ainda não foram atribuídos prémios. Ao seleccionar um desses passatempos o operador de Backoffice é redireccionado para a página `passatempoVencedor.aspx`. Nesta página todas as participações num passatempo são listadas e é dada a possibilidade de atribuir os prémios disponíveis para esse passatempo.

5 Implementação

Após um levantamento inicial dos requisitos desenvolveu-se um protótipo não-funcional, essencialmente gráfico da Loja Web, Cartão Virtual e Backoffice apresentando-o aos *stakeholders* na tentativa de criar uma base, um consenso, sobre o que era visualmente e estruturalmente pretendido. Partindo dessa base, o desenvolvimento da aplicação foi efectuada com recurso ao lançamento de versões beta da Loja Web, Cartão Virtual e Backoffice (disponibilizadas aos *stakeholders*) para que estas pudessem sofrer melhoramentos e possíveis falhas serem corrigidas mais facilmente, tendo em vista a obtenção de uma versão final estável.

Apesar de diferentes, a Loja Virtual, o Cartão Virtual e o Backoffice partilham tabelas, lógica de negócio e mesmo componentes do interface gráfico e isso reflectiu-se do desenvolvimento da aplicação (ao desenvolver a Loja Web, por exemplo, esteve-se já a desenvolver parte do Cartão Virtual e do Backoffice). Assim, torna-se mais fácil explicar a implementação, à luz do especificado no capítulo anterior, a três níveis interligados: base de dados, lógica de negócio e interface gráfico.

5.1 Camada de base de dados

A camada de base de dados foi desenvolvida sobre o SQL Server 2000 e consistiu:

- na criação de um conjunto de tabelas que suportassem e guardassem os dados;
- em *triggers* que permitiram definir restrições ou acções adicionais ao ser inserida, actualizada ou apagada informação nas tabelas;
- em *stored procedures* que efectuem operações de consulta, inserção, alteração ou eliminação de informação na base de dados ao serem invocados, e através dos quais se efectua a comunicação entre a camada de base de dados e a camada de lógica de negócio.

5.1.1 Tabelas

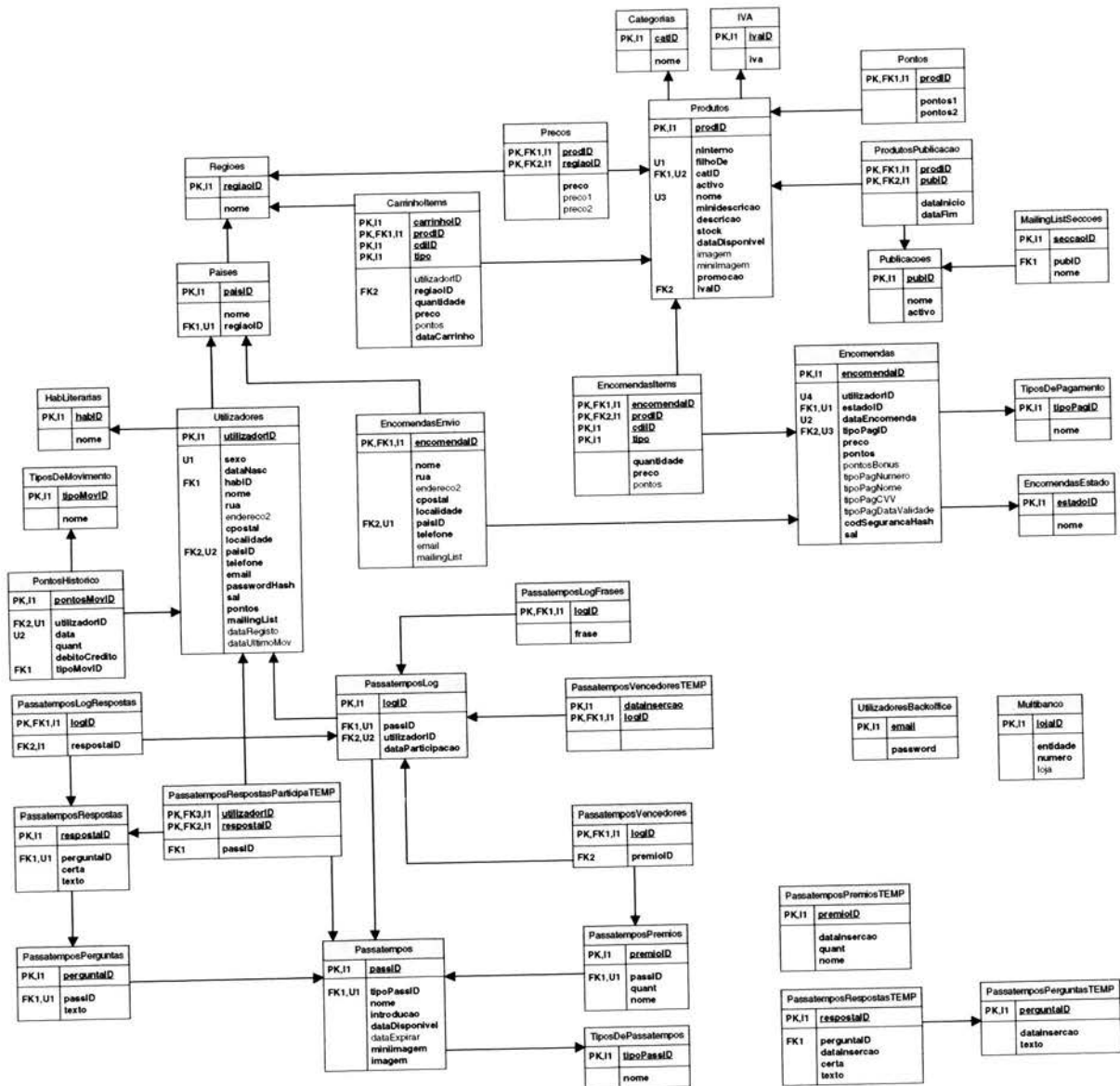


Diagrama 28 - Modelo ER da base de dados

Como se pode verificar a maior parte das 36 tabelas corresponde a uma transposição do que tinha sido especificado anteriormente no modelo de classes do domínio.

Importa, no entanto, clarificar o significado das tabelas e particularidades de alguns dos seus campos.

CarrinhoItens (onde são guardados os produtos colocados no carrinho de compras):

- *carrinhoID*: é o identificador do carrinho e é igual ao identificador de sessão do utilizador;
- *tipo*: tipo de preço (0, preço sem usar pontos; 1, preço com pontos do tipo 1; 2, preço com pontos do tipo 2);
- *cdID*: código adicional para produtos do CDI (0, quando o produto não é desse tipo);

- **utilizadorID**: número do cartão do utilizador registado a que pertence o carrinho (*null*, quando não é de um utilizador registado).

Categorias (as categorias de produtos – Livros ou Vídeo, por exemplo).

Encomendas (onde são guardadas as encomendas efectuadas):

- **utilizadorID**: número do cartão do utilizador registado a que pertence a encomenda (-1, quando a encomenda não é de um utilizador registado);
- **tipoPagDataValidade**: data de validade dos cartões de crédito dados como forma de pagamento (para simplificar a informação é guardada como uma data, juntando-se ao ano e mês fornecidos o dia 20);

EncomendasEnvio (onde são guardados os dados de envio para as encomendas).

EncomendasEstado (os estados em que uma encomenda se pode encontrar – Pendente ou Expedida, por exemplo).

EncomendasItens (onde são guardados os itens – produtos – pertencentes às encomendas):

- **tipo**: tipo de preço (0, preço sem usar pontos; 1, preço com pontos do tipo 1; 2, preço com pontos do tipo 2).

HabLiterárias (as habilitações literárias que um utilizador registado pode possuir – Bacharelato ou Doutoramento, por exemplo).

IVA (os tipos de IVA atribuíveis aos produtos).

MailingListSeccoes² (as secções das publicações que um utilizador pode escolher para receber por e-mail).

Multibanco (os códigos de entidade necessários para formar as referências de pagamento Multibanco).

Países (os países que o utilizador pode escolher – ao se registar, encomendar ou para definir os preços a visualizar na Loja Web).

² Esta tabela não tem utilização prática pois a parte de *mailing* de notícias não faz parte do projecto

PassatempoRespostas (onde são guardadas as respostas de passatempos tipo “perguntas e respostas” e “perguntas e respostas + frase”).

Passatempos (onde são guardados os passatempos):

- *dataExpirar*: 01-01-2050 é a data temporária atribuída para passatempos do tipo “primeiros x a responder/clicar”;

PassatemposLog (onde são guardadas as participações dos utilizadores em passatempos).

PassatemposLogFrases (onde são guardadas as frases dadas pelos utilizadores nos passatempos do tipo “frase”).

PassatemposLogRespostas (onde são guardadas as respostas dadas pelos utilizadores nos passatempos do tipo “perguntas e respostas” ou “perguntas e respostas + frase”).

PassatemposPerguntas (onde são guardadas as perguntas de passatempos do tipo “perguntas e respostas” e “perguntas e respostas + frase”).

PassatemposPerguntasTEMP (onde são guardadas temporariamente – pelo Backoffice – as perguntas de passatempos do tipo “perguntas e respostas” e “perguntas e respostas + frase” antes de serem inseridas na tabela PassatemposPerguntas):

- *dataInsercao*: data em que a informação foi inserida na tabela (serve como garantia adicional que os dados correctos estão a ser inseridos na tabela PassatemposPerguntas).

PassatemposPremios (onde são guardados os prémios dos passatempos).

PassatemposPremiosTEMP (onde são guardados temporariamente – pelo Backoffice – os prémios dos passatempos antes de serem inseridos na tabela PassatemposPremios):

- *dataInsercao*: data em que a informação foi inserida na tabela (serve como garantia adicional que os dados correctos estão a ser inseridos na tabela PassatemposPremios).

PassatemposRespostasParticipaTEMP (onde são guardadas temporariamente as respostas dadas pelos utilizadores a passatempos do tipo “perguntas e respostas” e “perguntas e respostas + frase” antes de serem inseridas na tabela PassatemposLogRespostas).

PassatemposRespostasTEMP (onde são guardadas temporariamente – pelo Backoffice – as respostas de passatempos tipo “perguntas e respostas” e “perguntas e respostas + frase” antes de serem inseridas na tabela PassatemposRespostas):

- *dataInsercao*: data em que a informação foi inserida na tabela (serve como garantia adicional que os dados correctos estão a ser inseridos na tabela *PassatemplosRespostas*).

PassatemplosVencedores (onde são guardados os vencedores dos passatemplos).

PassatemplosVencedoresTEMP (onde são guardados temporariamente – pelo Backoffice – os vencedores dos passatemplos antes de serem inseridas na tabela *PassatemplosVencedores*):

- *dataInsercao*: data em que a informação foi inserida na tabela (serve como garantia adicional que os dados correctos estão a ser inseridos na tabela *PassatemplosVencedores*).

Pontos (onde são guardados os pontos disponíveis para os produtos – os pontos do tipo 1 e tipo 2).

PontosHistorico (onde são registados os ganhos e gastos de pontos pelos utilizadores registados).

Precos (onde são guardados os preços dos produtos – preços sem pontos, usando pontos do tipo 1 e usando pontos do tipo 2).

Produtos (onde são guardados os produtos):

- *filhoDe*: usado para diferenciar produtos-colecção dos restantes (0, produto que representa toda uma colecção; -1, produto não pertencente a uma colecção; outro número, produto pertencente à colecção cujo código é este numero).

ProdutosPublicacao (onde é mantida a relação entre produtos e publicações as quais pertencem):

- *dataInicio*: data na qual o produto foi inserido como pertencente a uma publicação;
- *dataFim*: data na qual outra publicação “assumiu” o produto como seu (ou *dataInicio* adicionada de 50 anos caso o produto pertença apenas a uma publicação).

Publicacoes (as publicações da Global Notícias).

Regiões (as regiões correspondentes a diferentes preços de envio).

TiposDeMovimento (os tipos de movimentos que levam ao ganhar ou perder de pontos – Encomenda criada ou Encomenda cancelada, por exemplo).

TiposDePagamento (os tipos de pagamento aceites para as encomendas – Multibanco ou Cheque, por exemplo).

TiposDePassatempos (os tipos de passatempos existentes – Click ou Perguntas e Respostas, por exemplo).

Utilizadores (onde são guardados os utilizadores registados no Cartão Virtual):

- *dataUltimoMov*: data do último movimento de criação de conta, alteração de dados pessoais, alteração de *password*, criação ou cancelamento de encomendas, ou participação em passatempos por parte do utilizador.

UtilizadoresBackoffice (os utilizadores com acesso à secção de Backoffice).

5.1.2 Triggers

Foram definidos os seguintes *triggers*:

WriteCacheCategorias em Categorias

WriteCacheHabLiterarias em HabLiterarias

WriteCachePaises em Paises

WriteCachePub em Publicacoes

WriteCacheTiposPagamento em TiposDePagamento

WriteCacheTiposPassatempos em TiposDePassatempos

Ao se inserir, apagar ou modificar os dados nestas tabelas são gerados ficheiros que estão a ser monitorizados pelo ASP.NET. Ao verificar que estes ficheiros são alterados os dados existentes na cache são refrescados.

PassatemposLogFrasesCheck em PassatemposLogFrases

Verifica se para a participação inserida o passatempo existe e é do tipo Frase ou Perguntas e Respostas + Frase

PassatemposLogRespostasCheck em PassatemposLogRespostas

Verifica se para a participação inserida o passatempo existe e é do tipo Perguntas e Respostas ou Perguntas e Respostas + Frase

PassatemposPerguntasCheck em *PassatemposPerguntas*

Verifica se para a pergunta inserida o passatempo existe e é do tipo Perguntas e Respostas ou Perguntas e Respostas + Frase

PassatemposRespostasCheck em *PassatemposRespostas*

Verifica se para a resposta inserida o passatempo existe e é do tipo Perguntas e Respostas ou Perguntas e Respostas + Frase

PassatemposVencedoresCheck em *PassatemposVencedores*

Verifica se para o vencedor inserido o prémio atribuído pertence ao passatempo participado

5.1.3 *Stored Procedures*

Existem mais de 200 *stored procedures* que são usados para operações de consulta, inserção ou eliminação de dados existentes nas tabelas. Todos os *stored procedures* são chamados a partir de métodos da parte de lógica de negócio ou por outros *stored procedures*, chamados por esses métodos. Por os métodos terem nomenclaturas semelhantes aos dos *stored procedures* que invocam, mais informações sobre o que faz cada *stored procedure* poderá ser consultada nessa parte do relatório. O único *stored procedure* que não é directamente chamado por um método é *UtilizadorPontosHistorico* sendo, no entanto, invocado por vários outros *stored procedures* quando é necessário fazer uma entrada na tabela de histórico de pontos ganhos ou gastos.

Grande parte das *stored procedures* são simples *queries* à base de dados usando um ou mais parâmetros de *input*, devolvendo os dados individualmente com o uso de parâmetros *output* (quando a lógica de negócio espera, por exemplo, um inteiro, ou pretende encapsular os dados fornecidos num objecto) ou devolvendo toda a *query* sem individualizar a resposta por parâmetros (normalmente quando se pretende mostrar listagens).

Existem alguns casos em que foi necessário recorrer a cursores, nomeadamente para a criação e cancelamento de encomendas, devido à necessidade de subtrair/adicionar aos stocks ou na inserção/alteração de passatempos.

A razão da existência de tabelas temporárias prende-se com o facto de o SQL Server não suportar *arrays*, mas também com o uso de transacções em várias *stored procedures* para garantir que se uma das instruções dentro da transacção falha a integridade dos dados não fica comprometida, sendo executado um *rollback* (para o caso do uso das tabelas temporárias, por exemplo, se após a inserção de perguntas na tabela de perguntas, a inserção de respostas na tabela de respostas falhar, não ficaremos com um passatempo com perguntas sem respostas, tudo é reposto ao estado inicial).

De referir também a necessidade, de em algumas *queries* em que se pretendia devolver os "primeiros x elementos", e uma vez que o SQL Server não suporta na sua sintaxe algo do tipo

```
SELECT TOP @numeroParam ...
```

se ter de criar o texto com o SQL pretendido e executá-lo, como se mostra neste exemplo:

```
DECLARE @vSQL varchar(300)
SET @vSQL = 'SELECT TOP ' + convert(varchar, @numeroParam) + ' ...
EXEC(@vSQL)
```

5.2 Camada de lógica de negócio

A camada de lógica de negócio é composta por duas partes:

- objectos específicos ao contexto da aplicação (Produto ou Passatempo, por exemplo);
- métodos estáticos usados para aceder à base de dados (aos *stored procedures*).

Os objectos são usados para encapsular dados devolvidos pela base de dados ou para guardar temporariamente os dados (recorrendo à funcionalidade Session), na alteração de produtos e passatempos.

Os métodos são invocados nos ficheiros de *codebehind* existentes na camada de interface gráfico. Quando os *stored procedures* chamados são *queries* os dados são encapsulados nos objectos, em *SqlDataReaders* ou *DataSets* (normalmente para listagens), ou em inteiros, *strings* ou booleanos.

Objectos e métodos foram implementados em ficheiros C# num mesmo projecto do VisualStudio.NET para que com o DLL gerado, os projectos referentes à Loja Web, Cartão Virtual e Backoffice pudessem aceder à base de dados ou criar objectos conforme necessário (deste projecto farão também parte outros ficheiros que constam da directoria Misc e cujos métodos são partilhados entre páginas do interface gráfico).

Em seguida faz-se uma breve descrição dos objectos e métodos da camada de lógica de negócio.

5.2.1 Publicacao



Ilustração 7 - Objecto Publicacao (Publicacao.cs)

O objecto Publicacao define uma publicação do grupo Global Notícias e possui um código (o seu código na base de dados) e um nome.

O método GetCachePublicacoes introduz num *DataSet* todas as publicações activas (para que o *DataSet* possa ser guardado na cache do ASP.NET e evitar acessos desnecessários à base de dados).

5.2.2 Categoria



Ilustração 8 - Objecto Categoria (Categoria.cs)

O objecto Categoria define uma categoria de produto (Livros ou Vídeo, por exemplo) e possui um código identificativo e um nome.

O método GetCacheCategorias introduz num *DataSet* (a guardar na cache do ASP.NET) todas as categorias existentes na base de dados para evitar acessos desnecessários.

5.2.3 Pontos



Ilustração 9 - Objecto Pontos (Pontos.cs)

O objecto Pontos expressa os dois tipos de pontos que cada produto pode ter e que podem ser utilizados pelos utilizadores registados para comprar na Loja Web com descontos. Cada objecto é composto de dois inteiros, um para cada tipo de pontos.

5.2.4 Preço

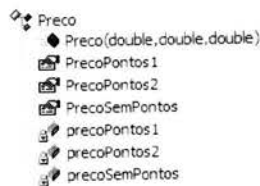


Ilustração 10 - Objecto Preço (Preco.cs)

O objecto Preço encapsula os diferentes tipos de preços que cada produto pode ter: o preço se for comprado sem usar pontos, o preço no caso de se usar pontos do tipo 1 e o preço no caso de se utilizar pontos do tipo 2. Cada uma destas variáveis é do tipo *double*.

5.2.5 Regiao



Ilustração 11 - Objecto Regiao (Regiao.cs)

O objecto Regiao representa uma região de envio para os produtos da Loja Web, e é composto por o código da região de envio e o respectivo preço associado a essa região.

O método GetRegiao devolve o nome da região de envio actualmente escolhida, para mais fácil identificação no cabeçalho das páginas da Loja Web.

5.2.6 Iva



Ilustração 12 - Objecto Iva (Iva.cs)

O objecto Iva representa o IVA que um determinado produto tem associado e possui um inteiro (o valor da percentagem de IVA) e um *char* (o código de IVA existente na base de dados).

O método GetTiposDeIva devolve os diferentes tipos de IVA existentes na base de dados.

5.2.7 Produto

O ficheiro Produto.cs é composto pelo objecto Produto e pelos métodos de acesso à base de dados mais directamente relacionados com esta temática.



Ilustração 13 - Objecto Produto (Produto.cs)

Um objecto Produto representa um produto e, como tal, possui um booleano indicativo de que o produto está activo, a data em que o produto se tornou disponível e apareceu no *site* da Loja Web, um código identificativo, o seu número interno (código SAP), o nome e descrições do produto, as imagens associadas ao produto, a publicação e categoria a que pertence, o seu stock, o IVA, um inteiro indicando se o produto pertence a uma colecção (e qual), os pontos, um *array* de objectos Regiao para obter os vários preços do produto, um booleano indicando se o produto deve aparecer na página principal da Loja Web e outro booleano indicando se o produto é o “produto-pai” de uma colecção cujos “filhos” ainda não estão todos disponíveis (impossibilitando a compra da totalidade dessa colecção).

Os vários construtores deste objecto são utilizados ao ver o detalhe do produto na Loja Web (na versão mais pequena), ou para a visualização dos *previews* na parte de inserção/alteração dos produtos no Backoffice (nas versões maiores).


```

◆ Altera(int,int,Publicacao,Categoria,string,string,string,int,DateTime,int,Iva,Pontos,Regiao[],byte[],byte[],bool,bool)
◆ AlteraStock(int,int)
◆ ConsultaByCat(int,int,string,int)
◆ ConsultaByPub(int,int,string,int)
◆ ConsultaByPubCat(int,int,int,string,int)
◆ ConsultaTodos(int,string,int)
◆ Elimina(int)
◆ GetByCat(int,int,int,int)
◆ GetByPub(int,int,int,int)
◆ GetCriticos(int,int,int)
◆ GetDetalle(int,int)
◆ GetDisponivel(int)
◆ GetFilhos(int,int,int,int)
◆ GetImagem(int)
◆ GetMiniImagem(int)
◆ GetNByCat(int)
◆ GetNByPub(int)
◆ GetNConsultaByCat(int,string)
◆ GetNConsultaByPub(int,string)
◆ GetNConsultaByPubCat(int,int,string)
◆ GetNConsultaTodos(string)
◆ GetNCriticos(int)
◆ GetNFilhos(int)
◆ GetNome(int)
◆ GetNProcuraByCat(int,string)
◆ GetNProcuraByPub(int,string)
◆ GetNProcuraTodos(string)
◆ GetPais(int)
◆ GetPontos(int,int)
◆ GetPromo()
◆ GetPromoRandom(int)
◆ GetStock(int)
◆ GetSuperDetalle(int)
◆ Insere(int,Publicacao,Categoria,string,string,string,int,DateTime,int,Iva,Pontos,Regiao[],byte[],byte[],bool,bool)
◆ ProcuraByCat(int,string,int,int,int)
◆ ProcuraByPub(int,string,int,int,int)
◆ ProcuraTodos(string,int,int,int)
◆ PromoReset(int)
◆ PromoSet(int)

```

Ilustração 14 - Restantes métodos de Produto.cs

Alguns dos métodos presentes neste ficheiro são exclusivamente utilizados na secção de Backoffice:

- Através de *Insere*, *Altera* e *Elimina*, é possível inserir, alterar ou eliminar produtos.
- Para procurar produtos com o intuito de os poder editar, são usados *ConsultaByCat*, *GetNConsultaByCat*; *ConsultaByPub*, *GetNConsultaByPub*; *ConsultaByPubCat*, *GetNConsultaByPubCat*; *ConsultaTodos*, *GetNConsultaTodos*. A escolha do método a usar depende da escolha de parameterizar por categoria, publicação, ambos, ou nenhum (procurando em todos os produtos). Assim, os métodos cujo nome se inicia por “GetN” devolvem o número total de produtos encontrados na pesquisa (usado para calcular o número de páginas e quais os valores a mostrar em cada página).
- *PromoSet* e *PromoReset* são usados para definir se um produto deve ou não aparecer na página principal do *site* da Loja Web.
- *GetCriticos* e *GetNCriticos* são utilizados para ver uma listagem dos produtos cujo stock se encontra abaixo de um determinado nível (mais uma vez *GetNCriticos* devolve o tamanho da listagem e *GetCriticos* a listagem em si). Dentro da listagem, o utilizador pode usar *AlterarStock* para modificar o stock de um produto para um novo valor.
- O método *GetPais* é chamado na parte de inserção/alteração de produto para popular uma *dropdownlist* onde se encontram os produtos que podem ser colecções (os que já o são e os que não pertencem a nenhuma colecção mas que se tornariam num produto-colecção se forem escolhidos).

- O método `GetDisponivel` (usado para visualização no *preview*) permite verificar se uma colecção tem todos os seus elementos disponíveis.
- `GetSuperDetalhe` devolve todos os pormenores referentes a um produto para poder popular os campos mostrados quando se deseja alterar um produto.

Outros métodos são chamados por páginas do *site* da Loja Web:

- As listagens de produtos existentes são dadas pelos métodos `GetByCat`, `GetNByCat`; `GetByPub`, `GetNByPub`; `ProcuraByCat`, `GetNProcuraByCat`; `ProcuraByPub`, `GetNProcuraByPub`; `ProcuraTodos`, `GetNProcuraTodos`. O método a usar depende do tipo de listagem pretendida, se de uma determinada categoria ou publicação, se proveniente de uma procura por parte do utilizador (pesquisando uma expressão por todos os produtos ou definindo apenas a pesquisa dentro de uma publicação ou categoria).
- Ao se escolher ver os produtos de uma determinada colecção, os métodos `GetFilhos` e `GetNFilhos` são chamados para formar uma listagem, sendo o mecanismo semelhante ao descrito anteriormente.
- O método `GetDetalhe` permite ver alguns detalhes do produto (nome, preço, pontos, descrição mais detalhada e uma imagem alusiva), antes de se adicionar um produto ao carrinho.
- `GetPontos` devolve os pontos de um certo tipo, associados a um produto, e é usado para verificar se um utilizador tem pontos suficientes para adicionar um produto ao carrinho.
- O método `GetPromo` permite ver os produtos que figuram na secção “em destaque”, existente na página principal.

O método `GetPromoRandom` escolhe aleatoriamente um dos produtos que se encontram “em destaque” na Loja Web para ser mostrado no *site* do Cartão Virtual.

`GetName` devolve o nome de um produto e `GetImagem` e `GetMiniImagem` devolvem os *arrays* de bytes que compõem as imagens alusivas ao produto

5.2.8 Carrinho

A funcionalidade “carrinho de compras”, habitual nos *sites* de comércio electrónico, tem os seus dados guardados na base de dados. O ficheiro `Carrinho.cs` contém os métodos relacionados com esta funcionalidade.

```

Carrinho
  ● Actualiza(string,int,int,int,int)
  ● ActualizaRegiao(string,int)
  ● ActualizaRegistado(string,int,int,int,int,int)
  ● Adiciona(string,int,int,int)
  ● AdicionaRegistado(string,int,int,int,int,int,int)
  ● CarrinhoEsvazia(string)
  ● ContaItems(string)
  ● GetCarrinhoID()
  ● GetItems(string)
  ● MarcaComoRegistado(string,int)
  ● RemoveItem(string,int,int,int)

```

Ilustração 15 - Métodos de `Carrinho.cs`

Assim, `GetCarrinhoID` devolve o identificador do carrinho, guardado numa *cookie* ou, se a *cookie* ainda não existir, cria-a e atribui-lhe um novo valor (o identificador da sessão).

Os itens são adicionados ao carrinho usando `Adiciona` ou `AdicionaRegistado`, as suas quantidades modificadas por `Actualiza` ou `ActualizaRegistado` (quer se tratem de carrinhos pertencentes a utilizadores registados ou não-registados) e são eliminados por `RemoveItem`. Em ambos os casos os argumentos utilizados incluem o identificador do carrinho, o código do produto e o tipo de preço do item, podendo ser necessário fornecer o identificador da região de envio, quantidades e pontos a gastar, bem como o código do utilizador. A visualização dos itens existentes no carrinho é executada com recurso a `GetItens`.

Cada vez que o utilizador muda de região, os preços no carrinho são alterados com uma chamada a `ActualizaRegiao`. De igual modo, se forem adicionados itens ao carrinho e posteriormente o utilizador se autenticar, `MarcaComoRegistado` é chamado para associar o carrinho a um utilizador.

O método `ContaItens` é usado várias vezes para verificar se o carrinho está ou não vazio.

Quando o utilizador termina a sessão é invocado `CarrinhoEsvazia` que, como o nome indica, apaga o conteúdo do carrinho.

5.2.9 Pais



Ilustração 16 - Objecto Pais (Pais.cs)

O objecto `Pais` representa um país e é constituído pelo nome do país e um código identificativo.

O método `GetCachePaises` coloca num *DataSet* (posteriormente guardado na cache do ASP.NET) todas as países existentes na base de dados para evitar acessos desnecessários.

5.2.10 DadosEnvio



Ilustração 17 - Objecto DadosEnvio (DadosEnvio.cs)

O objecto `DadosEnvio` pretende representar o conjunto de dados comuns exigidos quando um utilizador se regista no Cartão Virtual, sendo estes constituídos pelo nome, morada (rua, o campo adicional de endereço, o código postal, a localidade e país), telefone, e-mail e um booleano que indica se o utilizador deseja receber informação promocional adicional por parte da Global Notícias.

5.2.11 HabLiterarias



Ilustração 18 - Objecto HabLiterarias (HabLiterarias.cs)

O objecto `HabLiterarias` refere-se às habilitações literárias que um utilizador possui e é composto pelo nome da habilitação e um código identificativo da habilitação.

O método `GetCacheHabLiterarias` introduz num *DataSet* (a guardar na cache do ASP.NET) as habilitações literárias existentes para evitar acessos desnecessários à base de dados.

5.2.12 Utilizador



Ilustração 19 - Objecto Utilizador e restantes métodos de Utilizador.cs

Um objecto Utilizador representa um utilizador registado no Cartão Virtual e é constituído pela informação que consta num objecto DadosEnvio, as habilitações literárias, o sexo e a data de nascimento.

Os métodos contidos em Utilizador.cs permitem adicionar um novo utilizador à base de dados com os dados fornecidos aquando do registo (incluindo a *password* já sobre forma de *hash* e o “sal” associado – ambos obtidos graças a métodos existentes em Misc.Security). Permitem igualmente alterar os dados existentes do registo graças a AlteraDados, sendo que estes dados não incluem a alteração da *password*, processada pelo método AlteraPassword (recorrendo, mais uma vez, a Misc.Security). Tanto ao registar-se como ao alterar os dados na Conta Pessoal, VerificaEmail é chamado para garantir que o e-mail fornecido não existe já na base de dados. De igual modo, VerificaEmailDataNasc verifica se existe na base de dados um utilizador registado cujo e-mail e data de nascimento correspondem aos fornecidos na parte de recuperação de *password* da Conta Pessoal.

A autenticação ocorre graças a LoginEmail ou LoginCartao, pois o utilizador pode autenticar-se fornecendo a *password* e o seu e-mail ou o seu número de Cartão Virtual. Nestes métodos, a *password* sob forma de *hash* é, juntamente com o “sal”, retornada pela base de dados e, com os dados fornecidos e o “sal” proveniente da base de dados, é composta (com a ajuda de métodos presentes em Misc.Security) uma nova *password* sob forma de *hash* para ser confrontada com a guardada na base de dados. LoginBackoffice verifica se o e-mail e *password* fornecidos correspondem a campos da tabela UtilizadoresBackoffice.

O preenchimento de campos existentes na página de alteração de dados, e na página de fornecimento de dados de envio ao se finalizar uma encomenda, é garantida pela informação devolvida pelo método Get.

Os métodos `GetNome` e `GetSexo` são usados para visualizar uma mensagem de boas-vindas no cabeçalho da Loja Web e Cartão Virtual. `GetPontos` e `GetPontosTemp` são igualmente utilizados para mostrar os pontos que um utilizador possui e que irá gastar com os artigos existentes no carrinho de compras, bem como para garantir que existem pontos suficientes para adicionar produtos ao carrinho e colocar uma encomenda.

`GetRegiaoID` é usado para retornar o código da região de envio a que o utilizador pertence para que, na Loja Web, esta possa ser visualizada de acordo com a origem do utilizador, sempre que este se autentifica, ou aquando do registo ou alteração de dados.

O método `GetEmail` devolve o endereço electrónico do utilizador registado para lhe poder enviar um e-mail com a *password* recuperada. Também é utilizado para, quando se alteram os dados pessoais e o mesmo e-mail é fornecido, assegurar que a invocação do método `VerificaEmail` não ocorra.

O conjunto composto por `GetHistoricoPassatemp`, `GetNHistoricoPassatemp` e `GetHistoricoPontos`, `GetNHistoricoPontos` são responsáveis pelas listagens dos passatemp participados e pontos ganhos/gastos que são apresentadas secção da Conta Pessoal do Cartão Virtual.

5.2.13 Estado



Ilustração 20 - Objecto Estado (Estado.cs)

O objecto `Estado` é a representação do estado em que se encontra uma encomenda (Pendente ou Paga, por exemplo) e contém um código identificativo e o nome do estado.

5.2.14 TipoPagamento



Ilustração 21 - Objecto TipoPagamento (TipoPagamento.cs)

O objecto `TipoPagamento` refere-se aos tipos de pagamentos possíveis de usar para as compras na Loja Web (Cheque ou Multibanco, por exemplo) e os dados referentes ao pagamento introduzidos pelos utilizadores para efectuarem essas mesmas compras.

Os argumentos incluem o nome e código identificativo de um tipo de pagamento, utilizados na versão menor do construtor.

Relativamente aos dados fornecidos pelos utilizadores, estes são encapsulados nos outros construtores, quer se trate de informação referente a um cheque (apenas é necessário saber o número) ou referente a um cartão de crédito (além de o número, é exigido o nome escrito no cartão, a data de validade e o CVV). A outra forma de pagamento, o pagamento via Multibanco, não necessita de dados adicionais por parte do utilizador e não possui nenhum construtor específico.

O método `GetCacheTiposDePassatempo` introduz num *DataSet* (a guardar na cache do ASP.NET) os tipos de passatempos existentes para evitar acessos desnecessários à base de dados.

5.2.15 Encomenda



Ilustração 22 - Objecto Encomenda e restantes métodos de Encomenda.cs

Uma Encomenda é constituída por um conjunto de itens (os produtos guardados num *DataSet* para aparecerem mais facilmente em listagens), os dados referentes à morada de envio da encomenda, o estado em que a encomenda se encontra, os dados relativos ao tipo de pagamento utilizado, a data na qual a encomenda foi colocada e o utilizador que a efectuou.

O método `GetRegiaoID` determina qual o identificador de região de um determinado país e é usado para confrontar a escolha de país de envio (e da sua região) com a região de envio actualmente escolhida, sendo que se forem diferentes um aviso é emitido.

Os métodos cujo nome se inicia por “FinalizaEncomenda” contêm os dados necessários para a colocação de uma encomenda e são invocados com esse propósito, sendo que existem *stored procedures* diferenciados para diferentes tipos de pagamento e para utilizadores registados ou não-registados. Se o pagamento escolhido para a encomenda tiver sido Multibanco, um método existente em `Misc.Security` invocará `GetPagamentoEntidade` e `GetPagamentoLoja` que devolvem valores necessários para a geração das referências de pagamento.

Usando `GetEncomendasFrom` (e recorrendo a `GetNEncomendasFrom`) é possível ver todas as encomendas de um determinado utilizador (para listar no seu histórico de encomendas). Os detalhes das encomendas (os itens que contém, mas também o seu estado ou a data de encomenda, por exemplo) são obtidos invocando os métodos `GetEncomendaSuperDetalhe` e `GetEncomendaEstado`.

O utilizador que deseja ver o estado da sua encomenda (utilizando o código que lhe foi fornecido aquando da colocação desta) tem de fornecer dados específicos cuja validade será confirmada pela invocação de `LoginEstado`. O acesso à página do estado da encomenda está ainda condicionado a uma segunda verificação na qual é usado o método `GetSal`.

`GetBackEncomendasNcli`, `GetBackEncomendasNenc` e `GetBackEncomendasTempo` são usados na zona de gestão de encomendas do Backoffice para visualizar as listagens das encomendas, variando o método chamado consoante se pretenda ver as encomendas de um cliente (num espaço de tempo), uma determinada encomenda cujo código se conhece, ou todas as encomendas num espaço de tempo e num estado específico.

O método `CheckTemCDI` verifica se uma encomenda possui produtos da categoria CDI para determinar se um e-mail deverá ser enviado para o CDI para este processar a encomenda.

A mudança de estado de uma encomenda acontece com `SetCancelada`, `SetPendente`, `SetPaga` ou `SetExpedida`, consoante esta é cancelada, passa a pendente ou é dada como paga ou expedida.

5.2.16 Estatística



Ilustração 23 - Métodos de Estatistica.cs

O ficheiro estatistica.cs contém os métodos utilizados na secção de estatísticas do Backoffice. Assim, e normalmente limitados a um espaço de tempo definido, as estatísticas referentes a registos no Cartão Virtual são dadas pelos métodos iniciados por “GetCartao”, as referentes a participações nos passatempos são dadas pelos métodos cujo nome começa por “GetPassatempos”, estatísticas sobre as encomendas recebem os dados pelos métodos iniciados por “GetEncomendas” e relativamente aos pedidos de informações adicionais pela Global Notícias os métodos usados começam por “GetMailing”. Em todos os casos a informação pode ser obtida por faixas etárias, habilitações literárias, sexo, país ou localidade portuguesa. Nas estatísticas relativas à Loja Web, também se podem ver as encomendas por publicação (GetEncomendasTotalPub), os melhores clientes (GetEncomendasTotalClientes) e os produtos mais vendidos (GetProdutosTotalMensal). As estatísticas possíveis relativas a passatempos incluem os passatempos mais participados (GetPassatemposTotais) e os utilizadores que mais participam (GetPassatemposTotalParticipa).

Adicionalmente, na secção de estatísticas é incluída a possibilidade de ver os e-mails dos utilizadores que pediram para receber informação adicional da Global Notícias. A listagem de endereços electrónicos pode ser obtida através dos métodos GetMailingTodos, GetMailingRegistados ou GetMailingNaoRegistados, consoante se deseje todos os e-mails ou apenas os de utilizadores registados ou não-registados.

5.2.17 MailingList

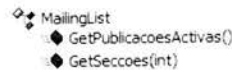


Ilustração 24 - Métodos de MailingList.cs

Os métodos existentes neste ficheiro são utilizados pela página do Cartão Virtual para revelar as publicações que enviam notícias por e-mail e as secções das publicações que estão disponíveis para escolha.

5.2.18 Passatempos



Ilustração 25 - Objecto Passatempo (Passatempo.cs)

Um objecto Passatempo é de um determinado tipo, tem associado uma imagem (e uma versão menor desta) e possui uma data na qual estará disponível e outra na qual terminará. Quando os passatempos forem de um tipo que só termina após um determinado número de participações correctas (correspondente ao número de prémios existentes), é temporariamente atribuída a data de expiração de 1 de Janeiro de 2005 (sendo esta corrigida após terminado o passatempo). Estão disponíveis cinco construtores, sendo as versões menores usadas no *site* do Cartão Virtual e as maiores no Backoffice. Relacionado com o Backoffice estão PremiosArr, PerguntasArr e RespostasArr, objectos do tipo *ArrayList* onde são guardados os prémios, perguntas e respostas inseridos ou alterados no Backoffice.

◆ AlteraClick(int, DateTime, Passatempo)	◆ GetParticipantesFrase(int)
◆ AlteraClickPrimeiros(int, DateTime, Passatempo)	◆ GetParticipantesPergResp(int)
◆ AlteraFrase(int, DateTime, Passatempo)	◆ GetParticipantesPergRespFrase(int)
◆ AlteraPergResp(int, DateTime, Passatempo)	◆ GetPerguntas(int)
◆ AlteraPergRespFrase(int, DateTime, Passatempo)	◆ GetPremios(int)
◆ AlteraPergRespPrimeiros(int, DateTime, Passatempo)	◆ GetRandom(int)
◆ AtribuiPremio(DateTime)	◆ GetRespostas(int)
◆ CheckJaParticipou(int, int)	◆ GetRespostasCertas(int)
◆ CheckJaTerminou(int)	◆ GetRespostasDadas(int)
◆ CheckPremiosDisp(int)	◆ GetRespostasDaPergunta(int)
◆ CheckPrimeiros(int)	◆ GetSemVencedores(int, int)
◆ CheckPrimeirosDisp(int)	◆ GetUltimos()
◆ CheckVencedorExiste(int)	◆ GetVencedores(int)
◆ ConsultaByPass(int, int, string, int)	◆ InserirClick(DateTime, Passatempo)
◆ ConsultaTodos(int, string, int)	◆ InserirClickPrimeiros(DateTime, Passatempo)
◆ Elimina(int)	◆ InserirFrase(DateTime, Passatempo)
◆ GetActivos()	◆ InserirPergResp(DateTime, Passatempo)
◆ GetDetalhe(int)	◆ InserirPergRespFrase(DateTime, Passatempo)
◆ GetFraseDada(int)	◆ InserirPergRespPrimeiros(DateTime, Passatempo)
◆ GetFrasesVencedoras(int)	◆ InserirPergunta(DateTime, Pergunta)
◆ GetImagem(int)	◆ InserirPremio(DateTime, Premio)
◆ GetIntroducao(int)	◆ InserirResposta(DateTime, Resposta)
◆ GetMiniImagem(int)	◆ InserirVencedores(int, int, DateTime)
◆ GetVConsultaByPass(int, string)	◆ ParticipaClick(int, int)
◆ GetVConsultaTodos(string)	◆ ParticipaFrase(int, int, string)
◆ GetVNome(int)	◆ ParticipaPergResp(int, int)
◆ GetVSemVencedores()	◆ ParticipaPergRespFrase(int, int, string)
◆ GetParticipantesClick(int)	◆ ParticipaRespostaInserir(int, int, int)
◆ GetParticipantesFrase(int)	◆ SetDataPrimeiros(int)

Ilustração 26 - Restantes métodos de Passatempo.cs

De entre os métodos relativos a passatempos existem alguns relativos à sua inserção ou alteração através do Backoffice. A escolha de qual método utilizar está dependente do tipo de passatempo: participação por *click* num botão (durante um determinado tempo ou terminando após um número de participações), escrita de frase, resposta a perguntas de escolha múltipla (durante um determinado tempo ou terminando após um número de participações correctas), ou escrita de frase e resposta a perguntas de escolha múltipla. Durante a inserção ou alteração de passatempos também são utilizados os métodos *InserirPremio* e, possivelmente, *InserirPergunta* e *InserirResposta* (que inserem os dados em tabelas temporárias sendo estes posteriormente passados às tabelas definitivas quando se chama, logo de seguida, o método de inserção ou alteração do passatempo). O uso de tabelas temporárias permite uma maior fiabilidade da aplicação, sendo facilmente efectuado o *rollback* em caso de erro inesperado. Para se alterar um passatempo é preciso procurá-lo e escolhe-lo de uma listagem. Essa listagem é obtida por *ConsultaTodos* ou *ConsultaByPass* (caso se limite a pesquisa a um tipo de passatempo). Na página de alteração de passatempos, a eliminação de passatempos ainda não iniciados é executada por *Elimina*. Ainda no Backoffice é disponibilizada uma listagem, dos passatempos terminados e ainda sem vencedores definidos, graças ao método *GetSemVencedores*. Na página de atribuição dos vencedores, e para visualizar a listagem de participações dos utilizadores num passatempo, é chamado, consoante o tipo de passatempo, um dos métodos iniciados por “*GetParticipantes*”. Para a atribuição dos prémios propriamente dita, os vencedores são guardados numa tabela temporária graças ao método *InserirVencedores* para depois serem passados à tabela definitiva por *AtribuiPremio*.

Para se ver no *site* do Cartão Virtual os passatempos activos, e os que mais recentemente terminaram, são usados os métodos *GetActivos* e *GetUltimos*. O método *GetRandom* devolve um dos passatempos activos para ser mostrado na página principal do Cartão Virtual.

A visualização das várias características de um passatempo acontece usando métodos como *GetNome*, *GetImagem*, *GetMiniImagem*, *GetIntrodução*, *GetPremios*, *GetPerguntas* ou *GetRespostasDaPergunta*. Contrariamente ao método *GetRespostas* (usado pelo Backoffice

para devolver todas as respostas do passatempo para edição deste), `GetRespostasDaPergunta` devolve o conjunto de respostas associadas a uma das perguntas que pertencem ao passatempo.

Antes de uma participação decorrem várias verificações. O método `CheckJaParticipou` verifica se o utilizador autenticado já participou naquele passatempo (apenas é permitida uma participação por passatempo). `CheckPrimeiros` verifica se o passatempo é limitado por número de participações (se tal acontecer os métodos `CheckPremiosDisp` e `CheckPrimeirosDisp` são chamados para garantir que o número de participações correctas não ultrapassa o número de prémios disponíveis). Nos passatempos limitados por tempo é necessário garantir que o passatempo continua activo (`CheckJaTerminou`).

A participação nos passatempos é efectuada por `ParticipaClick`, `ParticipaFrase`, `ParticipaPergResp` ou `ParticipaPergRespFrase`. Anteriormente, e para os passatempos que envolvem respostas, é chamado o método `ParticipaRespostaInsere` que escreve as respostas dadas numa tabela temporária. Após a participação, e caso se trate de um passatempo que termina após um número de participações, é invocado `SetDataPrimeiros` que guarda na base de dados a data na qual o passatempo terminou.

Para verificar se os passatempos já terminados têm vencedores atribuídos, passíveis de serem revelados no *site* do Cartão Virtual por `GetVencedores`, é invocado `CheckVencedorExiste`. Se os vencedores estiverem atribuídos e o passatempo envolver frases, é possível ver as frases vencedoras graças a `GetFrasesVencedoras`. Mesmo sem vencedores atribuídos, para os passatempos relativos a perguntas de escolha múltipla, são reveladas as respostas correctas ao passatempo terminado usando `GetRespostasCertas`. Os utilizadores registados podem também visualizar estas informações no detalhe das suas participações: `GetDetalhe` fornece alguma da informação relativa ao passatempo; os vencedores, respostas correctas e frases vencedoras são mostrados; possivelmente também são mostradas a frase (`GetFraseDada`) ou respostas dadas (`GetRespostasDadas`) pelo utilizador ao passatempo.

5.2.19 Pergunta

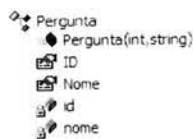


Ilustração 27 - Objecto Pergunta (Pergunta.cs)

Uma Pergunta é composta pelo texto da pergunta e por um código identificador.

5.2.20 Premio



Ilustração 28 - Objecto Premio (Premio.cs)

O objecto `Premio` é composto por um código que o identifica, pelo nome do prémio propriamente dito e pela quantidade de prémios deste tipo que se encontra disponível para serem atribuídos aos vencedores dos passatempos.

5.2.21 Resposta



Ilustração 29 - Objecto Resposta (Resposta.cs)

Uma `Resposta` possui um código identificativo, o texto da resposta e um booleano que indica se é uma resposta correcta. Uma `Resposta` está associada a uma `Pergunta`.

5.2.22 TipoPassatempo



Ilustração 30 - Objecto TipoPassatempo (TipoPassatempo.cs)

Este objecto pretende representar um tipo de passatempo existente e, como tal, é apenas constituído pelo nome do tipo de passatempo e pelo seu código.

5.2.23 Misc.GUI



Ilustração 31 - Métodos de GUI.cs

O ficheiro GUI inclui métodos que introduzem *Javascript* nas páginas do interface gráfico, alterando determinados comportamentos predefinidos.

Os métodos `setDefaultButton` ajudam a associar o evento de pressionar a tecla *Enter* num certo controlo (uma caixa de texto, por exemplo) ao “click” num determinado botão. O método `setFocus` coloca o foco num controlo. O método `confirm` é usado na parte de Backoffice e mostra uma *message box* de confirmação quando se carrega num botão que significa a mudança de estado de uma encomenda.

5.2.24 Misc.MailingList

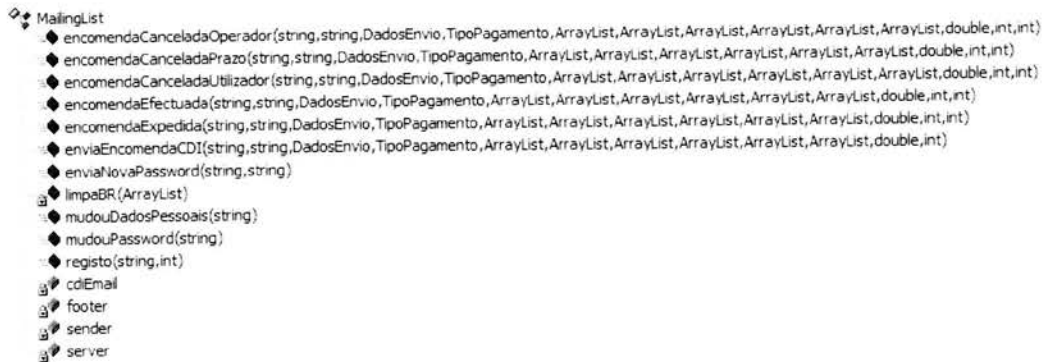


Ilustração 32 - Métodos de MailingList.cs

Os métodos deste ficheiro são invocados sempre que se pretende enviar um e-mail. Existe um rodapé que é incluído em todos os e-mails, e também estão predefinidos o servidor de mail, o endereço de quem envia as mensagens (“sender”) e o e-mail para o qual se enviará o resumo das encomendas quando estas incluem produtos do CDI. Um e-mail é enviado quando um utilizador se regista no Cartão Virtual, quando este muda os dados pessoais ou a sua *password*, ou quando este pede uma nova *password* temporária. Quando uma encomenda é efectuada ou muda para certos estados (cancelada, prazo de pagamento ultrapassado, paga ou expedida) também é enviado um e-mail que inclui o detalhe da encomenda (os dados de envio e de pagamento e os vários itens da encomenda divididos, coluna a coluna, em *ArrayLists*). O detalhe da encomenda também é incluído no e-mail enviado para o CDI quando uma encomenda que necessita de ser processada possui produtos dessa categoria.

5.2.25 Misc.Picture

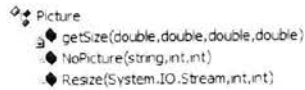


Ilustração 33 - Métodos de Picture.cs

Os métodos de Picture.cs são invocados no Backoffice para obter as imagens que ficarão associadas a produtos ou passatempos.

O método NoPicture é usado quando não se possui nenhuma imagem em particular para o produto ou passatempo. A localização de um ficheiro de imagem existente no servidor (que indica que não existir imagem para o produto) é fornecida, assim como as dimensões máximas pretendidas para a imagem ser posteriormente guardada na base de dados. Com as dimensões dadas e com as dimensões da imagem do ficheiro, e para impedir que a imagem fique deformada, é calculado o tamanho óptimo através do método getSize. É criado um objecto *Bitmap* com as dimensões desejadas onde o ficheiro é colocado. O *Bitmap* é guardado num *stream* (no formato *Jpeg*) o que permite copiar o seu conteúdo para um *array* de bytes, passível de ser guardado na base de dados.

De forma semelhante funciona Resize. A diferença entre este método e o anterior reside no facto de neste existir um ficheiro que se deseja associar ao produto ou passatempo que, encontrando-se no computador do cliente, é lido para um *stream* (fornecido nos argumentos).

5.2.26 Misc.Security

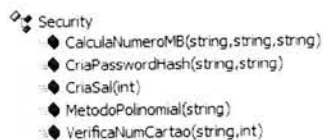


Ilustração 34 - Métodos de Security.cs

No ficheiro Security.cs existem métodos que estão relacionados com a segurança.

CriaSal devolve um número aleatório criptograficamente forte graças ao *RNGCryptoServiceProvider*. Este número é chamado de “sal” e é utilizado com o propósito de ser guardado na base de dados juntamente com as *passwords* para evitar ataques do tipo “SQL Injection”. CriaPasswordHash fornece um *hash* correspondente a uma *password* graças a um método de *FormsAuthentication*. Ambos os métodos são usados aquando do registo de novos utilizadores, pedido de nova *password* temporária e quando uma encomenda é colocada (para obter e guardar o código necessário para os utilizadores não-registados poderem ver o estado da encomenda). CriaPasswordHash é também utilizada para confrontar as *passwords* fornecidas (para a autenticação de utilizadores e para ver o estado de encomendas) com as guardadas na base de dados.

O método CalculaNumeroMB computa a referência de pagamento que o cliente necessita para efectuar um pagamento por Multibanco da sua encomenda, chamando a certo ponto o MetodoPolinomial para obter a referência pretendida.

VerificaNumCartao possibilita verificar se o número de cartão de crédito disponibilizado como forma de pagamento está bem formado, e será válido segundo as normas das entidades emissoras dos cartões.

5.3 Camada de interface gráfico

A implementação da camada de interface gráfico envolveu o desenvolvimento da parte de design dos *sites* e a produção de algum código de suporte às páginas nos ficheiros de *codebehind*.

Para o design das páginas foram utilizados *Cascading Style Sheets* (CSS) que permitiram manter certos parâmetros, como os tipos de letras, cores ou posicionamento de certos controlos, num local centralizado. Todas as páginas da Loja Web, Cartão Virtual e Backoffice foram produzidas tendo em vista a correcta visualização num monitor com resolução de 800x600 e possuem *Web User Controls* que representam o cabeçalho e rodapé, sendo que a Loja Web também usa um controlo para um menu lateral. O conteúdo de cada página é colocado entre o cabeçalho e o rodapé, mas a largura do espaço disponibilizado depende da existência, ou não, do menu lateral. Este permite às páginas da Loja Web manterem sempre uma altura mínima mas retira algum do espaço destinado ao conteúdo da página. Por outro lado, as páginas do Cartão Virtual e Backoffice têm mais espaço para o conteúdo mas necessitam de usar tabelas para formatarem as páginas e lhes darem uma altura mínima.



Ilustração 35 – Páginas principais da Loja Web, Cartão Virtual e Backoffice

Nos ficheiros de *codebehind* efectua-se a gestão de alguns aspectos das páginas aspx. A maioria do código destes ficheiros limita-se a chamar métodos da camada da lógica de negócio para associar os dados que estes fornecem aos controlos existentes na página. A invocação de métodos pode acontecer sempre que a página é carregada, só quando é carregada pela primeira vez ou na sequência de eventos. Neste particular destaca-se a situação de mudança de opção em *dropdownlists* (por exemplo, ao mudar o ano ou mês, os dias disponíveis para serem escolhidos são calculados, garantindo a introdução de datas válidas) ou em *checkboxes* (como no caso da escolha do tipo de pagamento ao finalizar uma encomenda, que esconde ou revela os controlos consoante os dados que são necessários fornecer). O comportamento de mostrar ou esconder controlos é muito utilizado ao longo da aplicação, sendo os controlos muitas vezes reunidos numa tabela que, definida a propriedade "runat=server", também pode ou não ser visível (um exemplo é o *preview* disponibilizado ao introduzir ou alterar produtos ou passatempos). Alguns métodos foram produzidos localmente

nos ficheiros de *codebehind*, possuindo nomes e funcionalidades semelhantes. Assim, existem métodos “bind” em várias páginas, normalmente referentes a listagens em *DataGrids* (que fazem a sua paginação automática do controlo), mas que necessitam de ser associadas a dados (por este método) sempre que a página muda. Também existem métodos que limpam mensagens de erro que possam ter sido mostradas e que não façam sentido no contexto actual, e em quase todas as *DataGrids* que envolvam a visualização de um total existe um método que o obtém através da leitura dos campos da *DataGrid*, linha a linha, somando os valores relevantes.

O processamento lógico recorre, por vezes, a objectos *Session* para fazer persistir informação ao longo de várias páginas. Estes objectos também são usados no Backoffice para guardar a informação dos produtos ou passatempos antes destes serem inseridos na base de dados e para mostrar os *previews*.

No *codebehind* das páginas onde ocorre o envio de e-mails a recolha da informação necessária a constar da mensagem é recolhida e o método da lógica de negócio responsável pelo envio do e-mail é invocado através de um *thread*. Desta forma evita-se que um potencial erro momentâneo no servidor de mail prejudique a navegação posterior com a ocorrência de um erro fatal.

Um caso específico de produção de código, vital para o processamento da página, é a situação de produtos.aspx.cs. A página correspondente mostra a listagem de produtos, por publicação, por categoria ou por pesquisa, da Loja Web. Visto a listagem pretendida usar uma *DataList* e esta não possuir a funcionalidade de paginação existente nas *DataGrids*, foi necessário produzir uma solução que lê variáveis do URL, verifica o número de produtos (e páginas) que correspondem ao critério (por publicação, categoria ou pesquisa) e mostre uma das páginas da listagem de produtos com a opção para mudar para outras páginas, se estas existirem.

Para alguns controlos é preciso especificar na própria página, através de um excerto de código, qual o campo a mostrar (por exemplo, uma *DataList* à qual foi associado o resultado de uma *query* precisa deste expediente para identificar que controlos apresentam os diferentes campos do resultado) ou como o resultado deve ser formatado (como dinheiro ou percentagem, por exemplo).

Outra funcionalidade implementada nesta camada é o uso de cache.

Por um lado, o resultado das chamadas a alguns *Web User Controls* (o menu da Loja Web, o cabeçalho do Backoffice e os rodapés da Loja, Cartão e Backoffice) são colocados na cache. Também a página picReader.aspx da Loja Web e do Cartão Virtual é guardada. O menu e a página picReader vão guardar versões diferentes das páginas mediante os parâmetros existentes no URL (diferentes publicações escolhidas e diferentes produtos ou passatempos têm resultados diferentes). Para evitar dar um carácter permanente ao resultado guardado este é refrescado passado algum tempo.

Por outro lado, alguns dados existentes na base de dados e que raramente serão modificados (como países ou publicações do grupo Global Notícias), são lidos para um *dataset* e colocados na cache do ASP.NET quando a aplicação é iniciada. Para este procedimento acontecer foi necessário definir métodos no ficheiro Global.aspx que sejam chamados pelo seu evento *Application_start*. Os métodos guardam em diferentes itens da cache os diferentes dados, estando a validade de cada item dependente de um ficheiro, constantemente monitorizado pelo ASP.NET, que modificado (por um *trigger* da base de dados, despoletado

quando os dados são mudados) leva a nova invocação do método e consequente refrescamento da cache.

Nos ficheiros de configuração (*web.config*) de cada projecto foram definidas algumas variáveis globais como a localização dos ficheiros utilizados no mecanismo de cache, por exemplo, mas também quais as páginas de erro, de acesso limitado a utilizadores autenticados e quais os mecanismos de autenticação (através de um formulário presente numa página *web* - a página *login.aspx* - no caso da Loja Web e Cartão Virtual, ou através do IIS para o Backoffice³).

5.4 Windows Service

A aplicação desenvolvida necessitou de recorrer a um Windows Service para resolver a problemática colocada pelos diferentes prazos de pagamento existentes para as encomendas efectuadas na Loja Web. Adicionalmente, este serviço também limpa entradas obsoletas na tabela do carrinho de compras existente na base de dados.

Para atingir os propósitos foi criado um projecto para o Windows Service chamado *AutoPrazoEncomendas* e adicionado o DLL que contém os métodos da lógica de negócio.

No código foi definido um objecto do tipo *Timer* (sendo-lhe atribuído um valor de intervalo) e um evento, invocado a cada intervalo de tempo ultrapassado. Quando o serviço é iniciado o método *CancelaEncomenda* é chamado e o *Timer* activado. Após um determinado intervalo de tempo é invocado o evento do *Timer* que também chama *CancelaEncomenda*. O método *CancelaEncomenda* chama *Encomenda.SetPrazo* e o *stored procedure* que é executado coloca o estado das encomendas (novas ou pendentes, cujo tipo de pagamento usado foi Cheque ou Multibanco e que tenham ultrapassado o prazo máximo de pagamento dado) em “Prazo de pagamento ultrapassado”. Se as encomendas forem feitas por utilizadores registados e pontos tenham sido gastos, estes são repostos e é colocada uma entrada na tabela de histórico de pontos indicando a situação. Adicionalmente, entradas antigas na tabela do carrinho de compras são limpas por este procedimento. O *stored procedure* devolve a listagem das encomendas alteradas e, para cada uma delas, é chamado o método *Misc.MailingList.encomendaCanceladaPrazo*.

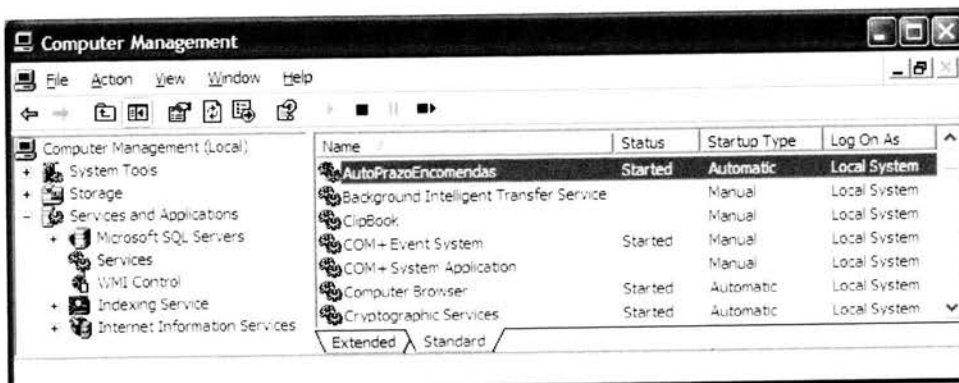


Ilustração 36 - Windows Service a correr na Management Console

³ A autenticação ao Backoffice foi, durante o período do estágio, efectuada com recurso a um formulário razão pela qual se manteve algumas referências a esse facto (tabelas e métodos usados para o efeito)

Para correr o serviço é necessário instalá-lo no servidor, correndo primeiro o utilitário da .NET Framework “installutil.exe” com o executável gerado pelo projecto como parâmetro. Após a instalação do serviço este é colocado a correr através da *Microsoft Management Console*.

5.5 Web Service

Aproveitando a facilidade de criação de Web Services pela plataforma .NET, e tendo em vista o possível uso por parte de outras páginas do grupo Global Notícias, foi criado um serviço que disponibiliza, por XML, o nome, uma pequena descrição e o endereço onde se encontra uma imagem alusiva a um dos produtos em destaque na Loja Web.



Ilustração 37 - Exemplo de uma página que invoca o Web Service

Num projecto à parte foi criado um Web Service chamado ProdPromo e incluído o DLL onde estão os métodos de lógica de negócio. Assim, foi possível criar um *método web* chamado Random que devolve um *DataSet* ao invocar *Produtos.GetPromoRandom*.

6 Conclusão

Os resultados obtidos à partir do trabalho desenvolvido durante o estágio permitem verificar que os objectivos delineados à partida foram cumpridos.

Foi projectada e implementada uma loja *online* que permite a venda de produtos promocionais das publicações da Global Notícias. A Loja Web desenvolvida possui todas as funcionalidades que se poderiam esperar de uma loja *online* como a visualização de produtos por categorias ou que correspondam a critérios de pesquisa, um carrinho de compras, um histórico das encomendas efectuadas e possibilidade de gestão destas, funcionalidades de registo e de autenticação, alteração dos dados pessoais e *password* fornecidos ou possibilidade de requisitar uma nova *password* temporária.

A Loja Web foi complementada com outro *site* para o “Cartão Virtual”. Os utilizadores registados no Cartão Virtual podem participar, através do *site*, em passatempos de variados tipos e ganhar prémios com as suas participações. O Cartão Virtual permite, tal como a Loja Web, o registo e autenticação dos utilizadores, bem como a manutenção dos dados fornecidos e pedidos de novas *passwords* temporárias. Além do histórico de encomendas efectuadas (com possibilidade de cancelar as encomendas ainda não processadas), o utilizador também pode ver o histórico de pontos acumulados/gastos e o histórico de passatempos participados (com informação detalhada sobre a sua participação).

Por último, a solução desenvolvida também contempla ferramentas de Backoffice associadas à Loja Web e ao Cartão Virtual. Usando um simples *browser web*, os operadores ganham acesso a uma zona de acesso restrito onde poderão gerir produtos (inserir, alterar ou modificar o stock), gerir passatempos (inserir, alterar ou atribuir os vencedores), gerir as encomendas efectuadas na loja *online* (alterar o seu estado, dando-as como pagas ou expedidas, por exemplo), ter acesso a dados estatísticos ao nível das encomendas, passatempos e registos no Cartão Virtual (permitindo saber, por exemplo, os produtos mais vendidos ou que faixa etária participa mais nos passatempos) ou obter os endereços electrónicos dos utilizadores que aceitaram receber mais informações de marketing da Global Notícias.

Uma das principais características do projecto é a divisão deste em camadas: base de dados, lógica de negócio e interface gráfico. Olhando para trás sobre as escolhas de tecnologias para implementar as diferentes camadas do projecto, verifica-se que a escolha da plataforma .NET e da base de dados SQL Server 2000 foram acertadas.

A base de dados é escalonável e proporciona os mecanismos necessários para o correcto funcionamento do projecto, como *triggers* ou *stored procedures*. Neste capítulo há que destacar, pela negativa, o facto de a sintaxe do SQL Server não suportar o uso de *arrays*. Em algumas situações durante o desenvolvimento tal foi notado, obrigando ao uso de tabelas temporárias para colmatar essa lacuna. A comunicação por ADO.NET entre a base de dados e o restante projecto é excelente, especialmente com o uso de *stored procedures*, que potenciam uma maior rapidez de execução e menor tráfego na rede.

O uso de tecnologia .NET permitiu fácil e rapidamente desenvolver a lógica de negócio e o interface gráfico. Em ficheiros de C#, a lógica de negócio cumpriu o seu papel de “cola” entre o interface e os dados: o contexto da aplicação foi convenientemente representado através de objectos, e métodos que invocam as *stored procedures* existentes na base de dados foram criados conforme necessário. Ao adicionar os DLL's resultantes da compilação dos ficheiros

da lógica de negócio aos projectos de ASP.NET da Loja Web, Cartão Virtual e Backoffice, tanto o contexto da aplicação como as operações sobre a base de dados ficaram acessíveis.

O interface em ASP.NET permitiu o uso de variados controlos e o recurso aos ficheiros de *codebehind* para os popular e definir acções a tomar aquando da ocorrência de certos eventos. Uma das características do ASP.NET utilizada foram os *Web User Controls*. Recorrendo a estes foi possível dividir uma página em partes e desenvolver estas partes em ficheiros separados. Isto permite, por exemplo, que modificações em cabeçalhos ou rodapés (que existem em todas as páginas) possam ser feitas num só local, repercutindo-se nas páginas de destino que, adicionalmente beneficiam de um código simplificado (uma *tag* basta para representar o código do *Web User Control*). Outra funcionalidade que se revelou útil para o projecto foi a cache do ASP.NET. Para tabelas que raramente serão modificados na base de dados e cujos dados são várias vezes requisitados pelo o interface, foi criado um *trigger* que gera/modifica um ficheiro aquando de inserções ou alterações. Através do ficheiro *Global.asax*, o ASP.NET é instruído a ir buscar inicialmente a informação à base de dados e a guardá-la na cache (aonde serão efectuados os pedidos de informação subsequentes). Para manter a informação na cache sempre actualizada, o ASP.NET monitoriza o ficheiro que é alterado pelo *trigger* e, quando tal acontece, actualiza a informação na cache. Por vezes, para guardar informação temporária ao longo de páginas, foram usados objectos do tipo *Session*, residentes no servidor e válidos, no máximo, até ao final de uma sessão. Similarmente, também se recorreu a *cookies* para guardar alguma informação não-confidencial nos computadores dos utilizadores. No capítulo da segurança, destacam-se a geração de números aleatórios criptograficamente fortes (usando um *cryptographic service provider* da .NET Framework) e o uso de métodos de *FormsAuthentication* que permitiram, por exemplo, criar o correspondente *hash* de uma *password*. Adicionalmente, no ficheiro de configuração (*Web.config*) de cada projecto é facilmente definido o tipo de autenticação a utilizar, bem como as páginas que só serão visíveis a utilizadores autenticados. As características dos ficheiros de configuração permitem também utilizá-los para definir variáveis globais ao nível da aplicação.

Durante o desenvolvimento foi necessário recorrer a *Javascript* para melhorar a usabilidade, uma vez que o comportamento dos controlos das páginas em ASP.NET nem sempre era o desejável e a plataforma não disponibiliza mecanismos próprios para os modificar. Também se verificou que a plataforma "peca" por não oferecer uma funcionalidade de paginação (igual à existente para *DataGrids*) para controlos do tipo *DataList*. Por essa razão a página que mostra a listagem de produtos na Loja Web teve que sofrer modificações ao nível do seu funcionamento lógico, permitindo aos utilizadores visualizar uma quantidade limitada de produtos por página e navegar entre as páginas da listagem. De facto, o controlo *DataGrid* revelou-se poderoso e simples, sendo utilizado tanto para listagens, como para edição de dados no Backoffice graças às propriedades de edição que possui.

A grande variedade de bibliotecas que compõe a .NET Framework foi uma vantagem para o projecto. Graças a elas foi possível redimensionar, proporcionalmente e para um tamanho predefinido, as imagens que o operador de Backoffice escolhia associar a um produto ou passatempo. De referir que o *upload* das imagens foi executado recorrendo ao controlo *HtmlInputFile*. Outras bibliotecas presentes permitem o envio de e-mails, e tal foi aproveitado para criar um conjunto de métodos que são chamados em situações predeterminadas como, por exemplo, após o registo de um utilizador, sempre que este actua sobre os seus dados na Conta Pessoal ou em situações como a colocação ou cancelamento de encomendas. Estes

e-mails também são enviados quando uma encomenda vê o seu estado modificado pelo operador do Backoffice. Um estado de encomenda que precisou de um tratamento especial foi o de “prazo de pagamento ultrapassado”. Foi necessário criar um Windows Service que corre no servidor e periodicamente procura por encomendas nessa situação e automaticamente as processa, modificando-lhe o estado e enviando um e-mail ao utilizador.

A facilidade com que podem ser desenvolvidos XML Web Services levou também à criação de um serviço que disponibiliza informação sobre um dos produtos em destaque na Loja Web. Os dados fornecidos são semelhantes aos presentes na página principal do *site* do Cartão Virtual, permitindo ver o nome, imagem e descrição de um produto. Este Web Service ou serviços semelhantes podem ser utilizados por alguns *sites* do grupo Global Notícias para a divulgação de produtos, trazer mais visitantes à Loja Web, podendo eventualmente aumentar o volume de vendas desta.

À data de escrita deste relatório a Loja Web e o Cartão Virtual ainda não se encontravam *online* e disponíveis aos internautas, apesar disso a parte de Backoffice estava já a ser utilizada pelo departamento de Marketing para inserção de dados na base de dados, libertando o departamento de Informática desse trabalho. O projecto desenvolvido está por isso, desde já, a ter um impacto sobre a organização. Quanto ao potencial sucesso e utilidade das restantes secções do projecto, estas só poderão ser realmente aferidas após o lançamento dos respectivos *sites*. De facto, sendo versões melhoradas e com mais funcionalidades de *sites* anteriormente existentes, o seu sucesso poderá ser medido pelo aumento de utilizadores registados, vendas de produtos e participações em passatempos (e é credível que tal aconteça graças à introdução de novos produtos e mais passatempos).

O trabalho desenvolvido para o projecto poderá, no entanto, ser objecto de melhoramentos e usufruir de mais funcionalidades. Um dos melhoramentos, à partida, mais óbvios é a integração com o sistema existente na loja física. Aquando do início do projecto os stocks dos produtos não estavam inseridos no sistema e foi decidido manter o *status quo* relativamente aos preços, com a definição de preços por região tendo estes já os portes de envio incluídos. Futuramente a integração e o estabelecimento de valores de portes de envio revelar-se-ia proveitosa por acabar com a duplicação de dados e reduzir a quantidade de informação que necessita de ser inserida no Backoffice. O projecto também poderia ser expandido para o campo da Internet móvel. Com o advento da terceira geração, a Global Notícias poderá, a médio prazo, utilizar a vantagem de pertencer ao grupo PT para vender os seus produtos através de um portal como o i9 da TMN. Apesar de esta tecnologia ainda não possuir muitos utilizadores, devido aos atrasos na sua implementação e ao preço dos terminais 3G, o portal i9 já permite a compras *online*, nomeadamente de bilhetes dos cinemas Lusomundo (com o pagamento a ser efectuado por Multibanco), por sinal outra empresa do grupo Portugal Telecom.

Em última análise, o estágio foi considerado como bastante útil pois permitiu uma experiência de integração num ambiente empresarial, a realização de um projecto aliciante e altamente motivador e, inerentemente, a aquisição e aprimoramento de várias competências, especialmente a nível tecnológico, que só um projecto de grande envergadura pode proporcionar.

Referências e Bibliografia

Bibliografia

- Bird, Danielle Voeller (2003). *Building Secure ASP.NET Applications*. Microsoft Press, Redmond, Washington, EUA. ISBN: 0-7356-1890-9
- Chappell, David (2002). *Understanding .NET: A Tutorial and Analysis*. Addison-Wesley, Boston, Massachusetts, EUA. ISBN: 0-201-74162-8
- Davis, Harold (2002). *Visual C# .NET Programming*. Sybex, Alameda, Califórnia, EUA. ISBN: 0-7821-4046-7
- Duthie, G. Andrew; MacDonald, Matthew (2003). *ASP.NET in a nutshell, Second Edition*. O'Reilly, Sebastopol, Califórnia, EUA. ISBN: 0-596-00520-2
- Hurwitz, Dan; Liberty, Jesse (2003). *Programming ASP.NET, Second Edition*. O'Reilly, Sebastopol, Califórnia, EUA. ISBN: 0-596-00487-7.
- Ortiz, Jonothon; Ahmed, Mesbah; Faircloth, Jeremy (2002). *ASP.NET Web Developer's Guide*. Syngress Publishing, Rockland, Massachusetts, EUA. ISBN: 1-928994-51-2.
- Platt, David S. (2003). *Introducing Microsoft .NET, Third Edition*. Microsoft Press, Redmond, Washington, EUA. ISBN: 0-7356-1918-2
- Reilly, Douglas (2002). *Designing Microsoft ASP.NET Applications*. Microsoft Press, Redmond, Washington, EUA. ISBN: 0-7356-1348-6
- Riordan, Rebecca M. (2002). *Microsoft ADO.NET Step by Step*. Microsoft Press, Redmond, Washington, EUA. ISBN: 0-7356-1236-6

Referências Web

- <http://msdn.microsoft.com>
- <http://support.microsoft.com>
- <http://webservices.xml.com>
- <http://www.asp.net>
- <http://www.aspalliance.com>
- <http://www.builder.com>
- <http://www.codeproject.com>
- <http://www.c-sharpcorner.com>
- <http://www.devasp.net>
- <http://www.developer.com>
- <http://www.dotnet247.com>
- <http://www.experts-exchange.com>

- <http://www.gotdotnet.com>
- <http://www.microsoft.com/sql/>
- <http://www.microsoft.com/windowsserver2003/iis/>
- <http://www.netscape.com/security/techbriefs/>
- <http://www.ondotnet.com>
- <http://www.w3.org>
- <http://www.webservices.org>
- <http://www.webservicesarchitect.com>

ANEXO A: Imagens da aplicação

Neste anexo apresentam-se imagens do interface gráfico da aplicação desenvolvida nas suas vertentes Cartão Virtual, Loja Web e Backoffice. Algumas páginas foram omitidas devido à sua similaridade com outras (por exemplo, as de ajuda da Loja Web e Cartão Virtual, as do registo, autenticação e de outras funções da secção Conta Pessoal da Loja Web, o histórico de encomendas disponível no Cartão Virtual, algumas estatísticas existentes no Backoffice).

Cartão Virtual



Ilustração 38 - Página principal



Ilustração 39 - Registo no Cartão Virtual

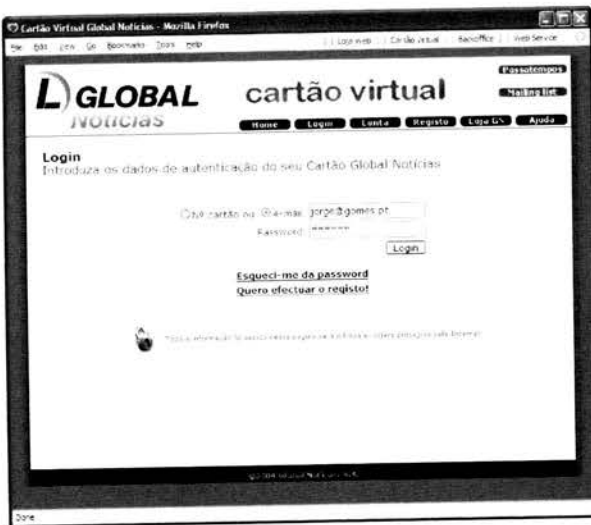


Ilustração 40 – Páginas de autenticação e de ajuda

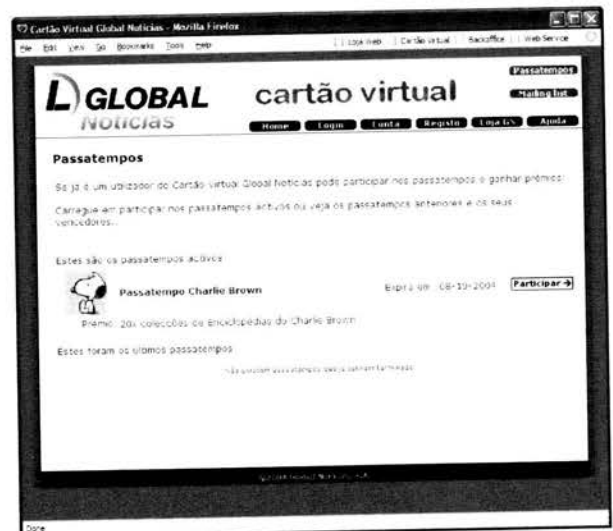
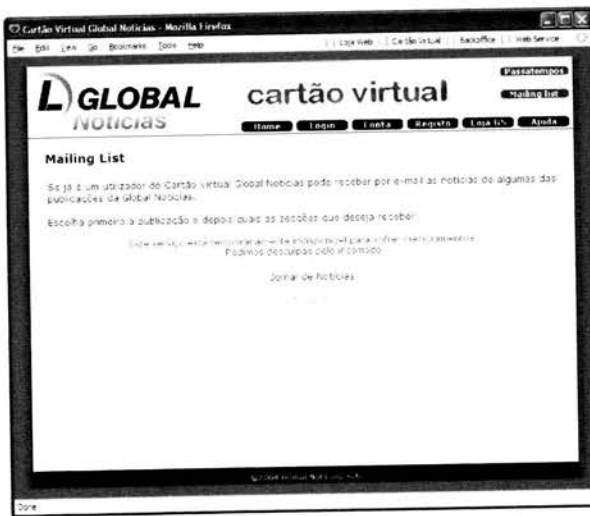


Ilustração 41 – Página de pedido de envio de notícias e principal dos passatempos



Ilustração 42 – Participação em passatempos do tipo click e frase

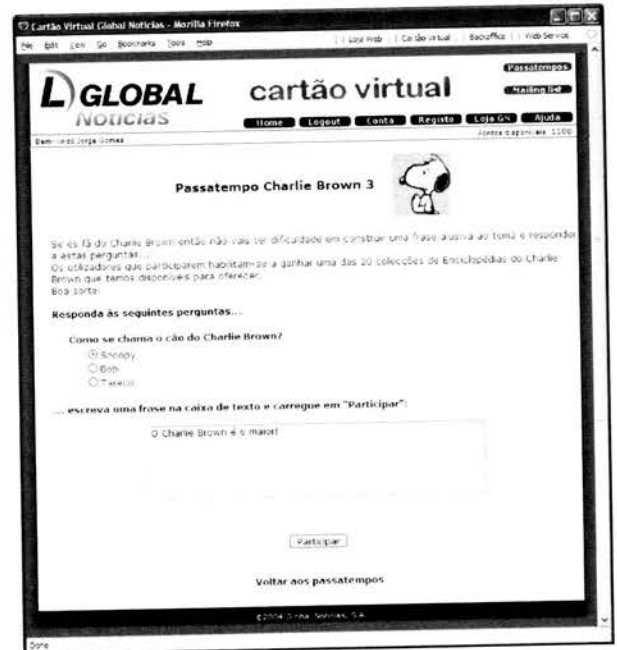


Ilustração 43 – Participação em passatempos do tipo “perguntas e respostas” e “perguntas e respostas + frase”



Ilustração 44 – Participação com sucesso num passatempo

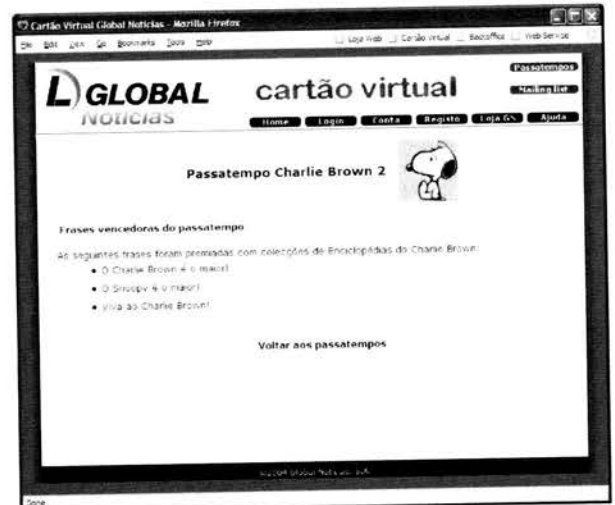


Ilustração 45 – Visualização dos vencedores e das frases vencedoras

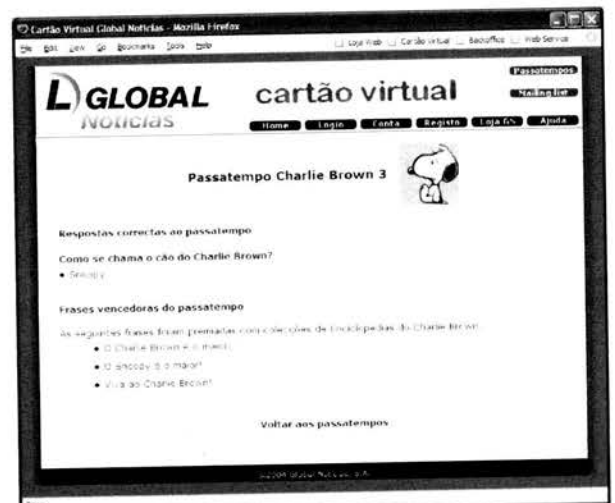
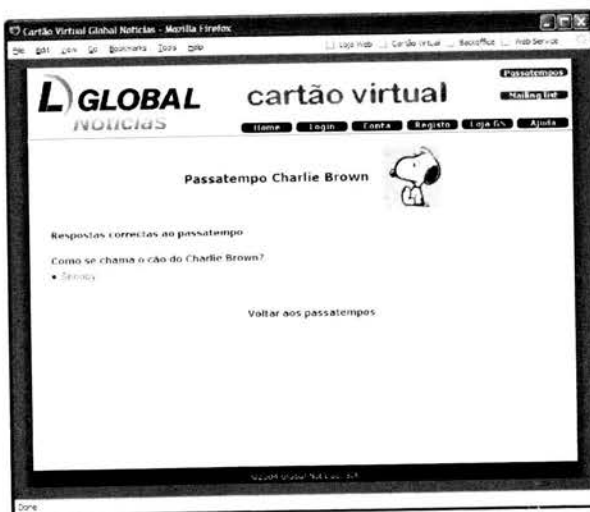


Ilustração 46 – Visualização das respostas correctas e das respostas correctas + frases vencedoras



Ilustração 47 – Página principal da secção Conta Pessoal

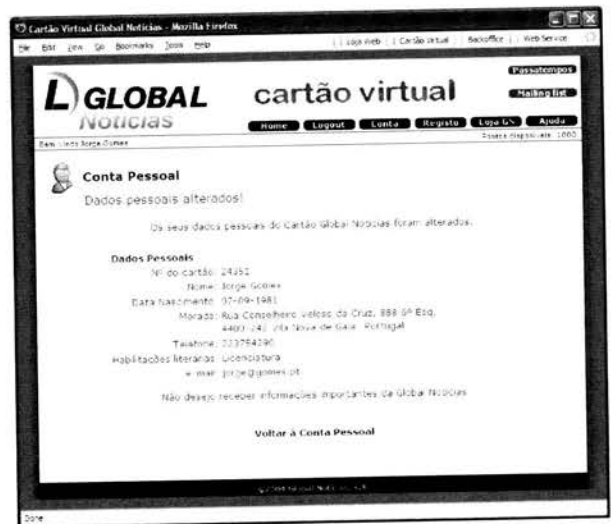
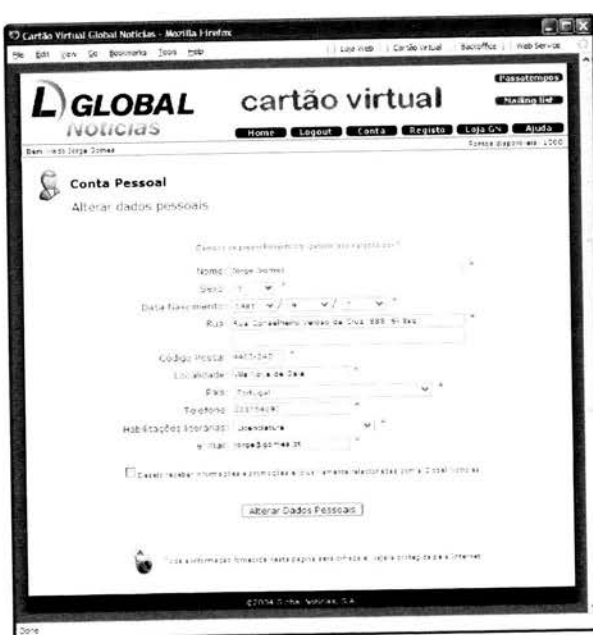


Ilustração 48 – Alteração dos dados pessoais



Ilustração 49 – Alteração da *password*



Ilustração 50 – Recuperação da *password*



Ilustração 51 – Histórico de passatempos

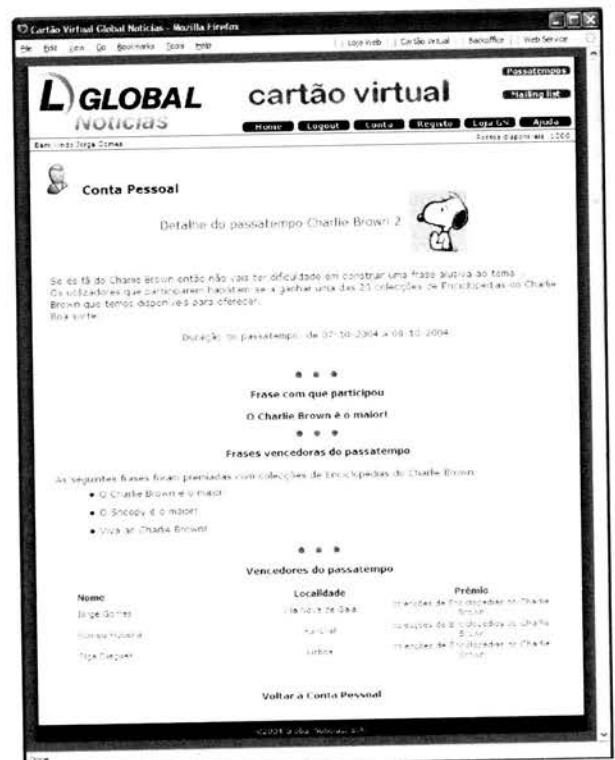


Ilustração 52 – Detalhe de passatempos do tipo “click” e “frase”

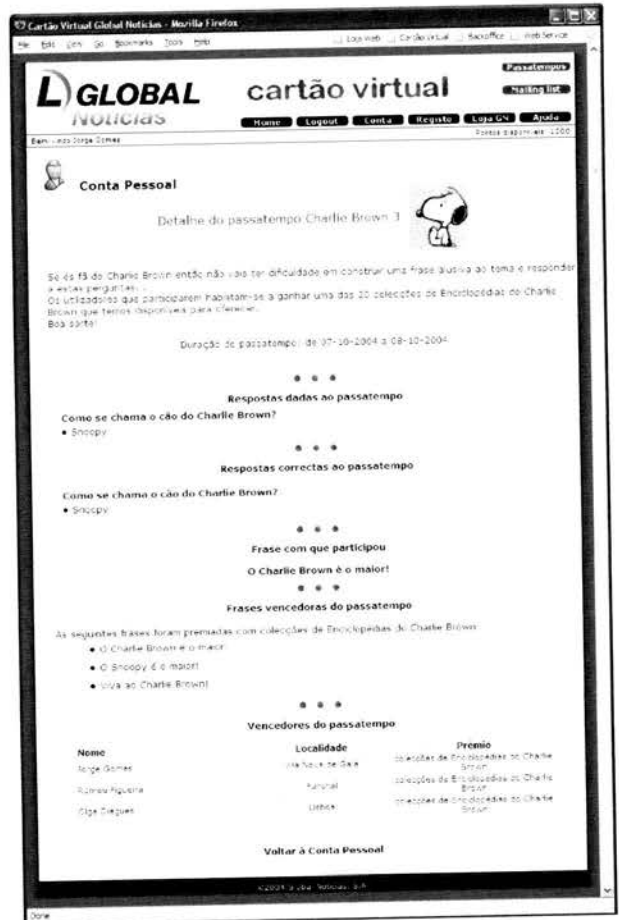
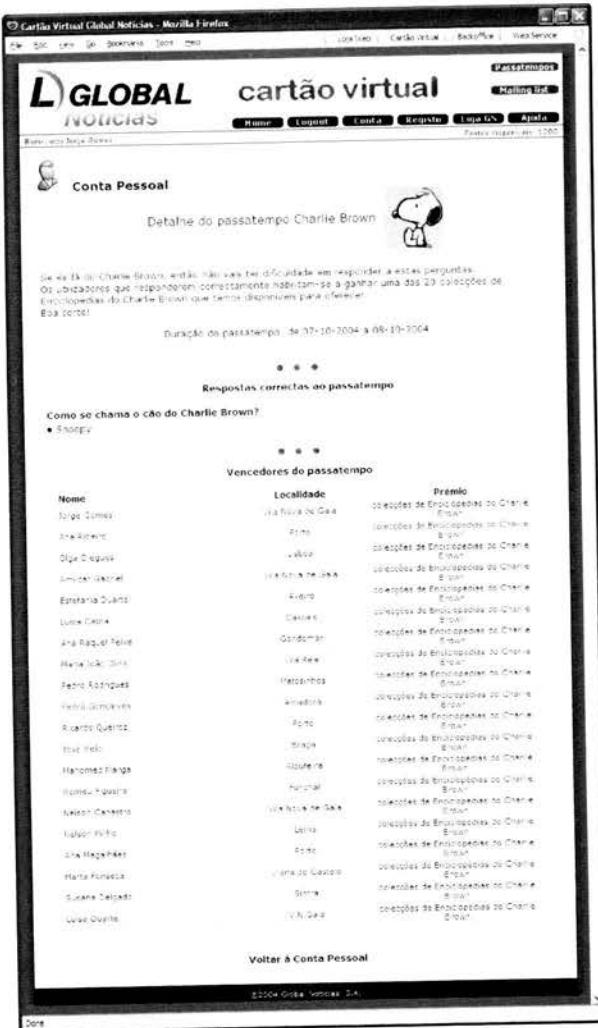


Ilustração 53 – Detalhe de passatempos do tipo “perguntas e respostas” e “perguntas e respostas + frase”

Loja Web



Ilustração 54 – Página principal da Loja Web



Ilustração 55 – Lista de produtos por publicação e por categoria



Ilustração 56 – Resultados de uma pesquisa e lista de produtos de uma colecção



Ilustração 57 – Escolha de uma região e detalhe de um produto



Ilustração 58 – Opções para utilizadores não-autenticados e autenticados (com poucos pontos)



Ilustração 59 – Carrinho de compras



Ilustração 60 – Finalizar a encomenda (fornecer dados de envio e pagamento)



Ilustração 61 – Opções de pagamento (cheque e Visa)



Ilustração 62 – Confirmação da encomenda e informações para efectuar pagamento



Ilustração 63 – Ver o estado de uma encomenda (através do código fornecido)



Ilustração 64 – Histórico das encomendas efectuadas e detalhe de uma encomenda



Ilustração 65 – Mensagem de confirmação de que uma encomenda foi cancelada

Backoffice



Ilustração 66 – Página principal do Backoffice

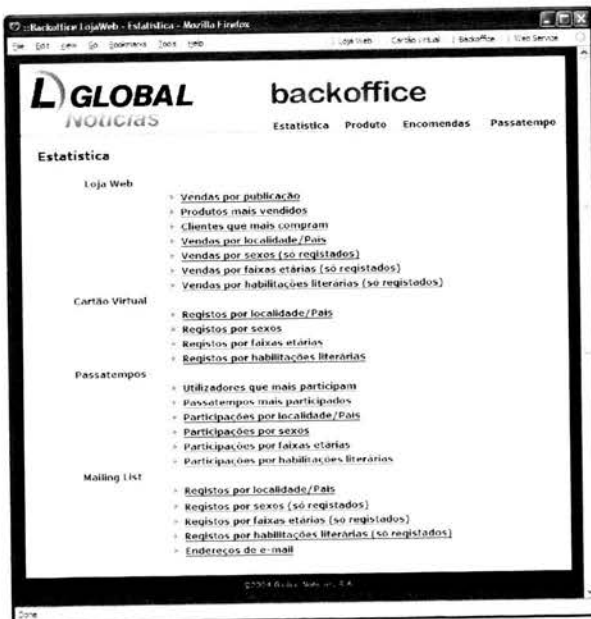


Ilustração 67 – Secção de estatística e estatística de vendas por publicação

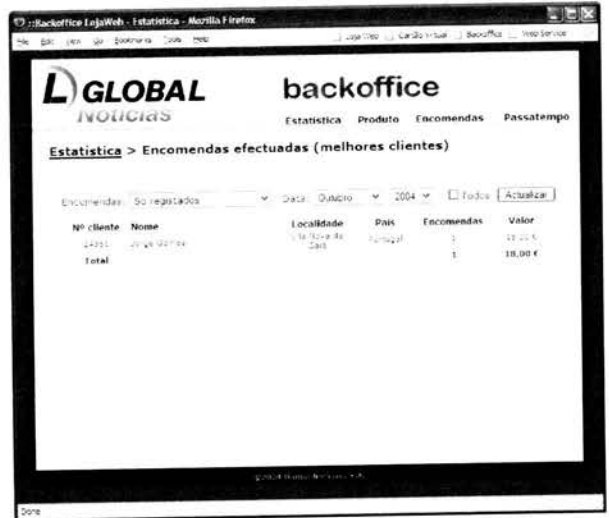
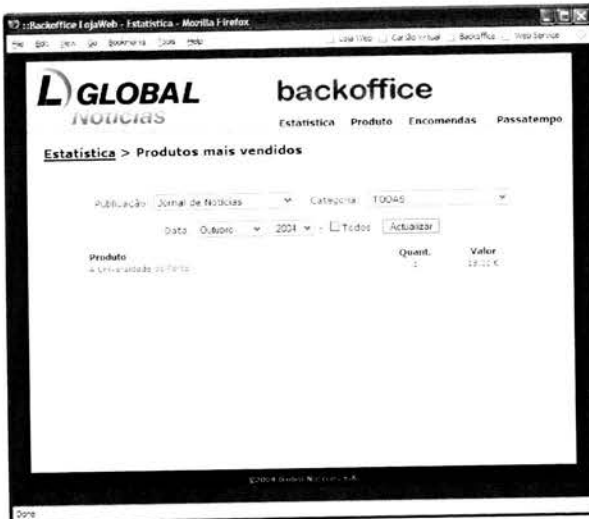


Ilustração 68 – Produtos mais vendidos e melhores clientes

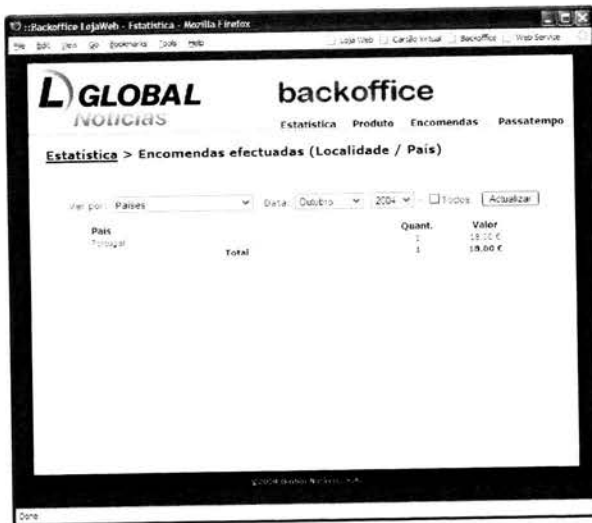


Ilustração 69 – Vendas por países e registos no Cartão Virtual por sexo

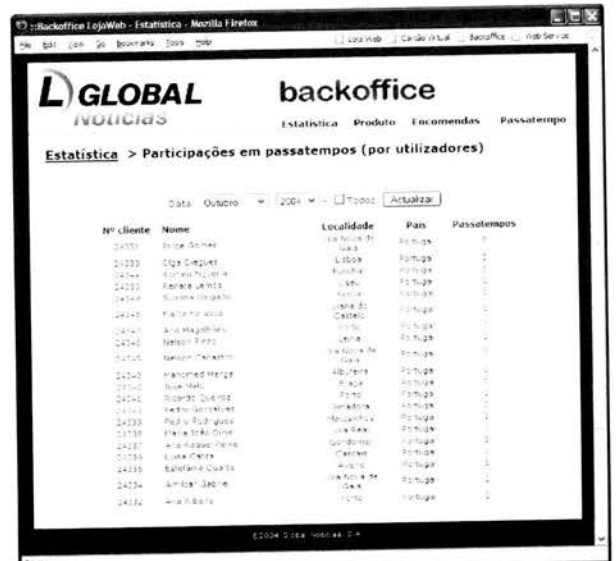


Ilustração 70 – Passatempos mais participados e utilizadores que mais participam

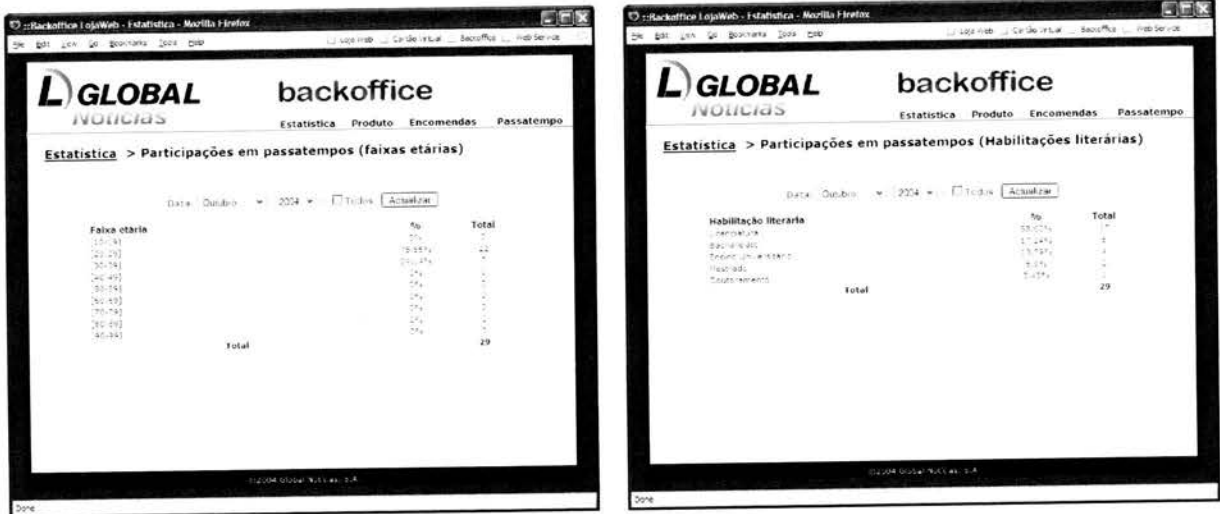


Ilustração 71 – Participações nos passatempos por faixas etárias e por habilitações literárias

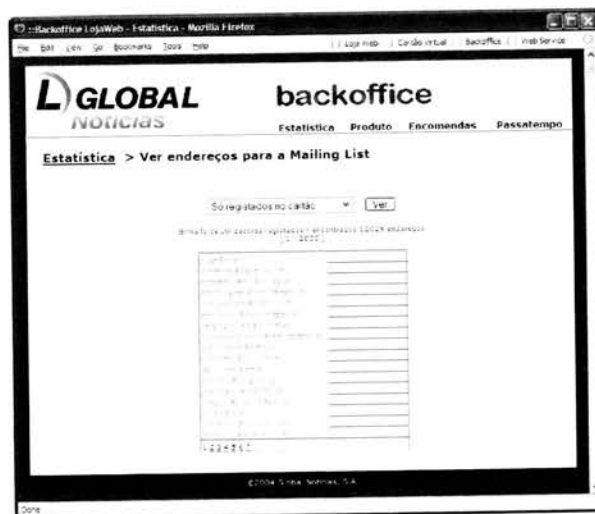


Ilustração 72 – Endereços de e-mail dos utilizadores que desejam receber informações adicionais da Global Notícias

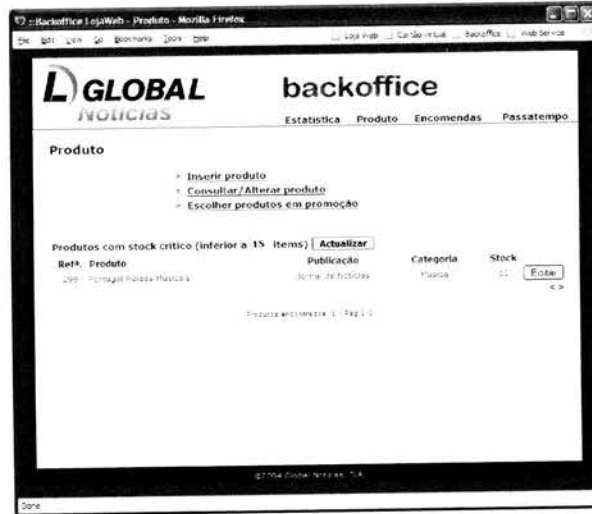


Ilustração 73 – Secção dos produtos

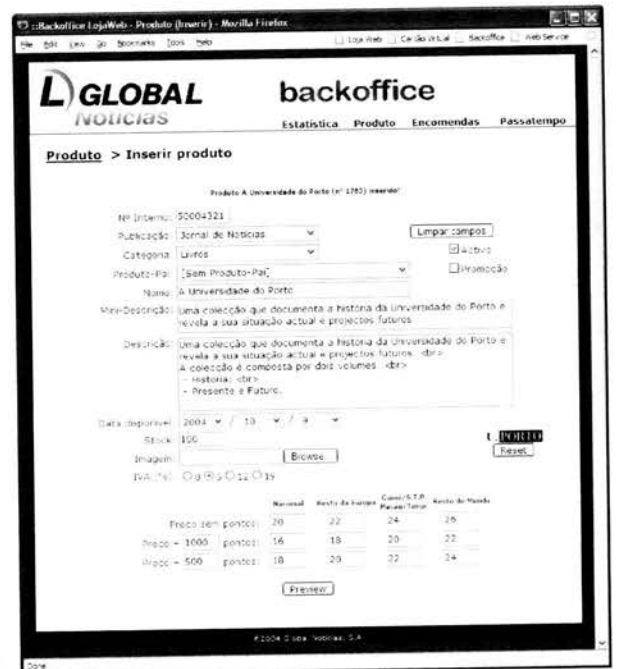
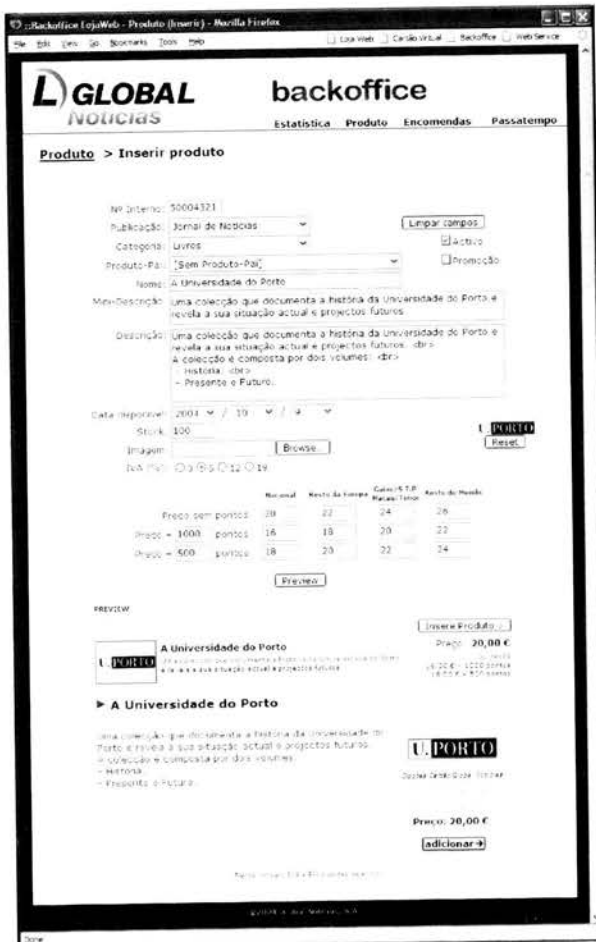


Ilustração 74 – Inserir um produto



Ilustração 75 – Procurar um produto para o alterar

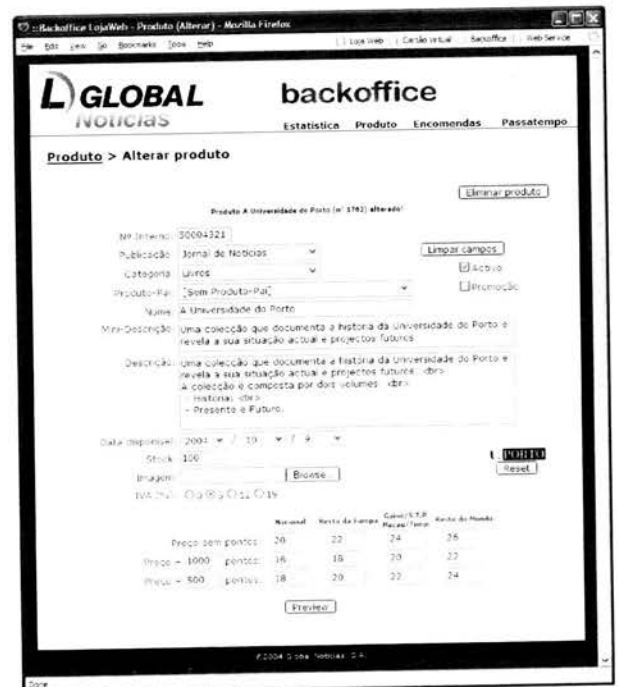
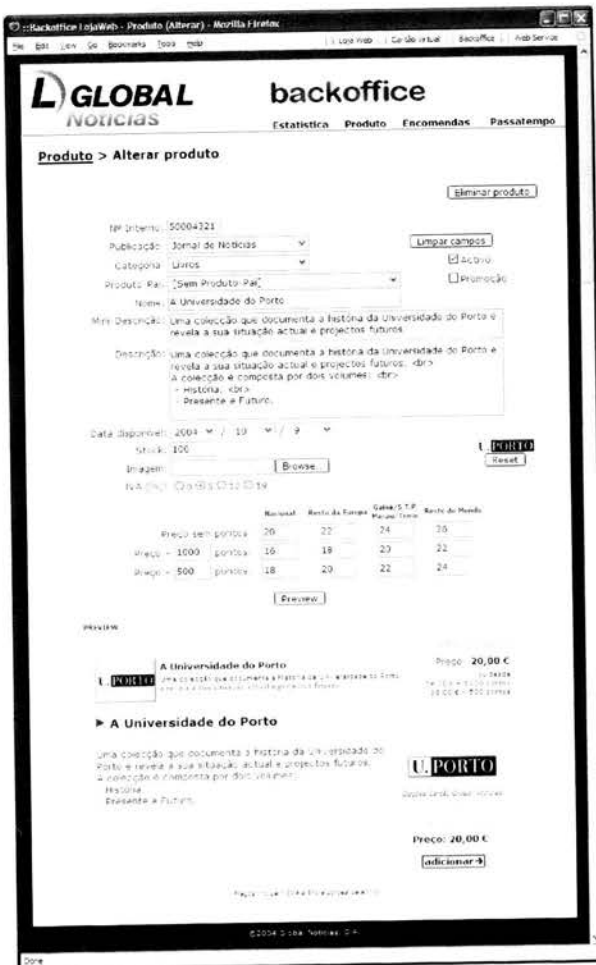


Ilustração 76 – Alterar um produto



Ilustração 77 – Edição da lista de produtos em destaque na Loja Web

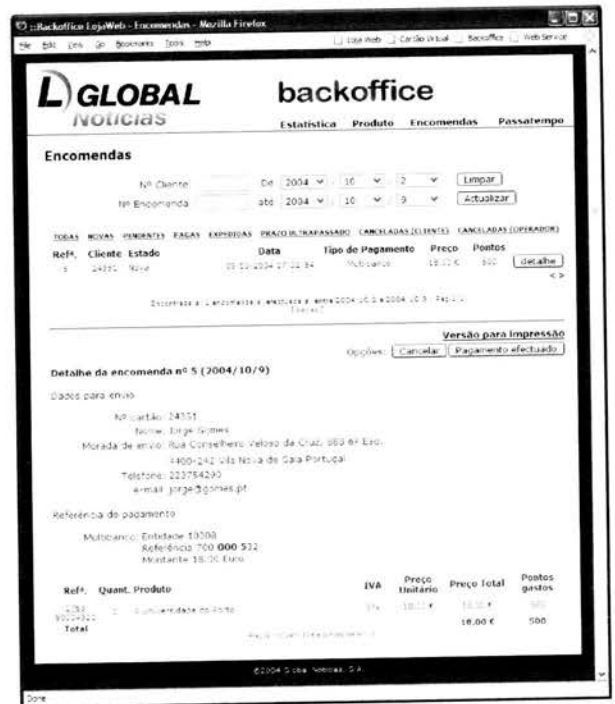
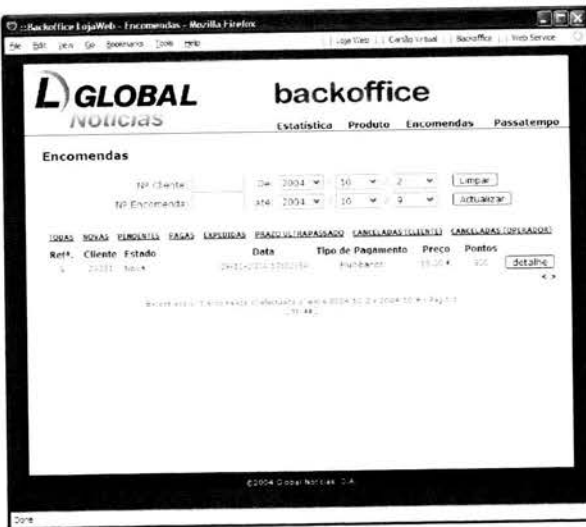


Ilustração 78 – Secção de processamento de encomendas e detalhe de uma encomenda



Ilustração 79 – Versão para impressão do detalhe e confirmação de mudança de estado



Ilustração 80 – Secção dos passatempos

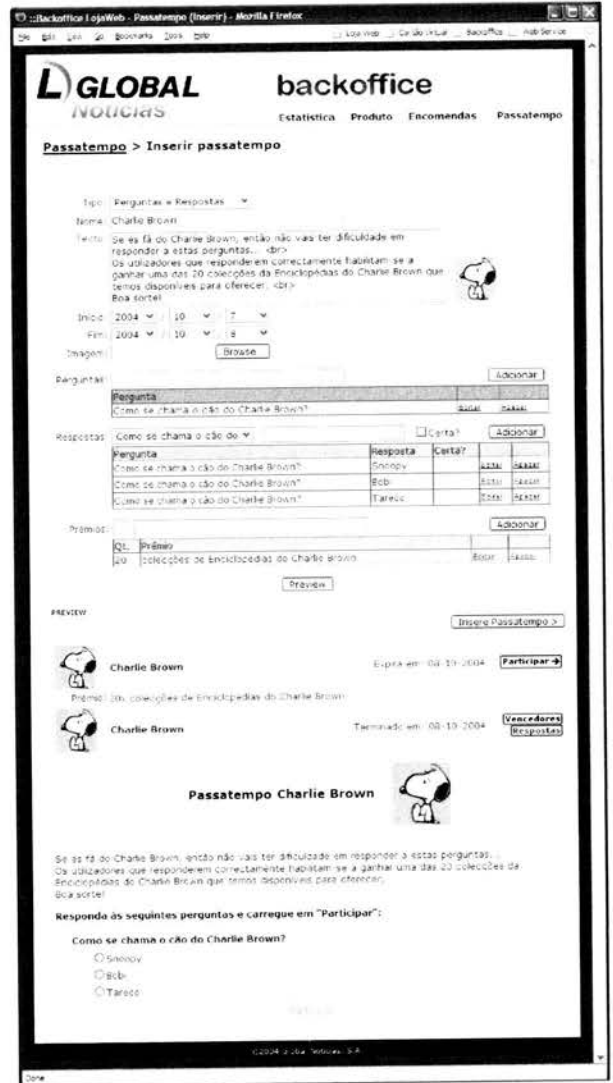
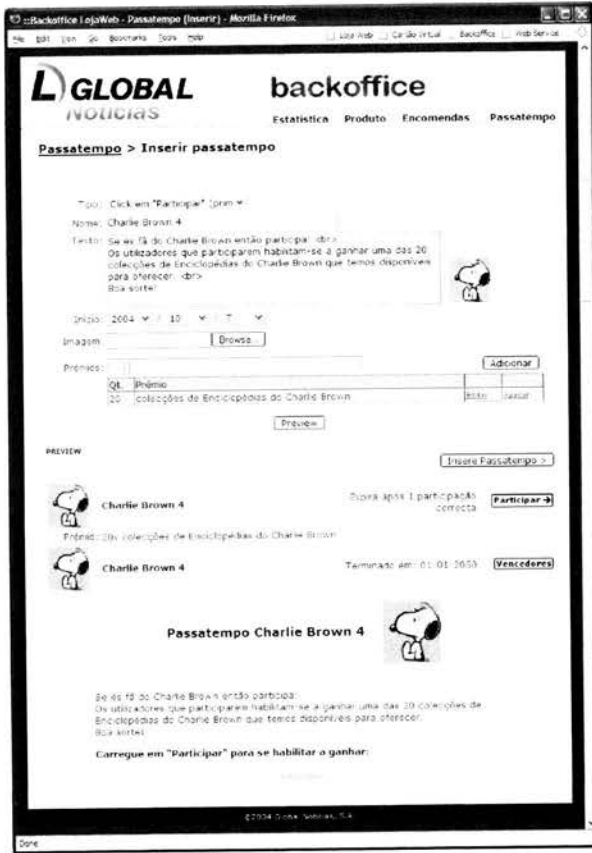


Ilustração 81 – Inserir passatempos do tipo “click” e “perguntas e respostas”



Ilustração 82 – Procurar um passatempo para o alterar

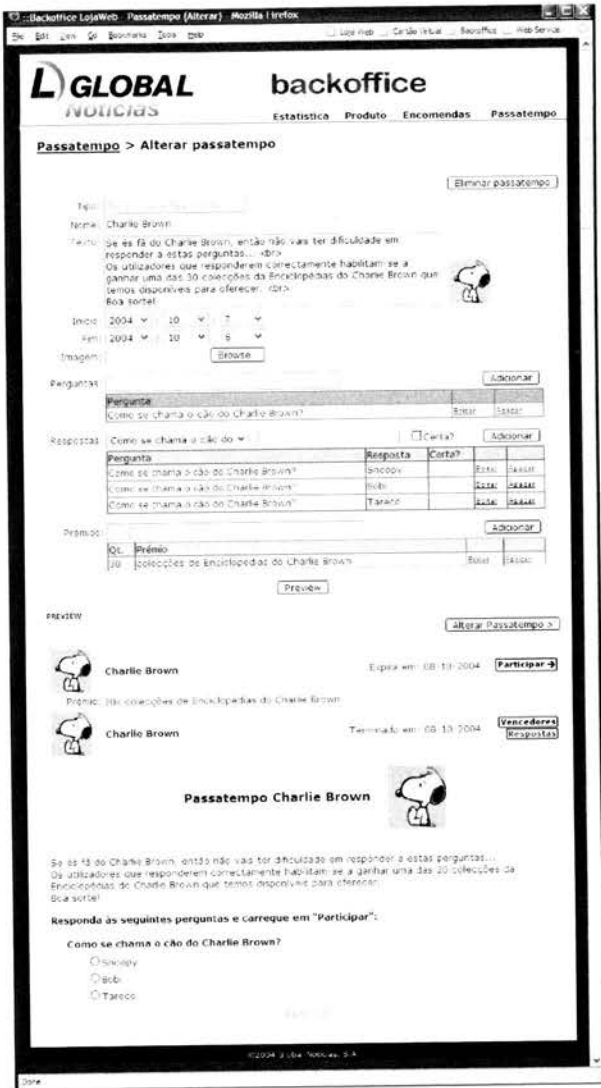


Ilustração 83 – Alterar um passatempo e pormenor da edição nas DataGrids



Ilustração 84 – Mensagem que confirma a eliminação de um passatempo

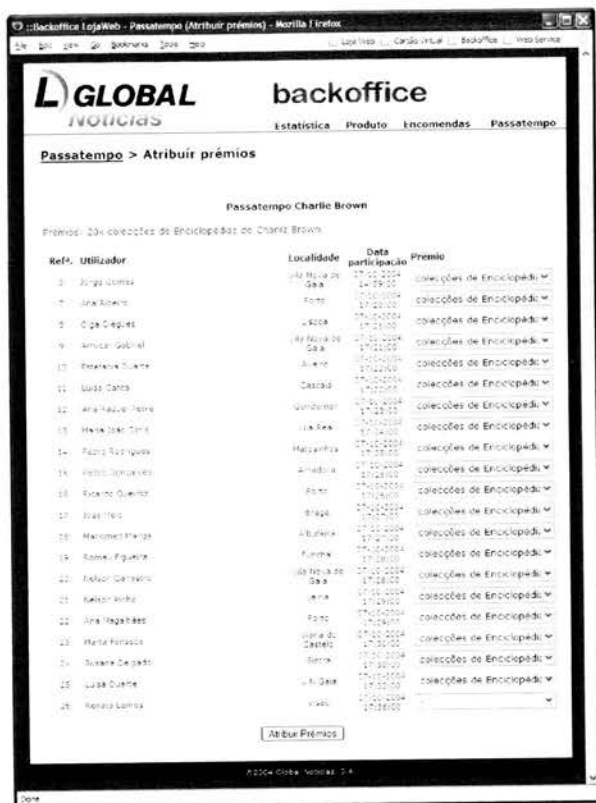


Ilustração 85 – Atribuição dos prémios de um passatempo





FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

BIBLIOTECA



0000081498