

André Filipe Ferreira Cardoso

Publicidade móvel adaptada ao utilizador

Universidade Fernando Pessoa – Porto, 2008

André Filipe Ferreira Cardoso

Publicidade móvel adaptada ao utilizador

Universidade Fernando Pessoa – Porto, 2008

Publicidade móvel adaptada ao utilizador

Por:

André Filipe Ferreira Cardoso

Orientador

Professor Doutor Nuno Magalhães Ribeiro

Dissertação apresentada à Universidade Fernando Pessoa como parte de requisitos para a obtenção do grau de Mestre em Engenharia Informática – ramo de Computação Móvel.

Resumo

Esta dissertação desenvolve um caso de estudo sobre a publicidade móvel adaptada ao utilizador. O trabalho desenvolvido consiste em duas aplicações: uma aplicação *Web* e uma aplicação móvel para Pocket PC. Este caso de estudo explora algumas das possíveis utilizações de tecnologias de computação móvel no sector da publicidade, nomeadamente na divulgação de anúncios, permitindo que empresas, ou outros anunciantes, consigam difundir a respectiva publicidade a um baixo custo e orientada ao público-alvo correcto.

A dissertação inicia-se com uma apresentação detalhada sobre as tecnologias e ferramentas subjacentes ao desenvolvimento e criação das aplicações, indicando as motivações que sustentaram a respectiva escolha. Posteriormente, faz-se uma descrição pormenorizada do desenvolvimento das aplicações móveis, indicando os passos mais importantes, sem descurar uma fase inicial da análise de requisitos para as aplicações. Assim, faz-se uma descrição do problema, bem como do método de engenharia de software aplicado para o desenvolvimento das aplicações. Para além disso determina-se igualmente a abordagem seguida para o desenvolvimento do mecanismo de sincronização de bases de dados entre as duas aplicações.

A seguir, descreve-se pormenorizadamente o conjunto de passos seguidos para a construção e desenvolvimento das aplicações com o objectivo de assegurar, e posteriormente avaliar por meio de uma experiência, que, tanto a aplicação *Web* como a aplicação móvel, permitem satisfazer as necessidades identificadas, tendo-se igualmente avaliado a interacção com o utilizador em termos de facilidade de uso e funcionalidades proporcionadas, usando para isso um questionário composto por 13 perguntas fechadas e 3 perguntas abertas, bem como um desafio que serviu para avaliar a eficiência e eficácia do conjunto de aplicações.

Por fim, concluiu-se que as aplicações desenvolvidas vieram contribuir de um modo positivo para uma gestão centralizada da distribuição de publicidade e de anúncios, sendo a mobilidade proporcionada um trunfo importante. Além disso, a interface desenvolvida foi igualmente bem aceite por parte dos participantes na avaliação das aplicações. Globalmente, as aplicações tiveram uma grande aceitação por parte dos

participantes no estudo realizado, constituindo uma solução tecnológica com potencialidades para trazer alguma inovação sector, já que permite adaptar a publicidade aos perfis de utilizador.

Abstract

This dissertation is about a case study that involves user adaptive mobile publicity. The work herein consists of two applications: one Web application and another application that targets the Pocket PC platform. This case study explore some of the possible utilizations of mobile technologies in the area of announcements and publicity, allowing companies or advertisers to create and post publicity at lower costs and post it to the correct target audience.

Initially, it presents the technologies and tools behind the creation and development of the applications, indicating the reasons that support the choice of programming technologies.

Later on, a detailed description is carried out, indicating the more important steps followed through a system analysis of the applications. It is also presented a description of the software engineering model used during the development of the applications. Furthermore, it is also described the approach followed in the development of the database synchronization and replication mechanism between the applications

Next, it is described in detail, the steps taken for the development of the applications with the aim of ensuring that both applications meet the needs identified. In order to evaluate the functionality and the ease of use, an experience was carried out, using a questionnaire composed of thirteen close questions and three open questions, as well a challenge that evaluates the efficiency and effectiveness of the applications.

In the end, it is concluded that the developed applications bring a positive contribution to a centralized management and distribution of advertisements and announcements. The mobility characteristic was regarded as an important factor and the interface was well accepted in the context of both applications. Finally, the applications were well accepted by all the participants in the study, proving that it has the potential to bring some innovation to advertising, as this solution allows to adapt advertisements based on user's profiles.

Aos Meus Pais

Agradecimentos

Agradeço aos meus pais pelo seu apoio incondicional durante toda a minha vida, pelo exemplo de dedicação e perseverança, pela compreensão, carinho e amor. Sem eles não teria chegado até este ponto e por isso esta dissertação é-lhes dedicada. Obrigado por acreditarem em mim.

Agradeço à minha namorada Raquel pelo apoio, companheirismo e, sobretudo, pelo amor incondicional que me reconforta e me dá forças para superar obstáculos. Obrigado pela compreensão demonstrada no decorrer deste desafio.

Agradeço à minha irmã, pelo apoio dado para enfrentar este desafio, encarando-o como um bem essencial para o meu desenvolvimento pessoal e intelectual.

Ao Professor Nuno Ribeiro, pela sua disponibilidade, paciência e brilhantismo ao ensinar, pelos conselhos e dedicação não só como orientador, mas também como professor e amigo.

A todos os meus amigos que souberam estar lá e que nunca me deixaram de apoiar e de incentivar.

A todos os professores do curso pela ajuda, apoio e passagem de conhecimentos ao longo do curso, nas disciplinas por eles leccionadas e não só.

Índice

Resumo.....	I
Abstract	III
Agradecimentos.....	V
Índice de figuras	IX
Índice de tabelas	XII
Índice de Exemplo de código	XIII
1. Introdução	- 1 -
1.1. Descrição do problema.....	- 2 -
1.2. Objectivos principais.....	- 3 -
1.3. Motivação.....	- 3 -
1.4. Metodologia	- 4 -
1.5. Estrutura do documento	- 6 -
2. Tecnologias para o desenvolvimento do trabalho	- 7 -
2.1. Tecnologias para o desenvolvimento de aplicações móveis	- 7 -
2.1.1. Plataforma Windows Mobile.....	- 8 -
2.1.2. <i>Mobile Device Center</i>	- 7 -
2.1.3. <i>Google Maps API</i>	- 8 -
2.1.4. Microsoft SQL server CE.....	- 10 -
2.2. Tecnologias para o desenvolvimento do servidor web.....	- 10 -
2.2.1. WAMP Server 2.0.....	- 10 -
2.2.2. HTML	- 13 -
2.2.3. CSS.....	- 14 -
2.3. XML.....	- 14 -
2.4. Publicação Web.....	- 15 -
2.5. Ferramentas de desenvolvimento de <i>software</i>	- 15 -
2.5.1. Visual Studio 2005 – SP1 – (C# CF .NET).....	- 15 -
3. Análise da aplicação “Pub Móvel”	- 18 -
3.1. Descrição do problema.....	- 18 -
3.2. Metodologia de desenvolvimento do software.....	- 20 -
3.2.1. Modelo Incremental	- 20 -
3.2.2. Prototipagem	- 22 -

3.3.	Análise e planeamento das aplicações de <i>software</i>	- 23 -
3.3.1.	Levantamento de requisitos funcionais	- 23 -
3.3.2.	Levantamento de requisitos não funcionais	- 26 -
3.3.3.	Interface do utilizador	- 27 -
3.3.4.	Mecanismo de replicação da base de dados	- 29 -
3.4.	Estruturas de navegação nas aplicações de software.....	- 30 -
3.5.	Concepção das bases de dados	- 33 -
3.5.1.	Base de dados da aplicação “Pub Móvel”	- 33 -
3.5.2.	Base de dados da aplicação “NewsMobile“	- 35 -
4.	Implementação das aplicações “Pub móvel” e “NewsMobile”.....	- 37 -
4.1.	Implementação da aplicação “PubMovel”	- 37 -
4.1.1.	Formulários da aplicação “PubMovel”	- 40 -
4.1.1.1.	Formulários genéricos	- 41 -
4.1.1.2.	Formulários Publicidade	- 45 -
4.1.1.3.	Formulários Pontos de interesse.....	- 49 -
4.1.1.4.	Formulários Favoritos	- 52 -
4.1.1.5.	Opção Visualizar Favoritos	- 60 -
4.1.1.6.	Classes auxiliares para o desenvolvimento da aplicação PubMovel.....	- 61 -
4.1.2.	Implementação da base de dados da aplicação “PubMovel”	- 63 -
4.2.	Implementação da aplicação <i>Web</i> ”NewsMobile”	- 64 -
4.2.1.	Páginas da aplicação <i>Web</i> “NewsMobile”	- 65 -
4.2.1.1.	Página Index.php.....	- 65 -
4.2.1.2.	Página AddTipoReg.php	- 67 -
4.2.1.3.	Página AddTipo.php.....	- 68 -
4.2.1.4.	Página RegUsername.php	- 68 -
4.2.1.5.	Página RegUserTipo.php	- 70 -
4.2.1.6.	Página Main.php.....	- 71 -
4.2.1.7.	Página Conta.php	- 72 -
4.2.1.8.	Página MudarPwd.php	- 72 -
4.2.1.9.	Página DetalhesConta.php	- 74 -
4.2.1.10.	Página DelConta.php.....	- 76 -
4.2.1.11.	Página Publicidade.php	- 77 -
4.2.1.12.	Página InsPub.php	- 78 -
4.2.1.13.	Página VerPub.php.....	- 80 -

4.2.1.14.	Página DeletePub.php	- 81 -
4.2.1.15.	Página Noticia.php	- 82 -
4.2.1.16.	Página ActPub.php.....	- 83 -
4.2.1.17.	Páginas em modo admin	- 85 -
4.2.1.18.	Página Index.php (Admin)	- 85 -
4.2.1.19.	Página Main.php (Admin).....	- 86 -
4.2.1.20.	Página Limpeza.php (Admin)	- 87 -
4.2.2.	Base de dados da aplicação “NewsMobile”	- 88 -
4.3.	Mecanismo de replicação da base de dados	- 90 -
5.	Estudo empírico.....	- 94 -
5.1.	Técnicas de avaliação.....	- 94 -
5.2.	Experiência.....	- 96 -
5.3.	Avaliação das aplicações.....	- 97 -
5.4.	Análise e discussão de resultados.....	- 100 -
5.4.1.	Perfil dos participantes	- 100 -
5.4.2.	Adequação do vocabulário.	- 102 -
5.4.3.	Facilidade de uso.....	- 103 -
5.4.4.	Funcionalidade	- 105 -
5.4.5.	Utilização da aplicação.....	- 107 -
5.4.6.	Análise dos resultados do desafio	- 108 -
6.	Conclusão.....	- 110 -
6.1.	Considerações finais.....	- 110 -
6.2.	Limitações	- 112 -
6.3.	Trabalho futuro.....	- 113 -
7.	Bibliografia	- 115 -
	Anexos.....	- 119 -
	Anexo I.....	- 120 -
	Anexo II	- 123 -
	Anexo III	- 130 -
	CD-ROM.....	- 131 -

Índice de figuras

Figura 1 - Exemplo de um PDA WM 5 (HTC P3300).....	- 8 -
Figura 2 - Exemplo de um PDA WM 6 (HTC Diamond)	- 8 -
Figura 3 - Biblioteca de classes da .NET CF (retirado de (Wigley, Wheelwright, Burbidge, MacLoed, & Sutton, 2003))	- 4 -
Figura 4 - Arquitectura da plataforma .NET CF (adaptado de Rubin & Yates (2003)).....	- 6 -
Figura 5 - API Javascript. Adaptado de (Hoffman-Andrews, 2008).....	- 9 -
Figura 6 - API Mapas estáticos. Adaptado de (Hoffman-Andrews, 2008)	- 9 -
Figura 7 - Cota de mercado de Servidores. Agosto 1995 - Dezembro 2007. (wss, 2007)	- 11 -
Figura 8 - Exemplo do IntelliSense no Visual Studio 2005	- 16 -
Figura 9 - Exemplo de integração com o M.S. SQL Server 2005 Mobile Edition.....	- 17 -
Figura 10 - Exemplo da listagem de erros e avisos	- 17 -
Figura 11 - Fases do modelo incremental (Fernandes, 2000)	- 21 -
Figura 12 - Processo de desenvolvimento usando a Prototipagem (adaptado de Pressman, 2001)- 22 -	
Figura 13 - Diagrama de Use Cases da aplicação PubMovel na fase de levantamento de requisitos.	- 24 -
Figura 14 - Diagrama de Use Cases da aplicação “NewsMobile” na fase de levantamento de requisitos.	- 25 -
Figura 15 - Diagrama de Use Cases da aplicação “NewsMobile” no módulo “administração” na fase de levantamento de requisitos.....	- 26 -
Figura 16 - Exemplo da utilização do modo <i>Full-Screen</i>	- 27 -
Figura 17 - Exemplo da utilização do <i>TabControl</i>	- 28 -
Figura 18 - Esquema de painéis de áreas	- 28 -
Figura 19 - Exemplo da aplicação "NewsMobile".....	- 29 -
Figura 20 - Replicação de base de dados da aplicação.....	- 30 -
Figura 21 - Mapa de navegação (PDA).....	- 32 -
Figura 22 - Mapa de navegação (NewsMobile – modo utilizador).....	- 32 -
Figura 23 - Mapa de navegação (NewsMobile - modo Admin).....	- 33 -
Figura 24 - Diagrama E-R da primeira base de dados (PubMovel).	- 34 -
Figura 25 - Diagrama E-R da segunda base de dados (PubMovel).....	- 35 -
Figura 26 - Diagrama E-R da base de dados (NewsMobile).....	- 36 -
Figura 27 - Caixa de diálogo de um novo projecto usando o MS Visual Studio 2005	- 38 -
Figura 28 - Ambiente de desenvolvimento no MS Visual Studio 2005 SP1	- 38 -
Figura 29 - Modo código no MS Visual Studio 2005	- 39 -
Figura 30 - Formulário Form1.cs da aplicação PubMovel.	- 41 -
Figura 31 - Formulário MenuAct.cs.....	- 42 -
Figura 32 - Formulário ServerData.cs	- 43 -
Figura 33 - Exemplo do Formulário XML.cs em modo Actualizar Notícias.....	- 44 -
Figura 34 - Exemplo do Formulário XML.cs em modo Actualizar Base de dados.	- 44 -
Figura 35 - Botão de apagar no formulário XML.cs	- 44 -
Figura 36 - Exemplo do formulário FormPub.cs na aplicação PubMovel	- 45 -
Figura 37 - Exemplo do formulário FormPubTipos.cs	- 46 -

Figura 38 - Exemplo do formulário FormPubPoi.cs	- 47 -
Figura 39 - Exemplo do formulário FormPubPoiEscolha.cs	- 48 -
Figura 40 - Exemplo do formulário FormPubNoticia.cs.....	- 48 -
Figura 41 - Exemplo do formulário FormTipo.cs	- 49 -
Figura 42 - Exemplo do formulário POI.cs.....	- 50 -
Figura 43 - Diferentes estados da checkBox do formulário PoiResult.cs	- 51 -
Figura 44 - Exemplo dos diferentes modos de visualização do PoiResult.cs	- 51 -
Figura 45 - Exemplo do formulário Favorito.cs.....	- 52 -
Figura 46 - Exemplo do formulário AdicionarFav.cs	- 53 -
Figura 47 - Exemplo do formulário AddTipo.cs.....	- 54 -
Figura 48 - Exemplo do formulário TiposExistentes.cs.....	- 55 -
Figura 49 - Exemplo dos diferentes modos de visualização do formulário AddFavDetalhe.cs	- 55 -
-	
Figura 50 - Exemplo da edição de Tipos.....	- 57 -
Figura 51 - Exemplo da edição de favoritos.....	- 57 -
Figura 52 - Exemplo de edição do tipo no formulário EditFav.cs.	- 58 -
Figura 53 - Exemplo de edição do nome do favorito no formulário EditFav.cs.	- 58 -
Figura 54 - Exemplo do formulário RemovFav.cs.....	- 58 -
Figura 55 - Exemplo da eliminação de um tipo no formulário RemoveFavTipoFav.cs.	- 59 -
Figura 56 - Exemplo da eliminação de um favorito no formulário RemovFavTipoFav.cs.....	- 59 -
Figura 57 - Exemplo do FormTipo.cs em modo de eliminação de favoritos.	- 60 -
Figura 58 - Criação de uma base de dados no MS Visual Studio 2005	- 63 -
Figura 59 – Exemplo da criação de uma tabela.....	- 64 -
Figura 60 – Exemplo de edição da tabela.....	- 64 -
Figura 61 - Exemplo da página de apresentação da aplicação NewsMobile.....	- 66 -
Figura 62 - Exemplo da página AddTipoReg.php.	- 67 -
Figura 63 - Exemplo da página AddTipo.php.....	- 68 -
Figura 64 - Exemplo da página RegUsername.php.....	- 69 -
Figura 65 - Exemplo de erro da página RegUsername.php.....	- 70 -
Figura 66 - Exemplo da página Main.php.....	- 71 -
Figura 67 - Exemplo da página Conta.php.....	- 72 -
Figura 68 - Exemplo de erro na comparação de passwords.	- 73 -
Figura 69 - Exemplo de erro na falta de preenchimento de campos.	- 73 -
Figura 70 - Exemplo de erro na falha da introdução da password actual.....	- 73 -
Figura 71 - Exemplo da página MudarPwd.php.....	- 74 -
Figura 72 - Exemplo da página DetalhesConta.php.....	- 75 -
Figura 73 - Primeira fase de eliminação da conta na página DelConta.php.....	- 76 -
Figura 74 - Segunda fase de eliminação da conta na página DelConta.php.....	- 77 -
Figura 75 - Exemplo da página Publicidade.php.	- 78 -
Figura 76 - Exemplo da página InsPub.php	- 79 -
Figura 77 - Exemplo da falha de campos preenchidos na página InsPub.php	- 79 -
Figura 78 - Exemplo do aviso de data posterior na página InsPub.php	- 79 -
Figura 79 - Exemplo da página VerPub.php.	- 80 -
Figura 80 - Exemplo da página DeletePub.php.....	- 82 -
Figura 81 - Exemplo da página Noticia.php.....	- 83 -

Figura 82 - Exemplo da página ActPub.php.	- 84 -
Figura 83 - Exemplo da página Index.php em modo Administrador.	- 86 -
Figura 84 - Exemplo da página Main.php em modo administrador.	- 87 -
Figura 85 - Exemplo da página Limpeza.php.	- 88 -
Figura 86 - Exemplo da utilização aplicação phpMyAdmin.	- 89 -
Figura 87 - Visualização das tabelas da base de dados usando o phpMyAdmin.	- 89 -
Figura 88 - Edição das tabelas usando o phpMyAdmin.	- 89 -
Figura 89 - Gráfico representativo da idade dos utilizadores entrevistados.	- 101 -
Figura 90 - Gráfico representativo da utilização das tecnologias.	- 101 -
Figura 91 - Gráfico representativo do local de utilização das tecnologias.	- 102 -
Figura 92 - Gráfico representativo da característica "Adequação do vocabulário".	- 102 -
Figura 93 - Gráfico da característica "Facilidade de uso" para as questões 2, 4, 5, 7 e 8.	- 104 -
Figura 94 - Gráfico da característica "Facilidade de uso" para as questões 3 e 6.	- 105 -
Figura 95 - Gráfico da característica "Funcionalidade" para as questões 9 e 10.	- 106 -
Figura 96 - Gráfico dos resultados do desafio 1.	- 109 -
Figura 97 - Gráfico dos resultados do desafio 2.	- 109 -

Índice de tabelas

Tabela 1 - Características do problema.....	- 19 -
Tabela 2 - Tabelas descritiva das funções presentes na classe Database.cs.....	- 63 -
Tabela 3 - Primeiro grupo de questões.....	- 98 -
Tabela 4 - Segundo grupo de questões.....	- 99 -
Tabela 5 - Terceiro grupo de questões.	- 99 -
Tabela 6 - Escala utilizada em cada questão.....	- 100 -
Tabela 7 - Resultados das questões para a característica "Adequação do vocabulário"	- 103 -
Tabela 8 - Resultado das questões associadas à característica "Facilidade de uso".....	- 103 -
Tabela 9 - Resultado das questões associadas à característica "Funcionalidade".....	- 106 -

Índice de Exemplo de código

Exemplo de código 1 - Código que permite fechar a aplicação PubMovel.	- 42 -
Exemplo de código 2 - Código que permite tornar <i>Enable</i> os restantes botões.	- 43 -
Exemplo de código 3 - Classe Program, responsável por iniciar a aplicação.	- 61 -
Exemplo de código 4 - Criação do nodo raiz.	- 91 -
Exemplo de código 5 - Ciclos de obtenção de registos.	- 91 -
Exemplo de código 6 - Exemplo da criação do ficheiro XML.	- 91 -
Exemplo de código 7 - Criação de ficheiros.	- 92 -
Exemplo de código 8 - Leitura do ficheiro XML.....	- 93 -

1. Introdução

Com o constante crescimento e evolução das tecnologias de informação, as empresas têm ao seu dispor um elevado número de ferramentas que devem tirar partido para o respectivo desenvolvimento no sector em que se inserem. Como é conhecido, muitas vezes o crescimento de uma empresa envolve uma boa estratégia de anúncios e de divulgação publicitária, fazendo chegar, de uma maneira não agressiva, toda a informação ao seu público-alvo, levando-o a conhecer e a interessar-se pelos produtos ou serviços comercializados.

Neste contexto, o desenvolvimento deste trabalho pretende apresentar uma solução que permita a empresas, que têm a necessidade de anunciar produtos ou serviços, diminuir o custo envolvido na publicidade fazendo uso das tecnologias de informação. Para além disso, a solução proposta nesta dissertação permite ainda, de um modo eficiente, orientar toda a informação divulgada para o respectivo público-alvo sem que acabe por ser considerada abusiva por parte de outras pessoas.

Com o intuito de tornar o trabalho investido na publicidade mais rentável é crucial atrair a atenção do cliente. Nenhuma das formas existentes para a publicidade, tais como, os jornais, a televisão, a rádio, o e-mail e os anúncios *on-line* são capazes de oferecer uma forma de publicidade individualizada compatível com as necessidades individuais de cada um. No entanto, com a introdução da publicidade móvel, o custo de adequar a publicidade ao público-alvo torna-se viável.

O grande objectivo é a entrega de publicidade ao público-alvo quando este precisa dela e quando este a quer receber, evitando assim a visualização ininterrupta dos mesmos anúncios e spots publicitários, impedindo que sejam consultadas as novidades que vão surgindo na mesma área de actividade.

Para isso, esta dissertação propõe analisar a introdução de duas aplicações para uma utilização rápida e genérica de publicidade: uma aplicação *Web* para o acesso principal à base de dados, permitindo que anunciantes se registem e façam uso da base de dados para realizar a difusão de notícias, que são encaradas como anúncios publicitários; e

uma aplicação móvel destinada à leitura das notícias, que permite gerir favoritos e realizar a criação de perfis de notícias.

1.1. Descrição do problema

Atendendo a que o fluxo de informação é, nos dias de hoje, incalculável, é necessário dispor de algum tipo de método de gestão de informação para que seja possível agrupá-la e torná-la mais acessível. A falta de um mecanismo eficiente para a manipulação da informação leva, muitas vezes, os profissionais de marketing a apostar na quantidade e na diversidade, fazendo com que o objectivo principal da publicidade seja perdido e não percebido por outros, fazendo assim com que seja descartada sem ser devidamente entendida.

Recolher e classificar essa informação poderá ser considerado como algo bastante complicado pois cada pessoa tem um perfil de preferências distinto e a sua classificação pode não se aplicar ao perfil de outra pessoa.

No caso particular de publicidade móvel, segundo Hristova & O'Hare, (2004), existem duas questões que por si só levantam problemas no desenvolvimento de um mecanismo de distribuição de notícias móveis. A primeira refere o direito de cada um em não ser incomodado, e a segunda envolve intrusões aborrecidas e não desejadas na vida de cada um, isto é, cada um tem o direito de possuir algum controlo sobre a quantidade e qualidade da informação que recebe.

O problema apresentado no âmbito do caso de estudo apresentado nesta dissertação refere-se à necessidade real de introduzir um mecanismo publicitário baseado em perfis, sendo que tal mecanismo representa um passo adiante no sentido de facilitar a obtenção de resultados de qualidade sempre que um utilizador efectua uma pesquisa de acordo com as suas preferências, sem que seja, no entanto, distraído com elementos externos e abusivos.

1.2. Objectivos principais

Tendo em conta os problemas introduzidos acima, pretende-se desenvolver uma abordagem a um sistema que se caracteriza por duas aplicações, a aplicação *Web* “NewsMobile” e a aplicação móvel “PubMoveI”, que têm como objectivo facilitar a criação e manipulação de anúncios publicitários, assim como facilitar a respectiva difusão para os utilizadores finais, tendo em vista um baixo custo e uma distribuição orientada para os dispositivos móveis do tipo Pocket PC, com vista a uma utilização baseada em perfis de utilizador.

É importante notar que os perfis de utilizadores não devem ser conhecidos pelos anunciantes nem publicados para aplicações externas, com o objectivo de assegurar a privacidade dos utilizadores.

Para além disso, deverá ser possível aliar às descrições dos anunciantes e dos seus anúncios um conjunto de imagens de mapas GPS para possibilitar a visualização da localização e mesmo do percurso a seguir, tendo em vista o aumento da qualidade dos dados fornecidos ao utilizador.

É igualmente necessário assegurar que, de acordo com Heinonem & Strandvik(2006) e Dickinger & Haghirian(2004), a utilização das aplicações permita que os utilizadores finais não tenham a necessidade fazer uso do tradicional e-mail e que não considerem os anúncios publicitários intrusivos e não desejadas.

Por fim, a aplicação móvel “PubMoveI” deverá permitir a introdução de dados pessoais por parte do utilizador, de modo a construir um perfil personalizado que seja materializado numa base de dados pessoal contendo os respectivos “favoritos”, permitindo, deste modo, categorizar todas as suas preferências de modo a filtrar a informação que receberá no futuro.

1.3. Motivação

Uma das principais motivações para a realização deste trabalho foi a possibilidade de aprender a utilizar e integrar diferentes tecnologias da informação, que se encontram em

grande expansão, e que se prevê que no futuro sejam largamente utilizadas. A integração entre tecnologias *Web* e tecnologias móveis é um desafio devido à complexidade dos protocolos, das tecnologias de desenvolvimento e dos próprios dispositivos em si, como é o caso dos dispositivos móveis do tipo Pocket PC em que é necessário ter em conta um conjunto de limitações que incluem a dimensão dos ecrãs e restrições a nível de memória e de poder computacional.

Para além disso, foi igualmente motivadora a aprendizagem de uma nova linguagem de programação, a linguagem C#, bem como a respectiva integração com a Framework .NET CF. Igualmente desafiante foi a integração de outras tecnologias de desenvolvimento *Web*, tais como, o PHP, o HTML, o CSS e ainda a integração entre diferentes bases de dados desenvolvidas em MySQL e em SQL Server Mobile.

Por fim, o desenvolvimento de um conjunto de aplicações concretas com vista a uma utilização real e realmente útil constituiu o grande objectivo na realização deste trabalho.

1.4. Metodologia

A metodologia empregue no desenvolvimento deste trabalho consistiu num conjunto de fases que se iniciaram pelo levantamento de requisitos, que permitiu compreender as dificuldades sentidas pelos utilizadores e formular os objectivos gerais do projecto.

Assim, foram realizadas pesquisas sobre as tecnologias existentes actualmente para a divulgação de notícias publicitárias, incluindo os *feeds* RSS¹, a utilização de mapas GPS, a publicidade recorrendo à utilização da rádio, televisão, jornais e *outdoors*, de modo a permitir comparar e reutilizar componentes essenciais para a criação do projecto. A fase de pesquisa teve como objectivo principal a realização de uma análise de requisitos de modo a definir quais seriam as funcionalidades pretendidas, o nível de desempenho exigido e outros aspectos relacionados com a interface do utilizador.

Foi ainda necessário definir o tipo de metodologia de engenharia de software a seguir, tendo sido seleccionados o modelo incremental e a prototipagem, devido às constantes

¹ RSS - Really Simple Syndication

mutações do projecto e ao conhecimento que se ia adquirindo com o decorrer das actividades de desenvolvimento. A utilização deste dois modelos permitiu que se fosse aumentando a diversidade e completando, cada vez mais, o projecto de modo a que este fosse implementado de acordo com as necessidades reais dos utilizadores.

Em paralelo, foi necessário analisar e aprofundar o estudo das características das tecnologias de computação móvel e das tecnologias *Web*, bem como das respectivas ferramentas de desenvolvimento, incluindo ainda a análise de mecanismos de replicação de bases de dados. Para tal, foi realizado um estudo sobre o desenvolvimento de aplicações móveis para Pocket PC e de aplicações *Web*, bem como da literatura específica que se refere no texto desta dissertação.

Em seguida, tratou-se de desenvolver e implementar as especificações das aplicações, determinando-se os algoritmos a desenvolver, assim como as respectivas regras e procedimentos, um trabalho que consiste na tradução das especificações para uma linguagem de programação.

A seguir identificaram-se os métodos de avaliação a aplicar, e desenvolveu-se um conjunto de testes sob a forma de uma experiência com utilizadores reais, com o intuito de testar a interface e as funcionalidades proporcionadas pelas aplicações, a lógica dos programas e o respectivo funcionamento externo junto dos utilizadores finais. Esta fase concretizou-se com a realização da avaliação das aplicações através de um estudo empírico que consistiu na utilização das aplicações para realizar um conjunto de desafios de modo a poderem testar as funcionalidades proporcionadas através da realização de pequenas tarefas de pesquisa e obtenção de informação, mediante o posterior preenchimento de um questionário que foi respondido pelos participantes que testaram a aplicação. A realização da experiência pretendeu igualmente avaliar a eficiência da utilização das aplicações comparando-as com outros tipos de tecnologias existentes actualmente.

Por fim, tratou-se de recolher e analisar os dados relacionados com as experiências realizadas e apurar um conjunto de conclusões sobre a utilidade das aplicações no que diz respeito ao preenchimento das necessidades dos utilizadores.

1.5. Estrutura do documento

Este documento encontra-se organizado em 6 capítulos com os seguintes conteúdos.

No **Capítulo 1** faz-se uma breve introdução ao tema da dissertação de mestrado, incluindo uma descrição do problema, os objectivos do trabalho, a metodologia aplicada e as respectivas limitações.

No **Capítulo 2** apresenta-se uma introdução às tecnologias de computação móvel, bem como às ferramentas de desenvolvimento utilizadas para o desenvolvimento dos protótipos de ambas as aplicações. Apresentam-se ainda as plataformas de desenvolvimento e os controlos utilizados na implementação das aplicações.

O **Capítulo 3** descreve a análise da aplicação, identificando o problema abordado e descrevendo o modelo de engenharia de *software* aplicado. Identifica-se ainda o conjunto de mapas de navegação das aplicações.

O **Capítulo 4** descreve a implementação associada ao desenvolvimento dos requisitos identificados no capítulo anterior, indicando igualmente o método de criação e replicação das bases de dados. Neste capítulo faz-se ainda uma descrição do conjunto de detalhes que conduziu à criação dos ecrãs das aplicações.

O **Capítulo 5** descreve o estudo empírico que serviu de base à avaliação das aplicações desenvolvidas. Este capítulo descreve os métodos de avaliação utilizados, analisa os dados recolhidos e relata os resultados obtidos durante a realização dos testes realizados por um conjunto de utilizadores finais.

No **Capítulo 6** são apresentadas as conclusões finais decorrentes da realização do trabalho, apontando os aspectos a serem desenvolvidos no futuro.

2. Tecnologias para o desenvolvimento do trabalho

Os actuais avanços tecnológicos, têm levado as empresas a reavaliar a respectiva postura quanto à sua localização, métodos de pesquisa e marketing utilizados, infraestrutura e posições a adoptar no futuro próximo relativamente à sua indústria.

Por isso a utilização das tecnologias de informação nas áreas da indústria e comércio já não é hoje vista como um avanço tecnológico, mas sim como uma necessidade, para que essas áreas possam avançar e permanecer competitivas em relação a outras.

A utilização de dispositivos móveis, juntamente com a liberdade que facilitam, permite que se explore um novo mercado de aplicações que possam proporcionar melhorias de funcionamento ao nível da gestão de informação e do respectivo processamento. Os dispositivos móveis mais comuns actualmente são os PDA ².

Ao analisar as características de um PDA, encontramos dispositivos de dimensões reduzidas com um poder computacional igualmente reduzido, mas com uma liberdade móvel muito superior à que é proporcionada por computadores portáteis. Actualmente, os PDA integram igualmente as funcionalidades de um telefone celular. Para além disso, possuem um sistema operativo, um ecrã táctil que facilita a sua utilização e estão normalmente equipados com tecnologias de transferências de dados tais como os infra-vermelhos, o *Bluetooth* e o *Wi-Fi*.

2.1. Tecnologias para o desenvolvimento de aplicações móveis

Neste trabalho foram desenvolvidas duas aplicações que permitem facilitar o fluxo de informação desejada entre as empresas e o seu público-alvo. A primeira, designada por “**PubMove!**” foi desenvolvida em C# .NET CF ³ com o recurso a duas bases de dados desenvolvidas em SQL Server CE ⁴ com o objectivo de ser instalada em dispositivos móveis do tipo PDA’s. A segunda aplicação, designada por “**NewsMobile**” foi

² PDA – *Personal Digital Assistants*

³ CF – Compact Framework

⁴ SQL Server CE – SQL Server 2005 Compact Edition

desenvolvida em PHP com recurso a uma base de dados desenvolvida em MySQL, tendo como objectivo ser disponibilizada através da Web.

2.1.1. Plataforma Windows Mobile

De acordo com a Microsoft(2008), um dispositivo móvel que corra uma versão do sistema operativo Windows Mobile ou WM permite ao utilizador fazer chamadas telefónicas, receber e enviar e-mails, manter-se a par do seu calendário, contactos, navegar na Internet, enviar e receber mensagens de texto e, ainda, compor ficheiros de texto no Microsoft Word, folhas de cálculo no MS Excel ou apresentações no MS PowerPoint. Para além disso, permite ainda que o utilizador desenvolva ou compre *software* desenvolvido por outras empresas de modo a poder adaptar o dispositivo às suas necessidades.

Actualmente existem três modelos do WM: o *Professional*, o *Standard* e o *Clássico*, existindo duas grandes diferenças entre os três modelos. A mais visível é a possibilidade de o dispositivo móvel possuir um ecrã táctil nas versões *Professional* e *Clássico*, enquanto que no modelo *Standard* tal não é possível. A outra grande diferença é que o WM Clássico é destinado a dispositivos sem a funcionalidade de telefone celular.

Neste momento, a última versão do WM lançada pela Microsoft é a versão 6 e, segundo a própria Microsoft, as aplicações desenvolvidas para a versão WM 5.0 deverão correr em WM 6.



Figura 1 - Exemplo de um PDA WM 5 (HTC P3300)



Figura 2 - Exemplo de um PDA WM 6 (HTC Diamond)

A Figura 1 e Figura 2 mostram dois modelos de PDA's com duas versões de WM diferentes, no entanto compatíveis entre si.

De acordo com Fox & Dox(2003), um conceito importante a ter em mente é que o Windows Mobile não é apenas um S.O.⁵ mas sim uma plataforma. Essencialmente, uma plataforma é constituída pela combinação de um conjunto de *hardware*, programas de aplicação, módulos, componentes da interface do utilizador e um sistema operativo. É precisamente a plataforma que define as características principais de uma classe de dispositivos, permitindo que os fabricantes e os engenheiros de software possam mais facilmente criar soluções à medida para esses dispositivos.

As características que presidiram a concepção do SO WM 5 e WM 6 foram, de acordo com Yao & Durant(2003), as seguintes:

- Pequena dimensão
- Portátil
- Modular
- Compatibilidade com a API Win32
- Conectividade
- Processamento em tempo real

Para além daquelas, o S.O. WM apresenta ainda as seguintes características de *software* básicas:

- Possui um ecrã *Today*, que pode ser encarado como um ambiente de trabalho em relação aos PC's Desktop. Nele pode apresentar-se um grande conjunto de informações sobre a data, a hora, os vários tipos de mensagens lidas e não-lidas, as tarefas e os compromissos do utilizador.
- Inclui versões reduzidas das aplicações do MS Office para WM tais como o MS Word, o MS Excel e o MS Powerpoint. Inclui ainda uma versão do IE⁶ para navegação e visualização de páginas *Web* e um leitor de conteúdos multimédia (Windows Media Player).

⁵ S.O. – Sistema Operativo

⁶ IE – Internet Explorer

- Para sincronismo de dados, o S.O. WM inclui o serviço *ActiveSync* que suporta ligações USB 2.0 ou *Bluetooth* entre um dispositivo móvel e um PC.

Em termos do suporte ao hardware, o WM inclui a algumas características importantes, tais como as seguintes:

- Suporte para *hardware* de armazenamento. Note-se que nem sempre é possível armazenar todas as aplicações na memória principal do PDA. De facto, é necessário possuir algum tipo de expansão de memória como é o caso dos cartões SD⁷, MiniSD, MicroSD e os SDHC⁸. Estes cartões permitem o armazenamento de ficheiros, contendo imagens, áudio, vídeo e também de ficheiros executáveis.
- Armazenamento em memória persistente, o que significa que a RAM no WM é usada apenas para executar aplicações, sendo os dados armazenados em ROM. Assim, o WM permite que os dados fiquem protegidos, mesmo que ocorra uma descarga total da bateria do dispositivo móvel.
- Suporte para USB 2.0 permitindo uma transferência de dados mais rápida. Esta funcionalidade é necessária quando se utilizam cartões de memória com grande capacidade e se lida com ficheiros de conteúdos multimédia, como os que se indicou acima.
- Suporte para os protocolos de comunicação sem fio mais comuns, tais como o *Wi-Fi*, o *Bluetooth* e os Infra-vermelhos.

2.1.1.1. *Classes de aplicação .NET CF*

Do ponto de visto do programador a parte interessante da .NET CF é o conjunto de bibliotecas de classes. Na realidade, a enorme selecção de classes e de API's constitui a fundação de qualquer aplicação .NET, uma vez que, proporcionam uma camada de abstracção entre os serviços do S.O. e o programador, permitindo adicionar funcionalidades de uma forma consistente independentemente da linguagem de programação ou da arquitectura do sistema. (Wigley, Wheelwright, Burbidge,

⁷ SD – Secure Digital

⁸ SDHC – Secure Digital High Capacity

MacLoed, & Sutton, 2003). A Figura 3 esquematiza a organização das classes da .NET CF.

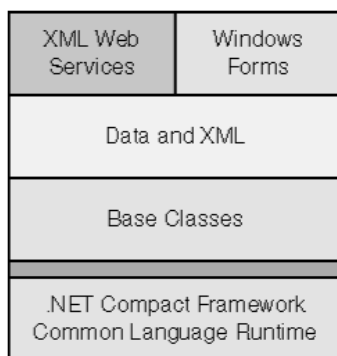


Figura 3 - Biblioteca de classes da .NET CF (retirado de (Wigley, Wheelwright, Burbidge, MacLoed, & Sutton, 2003))

De acordo com Wigley et al. (2003) e Miranda(2007) a organização das classes na plataforma .NET CF é composta do seguinte modo:

- **Classes Base**

Contém os tipo de dados básicos, colecções de classes tais como filas, *arrays* e *hash tables*, classes para I/O⁹, classes que suportam a segurança da arquitectura .NET CF e classes para a globalização das aplicações.

- **Dados e XML**

Contém as classes ADO.NET e as classes para formatação e *parsing* de dados XML.

- **XML Web Services**

Contém as classes que suportam os serviços Web em XML.

- **Windows Forms**

Contém as classes que suportam o desenvolvimento de interfaces gráficas do utilizador.

⁹ I/O – Input/Output

2.1.1.2. *Arquitectura da plataforma .NET CF*

A plataforma de desenvolvimento .NET CF é constituída por dois componentes fundamentais: o *common language runtime*(CLR) e a biblioteca de classes.

O CLR é o componente mais importante da NET CF, sendo responsável pelo tratamento do código no tempo de execução do programa, executando tarefas importantes como a gestão da memória e gestão de *threads*, enquanto assegura a segurança do código e respectiva optimização. (Roof & Fergus, 2003).

Usando o compilador JIT ¹⁰, faz-se a conversão das instruções IL ¹¹ para código máquina tornando-as compatíveis com o processador específico da máquina onde a aplicação está a correr.

De facto, fazendo uma análise mais profunda, a .NET CF funciona como uma máquina virtual, o que faz com que as aplicações sejam independentes da arquitectura do sistema.

A biblioteca de classes .NET CF é uma colecção de classes que podem ser facilmente reutilizadas facilmente quando se desenvolve uma aplicação. Em essência, estas classes são compostas por ficheiros DLL ¹² sendo delas que se retiram as funcionalidades pretendidas para as aplicações desenvolvidas. (Miranda, 2007)

Na plataforma de desenvolvimento .NET CF distingue-se ainda entre ***Managed Code*** e ***Native Code***. O *managed code* refere-se ao código que é controlado pelo CLR, e que é verificado e controlado a nível de erros de sintaxe e de apontadores de memória (*Garbage Collector*). Por outras palavras, trata-se do código que é desenvolvido pelos programadores. O *native code* refere-se ao código que não é controlado nem verificado, isto é, quando não existe nenhum tipo de gestão de memória. Este último é por vezes referido como código Win32, porque é específico e depende da arquitectura da máquina onde corre.

¹⁰ JIT – Just In Time.

¹¹ IL – Intermediate language.

¹² DLL – Dynamic link library.

Segundo (Rubin & Yates, 2003), a arquitectura .NET CF é constituída por três camadas de tecnologias: o *Common Language Runtime* (CLR), o compilador *Just-In-Time* (JIT) e o sistema operativo Windows CE. O diagrama de blocos da Figura 4 descreve como uma aplicação interage com a .NET CF e as suas respectivas camadas.

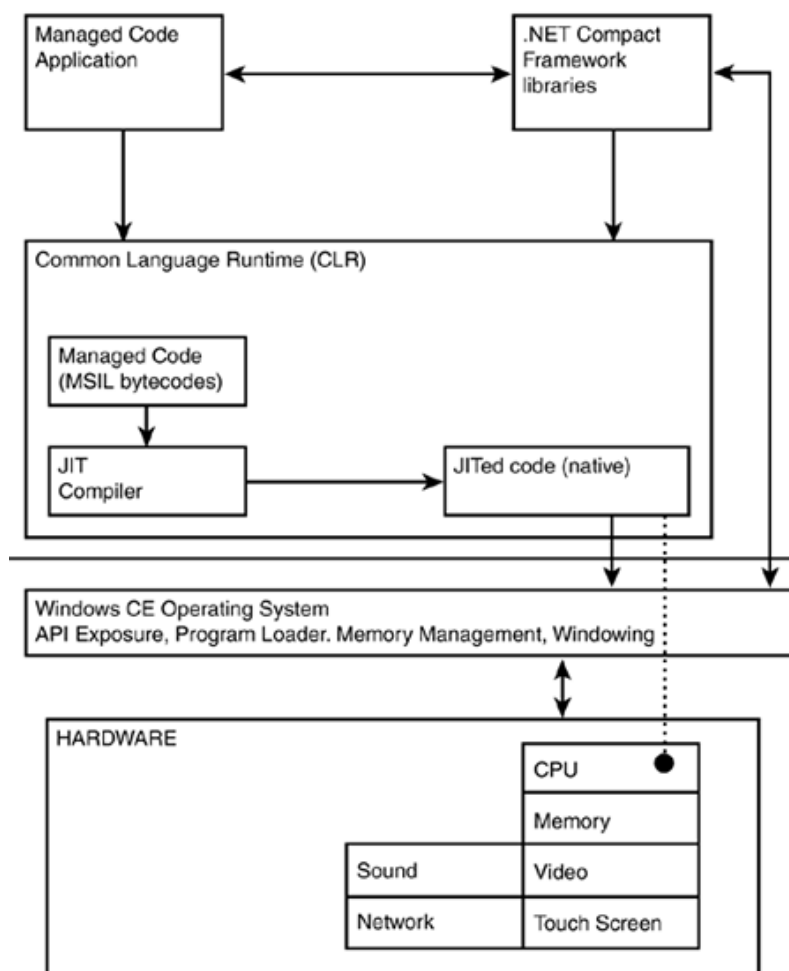


Figura 4 - Arquitectura da plataforma .NET CF (adaptado de Rubin & Yates (2003))

Como se pode observar na Figura 4, a primeira camada da arquitectura é a camada mais baixa, composta pelo *hardware*. Por outro lado todo o hardware existente no dispositivo móvel é controlado pela segunda camada, o S.O. Windows CE.

O Windows CE é, pois, responsável por efectuar a gestão de memória, o carregamento e a execução dos programas.

Por fim, o programa é executado e colocado em memória fazendo uso do CLR e das bibliotecas de classes.

2.1.1.3. Limitações

Durante a fase de implementação do projecto descrito neste trabalho, tornou-se evidente que a plataforma .NET CF¹³ apresenta algumas limitações quando comparada com a plataforma .NET completa ou mesmo quando comparada com os módulos e classes disponíveis na linguagem PHP.

Na verdade, na plataforma .NET CF, é possível executar, ou criar, quase todas as funcionalidade pretendidas. A diferença reside apenas na quantidade de classes que estão presentes na .NET CF. Isto sucede precisamente porque, segundo Wigley, et al (2003), a biblioteca de classes da .NET CF é composta por menos de 50% das classes existentes na plataforma .NET completa.

No entanto, mesmo não tendo todas as classes disponíveis, ou mesmo os métodos mais conhecidos, o programador consegue sempre encontrar uma alternativa baseada nas operações mais básicas.

A razão pela qual existe uma limitação de 50% nas classes disponíveis na .NET CF, torna-se evidente quando se toma em consideração que para um dispositivo móvel, não tem muito interesse disponibilizar métodos redundantes, ou pouco interessantes, para o tipo de dispositivo, sendo igualmente importante notar que existe um reduzido espaço de armazenamento disponível.

2.1.2. Mobile Device Center

O *Mobile Device Center* é a actualização do *ActiveSync* disponibilizada pela Microsoft no Windows Vista. Trata-se de um programa de sincronização mais moderno, mais completo e mais rápido, mantendo, no entanto um conceito de sincronização de dados muito semelhante ao seu antecessor.

¹³ CF – Compact Framework.

O *Mobile Device Center* permite sincronizar dados entre um PC¹⁴ e um dispositivo móvel, criando um perfil único por dispositivo. Essa sincronização pode ser realizada através de uma ligação *USB*¹⁵, *Bluetooth*¹⁶ ou, ainda por infra-vermelhos.

2.1.3. *Google Maps API*

A API do *Google Maps* é uma ferramenta que foi disponibilizada pela Google em 2005, e que tinha como principais características a pesquisa de endereços, locais públicos, ajuda na recolha de rotas, Geocoding¹⁷, visão de trânsito e visão de satélite.

Numa primeira fase, esta API foi criada usando exclusivamente Javascript.

No entanto em 2008, foram criadas mais duas novas formas de obtenção de mapas:

A API¹⁸ de mapas estáticos (*Static Maps API*) foi disponibilizada em Fevereiro de 2008 e a API de mapas para Flash (*Maps API for Flash*) foi disponibilizada em Maio de 2008.

A API de mapas estáticos teve uma boa aceitação por parte da comunidade de programadores de dispositivos móveis. A razão pela qual se verificou uma forte adesão a esta API é essencialmente, o espaço em memória que o mapa requer, passar a ser muito mais pequeno, assim como a quantidade de pedidos *http* que passou a ser muito menor. Este facto, aliado a uma boa capacidade de serviços, tais como o *zoom*, a utilização de múltiplos marcadores com diferentes cores, a possibilidade da criação de *polylines*¹⁹ com 50 pontos, e existência de auto zoom levou a que a API se tornasse muito útil para os programadores de aplicações móveis.

Comparando ambas as tecnologias mencionadas, as figuras Figura 5 e Figura 6 permitem verificar que, usando o Javascript obteve-se um tamanho total de 344 KB, que

¹⁴ PC – Personal Computer.

¹⁵ USB – Universal Serial Bus.

¹⁶ Bluetooth – Protocolo de comunicação de redes sem fio.

¹⁷ Geocoding ou Geocodificação é o processo de assinalar uma coordenada espacial ou latitude e longitude a um endereço. (InGlobe, 2006).

¹⁸ API - Application Programming Interface.

¹⁹ Polyline – Linha continua composta por um ou mais segmentos de recta. Adaptado de: (What is Polyline?, 2003).

demorou 4.45s, e que efectuou 18 pedidos *http*. Por outro lado, quando se utiliza a API de mapas estáticos, resulta num único pedido *http* que demorou 227 ms, e que apenas ocupou 66KB de espaço de armazenamento.

Request	Status	Source	Size	Duration
GET map-simple.hl	200 OK	code.google.com	615 B	227ms
GET maps?file=api	200 OK	maps.google.com	4 KB	236ms
GET main.js	200 OK	maps.google.com	75 KB	565ms
GET transparent.pi	200 OK	maps.google.com	95 B	319ms
GET poweredby.png	200 OK	maps.google.com	4 KB	291ms
GET vp?spn=0.040	200 OK	maps.google.com	252 B	265ms
GET mt?n=404&v=	200 OK	mt1.google.com	29 KB	237ms
GET mt?n=404&v=	200 OK	mt3.google.com	24 KB	233ms
GET mt?n=404&v=	200 OK	mt1.google.com	17 KB	638ms
GET mt?n=404&v=	200 OK	mt2.google.com	31 KB	1.03s
GET mt?n=404&v=	200 OK	mt0.google.com	30 KB	822ms
GET mt?n=404&v=	200 OK	mt2.google.com	21 KB	1.06s
GET mt?n=404&v=	200 OK	mt3.google.com	12 KB	1.26s
GET mt?n=404&v=	200 OK	mt1.google.com	29 KB	1.27s
GET mt?n=404&v=	200 OK	mt3.google.com	27 KB	1.65s
GET mt?n=404&v=	200 OK	mt0.google.com	2 KB	1.84s
GET mt?n=404&v=	200 OK	mt2.google.com	15 KB	1.84s
GET mt?n=404&v=	200 OK	mt0.google.com	30 KB	1.85s
18 requests			344 KB	4.45s

Figura 5 - API Javascript. Adaptado de (Hoffman-Andrews, 2008)

Request	Status	Source	Size	Duration
GET staticmap?cer	200 OK	maps.google.com	66 KB	227ms
1 request			66 KB	227ms

Figura 6 - API Mapas estáticos. Adaptado de (Hoffman-Andrews, 2008)

Ainda no que diz respeito à utilização de mapas, para dispositivos móveis, a velocidade também é um factor essencial. No entanto, de acordo com Hoffman-Andrews(2008), existem outros factores a ter em conta, como por exemplo o facto de a maioria dos dispositivos móveis ser incompatível com o Javascript, o seu poder computacional ser reduzido, e existir sempre alguma dificuldade no processo de mover o mapa.

Finalmente é importante notar que, para a utilização pessoal, é necessária uma chave própria que é facultada pela Google após o registo.

2.1.4. Microsoft SQL server CE

O Microsoft SQL Server 2005 CE é uma versão do Microsoft SQL Server 2005 destinada especificamente a dispositivos móveis. Trata-se de uma ferramenta de SGBD²⁰ que se encontra disponível e integrada com o Microsoft Visual Studio 2005, e que permite o acesso e gestão de bases de dados móveis.

Para se poder utilizar esta ferramenta é necessário fornecer o *namespace* (System.Data.SqlServerCe).

2.2. Tecnologias para o desenvolvimento do servidor web

Para que o projecto descrito nesta dissertação se tornasse viável foi necessário recorrer à utilização de um servidor que permitisse publicar a aplicação “NewsMobile” na *web*, uma vez que os utilizadores finais da aplicação teriam de receber informações dos anunciantes que pretendessem publicar os respectivos textos publicitários.

Para este fim, a forma escolhida para resolver esta questão passou pela utilização das linguagens HTML e PHP para a criação de um *web site* que controlasse os registos e a publicação de notícias que serão posteriormente descarregadas para o dispositivo móvel dos utilizadores mediante a sua necessidade.

Dado que a linguagem PHP é executada no servidor foi, por isso, necessário instalar um servidor que permitisse executar código PHP.

2.2.1. WAMP Server 2.0

O WAMP Server 2.0 é um ambiente de desenvolvimento Web em Windows. Este ambiente permite a criação de aplicações Web usando Apache, PHP e bases de dados MySQL. Para além disso, inclui de raiz um conjunto de ferramentas que facilitam a gestão de bases de dados, tais como o PHPMyAdmin e o SQLiteManager. (Bourdon)

²⁰ SGBD – Sistema de Gestão de Bases de Dados

A designação WAMP é um acrónimo para a combinação Microsoft Windows, Apache, MySQL e PHP, sendo que cada elemento do acrónimo WAMP disponibiliza uma plataforma de funcionalidade essencial:

- Microsoft Windows é o sistema operativo;
- Apache é o servidor web;
- MySQL é o SGBD (sistema de gestão de base de dados);
- PHP é a linguagem de scripting para desenvolvimento web orientada por objectos.

2.2.1.1. Apache

Criado em Fevereiro de 1995 por Rob McCool, o Apache tornou-se desde então um dos servidores http mais popular, tendo sido distinguido várias vezes como produto de excelência.

Este servidor é a principal tecnologia da Apache Software Foundation, que é responsável por vários projectos envolvendo tecnologia de transmissão, processamento e execução de aplicações via Web.

Actualmente o Apache continua a ser um dos servidores mais utilizados, como se pode verificar pela análise do gráfico da Figura 7.

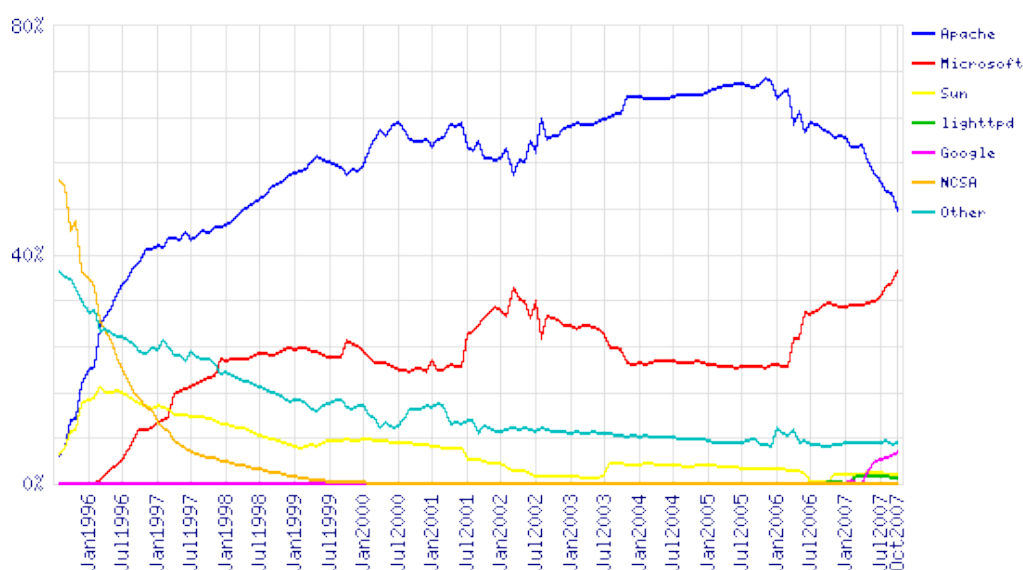


Figura 7 - Cota de mercado de Servidores. Agosto 1995 - Dezembro 2007. (wss, 2007)

É ainda importante referir que o servidor Apache é compatível com o protocolo *http* e as suas funcionalidades são mantidas através de uma estrutura de módulos, dando também a possibilidade de escrita de novos módulos ou de personalizar os módulos já existentes.

2.2.1.2. SGBD *MySQL*

O *MySQL* é um sistema de gestão de bases de dados relacionais, pelo que suporta *SQL*²¹, tratando-se de uma plataforma *open source* e sendo um dos SGBDs²² mais utilizados na actualidade.

A respectiva arquitectura é aberta, proporciona um alto desempenho e uma estabilidade que pode ser comprovada pela utilização em variados sistemas produtivos, como por exemplo os sistemas em produção da Airbus, Google, NASE, Sun Microsystems, Cisco, Nokia, Nortel e ministérios de vários países, incluindo a França, a Austrália e os EUA. Para além disso suporta ainda multiprocessamento e pode ser executada em várias multi-plataformas distintas, tais como MS Windows, Macintosh e Linux (*MySQL*, 1995-2008).

Mais especificamente, o dialecto *SQL* que suporta utiliza o standard ANSI *SQL92* e pode construir selecções entre tabelas de bases de dados diferentes.

Além disso, o suporte para aplicações é bastantes vasto, sendo desenvolvido para a maioria das linguagens de programação e permitindo desenvolver aplicações para aceder a bases de dados *MySQL* em C, C++, C#, JAVA, PERL, .NET (Neves & Ruas, 2005).

2.2.1.3. *PHP*

De acordo com Serrão & Marques(2004), o *PHP*²³ é uma linguagem de programação para o desenvolvimento de aplicações orientadas para a World Wide Web.

²¹ *SQL* - Structured Query Language

²² SGBDs – Sistema de Gestão de Bases de Dados

²³ *PHP* – *PHP*: Hypertext PreProcessor.

Em termos simples, o PHP é uma linguagem de *scripting* que é embebida no código HTML, permitindo que os programadores criem rapidamente um conjunto de páginas HTML dinâmicas. De facto, a grande parte da sua sintaxe é proveniente de linguagens de programação bastante conhecidas, tais como o C, a linguagem JAVA e o Perl (PHP, 2008).

Um script PHP é “marcado” recorrendo a *tags* próprias (<?php ?>), e apresenta aspectos distintos quando comparado com *scripts* escritos em Javascript, uma vez que o código HTML que é enviado para o cliente é gerado do lado do servidor. O cliente apenas recebe os resultados da execução do *script*.

A popularidade do PHP deve-se igualmente ao resultado da procura de um sistema funcional e de baixo custo. A sua extensibilidade e portabilidade a várias plataformas, a possibilidade de incorporar novos módulos/bibliotecas, a sua versatilidade e robustez, o facto de ser estruturado e de ter uma sintaxe similar a linguagens como C, o C++ e o PERL, a possibilidade de permitir uma integração com servidores Web, tais como o Apache e o IIS da Microsoft, e o facto de acelerar a concepção de *scripts*, tornam o PHP um adversário poderoso quando comparado com outras tecnologias. (Serrão & Marques, 2004)

2.2.2.HTML

A linguagem HTML (*Hyper Text Markup Language*) é uma linguagem de marcação utilizada para produzir páginas Web, sendo uma norma adoptada pela WWW²⁴. Esta linguagem permite, para além da formatação de texto para apresentação *web*, proporcionar comandos para a introdução de imagens, formulários, alteração de fontes, permitindo ainda desenvolver páginas multimédia que combinam informações nos mais variados formatos de texto, som, imagens e animações. As páginas HTML são interpretadas por navegadores Web, tais como, entre outros, o IE, o Firefox e o Opera.

Segundo (Ribeiro, 2004), “... a, HTML baseia-se na inserção de identificadores especiais no texto de um documento, de modo a delimitar porções de texto para as quais se pretende especificar um conjunto de propriedades, tais como a fonte do texto,

²⁴ WWW – World Wide Web.

a sua dimensão, o seu estilo e a sua cor. A utilização de marcas permite assim separar o processo de criação do conteúdo do processo utilizado para apresentar esse conteúdo. Por exemplo, no caso da linguagem HTML, a apresentação do conteúdo do documento é realizado por um programa, o browser, que interpreta e processa as marcas e as transforma em texto formatado.”

2.2.3.CSS

O CSS (*Cascading Style Sheets*) é um mecanismo utilizado para a construção do ambiente gráfico de páginas Web que permite o uso de uma técnica diferente da convencional (HTML) possibilitando uma redução no tempo de trabalho utilizando as *tags* e reaproveitando código.

Em essência, trata-se de uma linguagem de estilos que definem a disposição, ou *layout* de documentos HTML, controlando, por exemplo, fontes, cores, margens, linhas, alturas, imagens de fundo e posicionamentos (Vandré, 2007).

A utilização do CSS no projecto descrito nesta dissertação permitiu modificar a aparência dos elementos HTML/PHP para que a visualização Web do projecto se tornasse mais agradável e possuísse um aspecto profissional do ponto de vista do utilizador final.

2.3. XML

O XML (*Extensible Markup Language*) foi um projecto desenvolvido originalmente pelo consórcio W3C, e surgiu como resposta à necessidade crescente da organização de dados originalmente concebido para dar resposta aos desafios de grande escala da edição electrónica (Coelho, 2005) e (Serrão & Marques, 2004).

O XML proporciona um conjunto estruturado de “marcas” que identificam, ou descrevem, um dado conteúdo, independentemente do modo como ele vai ser apresentado, isto é, independentemente da respectiva formatação.

De facto, segundo o W3C(2008), o XML é um formato de texto muito flexível e simples que deriva do SGML – *Standard Generalized Markup Language*.

2.4. Publicação Web

Para a publicação da aplicação “NewsMobile”, desenvolvida no âmbito deste trabalho, foi necessário a utilização de um serviço que permitisse a gestão/utilização remota da interface Web da aplicação através de um endereço URL fixo.

Ora, sabendo que o endereço IP²⁵ pode mudar dinamicamente, dificultando a orientação dos utilizadores para a utilização da aplicação, optou-se por utilizar o serviço DynDNS que, quando devidamente parametrizado, permite criar um *hostname* que aponta para o IP dinâmico ou estático (DynDNS, 2008). O DynDNS proporciona igualmente um mecanismo de actualização do *hostname* com o IP dinâmico, permitindo assim uma orientação completa para a aplicação Web, permitindo utilizar sempre o mesmo URL²⁶.

2.5. Ferramentas de desenvolvimento de *software*

As ferramentas utilizadas para o desenvolvimento das aplicações descritas neste trabalho foram as seguintes: Visual Studio 2005 SP1 e o Windows Vista Notepad.

2.5.1. Visual Studio 2005 – SP1 – (C# CF .NET)

Do ponto de vista do desenvolvimento do código das aplicações de *software* descritas nesta dissertação, o Visual Studio 2005 é um ambiente integrado de desenvolvimento que permite a criação de aplicações para a plataforma .NET, fornecendo um conjunto de ferramentas para a programação de serviços Web, aplicações de *desktop* e aplicações móveis.

²⁵ IP- Internet Protocol.

²⁶ URL - Uniform Resource Locator.

Nesta plataforma, os programas são criados visualmente, adicionando-se controlos a uma janela, especificando as respectivas propriedades e as acções sobre esses controlos. Estas características evidenciam o facto de se designar por uma plataforma RAD²⁷. Para além disso tem também, totalmente integrada a plataforma .NET CF descrita anteriormente, que permite a criação de aplicações para *Pocket PC*, *SmartPhones* e outras plataformas baseadas em Windows CE.

As características principais que tornaram o Microsoft Visual Studio 2005 numa ferramenta de eleição para o desenvolvimento rápido de aplicações incluem o IntelliSense, a integração completa com o SQL Server e uma capacidade de listagem de erros/avisos e de depuração do código.

Na Figura 8 mostra-se um exemplo do funcionamento do IntelliSense, uma funcionalidade que disponibiliza sugestões de sintaxe e permite avançar o texto e completar palavras de forma automática. Esta ferramenta, embora simples, é muito prática e permite poupar muito tempo durante a escrita do código, evitando a procura de mais informações sobre a sintaxe e funcionalidade de um determinado método que se pretende utilizar no programa.

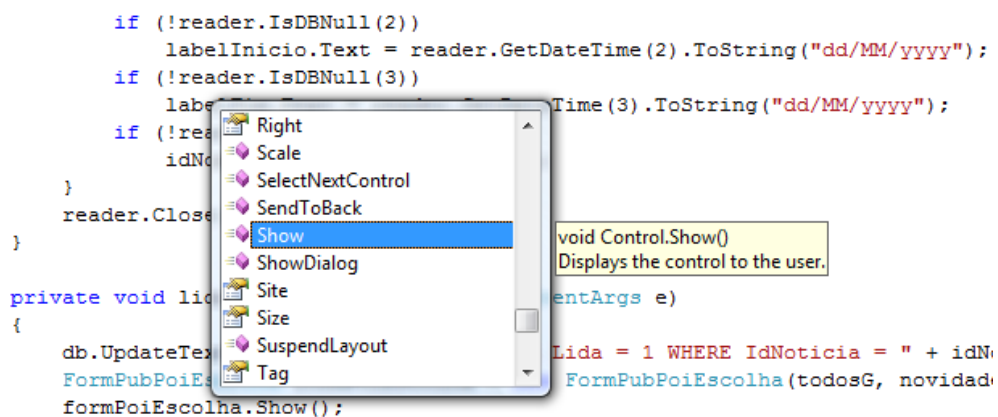


Figura 8 - Exemplo do IntelliSense no Visual Studio 2005

Para além disso, a possibilidade de interligação com bases de dados também é permitida, tal como foi referido anteriormente. A Figura 9 apresenta um exemplo do modo como o Visual Studio 2005 permite manipular e gerir, graficamente ou recorrendo a scripts SQL, toda a gestão das bases de dados.

²⁷ RAD – Rapid Application Development.

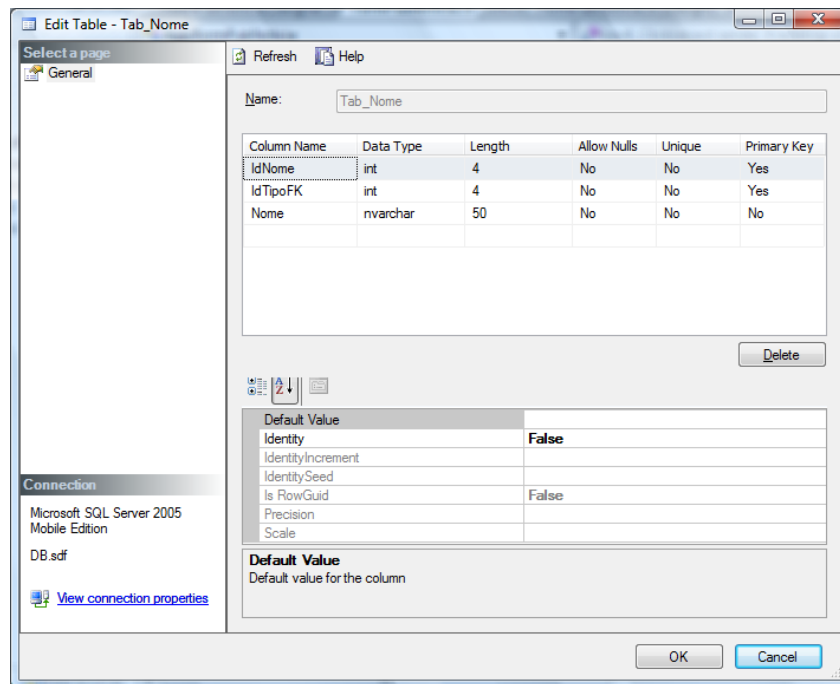


Figura 9 - Exemplo de integração com o M.S. SQL Server 2005 Mobile Edition

No entanto, mesmo dispo de ferramentas que auxiliam a escrita do código da aplicação, é sempre possível a ocorrência de erros. Neste âmbito, para uma resolução rápida e eficaz dos erros, o Visual Studio 2005 proporciona uma funcionalidades para a listagem de erros e avisos e de *debugging*. Assim todo o código é lido e verificado antes de ser executado. Caso se encontre algum erro, consegue-se localiza-lo com precisão. Esta funcionalidade permite que seja poupado muito tempo de depuração de código até encontrar o erro que por vezes é de simples correcção.

Na Figura 10 mostra-se um exemplo em que o editor indica onde se encontram presentes os erros e os avisos.

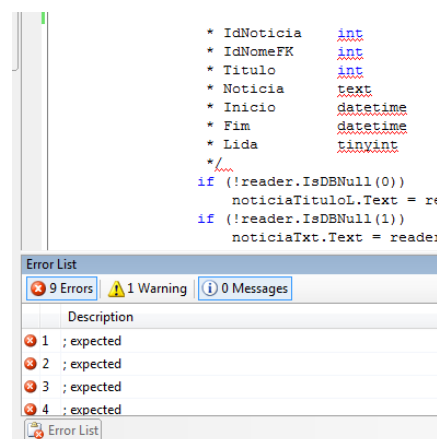


Figura 10 - Exemplo da listagem de erros e avisos

3. Análise da aplicação “Pub Móvel”

Neste capítulo faz-se a descrição do problema a abordar, identificando-se os perfis de utilizadores assim como os seus pré-requisitos.

Nesta fase foi necessário compreender qual a natureza do programa a desenvolver no âmbito do trabalho, e sua arquitectura sabendo qual a constituição do sistema, como irá funcionar, onde estará localizado, quais os intervenientes e qual a razão de funcionar de determinada forma.

3.1. Descrição do problema

De acordo com Obringer(2003), para as pessoas que trabalham em marketing, saber orientar a publicidade ao público-alvo correcto é considerado algo muito difícil de se concretizar, pois o desafio não depende tanto do que se anuncia mas sim de “a quem” se anuncia.

Teoricamente, o público-alvo é definido à partida e baseia-se em pontos-chave específicos, tais como as faixas etárias, os estratos sociais, as localizações geográficas, a dimensão de família, o género (masculino/feminino) e a respectiva personalidade.

No entanto, a publicidade pode não ser bem interpretada ou orientada, isto é, pode não ser encaminhada para o público tendo em conta os pontos-chave.

Actualmente é possível encontrar publicidade em todo lado, uma vez que é difundida em muitos meios, tais como os meios visuais incluindo os *outdoors*, os *flyers* e as revistas, e também os meios auditivos, incluindo a rádio e os respectivos anúncios. É, inclusivamente, possível a combinação destes meios como exemplo a publicidade que é difundida na televisão. Por outras palavras, faz-se uma difusão de publicidade que é encaminhada para todas as pessoas mesmo para aquelas que não a desejam receber.

Por isso gera-se um conflito, na medida que existem aqueles que têm interesse em receber uma determinada publicidade, ao mesmo tempo que existem outros que não têm

interesse em receber aquelas notícias. Em suma, quando para uns a publicidade é vista como útil para outros é encarada como *spam*.

Nos últimos anos têm sido realizados bastantes estudos relativamente à utilização de publicidade em dispositivos móveis, tais como o telefone celular, recorrendo à utilização de mensagens SMS²⁸ (Heinonem & Strandvik, 2006) e (Dickinger & Haghirian, 2004).

No entanto, estes estudos chegaram à conclusão que a aceitação de mensagens SMS publicitárias por parte dos utilizadores era fraca. Na realidade, muitos dos utilizadores, quando levados a comparar as tecnologias, preferiram o tradicional *direct mail* usando os PDA's²⁹, e o e-mail usando os *Desktop PC's*, (Heinonem & Strandvik, 2006), ao passo que outros indicaram que a publicidade enviada directamente para o telemóvel era considerada abusiva e incomodativa (Dickinger & Haghirian, 2004).

Para a resolução deste problema, há que ter em conta as diversas questões que se apresentam na Tabela 1.

Problema:	Orientar a publicidade correctamente para o público-alvo sem interferir com as preferências de outrem.
Objectivo:	Agrupar temas (notícias) de múltiplos anunciantes focando os respectivos pontos principais sem que exista intervenção/escolha por parte dos anunciantes.
Público-alvo:	Utilizadores comuns que desejem estar atentos a novidades/publicidades/anúncios nas respectivas áreas de preferência.
Impacto:	Melhor absorção/entendimento da publicidade por parte dos utilizadores.
Uma boa solução traria benefícios tais como:	<ul style="list-style-type: none"> • Não tornar abusivo o envio directo de publicidade para dispositivos móveis pessoais. • Adaptar os temas de acordo com as preferências do utilizador. As preferências do utilizador não são publicadas estabelecendo assim uma relação de confiança. • Aumento da eficiência na entrega da publicidade. • Diminuição do conjunto de ferramentas a utilizar para a obtenção de informações sobre a notícia.

Tabela 1 - Características do problema.

²⁸ SMS – Short Message Service

²⁹ PDA's – Personal Digital Assistants

3.2. Metodologia de desenvolvimento do software

Segundo Sommerville(2006) e Vasconcelos(2004) na sequência de acções compostas por métodos, ferramentas e procedimentos que permitem construir *software* é necessária a existência de modelos que definam as regras e os processos utilizados nesse sentido. A escolha de um determinado modelo depende da natureza do projecto, dos métodos, das ferramentas a usar e das acções de controlo necessários. Os modelos de engenharia de software mais conhecidos incluem os seguintes:

- Ciclo de Vida Clássico;
- Modelo em Cascata (ou *Waterfall*);
- Modelo Incremental;
- Prototipagem;
- Técnicas de quarta geração (RAD – *Rapid Application Development*).

Para o desenvolvimento das aplicações no âmbito do trabalho descrito nesta dissertação, foram seguidos dois modelos de engenharia de *software*, cuja aplicação se descreve com mais detalhes nas secções que se segue: o modelo incremental e a prototipagem.

3.2.1. Modelo Incremental

O modelo incremental combina as características do modelo em cascata com a filosofia iterativa subjacente à prototipagem. Como se mostra na Figura 11, este modelo de processo aplica sequências lineares de desenvolvimento de uma forma faseada. Cada sequência linear produz como resultado um incremento funcional do *software* final. No caso das aplicações desenvolvidas nesta dissertação a aplicação deste modelo resultou no conjunto das seguintes iterações:

- Na primeira iteração desenvolveu-se as funções de manipulação de dados (abrir e fechar base de dados, efectuar *queries*) e as funcionalidades básicas das aplicações, tais como a apresentação visual dos dados provenientes da base de dados.

- Na segunda iteração incluíram-se as capacidades de manipulação mais avançadas, tais como a pesquisa na base de dados com *queries* específicas para cada menu, incluindo as funcionalidades que permitem adicionar dados e eliminar dados directamente na base de dados.
- Na terceira iteração procedeu-se, à inclusão de imagens externas que facilitam a visualização de novos comandos.
- Na quarta iteração acrescentaram-se funcionalidades que permitem o controlo sobre as notícias e os registos dos utilizadores.
- Na quinta iteração efectuou-se o controlo sobre a replicação da base de dados.

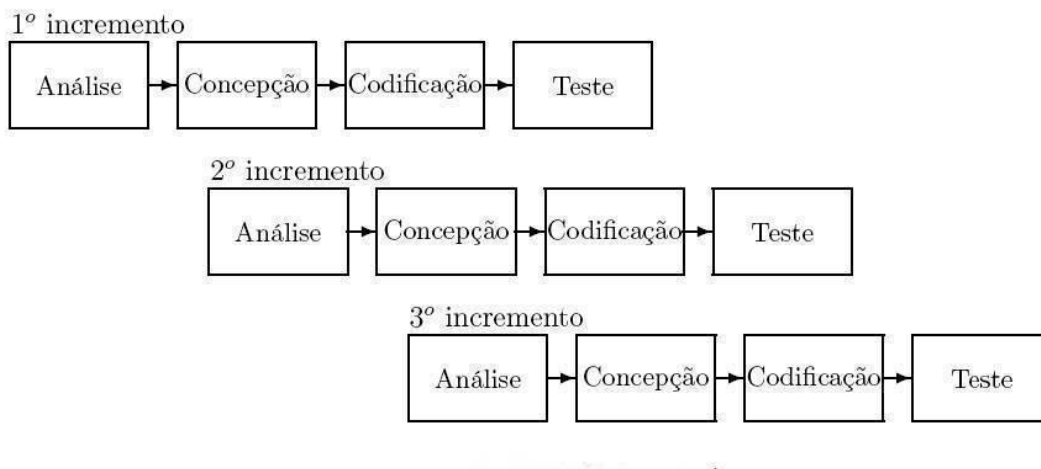


Figura 11 - Fases do modelo incremental (Fernandes, 2000)

A modelação incremental tem por ideia base a criação de uma estrutura mais simples do que uma complexa, baseando-se igualmente na noção de que é mais simples modificar uma estrutura já existente do que criar uma nova de raiz (a partir do nada). Assim, em vez de se tentar criar o modelo completo num só passo, as atenções, no início, concentram-se deliberadamente na incorporação dos aspectos mais críticos e importantes, sendo, posteriormente, o modelo repetidamente melhorado, até estar completo.

O modelo incremental, a exemplo do que acontece com a prototipagem, é, na sua essência, repetitivo e iterativo. Isto significa que as funcionalidades vão sendo acrescentadas até o sistema estar completamente descrito. No entanto, contrariamente ao que se verifica na prototipagem, o modelo incremental centra a sua acção na construção do produto final. Cada incremento representa uma versão reduzida do sistema final, que terá um contributo para a funcionalidade do sistema (Fernandes, 2000).

3.2.2. Prototipagem

A prototipagem é um modelo que permite, num curto espaço de tempo, apresentar um protótipo da solução final. Com este método é possível ter uma ideia concreta do produto que se pretende e ter a possibilidade de o refinar até se encontrar o produto final para as necessidades traçadas inicialmente.

Segundo Vasconcelos (2004), a prototipagem começa a tornar-se muito popular no mundo da engenharia de *software* já que é um paradigma que, por vezes permite poupar tempo e dinheiro. Contudo, para que este modelo funcione e tenha sucesso, é preciso que o protótipo seja elaborado o mais cedo possível, para que o analista tenha tempo de efectuar alterações no sentido de tornar o produto o mais eficiente possível.

A Figura 12 ilustra os processos iterativos do processo de prototipagem.

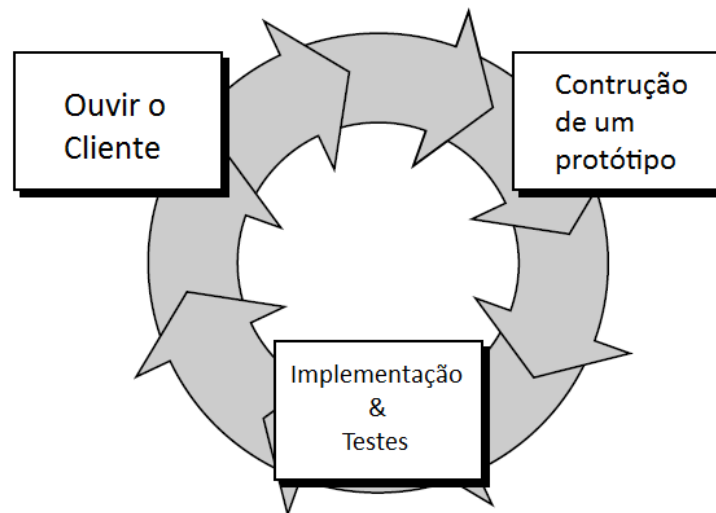


Figura 12 - Processo de desenvolvimento usando a Prototipagem (adaptado de Pressman, 2001)

O modelo da prototipagem é constituído por cinco fases principais, incluindo a recolha de requisitos, a concepção rápida, a construção do protótipo, a avaliação e o refinamento dos requisitos e a implementação do produto. Sabendo que é um processo iterativo, o ciclo repete-se as vezes necessárias até que o protótipo esteja refinado e satisfaça os requisitos iniciais.

3.3. Análise e planeamento das aplicações de *software*

Num processo de análise e planeamento reconhecem-se geralmente dois grandes tipos de requisitos: os funcionais e os não-funcionais.

Os requisitos funcionais descrevem as funções, tarefas e subtarefas que se esperam que o sistema realize. Incluem ainda outras funcionalidades que os utilizadores e analistas de sistemas esperam que o sistema execute. A identificação e definição de requisitos funcionais não é um exercício sobre o modo como o sistema suportará as funções, actividades e tarefas, mas sim um exercício detalhado sobre a forma como o sistema contemplará e perceberá os utilizadores.

Os requisitos não funcionais são aqueles que não dizem especificamente respeito às funcionalidades de um sistema. Este tipo de requisitos coloca restrições no sistema a desenvolver e no processo a usar. Por isso, os requisitos não-funcionais especificam as restrições externas às quais o produto deve obedecer. A análise dos requisitos não-funcionais inclui os requisitos de fiabilidade, segurança, adaptabilidade, portabilidade e desempenho (Rocha, 2004).

Descreve-se a seguir o desenvolvimento da análise realizada para as aplicações de software descritas nesta dissertação.

3.3.1. Levantamento de requisitos funcionais

Os requisitos funcionais correspondem à listagem de factores que um sistema deverá executar. Alguns são evidentes e criados com o conhecimento do utilizador, outros são ocultos e executados pelo próprio sistema sem que haja o conhecimento explícito do utilizador.

No que diz respeito aos requisitos funcionais da aplicação móvel, a aplicação “PubMovel” deverá ser capaz de:

- Fornecer notícias provenientes de um servidor *Web* onde se encontram registadas as empresas anunciantes.

- Proporcionar funcionalidades de consulta de notícias/anúncios, assim como informação relativa aos anunciantes. Deve ainda notificar o utilizador sempre que existam notícias/anúncios novos que façam parte da lista de preferências do utilizador.
- Permitir uma replicação correcta da base de dados de notícias e uma gestão anónima das preferências, adaptando-as de acordo com o perfil de utilizador. Deve permitir a visualização de imagens fazendo a integração com um serviço externo via *Web* de modo a permitir usufruir de uma ferramenta visual que permita ao utilizador visualizar a localização do anunciante.
- Permitir a adição, eliminação, actualização e visualização de novos locais definidos pelo utilizador final e encarados como preferências.

Na

Figura 13, é visível um diagrama de uses cases para a aplicação “PubMovel”.

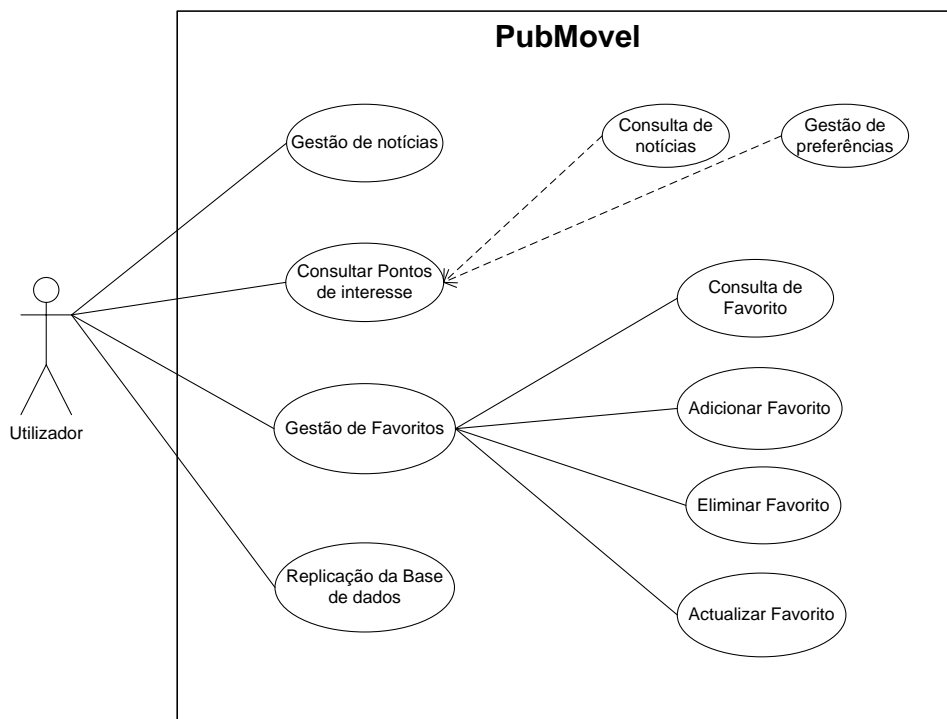


Figura 13 - Diagrama de Use Cases da aplicação PubMovel na fase de levantamento de requisitos.

A aplicação Web “NewsMobile” desenvolvida no âmbito desta dissertação deverá ser capaz de:

- Permitir a criação de contas de utilizadores, assim como a criação de tipos genéricos de pontos de interesse para melhorar a organização das notícias.

- Permitir a criação, edição, eliminação e visualização de publicidade, incluindo notícias e anúncios.
- Permitir a alteração de perfis de contas de utilizadores, bem como as respectivas *passwords* de acesso.
- Gerar os ficheiros de replicação da base de dados assim que se efectue um logout.

Na

Figura 14, é visível um diagrama de uses cases para a aplicação “NewsMobile”.

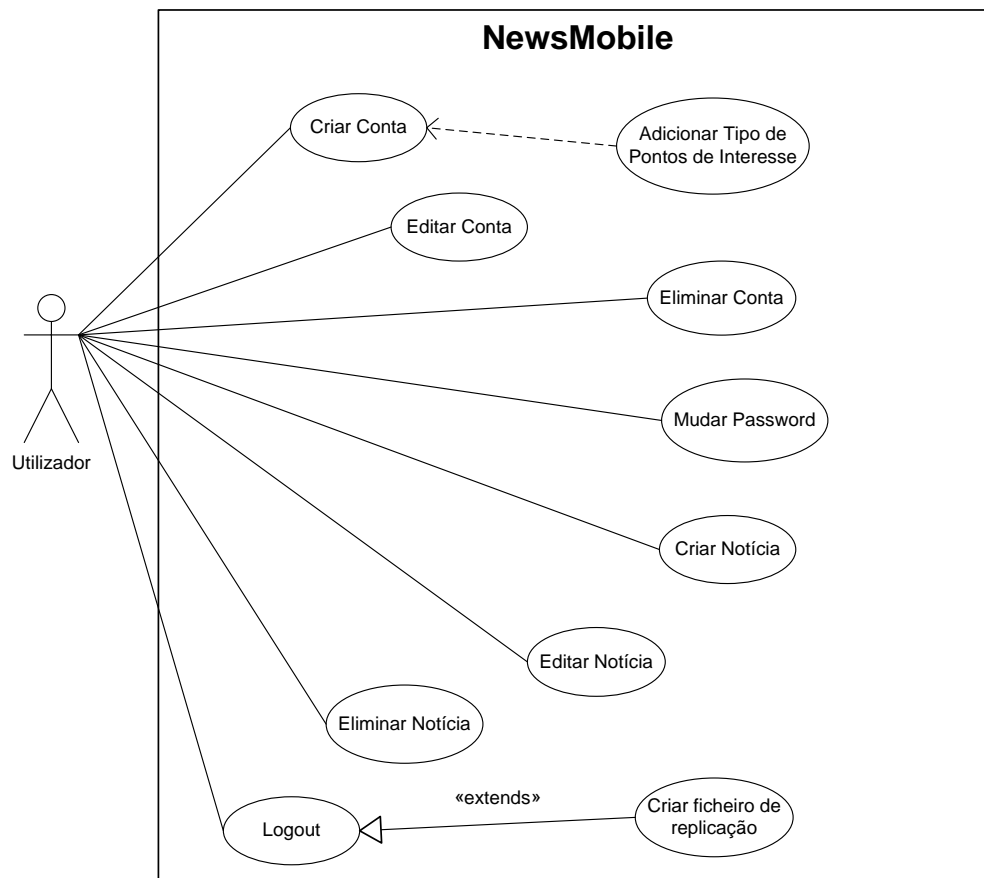


Figura 14 - Diagrama de Use Cases da aplicação “NewsMobile” na fase de levantamento de requisitos.

No caso da aplicação “NewsMobile”, existe ainda um módulo pertencente ao domínio de administração. O objectivo do administrador é controlar a validade de notícias, não permitindo acumular notícias no portal em que o prazo de validade já tenha expirado. Como é visível na Figura 15, o administrador do portal terá apenas a possibilidade de efectuar o login/logout assim como a limpeza de notícias.

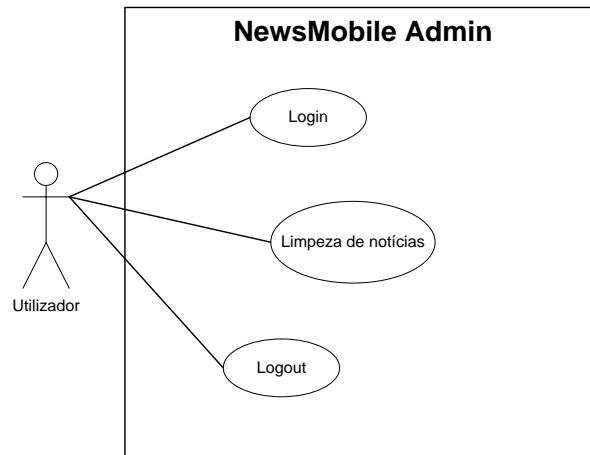


Figura 15 - Diagrama de Use Cases da aplicação “NewsMobile” no módulo “administração” na fase de levantamento de requisitos.

3.3.2. Levantamento de requisitos não funcionais

Os requisitos não-funcionais ou atributos de sistema, são requisitos que declaram restrições, ou atributos de qualidade para um software e/ou para o processo de desenvolvimento do sistema e de como este deve realizar os requisitos funcionais. Muitas vezes são propriedades ou qualidades que especificam características e melhoram o produto. A segurança, a precisão, a usabilidade e a performance são exemplos de requisitos não funcionais.

Em termos de requisitos não funcionais as aplicações “PubMove1” e “NewsMobile” deverão:

- Ser fáceis de usar e devem de exigir um grau de aprendizagem tão reduzido quanto possível.
- Permitir que, tanto um utilizador avançado como um utilizador inexperiente, sejam capazes de aceder facilmente aos menus, aumentando a eficiência da consulta e melhorando a orientação de publicidade para o público-alvo, fazendo com que o utilizador final consulte apenas as notícias que está interessado.
- Permitir uma instalação muito fácil da aplicação “PubMove1” no dispositivo móvel do utilizador final.
- Permitir um tempo máximo de funcionalidade do portal “NewsMobile”, em modo *on-line*, para que os utilizadores possam efectuar alterações assim que desejarem.

3.3.3. Interface do utilizador

Qualquer sistema informático destinado a ser utilizado por pessoas deve possuir uma interface simples e acessível de utilizar. O objectivo principal é conceber uma interface que contribua para aumentar o desempenho do utilizador (Barfield, 1993 e Faulkner, 1998 *cit. in* Ribeiro, 2004).

Por interface simples, entende-se uma interface que exige o menor esforço de aprendizagem por parte do utilizador (Ribeiro, 2004).

Na concepção da aplicação “PubMovel”, tomou-se em consideração as reduzidas dimensões de ecrã disponíveis num dispositivo móvel. O dispositivo usado para testar a aplicação foi um PDA HTC P3300 que contém um ecrã com as dimensões de 240×320 pixels. Com uma dimensão de ecrã tão reduzida, foi necessário ter em atenção a necessidade de não sobrecarregar os menus, tornando-os simples e leves, facilitando assim a leitura e a compreensão por parte do utilizador.

Assim, para cumprir o objectivo de fazer o aproveitamento do reduzido espaço disponível sem sobrecarregar o ecrã, utilizou-se o modo de visualização *full-screen*, que permite uma visão completa dos menus retirando alguns pormenores do próprio sistema operativo incluindo, a afixação das horas, o *StartMenu* e o botão de fecho da aplicação.

Utilizou-se ainda outra opção, designada por *TabControl* que permite recorrer a separadores de modo a repartir a informação dentro do mesmo ecrã. A Figura 16 e a Figura 17 apresentam, respectivamente exemplos da utilização do modo *full-screen* e do *TabControl*, mostrando o conteúdo dos separadores “Info” e “Notícias”.

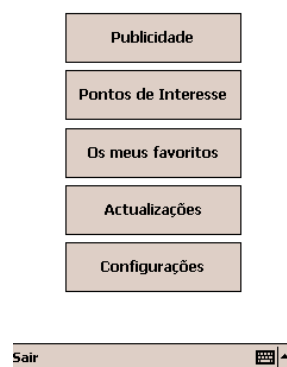


Figura 16 - Exemplo da utilização do modo *Full-Screen*.

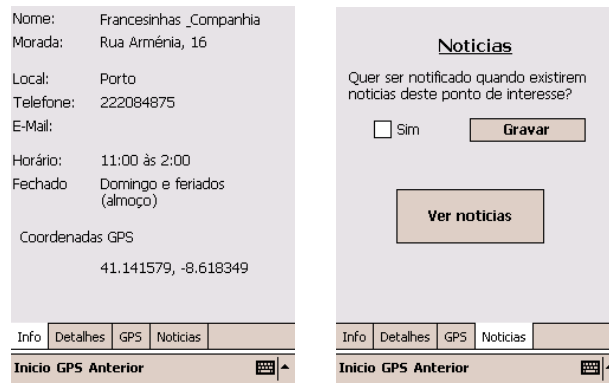


Figura 17 - Exemplo da utilização do *TabControl*.

No que diz respeito à aplicação “NewsMobile”, facilitou-se a interação entre o cliente e a aplicação através da, criação de três áreas de destaque:

Duas das áreas dinâmicas e uma terceira área estática, como se ilustra na Figura 18.

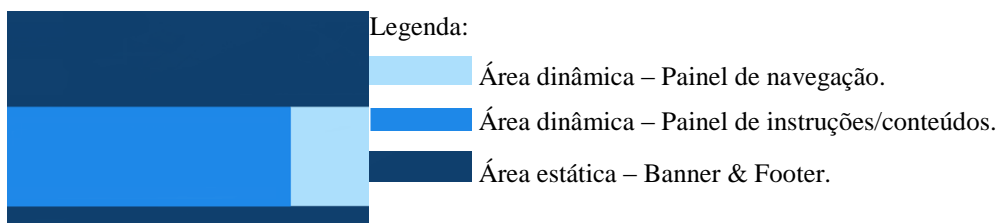


Figura 18 - Esquema de painéis de áreas

Como se pode verificar na Figura 19, a área estática contém um cabeçalho (*banner*) e um rodapé para a *interface* da aplicação. No que diz respeito às duas áreas dinâmicas, uma delas apresenta os conteúdos pedidos pelo utilizador, fornecendo igualmente instruções sobre o modo de utilização da aplicação “NewsMobile”. Por fim, a restante área dinâmica foi criada de modo a ir apresentando todas as opções de acordo com a navegação por meio dos menus.



Figura 19 - Exemplo da aplicação "NewsMobile".

3.3.4. Mecanismo de replicação da base de dados

Para a replicação da base de dados entre o servidor e os dispositivos móveis clientes, recorreu-se à linguagem XML.

As razões que levaram à sua utilização para um processo delicado, devem-se ao facto do XML ser uma linguagem estruturada e de permitir uma fácil partilha de informação através da internet.

Atendendo a que o objectivo do projecto não é o de avaliar a eficiência da replicação dos dados, mas sim o de implementar um processo simples e funcional, idealizou-se um processo que realiza a criação de um conjunto de ficheiros XML sempre que se finaliza uma sessão do lado servidor. Mais especificamente, sempre que ocorre o final de cada sessão, a aplicação cria automaticamente dois ficheiros: o ficheiro `db.xml` e o ficheiro `news.xml`:

- O ficheiro `db.xml` contém toda a estrutura da base de dados a ser replicada.
- O ficheiro `news.xml` contém apenas as notícias que devem ser replicadas.

Assim, esses ficheiros ficam disponíveis no servidor, podendo ser descarregados pelo utilizador final para o respectivo dispositivo móvel sempre que o utilizador pretenda actualizar os conteúdos da base de dados do dispositivo móvel. Note-se que a replicação da base de dados é sempre feita no sentido do servidor para o dispositivo móvel como se ilustra na Figura 20, e nunca no sentido contrário. O processo consiste na realização do download do ficheiro XML para o dispositivo móvel (PDA) permitindo que posteriormente, seja “lido” e interpretado pela aplicação “PubMovel”.

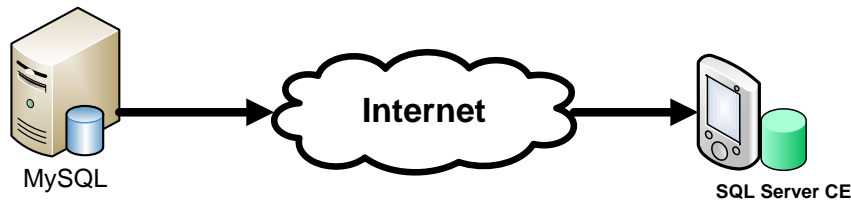


Figura 20 - Replicação de base de dados da aplicação

À medida que o ficheiro XML é lido pela aplicação móvel, está-se a popular a base de dados local do dispositivo móvel para possibilitar a integração dos registos existentes com os registos actualizados obtidos da base de dados principal.

Depois de terminada a leitura do ficheiro XML, o utilizador tem a opção de eliminar o ficheiro XML. De modo a poupar recursos de armazenamento no dispositivo móvel.

3.4. Estruturas de navegação nas aplicações de software

O mapa de navegação tem como objectivo principal mostrar, de um modo estruturado toda a informação que é possível apresentar ao utilizador. É particularmente útil para ilustrar a organização e as ligações entre os vários ecrãs de uma aplicação e para representar de um modo claro o modo como todos os ecrãs se interligam. O mapa de navegação permite assim, definir todos os percursos possíveis que o utilizador pode seguir quando está a utilizar a aplicação.

Segundo Vaughan (cit, in Ribeiro, 2004), existem quatro estruturas fundamentais a nível da navegação para a organização de uma aplicação.

- ✓ **Linear** – O utilizador navega sequencialmente de um ecrã para o seguinte ou para o anterior.
- ✓ **Hierárquica** – O utilizador navega ao longo dos ramos de uma árvore que reflecte uma organização lógica do conteúdo.
- ✓ **Não-linear** – O utilizador navega livremente por todo o conteúdo da aplicação, sem lhe ser imposta qualquer restrição, podendo escolher em cada instante, o percurso que deseja para consultar a informação disponível.
- ✓ **Composta** – O utilizado pode navegar livremente, mas existem ocasiões em que encontra restrições, tais como apresentações lineares de sequências de ecrãs contendo informação considerada crítica ou, ainda uma hierarquia de ecrãs.

O tipo de estrutura utilizada no desenvolvimento das aplicações desenvolvidas nesta dissertação foi a estrutura composta, uma vez que existem, nestas aplicações, percursos sequenciais e hierárquicos.

Apresenta-se de seguida uma descrição dos mapas de navegação das aplicações “PubMoveI” e “NewsMobile”. Note-se que os nomes contidos nas caixas que integram os mapas de navegação da Figura 21, da Figura 22 e da Figura 23 representam os ecrãs no mapa de navegação, sendo alusivos aos conteúdos que compõem esses ecrãs nas aplicações.

A Figura 21 representa o mapa de navegação possível da aplicação “PubMoveI”. A Figura 22 e a Figura 23 representam o mapa de navegação da aplicação “NewsMobile”, respectivamente, em modo utilizador (Cliente) e em modo Administrador.

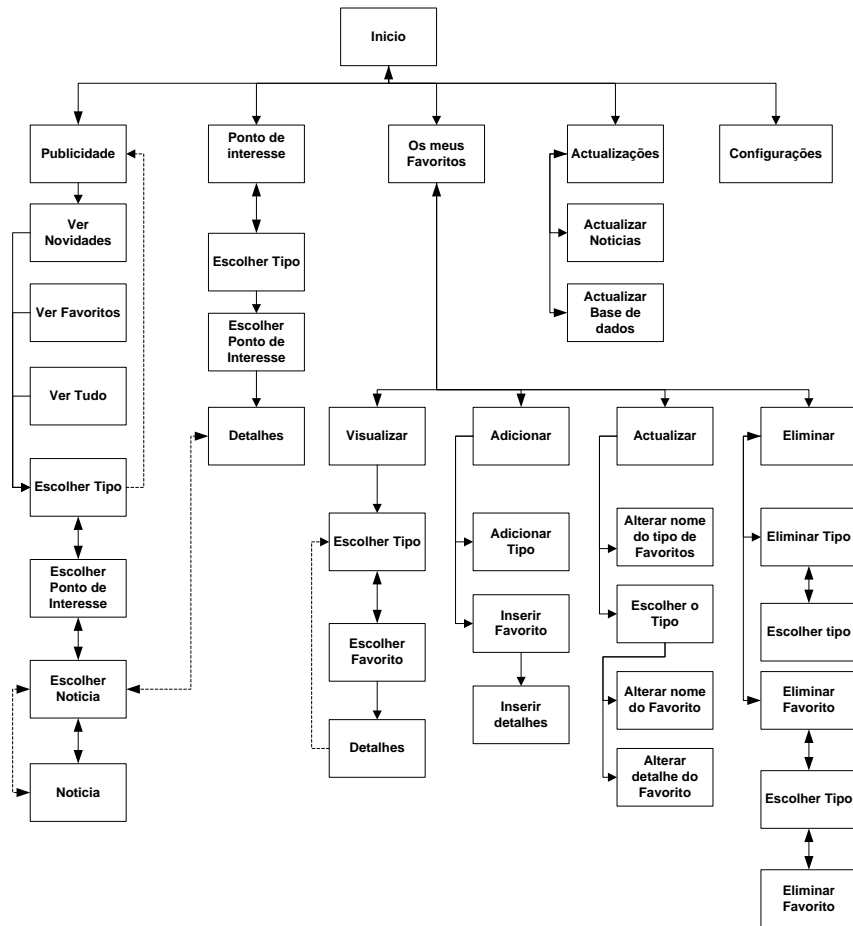


Figura 21 - Mapa de navegação (PDA).

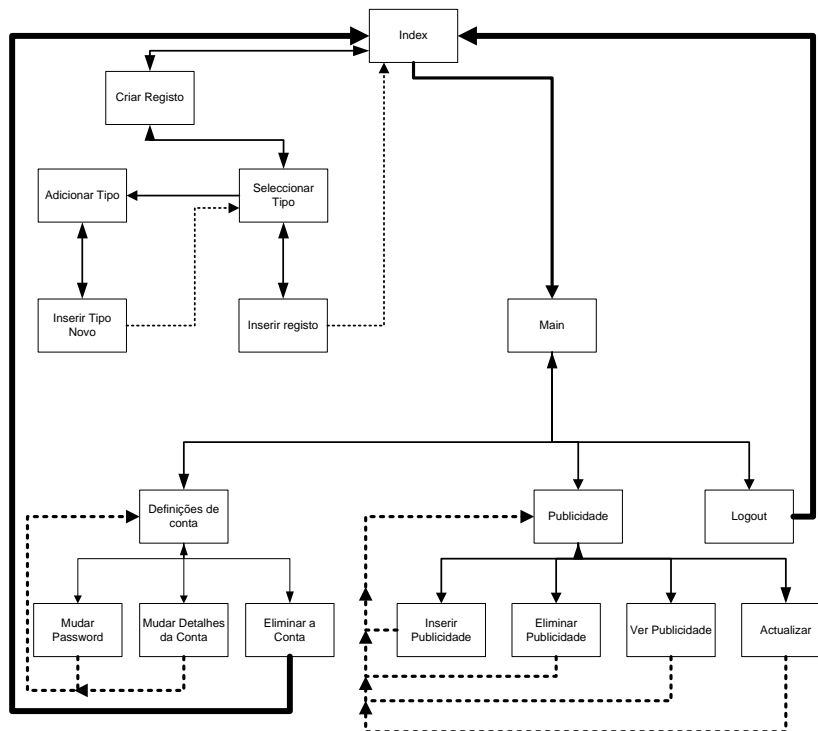


Figura 22 - Mapa de navegação (NewsMobile – modo utilizador).

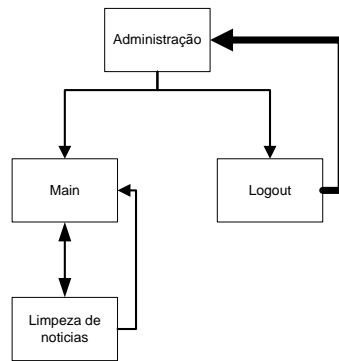


Figura 23 - Mapa de navegação (NewsMobile - modo Admin).

3.5. Concepção das bases de dados

Na concepção e desenvolvimento das bases de dados que suportam as duas aplicações descritas nesta dissertação utilizou-se o Modelo Entidade-Relacionamento (E-R), que se baseia numa percepção do mundo real representada por um conjunto de objectos denominados por Entidades e pelos relacionamentos que se podem estabelecer entre esses objectos. Mais especificamente, no modelo E-R, uma Entidade é um objecto que pode ser identificado de forma unívoca em relação a todos os outros objectos.

Apresenta-se a seguir um conjunto de diagramas E-R que ilustram os relacionamentos existentes entre as entidades da base de dados utilizada pelas aplicações. É importante notar que, nos diagramas que se seguem, as caixas rectangulares representam as entidades que deram origem à criação de uma tabela com o mesmo nome e com os atributos identificados. Os losangos representam os relacionamentos entre as entidades. Nos diagramas expostos adiante, apresenta-se ainda, ao lado de cada entidade, uma descrição dos respectivos atributos.

3.5.1. Base de dados da aplicação “Pub Móvel”

Numa primeira fase foi concebida a base de dados da aplicação “PubMovel”. Foram definidas as seguintes entidades: Tab_Tipo, Tab_Nome, Tab_Noticia, Tab_Detalhe e System. A entidade System guarda apenas registos exclusivos para a aplicação, tais

como a última actualização da base de dados e o endereço da publicação Web do serviço “NewsMobile”, de modo a possibilitar a actualização da base de dados.

A entidade Tab_Nome guarda registos do nome e tipo de ponto de interesse de um anunciante. Esta entidade possui um relacionamento com a entidade Tab_Tipo que é de N para 1, uma vez que um cliente apenas pode pertencer a um único tipo. No entanto, a um tipo podem estar associados N anunciantes.

Existem ainda mais dois relacionamentos para a entidade Tab_Nome. O primeiro é um relacionamento de 1 para 1 entre a entidade Tab_Nome e a entidade Tab_Detalhe. De facto, um anunciante possui apenas uma descrição que pertence, evidentemente, apenas a esse anunciante. Por outro lado, o segundo relacionamento estabelece-se entre a entidade Tab_Noticia sendo do tipo de 1 para N: um anunciante pode emitir N notícias, mas uma determinada notícia pertence apenas a 1 anunciante.

A Figura 24 mostra o diagrama E-R da base de dados principal da aplicação “PubMovel”.

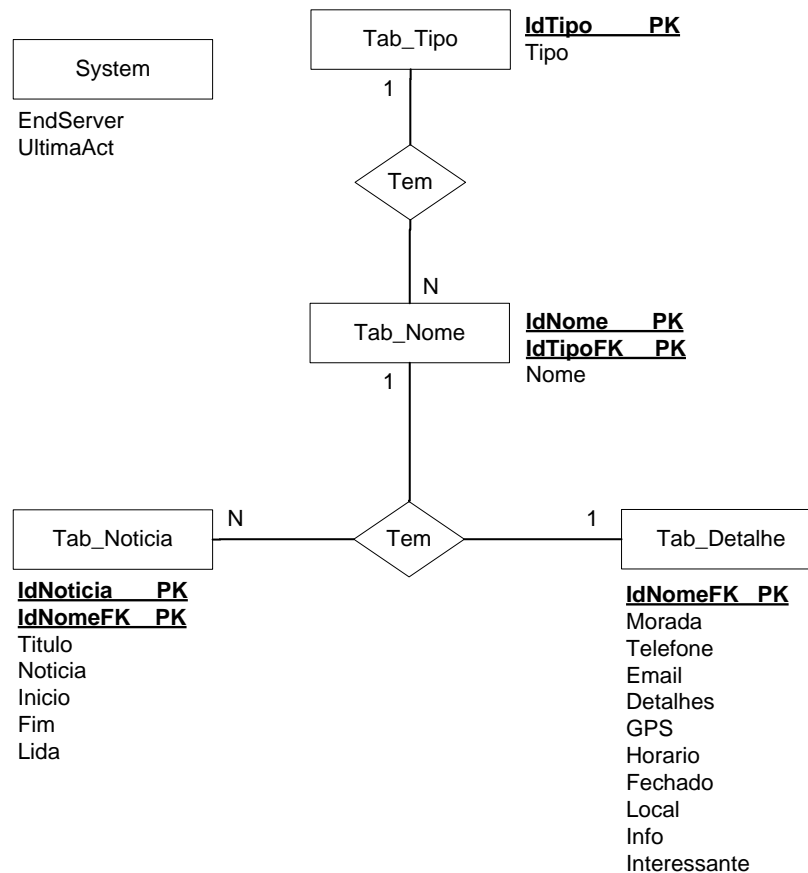


Figura 24 - Diagrama E-R da primeira base de dados (PubMovel).

Numa segunda fase, tendo em vista a implementação dos favoritos, que são elementos personalizados que dependem da preferência dos utilizadores, foi concebida uma segunda base de dados para a aplicação “PubMovel” tendo sido definido o seguinte modelo, representado na Figura 25 pelo respectivo Diagrama E-R.

É importante realçar que esta segunda base de dados é idêntica à primeira, variando apenas em termos das entidades System e Tab_Noticias. Isto deve-se ao facto de ser uma base de dados utilizada exclusivamente pelo utilizador final, não existindo a necessidade de se efectuar nenhum controlo sobre as actualizações, não existindo igualmente a possibilidade de obtenção de notícias.

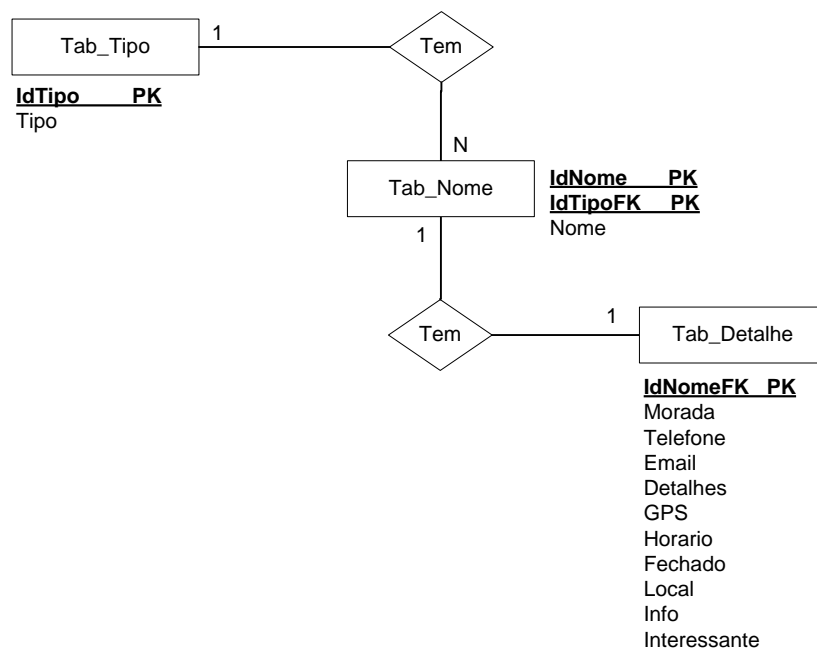


Figura 25 - Diagrama E-R da segunda base de dados (PubMovel).

3.5.2. Base de dados da aplicação “NewsMobile”

No âmbito desta dissertação, a concepção da aplicação “NewsMobile” foi desenvolvida com o objectivo de dar suporte à aplicação “PubMovel”. Por esse motivo, o desenvolvimento da respectiva base de dados baseou-se, naturalmente, na base de dados principal que foi concebida para a aplicação “PubMovel”.

Assim, a base de dados da aplicação “NewsMobile” é em tudo idêntica à que ficou exposta na secção anterior, excepto no facto de não conter a entidade System. Na realidade, nesta base de dados essa entidade foi substituída por uma outra entidade, designada por *Login*, que permite guardar os códigos de acesso à aplicação “NewsMobile”. A entidade *Login* possui um relacionamento com a entidade *Tab_Nome* do tipo 1 para 1, uma vez que um anunciante possui um único *Login* e 1 *Login* pertence exclusivamente a um anunciante.

A Figura 26 representa o diagrama E-R associado à concepção da base de dados da aplicação “NewsMobile”.

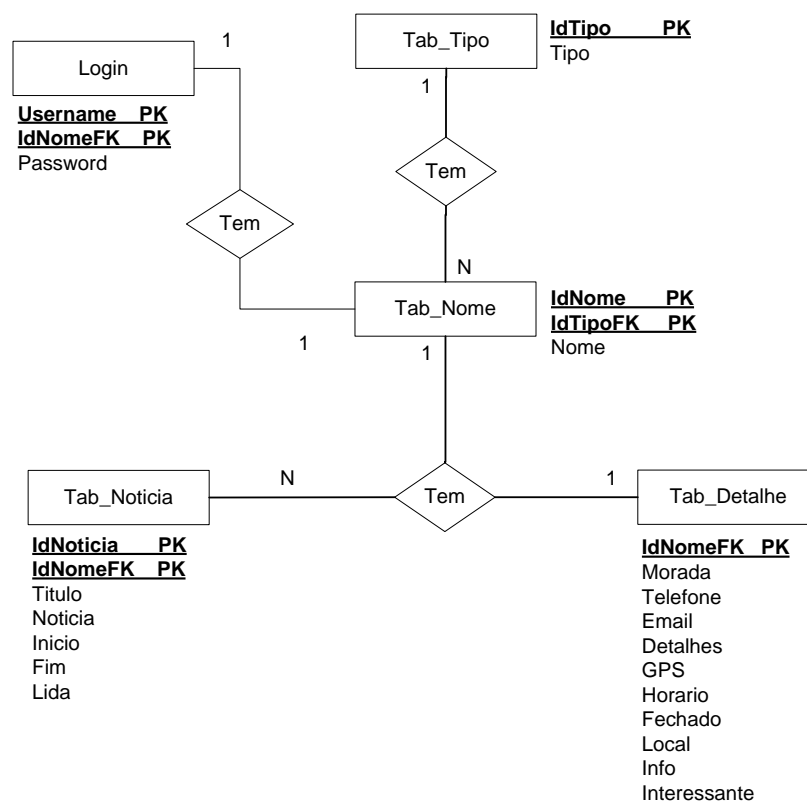


Figura 26 - Diagrama E-R da base de dados (NewsMobile).

4. Implementação das aplicações “Pub móvel” e “NewsMobile”

Este capítulo descreve o modo como as aplicações concebidas no capítulo 3 foram implementadas, identificando-se claramente todos os passos efectuados no desenvolvimento do código das aplicações, recorrendo às ferramentas de desenvolvimento identificadas no capítulo 2, de modo a satisfazer os requisitos das aplicações que se explicitaram no capítulo 3.

Assim, este capítulo apresenta uma descrição detalhada de todas as actividades relacionadas com a implementação das aplicações “PubMovel” e “NewsMobile”, recorrendo às funcionalidades das ferramentas de desenvolvimento que foram descritas com mais pormenor no capítulo 2.

4.1. Implementação da aplicação “PubMovel”

Para a implementação da aplicação móvel recorreu-se à ferramenta de desenvolvimento MS Visual Studio 2005 SP1, que proporciona um ambiente completo para a programação das aplicações. Na primeira vez que a ferramenta foi executada, foi indicado, usando o *wizard* de instalação, que a linguagem de programação a utilizar seria a linguagem C#, tendo-se indicado que seria utilizada no desenvolvimento de aplicações para dispositivos móveis, o que fez com que a ferramenta finalizasse a instalação e adicionasse o SDK³⁰ necessário para a implementação em dispositivos do tipo *smart device*.

Após a instalação da ferramenta, iniciou-se o processo de desenvolvimento das aplicações através da criação de um novo projecto do tipo C#, com a finalidade de ser executado em dispositivos móveis, tal como se mostra na Figura 27.

³⁰ SDK - *Software Development Kit*.

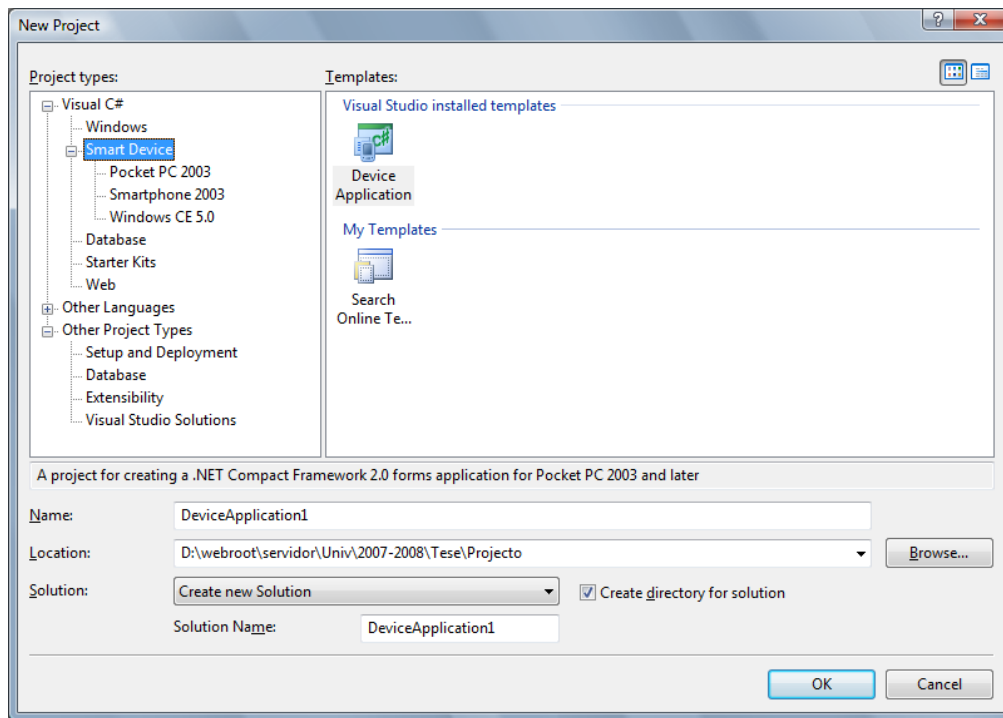


Figura 27 - Caixa de diálogo de um novo projecto usando o MS Visual Studio 2005

Ao efectuar a escolha, cria-se o ambiente de desenvolvimento como se pode observar na Figura 28. Por omissão, é criado um formulário (*form1.cs*) e uma classe (*Program.cs*) que tem como propósito servir de ponto de partida para o desenvolvimento da nova aplicação. Nesta classe já se encontra presente um método *Main()*.

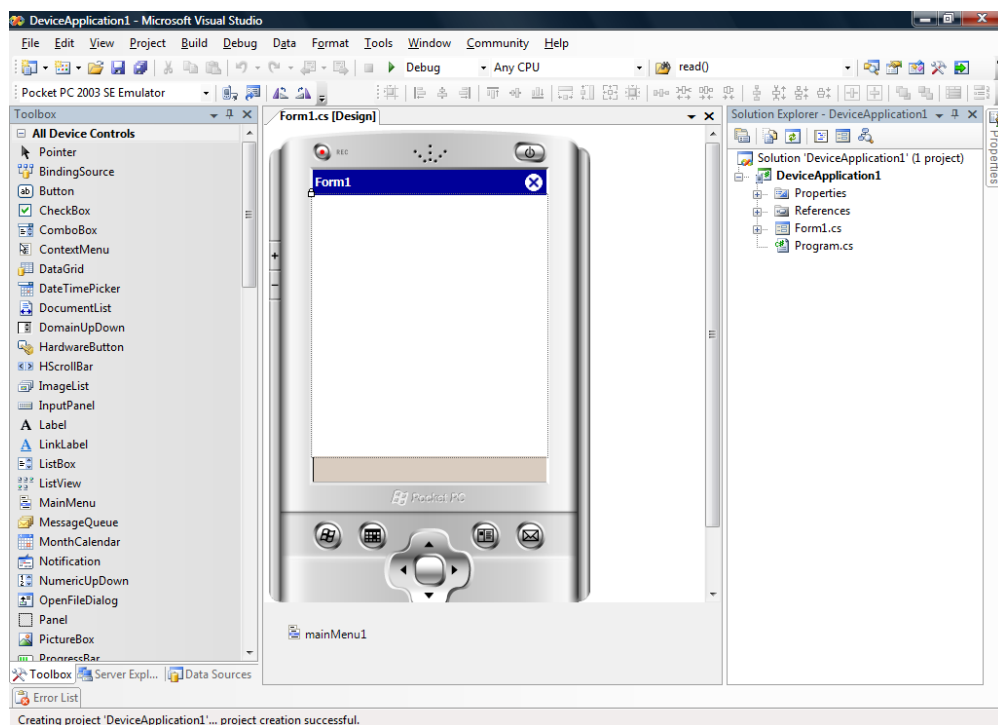


Figura 28 - Ambiente de desenvolvimento no MS Visual Studio 2005 SP1

No que diz respeito ao desenvolvimento das aplicações existem dois modos de edição da aplicação: o modo *design* e o modo código.

O modo *design*, ilustrado na Figura 29, permite ao programador a adição de controlos nos formulários arrastando o controlo necessário da *Toolbox* para uma imagem gráfica de um dispositivo móvel que representa o estado gráfico do formulário.

Por outro lado, no modo código, como se mostra na Figura 29, não são disponibilizadas ajudas gráficas. Neste modo não se permite, portanto, um desenvolvimento do tipo RAD (definir a sigla). Assim, é necessário escrever todo o código necessário para o desenvolvimento dos diferentes formulários que constituem a aplicação.

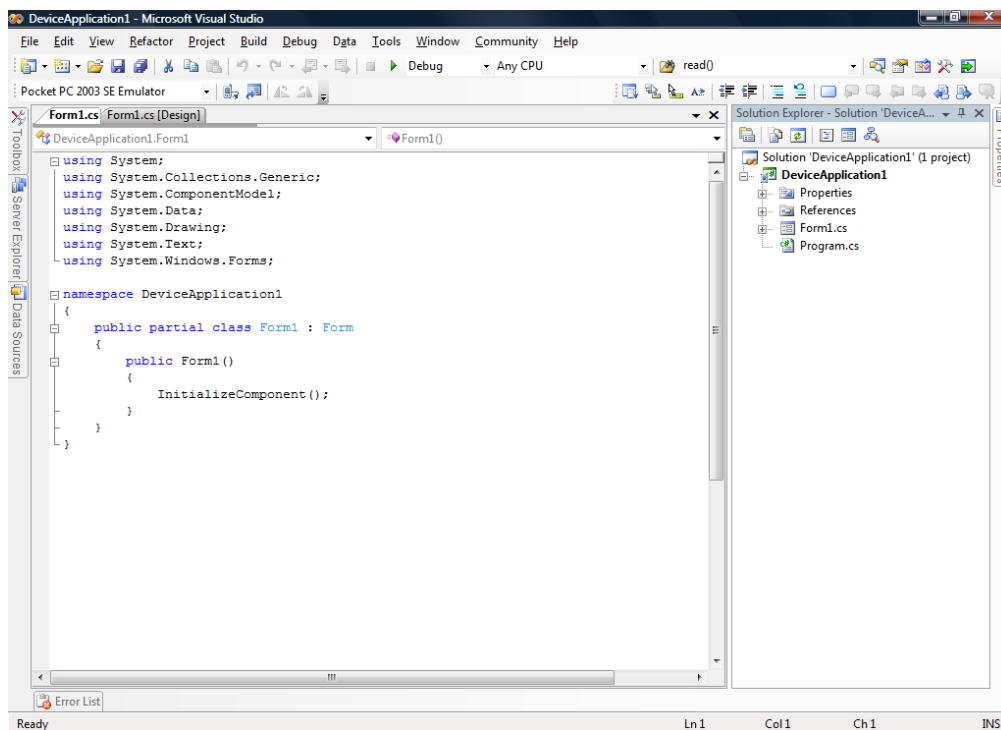


Figura 29 - Modo código no MS Visual Studio 2005

4.1.1. Formulários da aplicação “PubMovel”

Durante a implementação da aplicação “PubMovel” foram criados formulários e controlos que permitiram a construção de uma interface gráfica do utilizador, de modo a permitir que o utilizador navegue na aplicação. Para implementar todos os ecrãs que se apresentam no mapa de navegação da foram Figura 21, criados 20 formulários.

Os formulários da aplicação “PubMovel” que são apresentados ao utilizador ocupam completamente o ecrã em modo *full-screen*, o que faz com que seja possível interagir com um formulário de cada vez. Para que tal fosse possível, recorreu-se ao uso do controlo “`WindowState = FormWindowState.Maximized;`”.

Fez-se igualmente uso do menu Start Menu do .NET que existe em cada formulário, de modo a facilitar uma rápida transição entre menus. Genericamente, foram adicionados botões com a designação “Início”, que permitem em qualquer ponto da aplicação passar para o formulário inicial (Form1.cs). Em todo o caso, existem excepções, como é o caso do próprio formulário (Form1.cs) e o formulário TiposExistentes.cs, que apenas é do tipo informativo.

Os formulários foram ainda dotados de um botão adicional, designado por “Anterior”, que permite retornar ao formulário anterior, permitindo assim efectuar ajustes ou correcções no modo de visualização do utilizador sem que seja necessário repetir todo o processo desde o início. Neste caso, existem igualmente excepções no que diz respeito à inexistência do botão nos formulários, nomeadamente nos seguintes formulários: Form1.cs, MenuAct.cs, ServerData.cs, FormPub.cs, Favorito.cs, FormTipo.cs, Xml.cs, TiposExistentes.cs, AddFavDetalhe.cs e o EditFav.cs.

Do ponto de vista da funcionalidade oferecida ao utilizador, os formulários podem ser classificados em quatro tipos distintos:

- Formulários genéricos: são os formulários que funcionam como ponto de partida para a navegação ou parametrização da aplicação.
- Formulários de Publicidade: são os formulários que permitem a visualização dos anúncios ou notícias publicadas pelos anunciantes e que se encontram disponibilizadas na base de dados.

- Formulários de Pontos de interesse: são os formulários que permitem a navegação pelos pontos de interesse globais da aplicação.
- Formulários de Favoritos: permitem a navegação e parametrização de todos os pontos de interesse pessoais criados localmente pelo utilizador final.

Descreve-se em seguida com mais detalhe o conjunto de formulários que foram desenvolvidos, bem como as ferramentas utilizadas para o efeito.

4.1.1.1. Formulários genéricos

Os formulários genéricos que serão mencionados em seguida correspondem aos nomes visíveis no mapa de navegação de acordo com a Figura 21, como Início, Actualizações e o seu subconjunto em árvore, Actualizar Noticias, Actualizar Base de dados e Configurações.

4.1.1.1.1. Formulário *Form1.cs*

O Formulário *Form1.cs*, como se ilustra na Figura 30, é o formulário inicial de apresentação da aplicação. Este formulário disponibiliza um conjunto de botões do tipo (*System.Windows.Forms.Button*) que permite fazer a selecção dos menus que o utilizador pretende explorar.

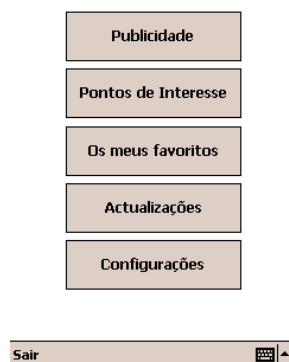


Figura 30 - Formulário *Form1.cs* da aplicação PubMovel.

Como a aplicação foi desenvolvida recorrendo ao modo *full-screen*, não é possível utilizar os controlos do WM para minimizar, ou mesmo para fechar, a aplicação. Por essa razão, foi adicionado ao menu StartMenu deste formulário um botão com a designação “Sair”, do tipo (System.Windows.Forms.MenuItem), que executa o Exemplo de código 1 que permite terminar a aplicação “PubMovel”, fechando todos os formulários.

```
private void menuItem1_Click(object sender, EventArgs e)
{
    Application.Exit();
    Application.DoEvents();
}
```

Exemplo de código 1 - Código que permite fechar a aplicação PubMovel.

4.1.1.1.2. *Formulário MenuAct.cs*

O formulário MenuAct.cs, permite ao utilizador escolher o tipo de actualização que deseja efectuar, como se mostra na Figura 31. Aqui utilizam-se *labels* do tipo (System.Windows.Forms.Labels) para adicionar títulos e instruções. Finalmente, a utilização de botões permite fazer uma separação do tipo de actualização que o utilizador deseja efectuar, isto é, permite decidir se deseja actualizar apenas as notícias ou a base de dados completa.

Escolher a opção a actualizar.

Actualizar Noticias

Actualizar Base de dados

Nota: Quando actualizar a Base de dados seja paciente até o processo de restauro acabar. Caso contrario a Base de dados poderá ficar danificada.

Inicio



Figura 31 - Formulário MenuAct.cs

4.1.1.1.3. Formulário *ServerData.cs*

O formulário *ServerData.cs* permite efectuar a gestão do URL que é guardado na base de dados e que permite, juntamente com o formulário *XML.cs*, realizar o *download* dos ficheiros de replicação da base de dados. Como se ilustra na Figura 32, é possível editar, limpar o endereço e gravar na base de dados novo endereço URL.

Quando este formulário é carregado, efectua-se uma *query* à base de dados que retorna uma *String* contendo o endereço URL que corresponde ao endereço do servidor. Ao utilizador só é permitido editar o URL quando pressiona o botão “Editar”, tornando a caixa do tipo (*System.Windows.Forms.TextBox*) editável, e colocando o estado dos botões “Limpar endereço” e “Gravar” em modo *Enable*, como se descreve no Exemplo de código 2.

```
private void editarB_Click(object sender, EventArgs e)
{
    saveB.Enabled = true;
    clean.Enabled = true;
    editarB.Enabled = false;
}
```

Exemplo de código 2 - Código que permite tornar *Enable* os restantes botões.

Definições do Servidor

Endereço do servidor:

http://192.168.0.104/teste/

Limpar endereço.

Editar Gravar

Inicio

Figura 32 - Formulário *ServerData.cs*

4.1.1.1.4. Formulário *XML.cs*

O formulário *XML.cs* é o aquele que permite iniciar a actualização, seja das notícias ou da base de dados completa. Possui uma *label* que recebe uma *String* consoante o tipo de actualização, como se mostra na Figura 33 e na Figura 34. Possui ainda um botão que

permite ao utilizador dar início à actualização e proporciona uma *textBox* onde é possível verificar os passos que se encontram a decorrer sempre que a aplicação está a replicar a informação da base de dados central para a base de dados local.

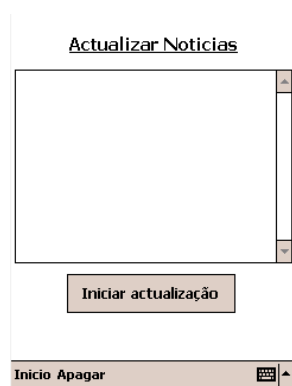


Figura 33 - Exemplo do Formulário XML.cs em modo Actualizar Notícias.

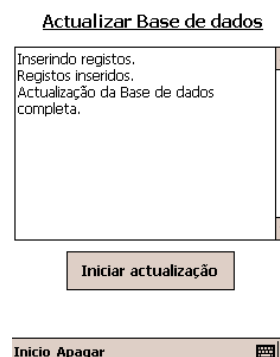


Figura 34 - Exemplo do Formulário XML.cs em modo Actualizar Base de dados.

Sempre que o utilizador executa a actualização, verifica-se em primeiro lugar se os ficheiros de replicação (DB.xml e News.xml) se encontram presentes no dispositivo móvel. Caso estejam presentes, faz-se uma actualização baseada nestes ficheiros. Caso contrário, a aplicação efectua a descarga dos ficheiros de actualização caso tenha ligação IP activa. Este *download* é realizado a partir do URL mencionado anteriormente no formulário ServerData.cs.

Logo que o download se conclua, os ficheiros são lidos e interpretados fazendo uma leitura estruturada do ficheiro XML. Durante essa leitura os registos são inseridos na base de dados local do dispositivo móvel.

Por fim, criou-se no menu startMenu um botão adicional que permite fazer a eliminação de ficheiros de replicação antigos, como se mostra na Figura 35, de modo a substituir ficheiros antigos por ficheiros de replicação mais actuais.

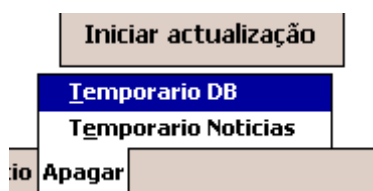


Figura 35 - Botão de apagar no formulário XML.cs

4.1.1.2. Formulários Publicidade

Os formulários Publicidade que serão mencionados em seguida correspondem aos nomes visíveis no mapa de navegação de acordo com a Figura 21, como Publicidade e o seu subconjunto em árvore, Ver Novidades, Ver Favoritos, Ver Tudo, Escolher Tipo, Escolher Ponto de Interesse, Escolher Notícia e Notícia.

4.1.1.2.1. Formulário FormPub.cs

O formulário FormPub.cs é o formulário inicial que permite visualizar a publicidade dos anunciantes que existe na base de dados local. Conforme se ilustra na Figura 36, existem três botões que permitem uma visualização da informação de acordo com as preferências do utilizador, e uma *textBox* que indica se a base de dados locais já foi actualizada., Caso tenha sido actualizada, a aplicação mostra há quantos dias se realizou a última actualização e indica quantas notícias novas se encontram presentes por cada ponto de interesse escolhido.

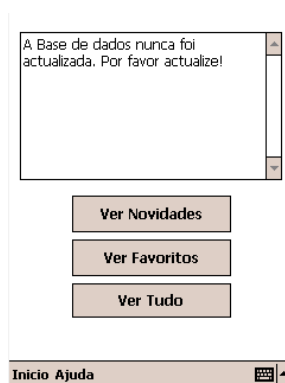


Figura 36 - Exemplo do formulário FormPub.cs na aplicação PubMoveI

No que diz respeito à visualização de notícias, foram definidos três modos de visualização, cada um permitindo proporcionar uma visualização mais confortável e adaptada às preferências do utilizador.

Assim, o botão “Ver Novidades” permite a visualização das notícias não lidas dos pontos de interesse que correspondem às preferências do utilizador. O modo como o utilizador pode definir os respectivos interesses encontra-se descrito mais adiante. A partir do momento em que a notícia passa ao estado *lida*, deixa de ser visível quando se selecciona o botão “Ver Novidades”.

O botão “Ver Favoritos” proporciona um modo de visualização semelhante ao modo anterior. Contudo, existe uma diferença: a possibilidade de visualização de notícias já lidas em conjunto com as notícias não lidas. No entanto, à semelhança do que sucede no caso anterior, apenas são visíveis as notícias correspondentes aos pontos de interesse seleccionados pelo utilizador.

O botão “Ver Tudo” permite apresentar todas as notícias, incluindo as notícias lidas, as notícias não lidas de todos os anunciantes existentes na base de dados local do dispositivo móvel, sem se filtrar a visualização de acordo com as preferências do utilizador.

Finalmente, e de modo a auxiliar o utilizador na compreensão da finalidade de cada botão, foi adicionado um botão “Ajuda” no startMenu, que mostra um *pop-up* com a descrição resumida de cada um dos botões existentes neste formulário.

4.1.1.2.2. *Formulário FormPubTipos.cs*

O formulário FormPubTipos permite a visualização dos tipos de pontos de interesse existentes, como é ilustrado na Figura 37. Trata-se de um formulário partilhado e dinâmico, pois os botões com os nomes dos tipos existentes são criados dinamicamente de acordo com o modo de visualização seleccionado no formulário FormPub.cs. Caso tenha sido escolhido o modo Ver Novidades ou o modo Ver Favoritos, irão surgir, neste formulário, botões para todos os tipos que possuem pontos de interesse com notícias. No caso de ter sido escolhido o modo de visualização “Ver Tudo” surgem botões para todos os tipos de pontos de interesse presentes na base de dados.

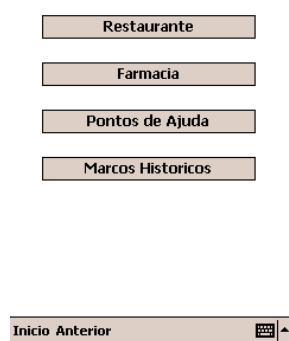


Figura 37 - Exemplo do formulário FormPubTipos.cs

4.1.1.2.3. Formulário *FormPubPoi.cs*

O formulário *FormPubPoi.cs* permite escolher um ponto de interesse para a visualização das respectivas notícias. Como se mostra na Figura 38, o nome do ponto de interesse surge directamente no botão, assim como a quantidade de notícias não lidas que estão relacionadas com esse ponto de interesse. À semelhança do que sucede no formulário *FormPubTipos.cs*, este formulário é partilhado pelos três modos de visualização de notícias referidos anteriormente.



Figura 38 - Exemplo do formulário *FormPubPoi.cs*

4.1.1.2.4. Formulário *FormPubPoiEscolha.cs*

O formulário *FormPubPoiEscolha.cs* permite escolher uma notícia referente ao ponto de interesse escolhido anteriormente no formulário *FormPubPoi.cs*. Uma vez mais, este formulário é igualmente partilhado pelos três modos de visualização, e adapta-se ao tipo de opção escolhida anteriormente pelo utilizador. Neste caso, para mostrar a informação respeitante à notícia escolhida, faz-se uma *query* à base de dados local do dispositivo móvel, em tudo semelhante à que se efectua no âmbito dos formulários descritos anteriormente (*FormPubTipos.cs* e *FormPubPoi.cs*). A informação retornada por esta interrogação á base de dados permite visualizar as notícias sob a forma de botões criados dinamicamente, como se mostra na Figura 39.

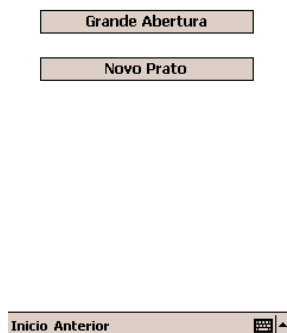


Figura 39 - Exemplo do formulário FormPubPoiEscolha.cs

4.1.1.2.5. Formulário FormPubNoticia.cs

Após todas as selecções de navegação terem sido feitas pelo utilizador, e logo que se tenha escolhido uma notícia para leitura, chega-se ao formulário FormPubNoticia.cs. Após a necessária *query* à base de dados, este formulário permite apresentar ao utilizador o conteúdo informativo da notícia escolhida.

Este formulário é constituído por *labels*, uma *textBox* e um botão, e permite a visualização dos seguintes parâmetros essenciais: o título da notícia, a data de inicio e fim da notícia, e ainda o corpo da notícia. Neste contexto, as *labels* mostram o título da notícia ou anúncio, assim como a respectiva duração ou validade (Data Início e Data Fim). Por outro lado, na *textBox* é possível visualizar o corpo da notícia. Por fim, o botão “Lido” efectua uma actualização da base de dados, marcando a notícia como lida. A Figura 40 mostra um exemplo do aspecto deste formulário.

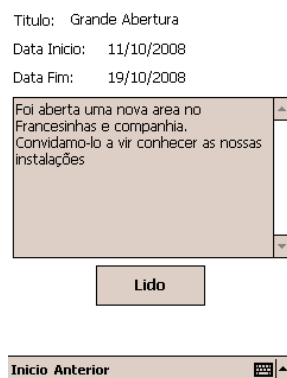


Figura 40 - Exemplo do formulário FormPubNoticia.cs

4.1.1.3. Formulários Pontos de interesse

Os formulários Pontos de interesse que serão mencionados em seguida correspondem aos nomes visíveis no mapa de navegação de acordo com a Figura 21, como Ponto de interesse e o seu subconjunto em árvore, Escolher Tipo, Escolher Ponto de Interesse e Detalhes.

4.1.1.3.1. Formulário *FormTipo.cs*

O formulário *FormTipo.cs*, ilustrado na Figura 41, mantém a mesma estrutura do formulário *FormPubTipos.cs* descrito anteriormente. Aqui faz-se uma *query* à base de dados e mostra-se, com o recurso a botões, os tipos de pontos de interesse existentes na base de dados local do dispositivo móvel.

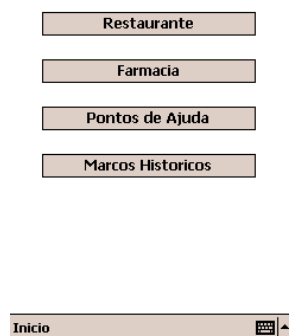


Figura 41 - Exemplo do formulário *FormTipo.cs*

4.1.1.3.2. Formulário *POI.cs*

Este formulário é semelhante ao formulário *FormPubPoi.cs* descrito anteriormente. Após ter sido seleccionado o tipo de ponto de interesse no formulário *FormTipo.cs*, também descrito anteriormente, é novamente feita uma *query* à base de dados, mostrando-se, através de botões, os pontos de interesse que pertencem ao subconjunto do tipo seleccionado. A Figura 42 mostra um exemplo da visualização do formulário em que se ilustra o subconjunto existente para o tipo principal “Restaurante”.



Figura 42 - Exemplo do formulário POI.cs

4.1.1.3.3. *Formulário PoiResult.cs*

Este formulário permite a visualização de toda a informação disponível sobre o ponto de interesse seleccionado no formulário anterior POI.cs, obtida a partir da realização de uma *query* à base de dados local do dispositivo móvel.

Devido à grande quantidade de informação disponível na base de dados, e tendo consideração as dimensões reduzidas do ecrã dos dispositivos móveis, recorreu-se à utilização do *tabControl* (`System.Windows.Forms.TabControl`) para se dividir a informação em separadores (normalmente designados por *tabs*). Em termos de organização, a informação é dividida e visualizada de acordo com os seguintes separadores:

- **Info:** um separador construído com base em *labels*, permitindo a apresentação do ponto de interesse e fornecendo informações do tipo nome, morada, local, telefone, o e-mail, qual o horário de funcionamento, qual o dia de descanso semanal e as coordenadas GPS.
- **Detalhes:** utilizando controlos do tipo *textBox* permite apresentar informações e detalhes sobre um ponto de interesse.
- **GPS:** permite a visualização de uma imagem originada pelo GoogleMaps baseada nas coordenadas do anunciante.
- **Noticias:** um separador construído com *labels*, botões e controlos do tipo *checkbox* (`System.Windows.Forms.Checkbox`) que permite accionar uma preferência sobre o ponto de interesse de modo notificar-se

automaticamente o utilizador sempre que existirem novidades. A notificação permitirá que o ponto de interesse esteja visível nos modos de visualização de notícias descritos anteriormente no formulário FormPub.cs. Note-se que a *checkbox* é dinâmica e muda o respectivo aspecto (marcado ou vazio) consoante o estado do respectivo registo da base de dados, como se ilustra na Figura 43. Finalmente, o botão existente *neste* separador permite que o utilizador seja direccionado para o formulário FormPubPoiEscolha.cs, de modo a permitir-lhe escolher e consultar as notícias associadas ao ponto de interesse seleccionado.



Figura 43 - Diferentes estados da checkBox do formulário PoiResult.cs

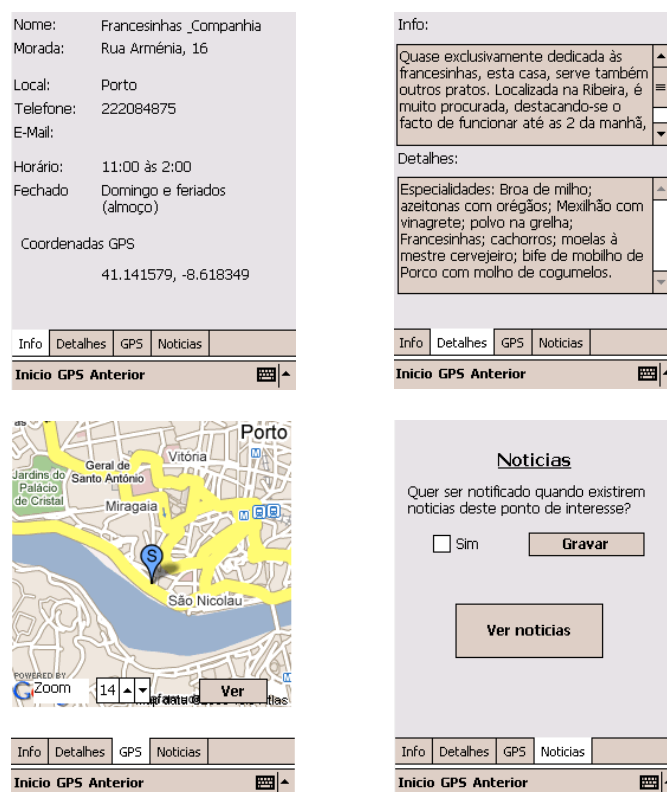


Figura 44 - Exemplo dos diferentes modos de visualização do PoiResult.cs

A Figura 44 mostra os diferentes separadores e estados do formulário PoiResult.cs. É ainda importante referir que no startMenu foi adicionado um botão “GPS” que permite a visualização do mapa GoogleMaps no *webBrowser* predefinido do WM, permitindo que o utilizador obtenha informações sobre o percurso a realizar para chegar até ao ponto de interesse seleccionado.

4.1.1.4. Formulários Favoritos

Os formulários Favoritos que serão mencionados em seguida correspondem aos nomes visíveis no mapa de navegação de acordo com a Figura 21, como Os Meus Favoritos e o seu subconjunto em árvore:

- Visualizar, Escolher Tipo, Escolher Favorito e Detalhes;
- Adicionar, Adicionar Tipo, Inserir Favorito, Inserir Detalhes;
- Actualizar, Alterar Nome do Tipo de Favoritos, Escolher o Tipo, Alterar o nome do Favorito e Alterar detalhes do Favorito;
- Eliminar, Eliminar Tipo, Escolher Tipo, Eliminar Favorito, Escolher Tipo e Eliminar Favorito.

4.1.1.4.1. Formulário Favorito.cs

A Figura 45 mostra o aspecto do formulário Favorito.cs. Este formulário funciona como o ecrã de apresentação e gestão dos favoritos, funcionando ainda como ponto de partida para que um utilizador visualize, adicione, actualize ou elimine os respectivos pontos de interesse favoritos. O formulário é constituído por quatro botões que encaminham o utilizador para os formulários correctos de acordo com a opção seleccionada.

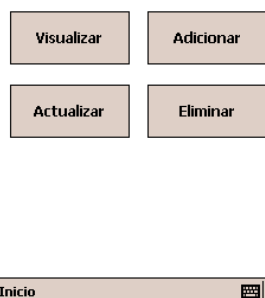


Figura 45 - Exemplo do formulário Favorito.cs

4.1.1.4.2. Formulário AdicionarFav.cs

Quando o utilizador selecciona a opção de adicionar um ponto de interesse favorito (botão “Adicionar” do formulário visível Figura 45), é direccionado para o formulário AdicionarFav.cs que se ilustra na Figura 46.



Tipo:

Nome:



Figura 46 - Exemplo do formulário AdicionarFav.cs

Este formulário é constituído por dois botões que permitem adicionar tipos de pontos de interesse (*Add Tipo*). Existe ainda outro botão, designado por “Inserir”, que permite fazer a inserção do novo favorito na base de dados. Para escolha do tipo é feita uma *query* à base de dados DB2.sdf, de modo a recolher todos os tipos de pontos de interesse existentes na base de dados. A informação resultante desta *query* é utilizada para popular uma *comboBox*, do tipo (System.Windows.Forms.ComboBox), que permite apresentar os tipos existentes de modo a que seja possível ao utilizador seleccionar facilmente o tipo pretendido para a inserção de um novo favorito. Para a inserção do nome do novo favorito é disponibilizada e uma caixa de texto.

Neste formulário o utilizador tem ainda a possibilidade de inserir o favorito escolhendo o tipo de favorito e introduzindo o respectivo nome. Caso o tipo do favorito correcto não se encontrar presente, o utilizador pode adicioná-lo à base de dados através do botão “Add Tipo”, que o remete para um formulário que permite a introdução de novos tipos de pontos de interesse, como se descreve a seguir.

4.1.1.4.3. Formulário *AddTipo.cs*

O formulário *AddTipo.cs*, ilustrado na Figura 47, permite a introdução de novas designações de tipos de pontos de interesse. Estes tipos são encarados como agregadores de favoritos, permitindo facilitar a gestão de pontos de interesse favoritos por parte do utilizador.

Toda a gestão dos tipo nos favoritos é efectuada por parte do utilizador. Procure ser sucinto para mais fácil

Ver Tipo já existentes

Nome do Tipo:

Guardar

Inicio Anterior

Figura 47 - Exemplo do formulário *AddTipo.cs*

Este formulário disponibiliza dois botões: o primeiro botão permite visualizar os tipos de pontos de interesse que se encontram registados na base de dados local “DB2.sdf” do dispositivo móvel. O outro botão permite realizar o registo de um novo nome de tipo de ponto de interesse, após a respectiva introdução na *textBox*. Faz-se aqui uma validação da informação omissa: caso não exista qualquer carácter escrito na *textBox*, a aplicação não permitirá a introdução de um novo registo, uma vez que este não possuiria uma designação.

4.1.1.4.4. Formulário *TiposExistentes.cs*

O formulário da Figura 48 é responsável pela visualização de todos os tipos de pontos de interesse favoritos existentes na base de dados local “DB2.sdf” do dispositivo móvel. Neste caso, trata-se de um formulário meramente informativo.

1: Restaurante

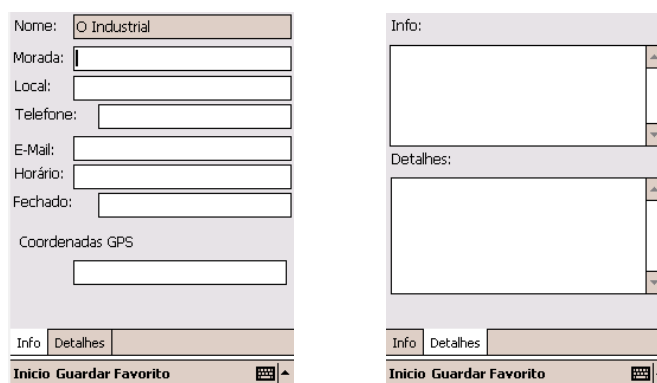
**Figura 48 - Exemplo do formulário TiposExistentes.cs**

4.1.1.4.5. *Formulário AddFavDetalhe.cs*

Após a introdução do nome do favorito no formulário AdicionarFav.cs, é apresentado ao utilizador o formulário AddFavDetalhe.cs. Este formulário é responsável pelo pedido de introdução de detalhes sobre o favorito introduzido pelo utilizador. Aqui, o utilizador preenche os detalhes relacionados com o ponto de interesse favorito a adicionar à base de dados, incluindo, entre outros, a morada, o local, o número de telefone para contacto.

Refira-se ainda que foi adicionado no StartMenu um botão “Guardar favorito” que permite a introdução e o registo dos detalhes associados ao ponto de interesse favorito na base de dados local do dispositivo móvel.

Considerando novamente a eficiência da apresentação de grandes quantidades de informação no ecrã de um dispositivo móvel, recorreu-se ao tabControl, como se ilustra na Figura 49.

**Figura 49 - Exemplo dos diferentes modos de visualização do formulário AddFavDetalhe.cs**

Na realidade, o formulário `AddFavDetalhe.cs` é um formulário que, à semelhança de outros formulários já mencionados anteriormente, possui uma dupla finalidade: permite a adição de um novo ponto de interesse favorito bem como a actualização dos dados de um ponto de interesse favorito já existente na base de dados.

A opção de actualização de favoritos pode ser escolhida através do botão “Actualizar”. No entanto, ao seleccionar esta opção, o utilizador terá, em primeiro lugar, que efectuar um conjunto de escolhas até que seja possível a edição dos detalhes associados ao ponto de interesse favorito. Esse conjunto de escolhas descreve-se com mais pormenor na secção que se segue.

4.1.1.4.6. *Formulário EditFav.cs*

Tendo introduzido um ponto de interesse favorito, o utilizador poderá julgar conveniente efectuar uma alteração no nome ou nos detalhes desse ponto de interesse favorito. Assim, e para que não seja forçado a eliminar o favorito, e ter que o criar novamente, foi desenvolvida uma área para actualizar os parâmetros de um determinado ponto de interesse favorito sem ser necessário eliminar o respectivo registo da base de dados.

Como foi mencionado anteriormente, ao seleccionar-se a opção “Actualizar” no formulário `Favorito.cs` o utilizador passa a dispor de permissões de actualização. O utilizador continua executar os mesmos passos na selecção do favorito à semelhança da selecção de pontos de interesse. Assim, faz-se um pedido para que o utilizador escolha o tipo de ponto de interesse favorito, seguido pela escolha do favorito propriamente dito. Caso o utilizador deseje apenas alterar os detalhes, é redireccionado para o formulário `AddFavDetalhe.cs` onde se realiza uma nova query à base de dados, apresentado todos os detalhes associados a esse favorito. Aí o utilizador pode efectuar as alterações pretendidas e guardá-las no mesmo registo da base de dados.

No entanto, o formulário `EditFav.cs` apenas permite a edição do nome dos tipos de favoritos ou o nome dos favoritos. Inicialmente mostram-se os tipos de favoritos que existem na base de dados, à semelhança do que sucede no caso do formulário `FormTipo.cs`. No entanto, neste caso adicionou-se à direita de cada tipo de ponto de

interesse um botão que permite a edição do nome dos tipos de pontos de interesse favoritos, tal como se mostra na Figura 50.

Para alterar outros parâmetros, utiliza-se o mesmo procedimento, basta seleccionar o botão correcto para se visualizarem os favoritos existentes na base de dados local que correspondam ao tipo seleccionado. A apresentação dos favoritos conta novamente com um botão “Editar”, para que seja possível editar o nome do favorito, como se mostra na Figura 51.



Figura 50 - Exemplo da edição de Tipos.

Figura 51 - Exemplo da edição de favoritos.

A partir do momento em que o botão “Editar” é seleccionado, o formulário EditFav.cs, adapta-se dinamicamente ao tipo de alteração a efectuar. Caso se trate da alteração do nome do tipo de ponto de interesse favorito, é apresentado numa *textBox* o texto correspondente à designação do tipo que se deseja alterar para edição, como se mostra se ilustra na Figura 52. Caso se trate da alteração do nome de um ponto de interesse favorito, apresenta-se o respectivo nome que se deseja alterar para edição, como está ilustrado na Figura 53. Neste caso foi igualmente adicionada uma *label* no formulário que mostra um texto de ajuda adaptado dinamicamente ao tipo de alteração a efectuar.

Alterar o nome do Tipo

Nota: Ao alterar o nome de um dos agrupadores de favoritos fará que todos os outros do mesmo tipo sejam vistos com o novo nome.

Restaurantes

Gravar

Início

Figura 52 - Exemplo de edição do tipo no formulário EditFav.cs.

Alterar o nome do Favorito

Nota: Vai alterar o nome de um dos seus favoritos.

O Industrial

Gravar

Início

Figura 53 - Exemplo de edição do nome do favorito no formulário EditFav.cs.

4.1.1.4.7. *Formulário RemovFav.cs*

Este formulário permite fazer a escolha de um tipo de favoritos ou de um favorito em particular, com o intuito de proceder à respectiva eliminação da base de dados local do dispositivo móvel.

Assim, ao seleccionar-se o botão “Eliminar” no formulário Favorito.cs, é apresentado ao utilizador o formulário removFav.cs, como se mostra na Figura 54.

Este formulário é constituído por uma *label* e dois botões, nomeadamente o botão “Remover Tipo” e o botão “Remover Favorito”, cada um deles implementando um comportamento diferente que se descreve com mais pormenor na secção que se segue.

Remover favoritos

Escolha se quer eliminar um conjunto/Tipo de favoritos ou se um favorito em particular

Remover Tipo

Remover Favorito

Início Anterior

Figura 54 - Exemplo do formulário RemovFav.cs.

4.1.1.4.8. Formulário *RemoveFavTipoFav.cs*

Em primeiro lugar, este formulário permite fazer a selecção do tipo de favoritos ou do favorito que se deseja eliminar.

Como se mencionou anteriormente, no formulário *RemovFav.cs* existem dois botões que, embora desencadeiem comportamentos diferentes na aplicação, possuem uma função comum, que é a de encaminhar o utilizador para o formulário *RemovFavTipoFav.cs*.

Assim, escolhendo o botão “Remover Tipo” é permitido ao utilizador escolher o tipo de favorito a eliminar. É importante notar que, após a selecção do tipo a eliminar, todos os favoritos que se encontram atribuídos com aquele tipo específico serão eliminados da base de dados local do dispositivo móvel. A Figura 55 mostra um exemplo deste formulário após a selecção do botão “Remover Tipo” no formulário *RemovFav.cs*.

No caso do botão “Remover Favorito” permite-se escolher um favorito específico para se eliminar. A Figura 56 mostra um exemplo deste formulário após a selecção do botão “Remover Favorito” no formulário *RemovFav.cs*. Uma vez mais, à semelhança do que sucede no caso da actualização de favoritos, é necessário que ocorra em primeiro lugar uma selecção do tipo de pontos de interesse favoritos para que seja possível chegar-se ao favorito correcto que se pretende eliminar.

Eliminar Tipo

Escolha o tipo de favoritos a eliminar.
ATENÇÃO ao eliminar um tipo vai eliminar todos os sub-conjuntos de favoritos da família do tipo

Restaurantes ▾

Eliminar


Início Anterior 

Figura 55 - Exemplo da eliminação de um tipo no formulário *RemoveFavTipoFav.cs*.

Eliminar Favorito

Escolha o Tipo do favorito para chegar até ao Favorito a eliminar. ATENÇÃO toda a informação do favorito escolhido irá ser eliminado

Restaurantes ▾

Eliminar


Início Anterior 

Figura 56 - Exemplo da eliminação de um favorito no formulário *RemoveFavTipoFav.cs*.

Finalmente, após a selecção do tipo de favorito, apresenta-se novamente o formulário FormTipo.cs. No entanto, na visualização dos favoritos é agora adicionado um botão (Eliminar) cuja selecção permite eliminar favorito da base de dados local do dispositivo móvel. A Figura 57 ilustra esse processo.



Figura 57 - Exemplo do FormTipo.cs em modo de eliminação de favoritos.

4.1.1.5. Opção Visualizar Favoritos

Para visualizar a lista de pontos de interesse favoritos existentes na base de dados local do dispositivo móvel reutilizou-se o conjunto de formulários de visualização dos pontos de interesse apresentados anteriormente. No entanto, neste caso estes formulários devem permitir a visualização dos favoritos e não dos pontos de interesse. Assim, a única diferença em relação aos formulários de visualização apresentados anteriormente é que, neste caso, trocou-se a base de dados “DB.sdf” pela base de dados “DB2.sdf”, tendo-se igualmente adaptado as *queries* de modo a que possibilitar a interpretação dos registos desta base de dados.

4.1.1.6. Classes auxiliares para o desenvolvimento da aplicação PubMovel.

4.1.1.6.1. Classe Program.cs

Esta classe apenas executa o método Main() que é responsável por executar a aplicação “PubMovel” através do Exemplo de código 3.

```
using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace App
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [MTAThread]
        static void Main()
        {
            Application.Run(new Form1());
        }
    }
}
```

Exemplo de código 3 - Classe Program, responsável por iniciar a aplicação.

4.1.1.6.2. Classe Database.cs

Para evitar que o mesmo código de acesso às bases de dados tenha de ser escrito várias vezes nos diversos formulários foi criada uma classe designada por Database.cs, cujo principal objectivo é o de agrupar o maior número de funções que são posteriormente disponibilizadas aos formulários que necessitam de as utilizar para implementar as funcionalidades correspondentes.

A Tabela 2 apresenta uma breve descrição das funções existentes nesta classe, acompanhadas pelo respectivo objectivo no âmbito do desenvolvimento da aplicação “PubMovel”.

Função	Descrição
selectTipo(String sql, String database)	Função que retorna um objecto do tipo <i>reader</i> para os formulários, de modo a disponibilizar o conjunto de tipos para os formulários que implementam escolhas. Permite ainda recolher os pontos de interesse que mereçam a preferência do utilizador e todas as informações referentes às notícias dos vários pontos de interesse.
selectPoi(int IdTipoFK, String database)	Função que retorna um objecto do tipo <i>reader</i> para o formulário contendo todos os registos, presentes na base de dados, de um ponto de interesse. A selecção desse ponto de interesse é feita através de um parâmetro (IdTipoFK) que o identifica e limita a query.
inserirTipo(String nomeTipo, String database)	Função que permite o registo de um novo tipo nas bases de dados locais do dispositivo móvel.
inserirRegistos(String sql, String database)	Função genérica que permite a inserção de registos nas bases de dados locais dos dispositivos móveis. A função permite que lhe passem sob a forma de um parâmetro o comando SQL a executar.
selectIdNome(String sql, String database)	Função que retorna um registo do tipo <i>inteiro</i> . Esta função é utilizada para a obtenção do registo do <i>IdNome</i> de modo a permitir a adição de favoritos.
SelectLink(String sql, String database)	Função genérica que retorna um registo do tipo <i>String</i> e que permite determinar o número de notícias não lidas que um determinado ponto de interesse possui, bem como determinar o URL que deve ser disponibilizado em certos formulários, como sucede, por exemplo, com o caso do formulário <i>ServerData.cs</i> .
InserirLink(String link)	Função que permite efectuar a actualização do

	registo do URL que permite realizar o <i>download</i> dos ficheiros de replicação da base de dados.
UpdateText(String sql, String database)	Função genérica que permite efectuar actualizações nos registos das bases de dados locais dos dispositivos móveis.
Delete(String sql, String database)	Função genérica que permite efectuar eliminações nos registos das bases de dados locais dos dispositivos móveis.

Tabela 2 - Tabelas descritiva das funções presentes na classe Database.cs

4.1.2. Implementação da base de dados da aplicação “PubMovel”

Para a criação e desenvolvimento das bases de dados da aplicação móvel “PubMovel” foi utilizada a ferramenta de desenvolvimento MS Visual Studio 2005. Assim, no projecto existente adicionou-se um novo item do tipo “*database file*” como se mostra na Figura 58.

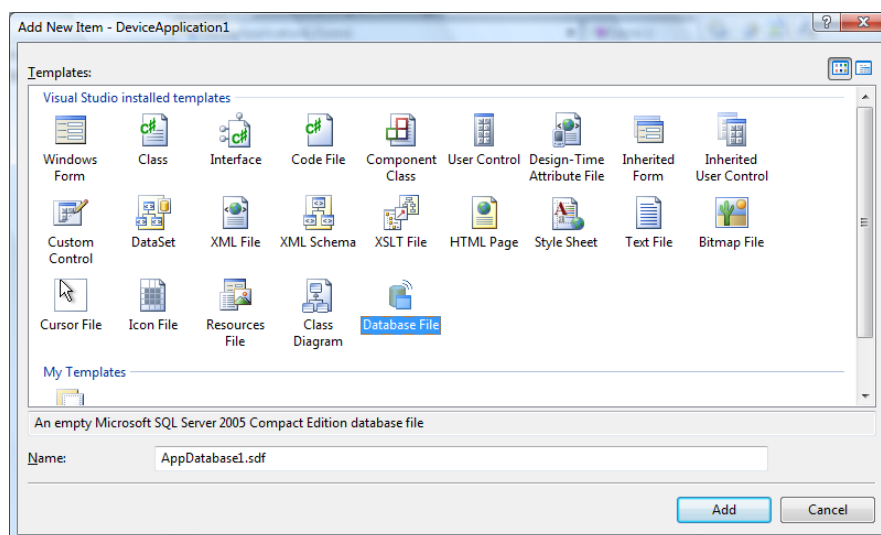


Figura 58 - Criação de uma base de dados no MS Visual Studio 2005

Em seguida, usando como ponto de partida as opções proporcionados pelo MS Visual Studio 2005, foram criados os objectos necessários para o desenvolvimento e criação das tabelas da base de dados. Assim, depois de se adicionar a base de dados ao projecto procedeu-se à criação das tabelas concebidas no capítulo 3 desta dissertação.

A Figura 59 e Figura 60 ilustram o processo de construção das tabelas da base de dados, incluindo a criação e a edição das propriedades de cada tabela.

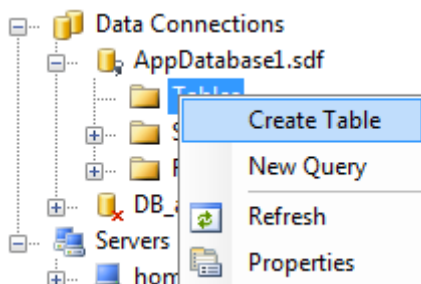


Figura 59 – Exemplo da criação de uma tabela.

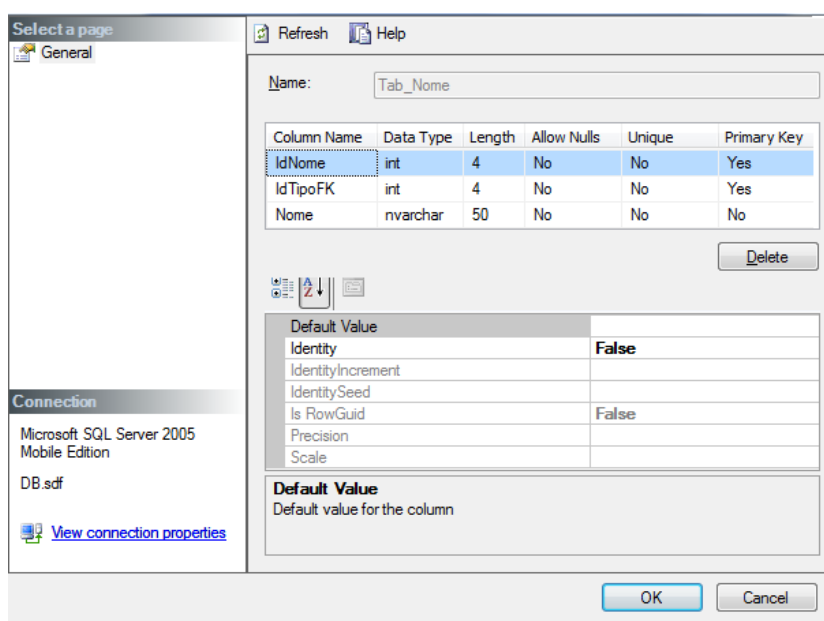


Figura 60 – Exemplo de edição da tabela.

4.2. Implementação da aplicação Web "NewsMobile"

Durante a implementação da aplicação "NewsMobile" foram criadas páginas dinâmicas com o recurso à linguagem de programação PHP, à linguagem CSS e à linguagem HTML. O objectivo principal foi a criação de um modelo que desse suporte à aplicação "PubMobile", isto é, que permitisse o registo de anunciantes, para que fosse possível darem-se a conhecer, bem como pudessem ter a possibilidade de anunciar as respectivas novidades através dos meios publicitários disponibilizados para o efeito.

As secções que se seguem detalham com algum pormenor o modo como foi implementada a aplicação “NewsMobile”.

4.2.1. Páginas da aplicação Web “NewsMobile”

Na implementação da aplicação *Web* “NewsMobile” foram utilizados formulários do tipo `<form method="post" action="">` tendo-se recorrido ao método POST para efectuar a passagem de dados de variáveis entre as várias páginas que constituem a aplicação, e que se descrevem nas secções que se seguem.

4.2.1.1. Página *Index.php*

Esta página é a página de apresentação da aplicação ao utilizador. Na construção desta página foi importante a utilização de um *template* CSS, para que a página inicial tivesse um aspecto atractivo e profissional do ponto de vista do utilizador, uma vez que o público-alvo desta aplicação é um público de natureza empresarial. Para este efeito, foi escolhido um *template* simples que permitisse orientar os utilizadores para a área central da página que foi dividida em duas áreas principais, tal como se ilustra na Figura 61.

A área da direita permite que o utilizador navegue na aplicação e a área da esquerda permite a visualização de dados e de instruções necessárias para a progressão da navegação pelos ecrãs da aplicação *Web*. A Figura 61 mostra um exemplo da página de entrada da aplicação “NewsMobile”.



Figura 61 - Exemplo da página de apresentação da aplicação NewsMobile.

Para efectuar o login, o utilizador necessita em primeiro lugar de fazer um registo na aplicação, utilizando a hiperligação “Criar Registo”. Mais abaixo, o utilizador encontra informação detalhada sobre o procedimento correcto para realizar o registo de um novo utilizador nesta aplicação.

No entanto, caso o utilizador já se encontre registado, apenas deverá introduzir as suas credenciais de modo a utilizar a aplicação em modo de edição. Poderá fazê-lo usando as duas caixas de texto na área do lado direito desta página. A primeira caixa de texto serve para a introdução do *username*, tratando-se de um controlo do tipo `<input type="text">`, a segunda caixa de texto serve para a introdução da *password do utilizador*, sendo do tipo `<input type="password">`. A única diferença entre os dois controlos é que, no caso da *password*, os caracteres introduzidos pelo utilizador são escondidos da visualização.

Quando o utilizador pressiona o botão de *login* é feita uma validação das suas credenciais. Caso tenha introduzido os valores correctamente é autorizado a utilizar a aplicação, caso contrário o utilizador é novamente reencaminhado para a página `Index.php` onde lhe é apresentada uma mensagem de erro de *login*. Explica-se a seguir o modo como essas validações são realizadas pela aplicação.

4.2.1.2. Página AddTipoReg.php

Para que o utilizador disponha das permissões necessárias para utilizar a aplicação “NewsMobile” é necessário efectuar o registo na aplicação, como se afirmou anteriormente. Assim, para se efectuar um registo de novo utilizador usando a hiperligação disponível na página Index.php, o utilizador é encaminhado para a página AddTipoReg.php como se mostra na Figura 62.

Nessa página pergunta-se o tipo de ponto de interesse em que o utilizador se deseja registar. Em seguida, é feita uma *query* à base de dados e o resultado dessa *query* é utilizado para popular um componente *List/Menu* do tipo `<select name="">`. Caso não exista ainda um tipo registado, é dada a possibilidade de o utilizador efectuar a introdução de um novo registo de tipo recorrendo à hiperligação “Adicionar novo tipo”.



Indique qual o ponto de interesse em que deseja ficar registado

Tipo: Restaurantes

Seguinte

Caso não exista o seu tipo por favor adicione. [Adicionar novo Tipo](#)

Pontos de interesse já registados

- Cafés
- Cinemas
- Farmácias
- Marcos Historicos
- Pontos de ajuda
- Restaurantes
- centros comerciais

Voltar

Figura 62 - Exemplo da página AddTipoReg.php.

Na página AddTipoReg.php disponibiliza-se ainda um *output* do conjunto de tipos já registados na base de dados de modo permitir ao utilizador verificar qual o tipo a escolher.

4.2.1.3. Página AddTipo.php

A página “AddTipo.php” foi desenvolvida para permitir a introdução de novos tipos na base de dados da aplicação “NewsMobile”. Para esse efeito foi criado um formulário usando uma *textBox* e um botão de acção que permite a adição de novos nomes de tipos na base de dados. A Figura 63 mostra um exemplo de utilização da página AddTipo.php.

Finalmente, após a introdução do nome do tipo de ponto de interesse faz-se um *INSERT* na tabela Tab_Tipo.



Por favor adicione o tipo na qual se deseja dar a conhecer.

Atenção: Qualquer tipo novo que seja adicionado será posteriormente analisado pelo administrador do portal e se necessário será modificado para que a aplicação contenha o mínimo de tipos específicos. Por isso é pedido que seja generico no tipo.

Tipo:

Copyright - 2008 Universidade Fernando Pessoa - All Rights Reserved. Blue-Skyscraper is a free template by Webdesign MediaUp

Figura 63 - Exemplo da página AddTipo.php.

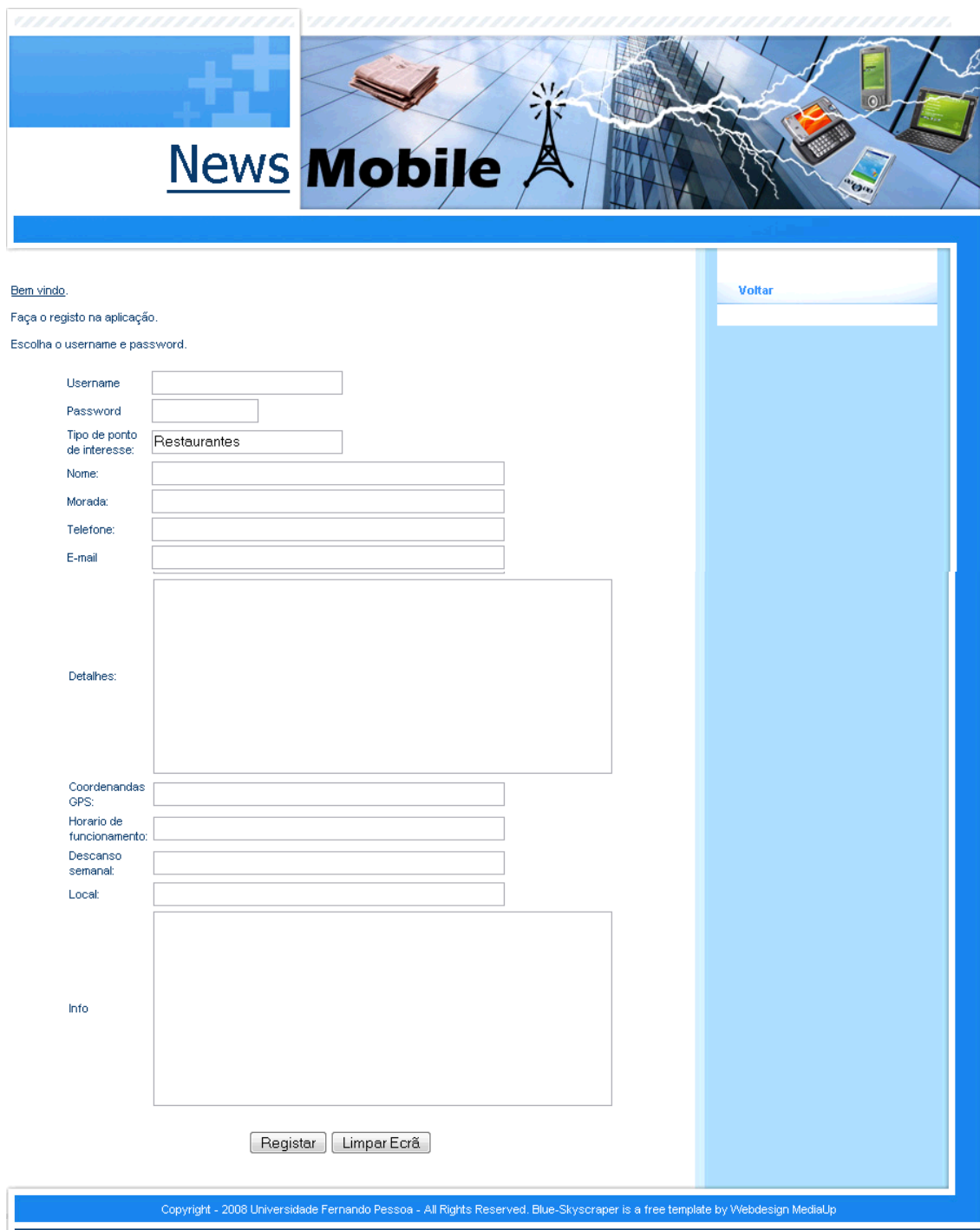
Logo que se finalize a introdução do tipo, o utilizador é encaminhado novamente para a página AddTipoReg.php, onde poderá então seleccionar um tipo para continuar com o registo. Feita a selecção, apresenta-se ao utilizador a página RegUsername.php cuja funcionalidade se descreve a seguir.

4.2.1.4. Página RegUsername.php

Após ter sido escolhido o tipo de ponto de interesse, disponibiliza-se ao utilizador uma série de parâmetros que deverão preencher para que fique completo o registo na

aplicação. Os parâmetros a preencher incluem, entre outros, os seguintes: *username*, *password*, nome, morada, como se mostra na Figura 64.

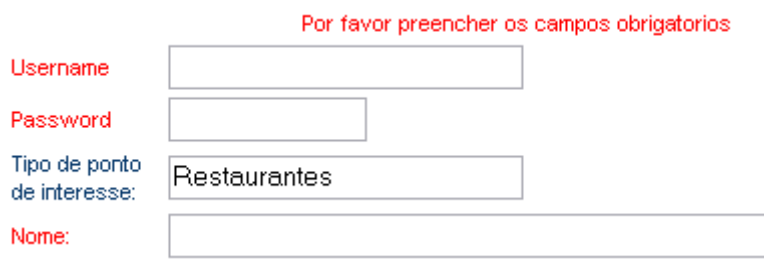
Tendo terminado a introdução de informação em todos os campos ou nos campos essenciais, tais como o *username*, a *password* e o nome, efectua-se o *POST* do formulário de modo a guardar todos os valores dos parâmetros inseridos pelo utilizador.



The image shows a web registration form for 'News Mobile'. The header features a blue banner with the text 'News Mobile' and a background image of a mobile phone, a laptop, and a tower. Below the banner, the form is divided into two main sections. The left section contains a registration form with the following fields: 'Username', 'Password', 'Tipo de ponto de interesse:' (with 'Restaurantes' selected), 'Nome:', 'Morada:', 'Telefone:', 'E-mail', 'Detalhes:', 'Coordenadas GPS:', 'Horario de funcionamento:', 'Descanso semanal:', and 'Local:'. The right section is a vertical sidebar with a 'Voltar' button at the top. At the bottom of the form, there are two buttons: 'Registrar' and 'Limpar Ecrã'. The footer contains the text: 'Copyright - 2008 Universidade Fernando Pessoa - All Rights Reserved. Blue-Skyscraper is a free template by Webdesign MediaUp'.

Figura 64 - Exemplo da página RegUsername.php.

No entanto, se, por lapso, o utilizador cometer algum erro na introdução dos valores de registo, ou casose esqueça de preencher os parâmetros essenciais referidos anteriormente, a aplicação valida os valores introduzidos, e a página RegUsername.php alerta para os erros cometidos pelo utilizador, como se mostra a Figura 65.



The image shows a registration form with the following elements:

- A red error message at the top: "Por favor preencher os campos obrigatorios".
- A label "Username" in red next to an empty text input field.
- A label "Password" in red next to an empty text input field.
- A label "Tipo de ponto de interesse:" in blue next to a dropdown menu showing "Restaurantes".
- A label "Nome:" in red next to a text input field.

Figura 65 - Exemplo de erro da página RegUsername.php.

Para facilitar a correcção dos erros, todos os dados introduzidos antes de o erro suceder são apresentados novamente no formulário. Isso simplifica o processo para o utilizador na medida em que implica uma redução quer no tempo dispendido no registo quer no tempo gasto na correcção de erros.

4.2.1.5. Página RegUserTipo.php

Após se verificar a introdução correcta dos dados referentes à criação do utilizador, o formulário da página RegUsername.php envia todas as variáveis para a página RegUserTipo.php. A função desta página é efectuar o registo de todos os dados fornecidos pelo utilizador, recorrendo, para o efeito, à utilização de transacções

A sequência de registo usando transacções é pois a seguinte:

- Efectuar a ligação à base de dados;
- Iniciar a transacção;
- Inserir o parâmetro “nome” na tabela Tab_Nome;
- Inserir os parâmetros de *login*, *username* e *password* na tabela *Login*;
- Inserir os parâmetros associados aos restantes detalhes, tais como, entre outros, a morada, o número de telefone, na tabela Tab_Detalhe;

- Caso todos os *INSERTS* tenham tido sucesso, efectuar o *COMMIT* e apresentar a página *Index.php*. Caso contrário, efectuar o *ROLLBACK* das alterações e apresentar novamente a página *RegUsername.php*;
- Fechar a ligação à base de dados.

4.2.1.6. Página *Main.php*

Mal o utilizador introduz as credenciais na página *Index.php*, essas credenciais são “enviadas” usando os formulários para a página *Main.php*. Essas duas variáveis são então utilizadas para efectuar a verificação de utilizador registado. Nessa altura faz-se uma nova *query* à base de dados da aplicação. Caso a validação falhe, o utilizador, como mencionado anteriormente, é novamente enviado para a página *Index.php* usando o comando PHP *header('location:index.php?error=1')*. Caso a validação tenha sucesso, faz-se mais uma *query* à base de dados de modo a permitir obter o nome do utilizador para que seja mostrado no ambiente de trabalho do utilizador, como se apresenta na Figura 66.



Figura 66 - Exemplo da página *Main.php*

4.2.1.7. Página Conta.php

Esta página foi desenvolvida para dar o suporte ao nível de gestão de conta do utilizador. A página é meramente informativa e permite a navegação, usando hiperligações, para outras páginas de controlo, tais como a mudança da *password* da conta, a mudança de detalhes da conta e a eliminação da conta. Cada uma das hiperligações mencionadas será descrita nas secções que se seguem. A Figura 67 mostra um exemplo de uma página Conta.php. Note-se pois que a página Conta.php é essencialmente uma variante da página Main.php, Em tudo semelhante àquela página apenas excepto na substituição das hiperligações da área dinâmica da página.



Figura 67 - Exemplo da página Conta.php

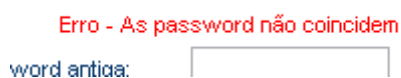
4.2.1.8. Página MudarPwd.php

O controlo do utilizador sobre as respectivas credenciais foi tomado em consideração no desenvolvimento da aplicação “NewsMobile”. Para a gestão de *passwords* foi criada a página MudarPwd.php que permite a alteração da *password* actual por uma nova. A Figura 71 mostra um exemplo da página.

No desenvolvimento da página, foi criado um formulário que apresenta três `textBox` do tipo (`<input type="password">`). A primeira é destinada para a *password* actual, as restantes são destinadas para a nova *password*.

No desenvolvimento da página foi criado um formulário que apresenta três controlos `textBox` do tipo `<input type="password">`. O primeiro controlo destina-se à introdução da *password* actual, os restantes controlos destinam-se à introdução de uma nova *password*.

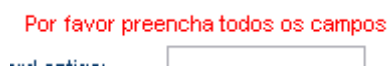
O motivo subjacente à criação de dois controlos `textBox` para a introdução de uma nova *password* é o facto de permitir ao utilizador um controlo mais preciso sobre a nova *password* evitando a ocorrência de enganos. Assim, se apenas existisse um campo para actualizar a *password*, a mudança da *password* poderia estar sujeita a um erro de escrita. Neste caso, sendo necessário introduzir a nova *password* em duas caixas de texto permite efectuar uma comparação entre o texto introduzido em ambas as `textBox`. Caso a comparação seja verdadeira, dá-se permissão sobre a alteração. Caso seja falsa, envia-se um aviso como se mostra na Figura 68.



Erro - As password não coincidem
word antiga:

Figura 68 - Exemplo de erro na comparação de passwords.

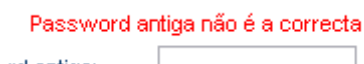
Para controlar a autenticação, apenas é possível efectuar a alteração de *password* inserindo inicialmente a *password* actual seguido de uma nova *password* e da respectiva confirmação. Caso não sejam preenchidos todos os campos é emitido um aviso, tal como ilustra na Figura 69.



Por favor preencha todos os campos
word antiga:

Figura 69 - Exemplo de erro na falta de preenchimento de campos.

Note-se, no entanto, que é ainda emitido um aviso sempre que exista uma falha na introdução da *password* actual, como se mostra na Figura 70.



Password antiga não é a correcta
word antiga:

Figura 70 - Exemplo de erro na falha da introdução da password actual.



The image shows a web page with a blue header and a main content area. The header features the text "News Mobile" in a large, bold font, with "News" in blue and "Mobile" in black. To the right of the text is a graphic of a mobile phone tower and various mobile devices. Below the header, the main content area has a white background with a blue border. It contains a form for changing a password. The form has the following elements:

- A heading: "Escolha a sua nova password."
- Three input fields labeled "Password antiga:", "Nova Password:", and "Confirmar Password:".
- A button labeled "Alterar" below the input fields.
- A button labeled "Voltar" in the top right corner of the form area.

At the bottom of the page, there is a footer with the text: "Copyright - 2008 Universidade Fernando Pessoa - All Rights Reserved. Blue-Skyscraper is a free template by Webdesign MediaUp".

Figura 71 - Exemplo da página MudarPwd.php.

Logo que todas as validações tenham sido efectuadas com sucesso, isto é, sempre que a nova password é comparada com sucesso, e quando a password actual é comparada com sucesso com a password obtida pela query à base de dados da aplicação, faz-se um UPDATE à base de dados de modo a ser actualizada a nova password introduzida pelo utilizador.

4.2.1.9. Página DetalhesConta.php

Quando um utilizador faz o registo de conta, fornece alguns dados que são importantes para permitir a realização de pesquisas por parte do utilizador da aplicação “PubMove!”. No entanto, com o decorrer do tempo, ou mesmo por lapso na introdução inicial, os dados poderão caducar ou mesmo ficar incorrectos. Assim, para permitir a correcção desses dados foi criada a página DetalhesConta.php, que permite efectuar actualizações aos parâmetros de conta do anunciante. Na Figura 72 mostra-se um exemplo da utilização desta página.

News Mobile

Tipo de ponto de interesse:

Nome:

Morada:

Telefone:

E-mail:

Detalhes:

Coordenadas GPS:

Horario de funcionamento:

Descanso semanal:

Local:

Info:

Voltar

Alterar

Copyright - 2008 Universidade Fernando Pessoa - All Rights Reserved. Blue-Skyscraper is a free template by Webdesign MediaUp

Figura 72 - Exemplo da página DetalhesConta.php.

Em termos de organização, a página é constituída por controlos *textBox* do tipo `<input type="text">`, e o seu funcionamento é semelhante ao funcionamento da página `RegUsername.php` descrita anteriormente. De facto, esta página possui o mesmo tipo de formulário. No entanto, existe uma *query* inicial à base de dados local do servidor que serve para popular todas as caixas de texto com a informação da conta do anunciante, de modo a permitir a visualização da informação actual para se efectuar a actualização necessária.

4.2.1.10. Página DelConta.php

Com o decorrer do tempo, os utilizadores finais da aplicação “NewsMobile” poderão deixar de pretender usufruir do serviço, e de manter uma conta aberta na aplicação. Neste caso será necessário proceder à eliminação da conta. Para esse efeito foi criada a página DelConta.php que permite, após dupla validação, eliminar da base de dados uma conta existente.

Mais especificamente, usando-se a área de navegação já mencionada anteriormente, o utilizador poderá chegar até à página DelConta.php, que, como se mostra na Figura 73 permite que um utilizador elimine uma conta.

A página integra um formulário que contém um botão cuja selecção remete novamente para a mesma página DelConta.php. No entanto, desta vez integrando um novo pedido de eliminação de conta. Este procedimento foi implementado para acautelar a selecção inadvertida do botão e a consequente eliminação da conta.



Figura 73 - Primeira fase de eliminação da conta na página DelConta.php.

Assim, apresenta-se ao utilizador um novo formulário reforçando o direito de eliminação de conta. A Figura 74 mostra um exemplo da página contendo o segundo formulário. Se o utilizador seleccionar novamente o botão “Eliminar Conta”, a conta será definitivamente removida da base de dados e todos os dados relativos a essa conta serão igualmente eliminados.



Figura 74 - Segunda fase de eliminação da conta na página DelConta.php.

Caso o utilizador, não deseje eliminar a conta, para sair desta página, basta usar a área de navegação seleccionando a hiperligação “Voltar”.

4.2.1.11. Página Publicidade.php

Um dos principais objectivos do desenvolvimento da aplicação “NewsMobile”, para além de permitir que um anunciante se dê a conhecer, é a possibilidade da publicação de novas notícias.

Para isso foi criada uma página designada por Publicidade.php, que permite efectuar o controlo e a gestão sobre os textos publicitários publicidades. A Figura 75 mostra um exemplo desta página.



Figura 75 - Exemplo da página Publicidade.php.

É precisamente através da página Publicidade.php que o utilizador ganha acesso à criação de novos textos publicitários, sejam anúncios ou notícias, bem como à respectiva actualização, visualização e eliminação. Descreve-se a seguir o modo como forma desenvolvidas as páginas que permitem a manipulação dos textos publicitários.

4.2.1.12. Página InsPub.php

No que diz respeito à criação dos textos publicitários usando a área de navegação da página Publicidade.php, o utilizador deve seleccionar a hiperligação “Inserir Publicidade”, sendo posteriormente encaminhado para a página InsPub.php que se mostra na Figura 76, sendo constituída por um formulário contendo um botão de acção, um conjunto de caixas de texto e listas do tipo *List/Menu* `<select name="">`.



The image shows a web interface for 'News Mobile'. At the top, there is a banner with the text 'News Mobile' and an illustration of a mobile phone, a laptop, and a radio tower. Below the banner is a form with the following fields:

- Título da notícia:
- Data Inicio: Dia: , Mes: , Ano:
- Data Fim: Dia: , Mes: , Ano:
- Notícia:

At the bottom of the form is a 'Guardar' button. On the right side of the form, there is a 'Voltar' button.

Figura 76 - Exemplo da página InsPub.php

Para introduzir um texto publicitário na base de dados o utilizador necessita de preencher os campos do formulário, incluindo o título, a data de início da notícia, a data de fim da notícia e o texto relacionada com a própria notícia.

Caso os campos não sejam introduzidos correctamente, ocorre uma validação, e a página proporciona um aviso sobre o modo como se deve proceder para corrigir a informação introduzida, como é o caso do exemplo que se mostra na Figura 77, em que os parâmetros não foram totalmente introduzidos, faltando o título ou mesmo o corpo da notícia.

Por favor preencha todos os campos no formulario

Figura 77 - Exemplo da falha de campos preenchidos na página InsPub.php

No entanto, não é somente na falha de campos preenchidos que pode existir um engano. Por exemplo, o utilizador, ao recorrer às listas dia, mês e ano, pode inadvertidamente colocar uma data de início posterior a data de fim da notícia que pretende introduzir. Nesta situação faz-se uma verificação e valida-se a introdução dos dados, avisando o utilizador do problema com as datas introduzidas, como se mostra na Figura 78.

Data invalida. Por favor verifique a data, pois a Data Fim é anterior a Data Inicio

Figura 78 - Exemplo do aviso de data posterior na página InsPub.php

Caso a informação introduzida em todos os campos esteja de acordo com as regras adequadas ao preenchimento do formulário, faz-se finalmente a inserção do texto publicitário na base de dados.

4.2.1.13. Página VerPub.php

Após a introdução de novos textos publicitários pode ser necessário visualizar as notícias ou anúncios introduzidos. Para este efeito adicionou-se uma página à aplicação “NewsMobile” que permite que o utilizador visualize as notícias que publicou. Na realidade, trata-se de uma funcionalidade muito útil, na medida em que permite que o anunciante se mantenha a par das notícias que publicou, e efectue correcções de erros na publicação que entretanto forem detectados. A Figura 79 mostra um exemplo do aspecto que a página VerPub.php apresenta ao utilizador.

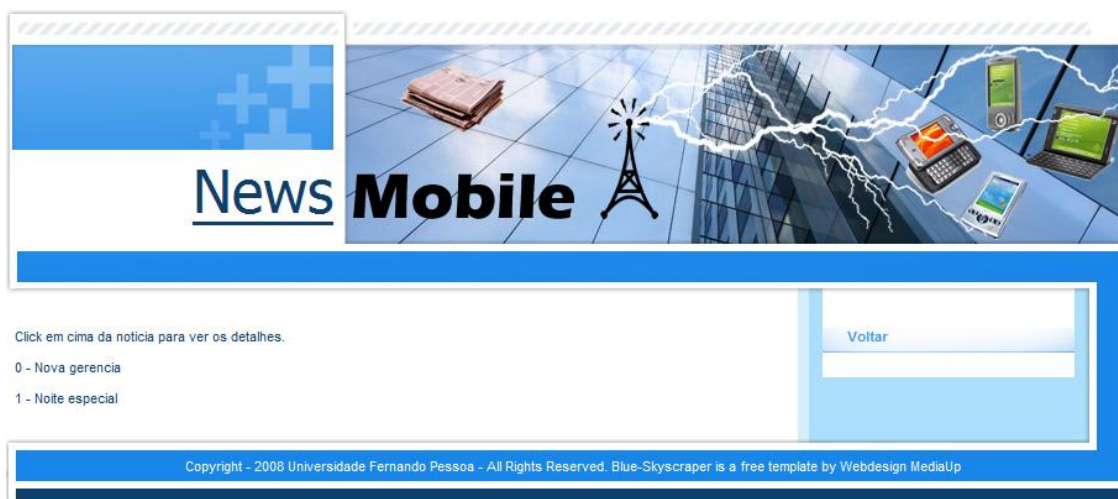


Figura 79 - Exemplo da página VerPub.php.

A visualização das notícias requer que se efectue uma *query* à base de dados. Tal *query* retorna os títulos das notícias que o anunciante possui na base de dados. Juntamente com esse registo (qual registo? Não se percebe) são criadas hiperligações que, ao serem seleccionadas, permitem que o utilizador visualize os detalhes da notícia.

4.2.1.14. Página DeletePub.php

Caso um anunciante julgue conveniente eliminar uma determinada notícia será necessário recorrer à página DeletePub.php disponibilizada pela aplicação “NewsMobile”.

Assim, usando novamente a área de navegação, o utilizador pode facilmente seleccionar a hiperligação “Eliminar Publicidade” que conduz à apresentação da página VerPub.php que contém uma listagem de todos os textos publicitários criados por um determinado anunciante. Note-se que tal listagem, muito embora seja visível na página VerPub.php, é diferente da listagem mencionada anteriormente, na medida em que as hiperligações são criadas de modo a permitir a abertura das notícias na página DeletePub.php.

Por sua vez, a página DeletePub.php é semelhante à página InsPub.php já descrita, com a excepção de três aspectos.

Em primeiro lugar, ao ser carregada a página DeletePub.php efectua-se uma *query* à base de dados que recolhe todos os registos de acordo com a notícia escolhida na página VerPub.php, de modo a que todos os campos da página DeletePub.php sejam preenchidos com a finalidade de permitir que o utilizador veja a notícia que vai eliminar.

Em segundo lugar, o botão “Eliminar” existente no formulário permite executar um comando SQL que irá eliminar a notícia escolhida na base de dados da aplicação “NewsMobile”.

Por fim, a utilização das listas do tipo *List/Menu <select name="">*, na área de “Data Início” e “Data Fim” foram substituídas por caixas de texto. A Figura 80 mostra um exemplo da utilização da página DeletePub.php.



Figura 80 - Exemplo da página DeletePub.php.

4.2.1.15. Página Noticia.php

Da aplicação “NewsMobile” foram já descritas as páginas InsPub.php e DeletePub.php, que permitem a inserção e a eliminação de uma notícia. No entanto, o utilizador poderá igualmente visualizar as notícias que se encontram presentes na aplicação. Para visualizar as notícias desenvolveu-se a página Noticia.php que é semelhante às páginas InsPub.php e DeletePub.php, já descritas anteriormente.

Para visualizar o texto publicitário, o utilizador apenas deverá usar a área de navegação, seleccionando a hiperligação “Ver Publicidade”. Esta hiperligação conduz à página VerPub.php, já descrita anteriormente. Assim, faz-se uma nova *query* à base de dados, apresentando, sob a forma de uma listagem contendo hiperligações, as notícias existentes. No entanto, a selecção de uma determinada notícia ou anúncio publicitário irá encaminhar o utilizador para a página Noticia.php, permitindo deste modo que o utilizador visualize a notícia publicada. Na Figura 81 é possível observar um exemplo de utilização da página Noticia.php.



Figura 81 - Exemplo da página Noticia.php.

À semelhança da página DeletePub.php, a página Noticia.php apresenta a notícia em caixas de texto. A única diferença entre as duas páginas é a seguinte: na página Noticia.php não se utilizam formulários nem botões porque a página Noticia.php é meramente informativa, não existindo a necessidade de se realizar a passagem de informação.

4.2.1.16. Página ActPub.php

Tendo já sido descrita a funcionalidade de inserção de notícias, bem como a respectiva eliminação e visualização, falta apenas referir que na aplicação “NewsMobile” é ainda possível actualizar uma determinada notícia.

Assim, desenvolveu-se a página ActPub.php para facilitar a gestão e a manipulação das notícias já existentes. Na verdade, seria bastante incómodo para o utilizador se fosse necessário eliminar uma notícia, ter de a introduzi-la novamente apenas porque ocorreu um erro ou um engano num único ponto, como por exemplo numa data, no título, ou mesmo no corpo da notícia.

Neste caso, seleccionando-se a hiperligação “Actualizar Publicidade” encaminha o utilizador para a página de selecção de notícias, onde se permite escolher a notícia a actualizar. Neste caso, faz-se uma vez mais separação das páginas, e mostra-se ao utilizador a página ActPub.php.

A página ActPub.php é semelhante à página InsPub.php, possuindo as mesmas caixas de texto, as mesmas listas e o mesmo tipo de formulário, sendo igualmente semelhante à página DeletePub.php. Neste caso, faz-se uma *query* à base de dados que permite a visualização dos dados da notícia para que seja possível a sua edição e respectiva correcção.

Após efectuar a alteração, o utilizador apenas deve seleccionar o botão “Guardar” para que seja executado um comando SQL do tipo UPDATE ao registo correspondente à notícia editada. A Figura 82 mostra um exemplo de utilização da página ActPub.php.



The screenshot displays the 'News Mobile' website interface. At the top, there is a banner with the text 'News Mobile' and an image of a mobile phone, a laptop, and a tower. Below the banner, the form is divided into several sections:

- Título da noticia:** A text input field containing 'Nova gerencia'.
- Data Inicio:** Three dropdown menus for 'Dia' (01), 'Mes' (01), and 'Ano' (2009).
- Data Fim:** Three dropdown menus for 'Dia' (05), 'Mes' (01), and 'Ano' (2009).
- Noticia:** A large text area containing the text: 'O restaurante Cosa Rio vai reabrir com nova gerência. Venha experimentar as novas especialidades da casa.'
- Buttons:** A 'Guardar' button at the bottom of the form and a 'Voltar' button on the right side.

Figura 82 - Exemplo da página ActPub.php.

4.2.1.17. Páginas em modo admin

De modo a assegurar que o utilizador final da aplicação “PubMovel” receba sempre um número mínimo de notícias expiradas criou-se um modo de administração na aplicação “NewsMobile” com o objectivo de permitir o controlo do prazo de validade das várias notícias.

O modo de administração é constituído por três páginas: Index.php, Main.php e Limpeza.php. estas páginas implementam um aspecto gráfico e um modo de navegação em tudo semelhante ao modo implementado na aplicação “NewsMobile”. Descreve-se em seguida com algum pormenor a forma como estas páginas foram implementadas.

4.2.1.18. Página Index.php (Admin)

Em primeiro lugar, a página Index.php disponibilizada em modo de administração varia muito pouco da página Index.php que foi descrita anteriormente. Na verdade, o aspecto gráfico mantém-se, existindo as mesmas áreas estáticas e dinâmicas que foram definidas anteriormente.

Como se mostra na Figura 83, não existe contudo qualquer hiperligação para a criação de novos utilizadores. Assim, apenas se disponibilizam duas caixas de texto para a introdução do *username* e *password* do administrador, e um botão “Login” que pertence ao formulário de validação de utilizador.



Figura 83 - Exemplo da página Index.php em modo Administrador.

4.2.1.19. Página Main.php (Admin)

Em segundo lugar, logo que o administrador do portal introduza correctamente as respectivas credenciais, a aplicação apresenta a página Main.php que é semelhante à página Main.php descrita anteriormente.

Assim, esta página proporciona o mesmo tipo de área de navegação, mas neste caso encontra-se restrita apenas à limpeza de notícias e à possibilidade de terminar a sessão “Logout”, como se mostra na Figura 84.



Figura 84 - Exemplo da página Main.php em modo administrador.

4.2.1.20. Página Limpeza.php (Admin)

Finalmente, a página Limpeza.php é constituída por um formulário que contém três listas para a introdução de datas, nomeadamente para o Dia, o Mês e o Ano. Na abertura da página apresenta-se a data actual e as listas são preenchidas automaticamente com a data actual.

O objectivo principal desta página é o de facilitar a eliminação da base de dados dos registos das notícias que sejam anteriores à data actual, ou que possuam uma data pré-definida pelo administrador. A Figura 85 mostra um exemplo da utilização da página Limpeza.php.



Figura 85 - Exemplo da página Limpeza.php.

4.2.2. Base de dados da aplicação “NewsMobile”

A criação e manipulação da base de dados da aplicação “NewsMobile” recorreu a uma ferramenta designada por phpMyAdmin que é instalada juntamente com o WAMP. Esta ferramenta permitiu acelerar a criação e o preenchimento inicial da base de dados da aplicação “NewsMobile”, de acordo com as tabelas concebidas no capítulo 3, tal como se ilustrou na Figura 26.

A Figura 86 mostra a página principal da ferramenta phpMyAdmin.

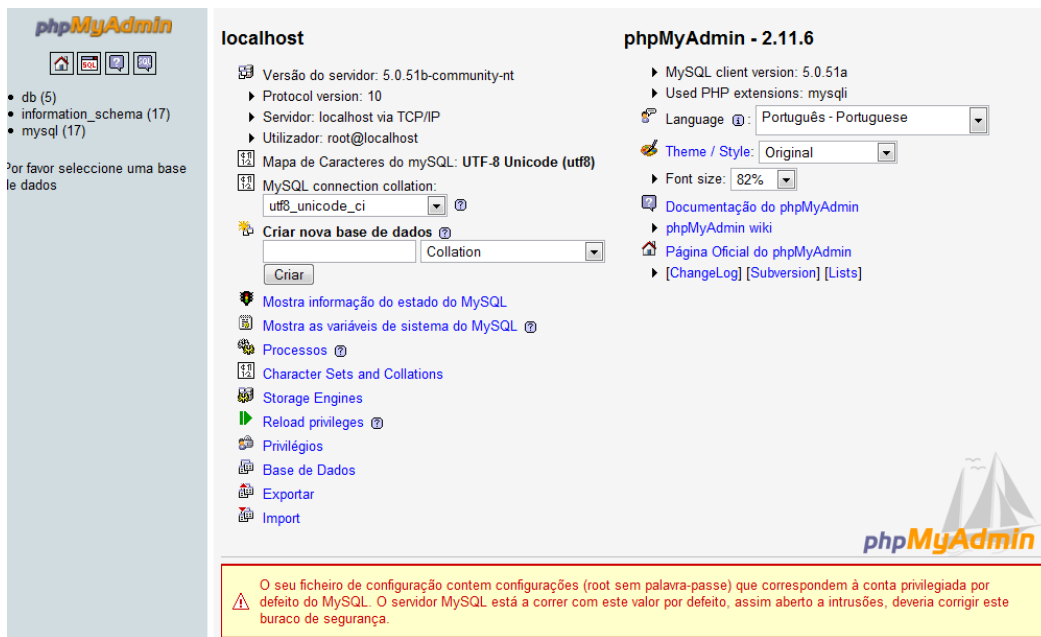


Figura 86 - Exemplo da utilização aplicação phpMyAdmin.

A Figura 87 e Figura 88 ilustram o modo como a ferramenta phpMyAdmin permite visualizar as tabelas criadas para cada base de dados, bem como o modo como se pode efectuar a respectiva edição e, ainda, a possibilidade de gerir os registos existentes nas várias tabelas.

	Tabela	Acções	Registos	Tipo	Collation	Tamanho	Suspensão
<input type="checkbox"/>	login		~16	InnoDB	utf8_unicode_ci	48.0 KB	-
<input type="checkbox"/>	tab_detalle		~16	InnoDB	utf8_unicode_ci	16.0 KB	-
<input type="checkbox"/>	tab_nome		~16	InnoDB	utf8_unicode_ci	32.0 KB	-
<input type="checkbox"/>	tab_noticia		~7	InnoDB	utf8_unicode_ci	32.0 KB	-
<input type="checkbox"/>	tab_tipo		~7	InnoDB	utf8_unicode_ci	32.0 KB	-
5 tabela(s)		Soma	~62	InnoDB	utf8_unicode_ci	160.0 KB	0 Bytes

↑ Todos / Nenhum Com os seleccionados:

Figura 87 - Visualização das tabelas da base de dados usando o phpMyAdmin.

Campo	Tipo	Collation	Atributos	Nulo	Defeito	Extra	Acções
<input type="checkbox"/> IdTipo	int(4)			Não		auto_increment	
<input type="checkbox"/> Tipo	varchar(50)	utf8_bin		Não			

↑ Todos / Nenhum Com os seleccionados:

Vista de impressão Vista de Relação Propor uma estrutura de tabela

Add field(s) No Fim da Tabela No Início da Tabela Depois

Índices:				Espaço ocupado	
Nome do Índice	Tipo	Quantidade	Acções	Tipo	Utilização
PRIMARY	PRIMARY	7		Dados	16,384 Bytes
Tipo	UNIQUE	7		Índice	16,384 Bytes
Criar um índice com <input type="text" value="1"/> coluna(s) <input type="button" value="Executa"/>				Total	32,768 Bytes

Figura 88 - Edição das tabelas usando o phpMyAdmin.

Mais especificamente, a gestão de registos permite introduzir, eliminar e os dados contidos nos registos, bem como efectuar a alteração de propriedades das tabelas, tais como os índices de incrementação automática, a selecção de chaves primárias e a definição de chaves únicas.

4.3. Mecanismo de replicação da base de dados

Para replicar as bases de dados entre as aplicações “NewsMobile” e “PubMovel” utilizou-se, como se referiu anteriormente, um conjunto de dois ficheiros XML. Nesta secção explica-se o modo como se implementou um mecanismo de replicação recorrendo aos ficheiros XML. Em primeiro lugar, desenvolveu-se a página CriaXML.php na aplicação “NewsMobile” para se realizar a criação dos ficheiros XML. Esta página é invocada sempre que um anunciante efectua um *logout* da aplicação “NewsMobile”. Assim, logo que a página é executada, extraem-se todos os registos das tabelas a replicar. Para isso, foi desenvolvida uma função designada por criarXML() que executa o código necessário à criação dos ficheiros db.xml e o news.xml.

O ficheiro db.xml contém todos os registos existentes na base de dados. Por sua vez, o ficheiro news.xml contém apenas os registos das notícias publicitárias. A razão pela qual se efectua a criação de dois ficheiros deve-se ao tipo de actualização que o utilizador pode desejar efectuar na aplicação “PubMovel”.

No que diz respeito à função criarXML() propriamente dita, sempre que é invocada sucede uma ligação à base de dados. Caso se consiga efectuar a ligação, é criado no mesmo instante um documento, usando o comando “\$doc = new XmlDocument('1.0', 'UTF-8');”, este comando permite adicionar a primeira linha no ficheiro XML do tipo <?xml version="1.0" encoding="UTF-8"?>. Esta primeira linha identifica o tipo de documento a ser lido do tipo XML da versão 1.0 e que possui uma codificação de caracteres em modo UTF-8. Assim irá permitir que quando o ficheiro XML for “lido” as *tags* XML sejam devidamente interpretadas

Em seguida, cria-se o nodo raiz como se demonstra no Exemplo de código 4, e iniciam-se os ciclos para a obtenção dos registos da bases de dados, como se ilustra no Exemplo de código 5.

```
$root = $doc->createElement('db');
$root = $doc->appendChild($root);
```

Exemplo de código 4 - Criação do nodo raiz.

```
foreach ($tabelas as $tabela)
{
    $table_id = $tabela;
    $query = "SELECT * FROM $table_id";
    $dbresult = mysql_query($query, $dbconnect);

    // Processar uma linha de cada vez
    while($row = mysql_fetch_assoc($dbresult)) {

        // adicionar um nodo por cada linha
        $occ = $doc->createElement($table_id);
        $occ = $root->appendChild($occ);

        // adicionar um nodo filho a cada campo
        foreach ($row as $fieldname => $fieldvalue) {
            $child = $doc->createElement($fieldname);
            $child = $occ->appendChild($child);
            $fieldvalue = mb_convert_encoding($fieldvalue,'UTF-8','ISO-8859-1');
            $value = $doc->createTextNode($fieldvalue);
            $value = $child->appendChild($value);
        }
    }
}
```

Exemplo de código 5 - Ciclos de obtenção de registos.

Este procedimento de criação de nodos é necessário para que se obtenha uma base de dados estruturada, tal como se mostra no Exemplo de código 6:

```
<Base de dados>
  <Tabela>
    <Atributo1>Registo1</Atributo1>
    <AtributoN>RegistoN</AtributoN>
  </Tabela>
</Base de dados>
```

Exemplo de código 6 - Exemplo da criação do ficheiro XML.

Finalmente, gera-se o ficheiro XML, tal como se mostra no Exemplo de código 7.

```
// Documento XML
$xml_string = $doc->saveXML();
// Abrir o documento em modo de escrita
$manipulador_arq = fopen("/wamp/www/PubMoviel/$file.xml", "w+");
// Escrever para o ficheiro
@fwrite($manipulador_arq, $xml_string);
```

Exemplo de código 7 - Criação de ficheiros.

A partir do momento em que o ficheiro XML é gerado, passa a ser possível descarregá-lo na aplicação “PubMoviel” de modo a replicar todos os registos existentes no ficheiro XML para a sua base de dados local do dispositivo móvel.

Do lado da aplicação móvel, quando o utilizador faz um pedido para efectuar actualizações, a aplicação “PubMoviel” descarrega o ficheiro XML como se explicou anteriormente, e grava o ficheiro localmente na memória interna do dispositivo móvel.

A partir desse momento é feita uma limpeza da base de dados local da aplicação “PubMoviel”, guardando-se apenas alguns registos em ficheiros temporários, tais como, os pontos de interesse escolhidos como parte das preferências do utilizador e as notícias que se já foram lidas pelo utilizador. O objectivo destes ficheiros temporários é o de permitir a actualização dos registos após a actualização da base de dados relativamente ao ficheiro XML. Estas últimas actualizações (que recorrem aos ficheiros temporários) evitam que o utilizador tenha que colocar explicitamente como lidas as notícias que já tinha lido anteriormente, bem como evitar que o utilizador necessite de voltar a parametrizar as suas preferências relativamente aos pontos de interesse.

A aplicação “PubMoviel” efectua a leitura do ficheiro XML fazendo o uso código que se mostra no Exemplo de código 8.

```
XmlTextReader reader = new
XmlTextReader((System.IO.Path.GetDirectoryName(xml));

reader.Read();
while (reader.Read())
{
    switch (reader.NodeType)
    {
        case XmlNodeType.Element:
        case XmlNodeType.Text:
        case XmlNodeType.EndElement:
    }
}
```

Exemplo de código 8 - Leitura do ficheiro XML.

Assim, usa-se o método `.Read()`, associado com o método `.NodeType`, para efectuar a leitura, definindo se o elemento que se está a ler é do tipo elemento `<Atributo1>`, do tipo texto `Atributo1`, ou se é do tipo fim do elemento `</Atributo1>`. Em seguida, tendo por base essa leitura, procede-se à execução de código que permite a introdução dos registos na base de dados através de comandos SQL.

5. Estudo empírico

Entre as fases envolvidas no desenvolvimento das aplicações, a fase do estudo empírico é tão importante quanto a fase de implementação ou a fase de análise e planeamento, uma vez que permite chegar a resultados obtidos experimentalmente, ou por observação de utilizadores, e não meramente a partir de deduções teóricas. Trata-se pois de uma fase do desenvolvimento das aplicações que se fundamenta na experiência.

O objectivo da realização de um estudo empírico prendeu-se com a necessidade de verificar se as aplicações cumpririam os requisitos definidos inicialmente. Para tal, o desenvolvimento do estudo empírico envolveu uma interacção, e a consequente avaliação, das aplicações por parte de um grupo de utilizadores finais, tendo em consideração a facilidade de uso e as funcionalidades proporcionadas por ambas as aplicações.

5.1. Técnicas de avaliação

Segundo Strauss (cit. In Ribeiro, 2004), existem quatro tipos de testes que se podem realizar na avaliação de uma aplicação:

- Testes dos módulos de *software*;
- Testes da integração do *software*;
- Testes de compatibilidade;
- Testes de facilidade de uso da interface do utilizador.

Os testes dos módulos de software deverão ser realizados pelo programador. Estes testes deverão incluir a utilização de dados de entrada que sejam tanto normais como anormais, de modo a assegurar que o código funciona correctamente, mesmo quando se fornecem dados de entrada menos correctos.

Os testes de integração de *software* referem-se à verificação do modo de funcionamento da aplicação quando todos os módulos forem combinados como parte integrante da aplicação final. São testes que deverão ser conduzidos por pessoas que não participaram

na construção da aplicação, pois o facto de desconhecerem o modo como foi desenvolvida contribui para manipularem a aplicação de uma maneira que não foi antecipada pelos programadores, identificando assim erros que não seriam identificados pelos próprios programadores.

Os testes de compatibilidade referem-se ao comportamento da aplicação sob diferentes circunstâncias. Deverá ser dada uma atenção particular aos requisitos mínimos para que a aplicação funcione integralmente sem erros.

Por fim, os testes de facilidade de uso da interface do utilizador, normalmente conhecidos por *usability tests*, destinam-se a identificar se a interface é de facto simples e acessível e se os controlos desempenham as funções que seria de esperar.

Segundo Faulkner cit. In (Miranda, 2007), existem duas abordagens diferentes a para avaliar a facilidade de uso. A primeira trata-se de uma avaliação analítica que consiste numa avaliação formal dos objectivos e tarefas a cumprir. A segunda é um método empírico que consiste numa análise do desempenho do utilizador através de entrevistas, experiências, questionários, desafios ou mesmo através da observação dos utilizadores enquanto estes realizam um conjunto de tarefas.

O método escolhido para este trabalho foi o método empírico, que consiste na avaliação do desempenho do utilizador em relação ao uso das aplicações, utilizando para esse fim um questionário e dois desafios.

O questionário é constituído por perguntas abertas e perguntas fechadas. As perguntas fechadas oferecem uma lista de respostas em que o entrevistado terá de responder de acordo com as opções fornecidas - trata-se portanto de um conjunto de perguntas bastante limitado. Já as perguntas abertas são subjectivas, e têm como finalidade obter a opinião pessoal do entrevistado relativamente à aplicação que esteve a testar. Este tipo de perguntas acarretam uma desvantagem, uma vez que o número de dados obtidos é variável, podendo ser muito extenso e, por isso, dificultar a análise de dados.

De acordo com Saunders cit. in (Miranda, 2007), a elaboração do questionário deve considerar as seguintes recomendações:

- As perguntas fechadas são melhores que as perguntas abertas. Numa pergunta aberta pode sempre existir um “não sei” como resposta.

- As perguntas devem ser estruturadas e progredir da generalidade para o detalhe. As primeiras perguntas deverão ser de fácil resposta deixando as mais difíceis para o fim.
- Os questionários não deverão ser muito longos e devem ser atractivos de modo a motivar o entrevistado a continuar.
- É importante testar o questionário antes de o usar. Isto permite ao entrevistador perceber qual é o tipo de dificuldades que o entrevistado pode encontrar para responder correctamente ao que é solicitado
- Não deverão existir respostas ambíguas.

No entanto, o resultado de um questionário poderá ser ambíguo para o entrevistador, pois este dificilmente saberá qual o tipo de facilidade que os entrevistados encontraram no decorrer da avaliação das aplicações. Para colmatar essa ambiguidade, no âmbito do trabalho desenvolvido para esta dissertação foram também criados dois desafios com o intuito de efectuar uma comparação entre a facilidade de adaptação entre os utilizadores e as aplicações com outras soluções, sendo tecnológicas ou não, que permitam a obtenção do mesmo tipo de registos.

O conjunto de análises deverá ser focado em unidades temporais, indicando quanto tempo o entrevistado demora a encontrar a informação, qual a facilidade de pesquisa e a respectiva eficiência em encontrar toda a informação pretendida.

5.2. Experiência

A experiência efectuada pelos participantes no estudo empírico consistiu na realização de um conjunto de dois desafios de pesquisa de informação que permitiu avaliar a facilidade de uso e o interesse das funcionalidades proporcionadas pelas aplicações desenvolvidas através de um questionário, e efectuar uma comparação do desempenho obtido utilizando as aplicações aqui desenvolvidas e utilizando as tecnologias *Web* existentes. O desempenho dos participantes foi medido através do tempo dispendido na obtenção do mesmo tipo de dados recorrendo à aplicação móvel ou recorrendo a uma aplicação *Web* convencional. As indicações fornecidas aos participantes na experiência encontram-se no Anexo I.

No âmbito do primeiro desafio pediu-se aos participantes que escolhessem um cinema contendo em cartaz um filme do seu agrado. No entanto, para além da escolha de um cinema e de um filme, os participantes teriam de recolher mais informações relevantes para esta tarefa, incluindo o título do filme, uma sinopse do filme, o nome do cinema, o endereço do cinema e a quantidade de bilhetes reservados.

No âmbito do segundo desafio pediu-se aos participantes que escolhessem um restaurante do seu agrado para planearem um jantar. No entanto, para além da escolha do restaurante o objectivo seria também a obtenção do endereço do restaurante, do número de reservas, da escolha do prato, caso se encontrasse disponível nas notícias, a indicação das especialidades do restaurante escolhido, e a visualização de um mapa indicando a localização do restaurante, usando para esse efeito as coordenadas GPS do local.

5.3. Avaliação das aplicações

Como se mencionou acima, foi desenvolvido um questionário, que pode ser consultado no anexo II. Este questionário permitiu avaliar, no contexto da experiência indicada acima, as opiniões dos participantes sobre a facilidade de uso e o interesse das funcionalidades proporcionadas pelas aplicações. Na Tabela 3, Tabela 4 e Tabela 5, definem-se as características que se pretendeu avaliar, bem como as questões que se conceberam para avaliar as características identificadas.

A construção do questionário observou a estrutura que se detalha a seguir.

O primeiro grupo é constituído por três questões, como se mostra na Tabela 3, que permitem fazer uma caracterização do participante de acordo com o seu perfil de utilizador, usando variáveis independentes deste estudo, tais como a idade, a frequência de utilização de um PDA e de um PC, e a identificação do local onde normalmente são utilizados.

Característica	Questões:
Perfil do utilizador	1) Indique, por favor, a sua idade.
	2) Indique, por favor, a frequência com que utiliza: a) Um computador; b) Um Pocket PC.
	3) Indique ainda o local onde normalmente utiliza o: a) Computador; b) Pocket PC.

Tabela 3 - Primeiro grupo de questões.

O segundo grupo de questões é constituído por dez perguntas de resposta fechada, como se mostra na Tabela 4, e permitem a recolha de opiniões sobre a facilidade de uso da interface das aplicações, bem como opiniões sobre a utilidade das funcionalidades proporcionadas pelas aplicações.

Característica	Questões:
Adequação do vocabulário.	1) O vocabulário utilizado em ambas as aplicações é adequado.
Facilidade de uso.	2) A localização dos botões e do texto que se apresenta no ecrã facilita a manipulação da informação: a) Na aplicação do computador; b) Na aplicação do Pocket PC.
	3) Qual foi o grau de dificuldade que sentiu na introdução dados: a) No computador; b) No Pocket PC.
	4) A função dos botões das aplicações é fácil de identificar:
	5) A localização da informação nos ecrãs da aplicação é fácil de recordar: a) Na aplicação do computador; b) Na aplicação do Pocket PC.
	6) Qual foi o grau de dificuldade que sentiu na utilização da aplicação: a) Na aplicação do computador; b) Na aplicação do Pocket PC.
	7) A dimensão do ecrã dificulta a leitura da informação.
	8) Os menus e botões da aplicação permitem uma boa navegação.
Utilidade das funcionalidades proporcionadas.	9) As aplicações fornecem a informação necessária para executar as tarefas: (tais como a obtenção de morada, telefone, localização e de notícias)

	10) Qual é a sua opinião em relação às tarefas que realizou recorrendo às aplicações: a) Julgo que se trata de uma forma simples de se publicar um anúncio/notícia. b) Julgo que se trata de uma forma simples e eficiente de recolher informação.
--	--

Tabela 4 - Segundo grupo de questões.

Por fim, o terceiro grupo de questões, ilustrado na Tabela 5, é constituído por três perguntas de resposta aberta onde se pretende obter informações e opiniões dos entrevistados sobre três aspectos importantes relacionados com as aplicações: a utilidade das aplicações, o tipo de implementações futuras que seriam necessárias do ponto de vista dos utilizadores e, por fim, a opinião sobre a utilização continuada das aplicações no futuro, se tal fosse possível.

Característica	Questões:
Utilização das aplicações	1) O que acha da aplicação em termos de utilidade.
	2) Na sua opinião que outras funcionalidades gostaria de ver implementadas no futuro?
	3) Qual é a sua opinião em relação à futura utilização desta aplicação?

Tabela 5 - Terceiro grupo de questões.

Os dados associados às várias questões de resposta fechada foram medidos de acordo com as escalas identificadas na Tabela 6.

Grupos	Questões	Escala									
		1	2	3	4						
1	2a e 2b	Nunca utilizei	1 vez por semana	2 a 3 vezes por semana	Diariamente						
2	1										
	2a e 2b										
	4										
	5						1	2	3	4	5
	7						Discordo totalmente	Discordo	Sem opinião	Concordo	Concordo totalmente
	8										
	9										
	10										
	3a e 3b						1	2	3	4	5
	6a e 6b						Muito difícil	Difícil	Mediano	Fácil	Muito fácil

Tabela 6 - Escala utilizada em cada questão

5.4. Análise e discussão de resultados

As secções que se seguem apresentam e discutem os resultados obtidos para cada característica em estudo. Inicialmente será feita uma introdução aos tipos de utilizadores que efectuaram os testes indicando o perfil médio. Posteriormente apresenta-se as secções da adequação do vocabulário, a facilidade de uso, a funcionalidade e a utilização da aplicação.

5.4.1. Perfil dos participantes

A Figura 89 mostra a distribuição de idades dos 10 utilizadores que participaram no teste das aplicações e realizaram o desafio. Como se pode verificar, os participantes apresentam-se numa faixa etária média entre os 21 e 30 anos, tendo o mais novo 16 anos e o mais idoso 62 anos. O intuito foi o de apresentar as aplicações a diferentes

faixas etárias de modo a que possibilitar uma análise tão realista quanto possível do que poderá ser a utilização das aplicações num ambiente comercial. Deste modo, são captados diferentes pontos de vista relacionados com a idade e com a experiência de cada utilizador.

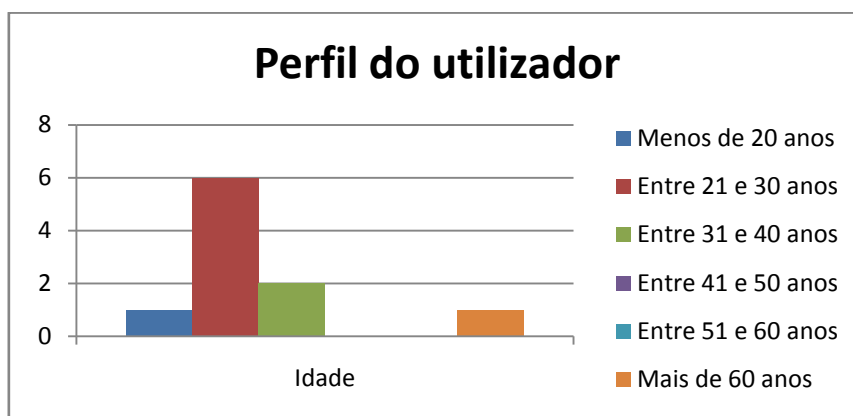


Figura 89 - Gráfico representativo da idade dos utilizadores entrevistados.

De modo a caracterizar os utilizadores de acordo com a experiência prévia com as tecnologias e respectiva utilização foram elaboradas questões inquirindo se utilizam o computador e/ou um Pocket PC, qual a sua frequência de utilização e a sua localização. Como se mostra na Figura 90 e na Figura 91, o conjunto de utilizadores que testaram as aplicações utiliza, no mínimo, o computador de 2 a 3 vezes por semana, o que fez com que se sentissem relativamente à vontade para executar os testes das aplicações. No entanto, o mesmo não se pode dizer quanto à utilização do Pocket PC, uma vez que apenas dois dos inquiridos utilizam o Pocket PC diariamente.

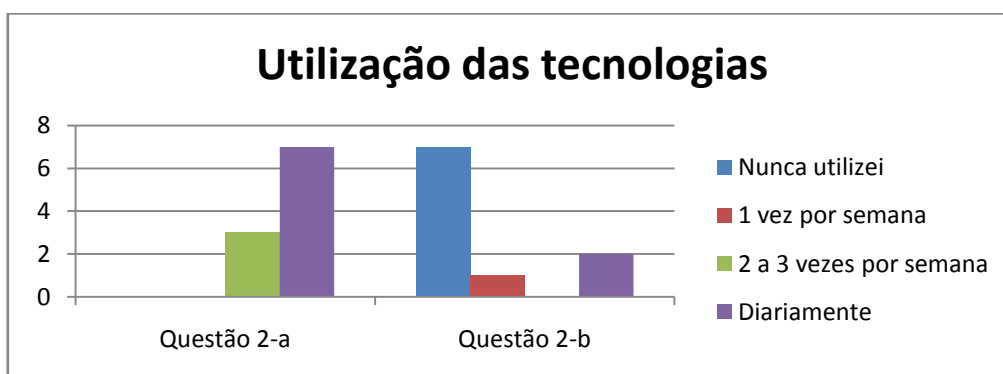


Figura 90 - Gráfico representativo da utilização das tecnologias.

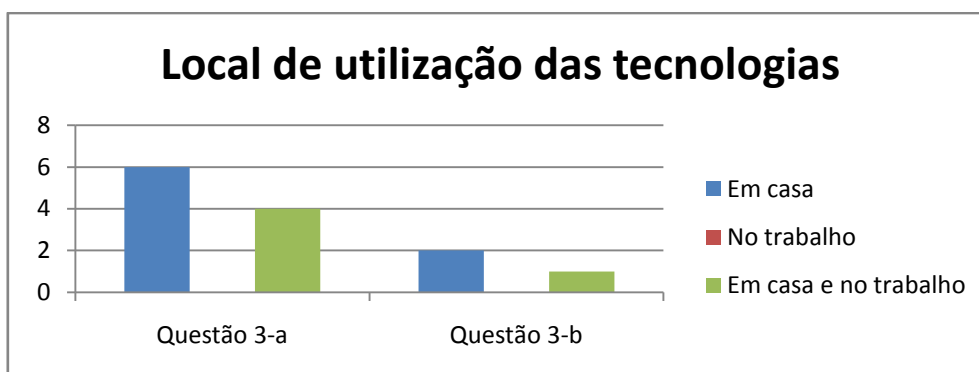


Figura 91 - Gráfico representativo do local de utilização das tecnologias.

Relativamente à localização da utilização das tecnologias, os dados recolhidos permitem-nos determinar que os participantes fazem o uso das tecnologias não só no trabalho mas também em casa, o que permite concluir que possuem alguma destreza tecnológica para além daquela que é exigida na avaliação deste trabalho.

5.4.2. Adequação do vocabulário.

Para a característica adequação do vocabulário, os resultados obtidos no questionário são apresentados a seguir. Como se observa na Figura 92 e na Tabela 7, os participantes dividiram as suas respostas em “Concordo Totalmente” e “Concordo”, apresentando uma média de 4,3 no que diz respeito à questão. Deste modo, é possível considerar que os participantes ficaram satisfeitos com o vocabulário usado nas aplicações e consideram-no adequado ao tipo de trabalho que desenvolvem, não constituindo, por isso, um impedimento à utilização das tecnologias.

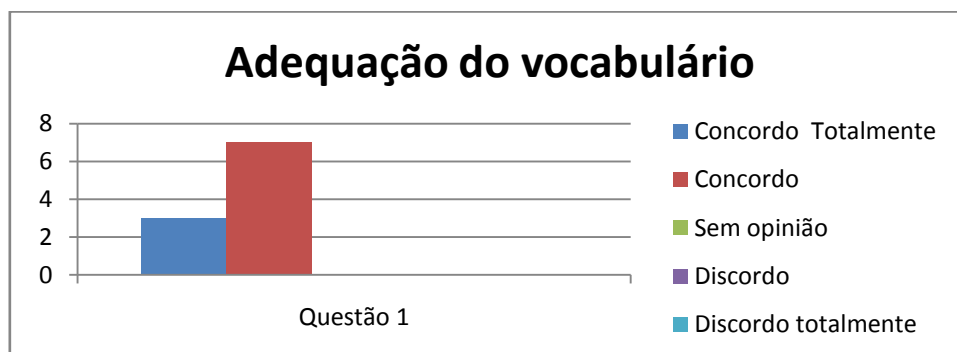


Figura 92 - Gráfico representativo da característica "Adequação do vocabulário".

Questões	Média	Mediana	Moda	Desvio Padrão
1) O vocabulário utilizado em ambas as aplicações é adequado.	4,3	4	4	0,483045892

Tabela 7 - Resultados das questões para a característica "Adequação do vocabulário"

5.4.3. Facilidade de uso

No que diz respeito à facilidade de uso, os resultados obtidos no questionário foram agrupados e são apresentados em seguida. A Tabela 8 e as figuras Figura 93 e Figura 94, mostram o conjunto de resultados obtidos para as questões relacionadas com a facilidade de uso das aplicações.

Questões	Média	Mediana	Moda	Desvio Padrão
2) A localização dos botões e do texto que se apresenta no ecrã facilita a manipulação da informação				
a) Na aplicação do computador	4,3	4	4	0,48304589
b) Na aplicação do PocketPC	4,3	4	4	0,48304589
3) Qual foi o grau de dificuldade que sentiu na introdução de dados:				
a) No computador:	4,3	5	5	0,9486833
b) No PocketPC	3,4	4	4	0,96609178
4) A função dos botões das aplicações é fácil de identificar:	4,5	4,5	5	0,52704628
5) A localização da informação nos ecrãs da aplicação é fácil de recordar:				
a) Na aplicação do computador	4,6	5	5	0,51639778
b) Na aplicação do PocketPC	4,3	4	4	0,48304589
6) Qual foi o grau de dificuldade que sentiu na utilização da aplicação:				
a) No computador:	4,1	4	5	0,87559504
b) No PocketPC	3,6	4	4	0,51639778
7) A dimensão do ecrã dificulta a leitura da informação.	1,9	2	1	0,87559504
8) Os menus e botões da aplicação permitem uma boa navegação.	4,2	4	4	0,42163702

Tabela 8 - Resultado das questões associadas à característica "Facilidade de uso".

Da análise da Figura 93 verifica-se que, globalmente, as aplicações são de fácil utilização, visto que 85% das respostas consistem em “Concordo Totalmente” e “Concordo”. De facto, verifica-se na Tabela 8 que, numa escala de 1 a 5, a localização da informação (média de 4,3 em ambas as aplicações), a facilidade na identificação dos botões (média de 4,3 na aplicação “NewsMobile” e 3,4 na aplicação “PubMovel”), a facilidade de navegação (média de 4,2 em ambas as aplicações), assim como as restantes questões obtiveram opiniões positivas.

Os participantes, quando questionados se a dimensão do ecrã dificulta a leitura da informação, indicaram, em 30%, que não possuíam uma opinião formada por não utilizar o Pocket PC, ao passo que 70% deram como resposta “Discordo” e “Discordo Totalmente”. Ao cruzar os dados com os perfis de utilizadores, é possível verificar que 70% dos participantes nunca utilizaram um Pocket PC. No entanto, a organização de botões da aplicação “PubMovel” foi capaz de permitir um manuseamento fácil da informação, mesmo utilizando um ecrã de dimensões reduzidas como é o caso do ecrã de um dispositivo do tipo PocketPC.

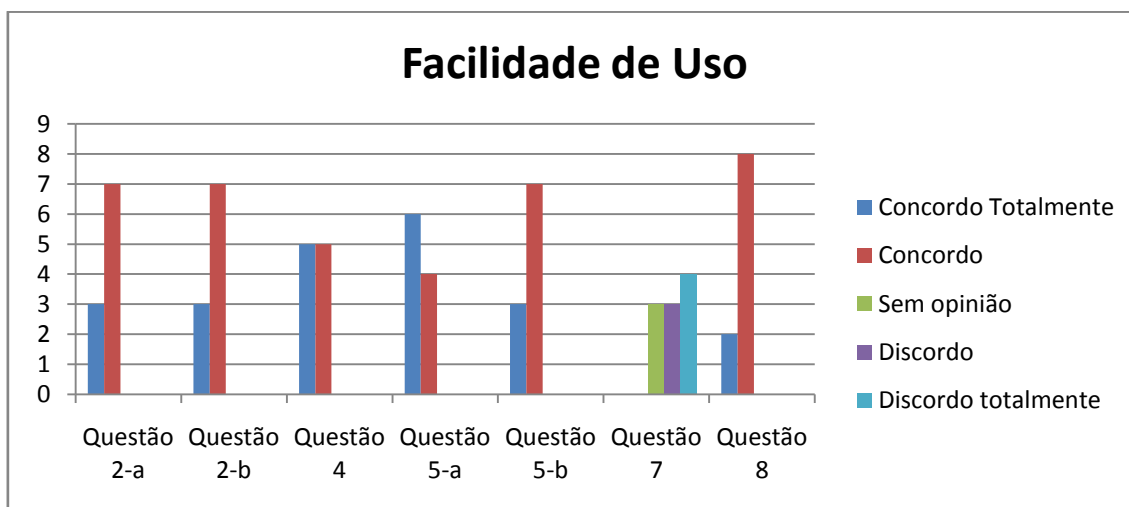


Figura 93 - Gráfico da característica "Facilidade de uso" para as questões 2, 4, 5, 7 e 8.

Investigando ainda a facilidade de uso das aplicações, e fazendo uma análise mais detalhada às questões 3 e 6 é possível tirar as seguintes conclusões.

Relativamente ao grau de dificuldade na introdução de dados, fazendo uso dos dados da Tabela 8 e da Figura 94, é possível verificar que existe uma maior facilidade na introdução dos dados na aplicação *Web* “NewsMobile” do que na aplicação de Pocket

PC “PubMovel”. De facto, a média obtida para a introdução de dados no computador é de 4,3 enquanto que na aplicação “PubMovel” é de apenas 3,4.

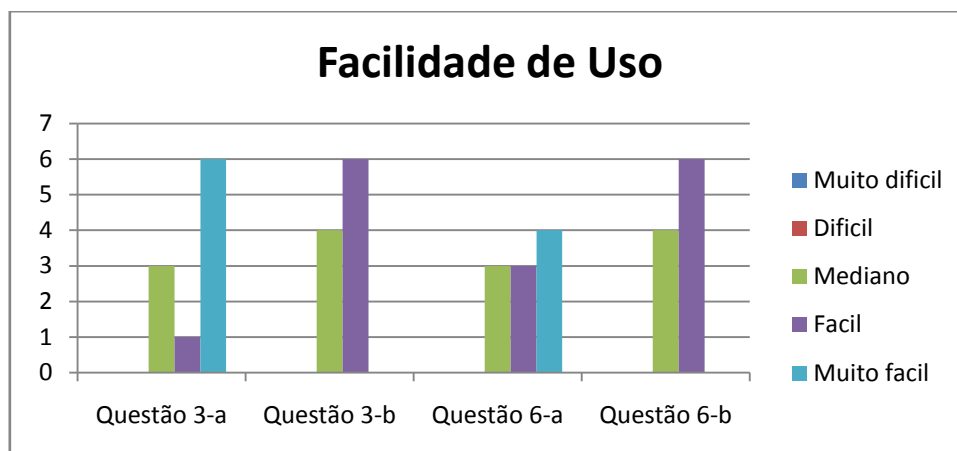


Figura 94 - Gráfico da característica "Facilidade de uso" para as questões 3 e 6.

Também é possível verificar que os participantes se sentiram mais à vontade na utilização da aplicação “NewsMobile” do que na utilização da aplicação “PubMovel”. Em termos percentuais, verifica-se que 70% dos participantes julga a aplicação no computador “Muito Fácil” e “Fácil”, enquanto que apenas 60% acha a aplicação “PubMovel” de utilização “Fácil”. No entanto, quando se analisam os valores médios, 4,1 para a aplicação “NewsMobile” e 3,6 para a aplicação “PubMovel”, encontram-se valores muito distintos que, uma vez mais, confirmam uma maior facilidade de utilização da aplicação “NewsMobile”.

A forte diferença nas pontuações obtidas quanto à utilização das aplicações permite concluir que os participantes estão muito mais habituados à utilização de aplicações *Web*. No entanto, não se deve ignorar a dificuldade envolvida no manuseamento da aplicação “PubMovel” devido precisamente ao facto da fraca experiência de utilização do Pocket PC por parte dos participantes.

5.4.4. Funcionalidade

Os resultados apresentados na Figura 95 e na Tabela 9 dizem respeito à característica funcionalidade. A análise da Figura 95 e dos valores da Tabela 9 permite verificar que, em termos de funcionalidade, a opinião dos participantes é bastante positiva, visto que,

numa escala de 1 a 5, os participantes concordaram que a aplicação fornece a informação necessária para a realização do trabalho com uma média de 4,6, tendo igualmente concordado com o facto de que se trata de uma maneira simples de se publicar uma notícia (com uma média de 4,8), e de se tratar de uma maneira simples e eficiente de recolha de informação (com uma média de 4,7). De facto, 70% dos participantes deram como resposta “Concordo Totalmente” e os restantes 30% deram como resposta “Concordo”. Isto permite concluir que as aplicações são relativamente funcionais, na medida em que os participantes concordaram que facilitam e fornecem a informação necessária para a realização dos desafios lançados no âmbito da experiência que se realizou.

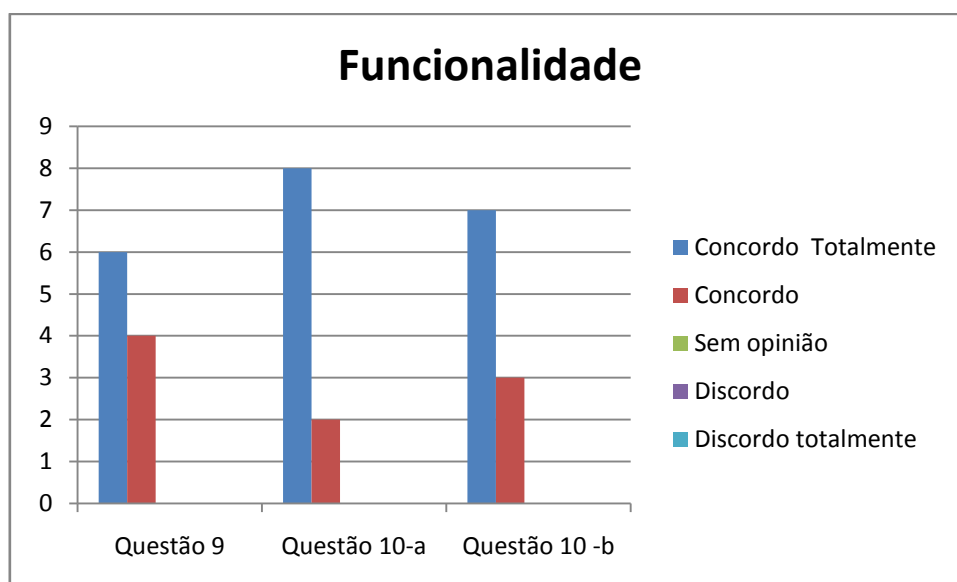


Figura 95 - Gráfico da característica "Funcionalidade" para as questões 9 e 10.

Questões	Média	Mediana	Moda	Desvio Padrão
9) As aplicações fornecem a informação necessária para executar as tarefas: (tais como a obtenção de morada, telefone, localização e de notícias).	4,6	5	5	0,51639778
10) Qual é a sua opinião em relação às tarefas que realizou recorrendo às aplicações?				
a) Julgo que se trata de uma forma simples de se publicar um anúncio/notícia:	4,8	5	5	0,42163702
a) Julgo que se trata de uma forma simples e eficiente de recolher informação:	4,7	5	5	0,48304589

Tabela 9 - Resultado das questões associadas à característica "Funcionalidade".

5.4.5. Utilização da aplicação

Esta secção analisa os resultados obtidos no terceiro grupo de questões apresentadas na Tabela 5. Trata-se de um conjunto de três questões de resposta aberta que pretendeu avaliar as opiniões dos participantes sobre novas funcionalidades a incluir nas aplicações e as respectivas opiniões sobre a utilização futura das aplicações.

Analisando as respostas dos participantes à questão 1, verifica-se que 100% das respostas considera as aplicações úteis, dividindo-se as opiniões pela mobilidade que a aplicação “PubMovel” permite, pela acessibilidade, pela rapidez na obtenção de informação e pela facilidade de expansão de negócios através de métodos publicitários.

A questão 2 serviu para que os participantes contribuíssem com sugestões de novas funcionalidades que gostariam de ver implementadas nas aplicações. Do total de 10 participantes apenas 3 utilizadores não fizeram sugestões. No entanto, atendendo às diferentes necessidades de cada um dos participantes, surgiram ideias bastante interessantes que não foram contempladas no protótipo inicial mas que, no futuro, poderão garantir um maior interesse na utilização da aplicação móvel. As funcionalidades sugeridas pelas respostas aos questionários consistem na possibilidade de efectuar reservas ou compras através da aplicação “PubMovel”, a introdução de um mecanismo de pesquisa por nomes, moradas e localidades e, por fim, completar a informação disponibilizada sobre cada ponto de interesse com a adição de uma fotografia do estabelecimento do anunciante.

No que diz respeito à questão 3, que pretendeu recolher a opinião dos participantes sobre a futura utilização da aplicação móvel, 4 dos participantes não responderam, indicando que necessitariam de mais tempo de utilização das aplicações de modo a possuírem uma opinião verdadeiramente formada relativamente a uma utilização futura. Indicaram também que, para uma utilização futura das aplicações, a versão actual se encontra limitada em termos de plataforma, pois apenas se encontra disponível na versão Pocket PC. Assim, sugeriram que fosse desenvolvida uma versão para outras plataformas de telefones celulares, o que certamente faria com que se tornassem utilizadores da aplicação móvel. Entre os restantes participantes a resposta acabou por ser unânime, indicando que a aplicação apresenta uma mais-valia na recolha de informação que rapidamente pode ser perdida ou esquecida, e que, quanto mais

abrangente fosse o número de serviços disponibilizados, maior seria a sua utilidade. Referiram ainda que a mobilidade proporcionada pela aplicação móvel poderá ser um trunfo quando comparada com outras tecnologias existentes para a pesquisa de anúncios e outra informação publicitária.

5.4.6. Análise dos resultados do desafio

Esta secção analisa os resultados obtidos na comparação do desempenho obtido utilizando as aplicações aqui desenvolvidas e utilizando as tecnologias *Web* existentes. Recorde-se que o desempenho dos participantes foi medido através do tempo dispendido na obtenção do mesmo tipo de dados recorrendo à aplicação móvel ou recorrendo a uma aplicação *Web* convencional.

No primeiro desafio pediu-se aos participantes que escolhessem um cinema com um filme do seu agrado acompanhado pelo registo de informações adicionais. Como se pode analisar na Figura 96, a utilização da aplicação “PubMovel” para o primeiro desafio permitiu um tempo médio para a obtenção da informação pedida de 56,6s, enquanto que a realização da mesma tarefa recorrendo a outra tecnologia, tais como a Internet, permitiu recolher a mesma informação com tempos médios de 125s.

Um dos principais meios alternativos à aplicação móvel que foram escolhidos pelos participantes para uma obtenção de dados foi a realização de pesquisas convencionais na Internet. No entanto, o objectivo da experiência consistia na comparação do desempenho em vários meios alternativos que disponibilizassem a informação pretendida. Desse modo, foi ainda pedido aos participantes que realizassem a mesma tarefa mas recorrendo à utilização das páginas amarelas e do jornal.

Analisando uma vez mais o gráfico ilustrado na Figura 96, verifica-se que, apesar de se apresentarem tempos semelhantes aos tempos obtidos com a utilização da aplicação “PubMovel”, os participantes não foram capazes de obter o mesmo volume de informação pedido.

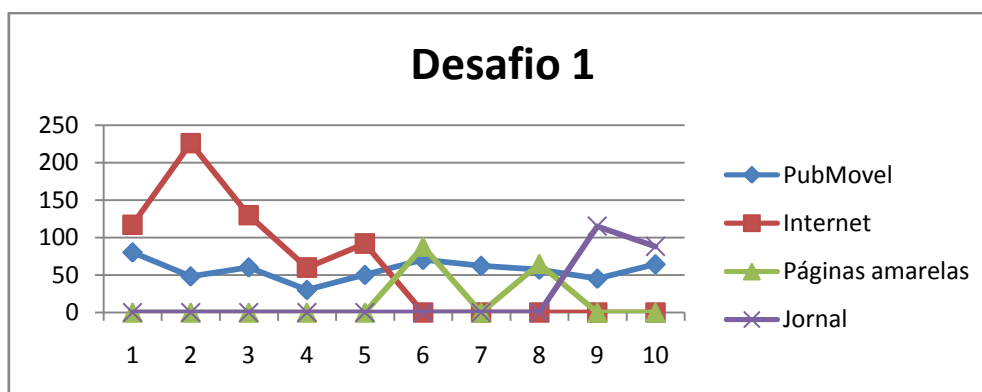


Figura 96 - Gráfico dos resultados do desafio 1.

No âmbito do segundo desafio pediu-se aos participantes que escolhessem um restaurante do seu agrado para planearem um jantar, recolhendo um conjunto de informação adicionais. Analisando a Figura 97, conclui-se, uma vez mais, que a aplicação “PubMovel” consegue proporcionar um desempenho bastante uniforme tendo um tempo médio de resolução do desafio de 54,5s, enquanto que, na segunda tecnologia mais usada (Internet) o tempo de resolução médio foi de 85,4s.

Em seguida, pediu-se novamente aos participantes que realizassem o mesmo desafio usando uma ferramenta de pesquisa alternativa. O meio escolhido foi as páginas amarelas. Uma vez mais o tempo dispendido foi superior ao obtido mediante a utilização da aplicação “PubMovel”, tendo resultado um tempo médio de 92,5s. É ainda importante referir que, com a utilização das páginas amarelas, o número de itens obtidos durante a pesquisa foi claramente inferior, tendo-se obtido apenas o nome do restaurante, o endereço do restaurante e o número de telefone de contacto para a realização de reservas.

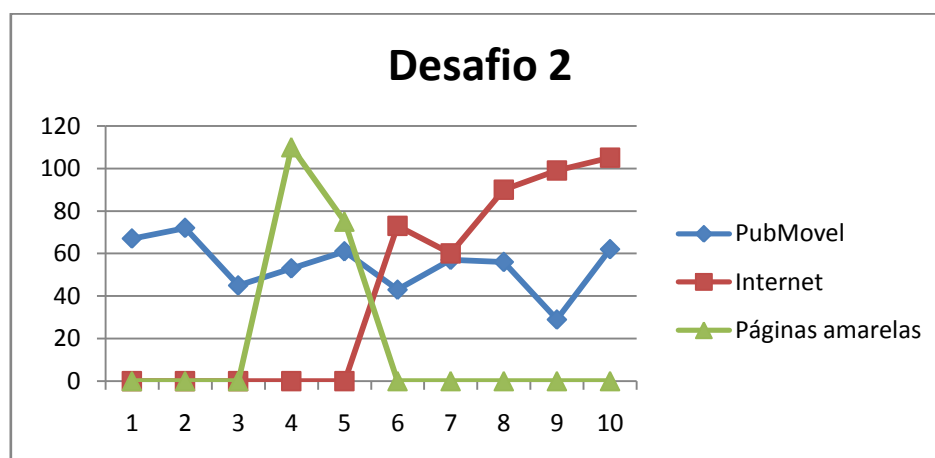


Figura 97 - Gráfico dos resultados do desafio 2.

6. Conclusão

Este capítulo apresenta as conclusões finais respeitantes ao desenvolvimento deste trabalho, indicando ainda possíveis aperfeiçoamentos às aplicações que poderão ser implementados no futuro. Apresenta-se igualmente o conjunto de limitações que foram encontradas no decorrer da realização deste trabalho.

6.1. Considerações finais

O objectivo principal desta dissertação de mestrado era a criação de um sistema de publicação de anúncios assim como a respectiva distribuição para dispositivos móveis, dando assim uma possibilidade mais abrangente para os anunciantes colocarem anúncios menos genéricos, adaptando-os ao público-alvo, e conferindo-lhes um tempo de “vida” muito mais limitado. Pretendeu-se igualmente investigar sobre a possibilidade de recorrer à utilização das tecnologias de informação para a redução dos custos relacionados com a publicidade, assim como a possibilidade de alargar essa mesma publicidade em áreas ou locais onde, anteriormente, e usando os métodos tradicionais, o custo/retorno desse investimento não se justificaria.

O objectivo, no que diz respeito ao utilizador final, era permitir que a mobilidade fosse um factor importante, e permitir que todas as publicações dos anunciantes chegassem ao utilizador final independentemente do local onde este se encontra. Pretendeu-se igualmente facilitar a criação de um perfil de preferências local relativamente aos tipos de pontos de interesse existentes na aplicação, bem como proporcionar ao utilizador a possibilidade de organizar os respectivos favoritos localmente. A utilização de um perfil de preferências local permitiria, pois, que o utilizador confiasse nas notícias que lhe seriam apresentadas pela aplicação, uma vez que corresponderiam às suas preferências, permitindo assim que apenas fosse alertado relativamente a notícias da sua preferência. Um segundo objectivo era o de permitir que a aplicação móvel fosse capaz de gravar localmente instâncias pré-definidas como favoritos do utilizador, e que este tivesse a possibilidade de os organizar face aos pontos de interesse, catalogando-os de acordo

com os seus interesses. Esta organização permitiria facilitar a gestão dos seus pontos de interesse locais, acelerando a posterior pesquisa e obtenção de informação.

A avaliação das aplicações desenvolvidas permitiu obter resultados importantes no que diz respeito às vantagens que a aplicação poderá trazer se implementada no âmbito da publicidade adaptativa. Os resultados demonstraram que as aplicações possuem potencial para desenvolvimento e aperfeiçoamento, visto que a possibilidade de se dispor de um dispositivo móvel que reunisse todos os dados referentes às preferências de publicidade de cada utilizador foi bem aceite pelos participantes no estudo empírico que se efectuou.

Por outro lado, as respostas à segunda questão do último grupo foram indicadoras que as aplicações despertaram o interesse dos participantes, já que foram sugeridas novas funcionalidade que os utilizadores gostariam de ver acrescentadas à aplicação com o intuito de a melhorar. Também as respostas à terceira questão desse mesmo grupo prevêem que as aplicações poderão ter um futuro promissor quando largamente utilizada pelo grupo de anunciantes, permitindo assim que os utilizadores finais possuam o máximo de pontos de interesse e notícias referentes a esses mesmos pontos de interesse.

Verificou-se igualmente que, apesar de alguns dos participantes sentirem dificuldades na utilização das aplicações, muito devido à pouca competência informática que relataram possuir, apresentaram mesmo assim respostas bastante positivas em relação à utilidade das aplicações e respectiva utilização no futuro, manifestando assim uma vontade de, pelo menos, tentarem acompanhar a evolução das tecnologias de computação móvel.

Embora o número de participantes no estudo não seja substancial (10), não representando pois uma amostra fiável de utilizadores, é mesmo assim possível retirar algumas conclusões dos dados recolhidos. Assim, pode concluir-se que a mobilidade que a aplicação móvel permite pode constituir um factor de desequilíbrio quando comparada com a utilização das tecnologias actuais na concepção e desenvolvimento da publicidade.

Por fim, é importante mencionar que o trabalho de aprendizagem para o desenvolvimento das aplicações foi considerável. Contudo, uma das principais

motivações na realização do trabalho foi precisamente a de aproveitar a oportunidade para adquirir conhecimentos e ganhar experiência no desenvolvimento de aplicações móveis para a plataforma Windows Mobile recorrendo para esse efeito à plataforma de desenvolvimento .NET *Compact Framework*, assim como na realização da interligação entre diferentes plataformas programáticas móveis e não móveis, nomeadamente plataformas *Web*. Neste âmbito, pode concluir-se que o trabalho desenvolvido foi muito proveitoso.

6.2. Limitações

Durante o desenvolvimento das aplicações foram encontradas algumas limitações que condicionaram alguns aspectos relacionados com a funcionalidade das aplicações, tais como, a impossibilidade da utilização de AJAX e Javascript no Internet Explorer do Windows Mobile 5.1.465 (Compilação 15673.3.3.1), a necessidade de se efectuar um registo para a obter a *free key* do Google Maps de modo a possibilitar a visualização dos mapas: de facto, segundo Google Maps API - FAQ(2008), a visualização dos mapas, mesmo recorrendo à *free key* contém uma limitação imposta pela Google que não permite a visualização de mais de 500,000 páginas de mapas por dia, limitando-se igualmente o número de pedidos de *geocode*.

No entanto, as limitações deste trabalho não se resumiram ao desenvolvimento das aplicações. De facto, foram encontradas limitações nas próprias ferramentas de desenvolvimento. Por exemplo, o MS Visual Studio 2005 não é totalmente compatível com o MS Windows Vista, sendo necessário a instalação do *Service Pack 1* do Visual Studio 2005 para resolver essa incompatibilidade. No entanto, o *upgrade* com o *Service Pack* por si só não resolve todos os problemas. O emulador presente no MS Visual Studio não permitia que o MS Windows Vista detectasse o emparelhamento em estado *cradle*. Assim, não era possível a partilha da ligação de internet que era fundamental para a realização dos testes do código desenvolvido. Assim, foi necessário recorrer a uma actualização do emulador para o Device Emulator 2.0. Toda a informação de apoio encontra-se disponível em (Anandba, 2007).

No que diz respeito à realização dos testes com os utilizadores, para além da limitação do número de participantes (10), foi igualmente limitadora a necessidade de explicar sempre qual o objectivo das aplicações assim como o respectivo funcionamento num PDA. Faltando alguma experiência na utilização de dispositivos móveis, os utilizadores acabavam sempre por colocar questões que, embora permitissem cumprir o objectivo, demonstram a falta de prática com dispositivos móveis do tipo Pocket PC.

Por fim, não foi possível aprofundar o nível da análise de resultados da avaliação das aplicações, bem como a realização de melhorias ao código devido à limitação temporal da realização deste trabalho de dissertação, bem como à limitação de páginas imposta para este trabalho.

6.3. Trabalho futuro

O resultado deste trabalho pode funcionar como ponto de partida para o desenvolvimento e melhoramento de uma aplicação que poderá, no futuro, ser bastante útil. Muitas ideias surgiram com o desenvolvimento do trabalho, assim como nas respostas aos questionários, que revelaram ser uma fonte de ideias e de funcionalidades adicionais. No entanto, devido às limitações indicadas anteriormente, tais funcionalidades não puderam ser implementadas em tempo útil. De qualquer modo, apresenta-se a seguir um conjunto resumido das ideias que surgiram para um desenvolvimento futuro deste trabalho.

Na aplicação do Pocket PC, sugere-se o desenvolvimento de um formulário de pesquisa por um conjunto de detalhes, incluindo o tipo, nome, morada, telefone, GPS,; sugere-se o desenvolvimento de uma funcionalidade que permita efectuar uma gestão mais interactiva com os mapas do Google Maps. Seria igualmente interessante implementar actualizações automáticas das notícias ou das bases de dados, assim como notificações sonoras de novas notícias relativas aos favoritos de um determinado utilizador. Para além disso, seria igualmente interessante desenvolver a possibilidade de integração entre a aplicação do Pocket PC e a aplicação *Web* com o intuito de permitir a introdução de novos anúncios publicitários usando a aplicação do Pocket PC.

No que diz respeito à aplicação *Web*, sugere-se a introdução de um nível de administração mais abrangente, tal como a gestão de notícias e respectivas datas, assim como a gestão de registos. Por fim, e pensando um pouco em termos de segurança, seria importante melhorar as questões relacionadas com a cifragem de dados tais como a *password* de acesso à aplicação, bem como dos ficheiros XML de replicação das bases de dados.

7. Bibliografia

Anandba. Device Emulator V2 is Live. [Em Linha] Disponível em <<http://blogs.msdn.com/anandba/archive/2007/03/13/device-emulator-v2-is-live.aspx>> Consultado em 21/05/2008.

Bourdon, R. WampServer 2.0 Presentation. [Em Linha] Disponível em <<http://www.wampserver.com/en/presentation.php>> Consultado em 10/11/2008.

Coelho, P. (2005). Programação com JSP. Lisboa: FCA – Editora Informática.

Dickinger, A., & Haghirian, P. (2004). An investigation and Conceptual Model of SMS Marketing. 37th Hawaii International Conference on System Sciences (p. 10). IEEE Computer Society.

DynDNS. (2008). Free Dynamics DNS. [Em Linha] Disponível em <<http://www.dyndns.com/services/dns/dyndns/>> Consultado em 10/11/2008.

Fernandes, J. (2000). MIDAS: Metodologia Orientada ao Objecto para Desenvolvimento. [Em linha] Disponível em <<http://www.di.uminho.pt/~jmf/PUBLI/papers/phd-Fernandes.pdf>> Consultado em 17/11/2008.

Fox, D., & Dox, J. (2003). Building Solutions with the Microsoft .NET Compact Framework: Architecture and Best Practices for Mobile Development. (Addison-Wesley, Ed.) Estados Unidos da América.

Google Maps API - FAQ. (2008). [Em Linha] Disponível em <http://code.google.com/apis/maps/faq.html#geocoder_limit> Consultado em 12/06/2008.

Heinonem, K., & Strandvik, T. (2006). How do consumers react to mobile marketing? IEEE Computer Society.

Hoffman-Andrews, J. (2008). Static & Flash Maps para dispositivos móveis e outros ambientes. [Em Linha] Disponível em <<http://sites.google.com/site/developerdaybrazil/google-developer-day-2008-brazil/static--flash-maps-para-dispositivos-m%C3%B3veis-e-outros-ambientes>> Consultado em 13/11/2008.

Hristova, N., & O'Hare, G. (2004). Ad-me: Wireless Advertising Adapted to the user location, device and emotions. Hawaii: IEEE.

InGlobe. (2006). Geocoding. [Em Linha] Disponível em <<http://www.inglobe.com.br/geocoding.asp>> Consultado em 12/11/2008.

Microsoft. (2008). Learn Windows Mobile. [Em Linha] Disponível em <<http://msdn.microsoft.com/en-us/windowsmobile/bb264342.aspx>> Consultado em 15/11/2008.

Miranda, C. (2007). Mobilidade na Gestão de Custos de Edifícios de Habitação. Porto.

MySQL. (1995-2008). MySQL Customers by Industry. [Em Linha] Disponível em <<http://www.mysql.com/customers/>> Consultado em 12/11/2008.

Neves, P., & Ruas, R. (2005). O Guia Prático do MySQL (1º Edição ed.). Lisboa: Centro Atlântico.

Obringer, L. (2003). Como funcionam os planos de marketing. [Em Linha] Disponível em <<http://empresasefinancas.hsw.uol.com.br/planos-de-marketing13.htm>> Consultado em 19/11/2008.

PHP. (2008). PHP: General Information. [Em Linha] Disponível em <<http://pt.php.net/manual/en/faq.general.php>> Consultado em 12 /11/2008.

Pressman, R. (2001). Software Engineering A practitioner's approach (5º Edição ed.). McGraw-Hill.

Ribeiro, N. (2004). Multimédia e Tecnologias Interactivas. Lisboa: FCA – Editora Informática.

Rocha, Á. (2004). O Essencial da Análise de Sistemas.

Roof, L., & Fergus, D. (2003). The Definitive Guide to the .NET Compact Framework. Estados Unidos da América: Apress.

Rubin, E., & Yates, R. (2003). Microsoft® .NET Compact Framework Kick Start. Estados Unidos da América: Sams Publishing.

Serrão, C., & Marques, J. (2004). Programação com PHP 4.3. Lisboa: FCA – Editora Informática.

Sommerville, I. (2006). Software engineering (8º Edição ed.). Addison Wesley.

Vandré, P. (2007). Revista PHP / CSS - O que é CSS?. [Em Linha] Disponível em <<http://www.revistaphp.com.br/artigo.php?id=119>> Consultado em 10/11/2008.

Vasconcelos, J. (2004). A Engenharia de Software. A Engenharia de Software - Uma Abordagem Orientada a Objectos.

W3C. (2008). Extensible Markup Language (XML). [Em Linha] Disponível em <<http://www.w3.org/XML/>> Consultado em 08/11/2008.

What is Polyline? (2003). [Em Linha] Disponível em <<http://www.webopedia.com/TERM/P/polyline.html>> Consultado em 12/11/2008.

Wigley, A., Wheelwright, S., Burbidge, R., MacLoed, R., & Sutton, M. (2003). Microsoft .NET Compact Framework (Core Reference). United States of America: Microsoft Press.

wss. (2007). October 2007 Web Server Survey. [Em Linha] Disponível em <http://news.netcraft.com/archives/2007/10/11/october_2007_web_server_survey.html> Consultado em 13/11/2008.

Yao, P., & Durant, D. (2003). Programming the .NET Compact Framework in C#. Prentice Hall.

Anexos

1. Instruções para a experiência.
2. Questionário para os participantes na experiência.
3. CD-ROM contendo as versões executáveis e o código-fonte correspondente às aplicações desenvolvidas.

Anexo I

Instruções para a experiência:

Comparações	Grupo A	Grupo B
Tecnológica	Não usa a aplicação	Usa a aplicação
Temporal	Tempo demorado a encontrar a informação.	
Facilidade	Facilidade de pesquisa.	
Eficiência	Consegue encontrar toda a informação pretendida.	

Desafios:

1. O desafio consiste em procurar um cinema que tenha um filme do seu agrado.

O objectivo para a conclusão do desafio será, para além de escolher um filme, resumir o respectivo argumento, a que hora existe uma sessão que lhe convém, qual o número de telefone para marcação de bilhetes e detalhes sobre o cinema em que o filme está em exibição, tais como, a respectiva localização.

Pode utilizar todas as ferramentas que tiver ao seu dispor para encontrar a informação pedida. Pedimos apenas, que no final, nos indique quais as ferramentas que utilizou.

Título do filme:	
Sinopse do filme:	
Hora de início da sessão:	
Hora de fim da sessão:	
Nome do cinema:	
Endereço do cinema:	
Número para marcação de bilhetes:	
Coordenadas GPS:	

2. O desafio consiste em procurar um restaurante do seu agrado.

O objectivo para a conclusão do desafio será, para além, de escolher o restaurante, conseguir indicar mais detalhes sobre o restaurante escolhido, tais como a respectiva localização e as especialidades do restaurante. Caso seja possível escolher antecipadamente, deverá ainda escolher um prato.

Pode utilizar todas as ferramentas que tiver ao seu dispor para encontrar a informação pedida. Pedimos apenas que, no final indique quais as ferramentas que utilizou.

Nome do restaurante:	
Endereço do restaurante:	
Número de reservas:	
Escolha do prato:	
Especialidades:	
Coordenadas GPS:	

Anexo II

Questionário:

Este estudo está a ser realizado no âmbito do projecto de conclusão de Mestrado em Computação Móvel na Universidade Fernando Pessoa. Desde já agradecemos a sua disponibilidade em contribuir para este trabalho.

Neste questionário todas as suas respostas serão estritamente confidenciais e anónimas, pelo que solicitamos o favor de não escrever o seu nome na folha do questionário. Os dados recolhidos destinam-se exclusivamente a tratamento estatístico.

Não existem respostas certas ou erradas. Somente importa a sua opinião em relação ao que sentiu enquanto experimentou a utilização das aplicações. De modo a obtermos dados correctos é muito importante que dê as suas opiniões mais sinceras: não se preocupe se as suas respostas irão causar boa impressão já que o questionário é anónimo.

1) Indique, por favor, a sua idade:

Menos de 20 anos	<input type="checkbox"/>	Entre 21 e 30 anos	<input type="checkbox"/>	Entre 31 e 40 anos	<input type="checkbox"/>
Entre 41 e 50 anos	<input type="checkbox"/>	Entre 51 e 60 anos	<input type="checkbox"/>	Mais de 60 anos	<input type="checkbox"/>

2) Indique, por favor, a frequência com que utiliza:

a) Um computador

Nunca utilizei	<input type="checkbox"/>	1 vez por semana	<input type="checkbox"/>
2 a 3 vezes por semana	<input type="checkbox"/>	Diariamente	<input type="checkbox"/>

b) Um PocketPC

Nunca utilizei	<input type="checkbox"/>	1 vez por semana	<input type="checkbox"/>
2 a 3 vezes por semana	<input type="checkbox"/>	Diariamente	<input type="checkbox"/>

3) Indique ainda o local onde normalmente utiliza:

a) Um computador (Se respondeu nunca utilizei na pergunta 2ª, salte para a pergunta 3b)

Em casa No trabalho Em casa e no trabalho

b) Um PocketPC (Se respondeu nunca utilizei na pergunta 2ª, salte para a próxima pergunta)

Em casa No trabalho Em casa e no trabalho

Para as seguintes afirmações, agradecemos que indique a sua opinião assinalando com uma cruz a caixa respectiva.

1) O vocabulário utilizado em ambas as aplicações é adequado.

Concordo totalmente Concordo Sem opinião Discordo Discordo totalmente

2) A localização dos botões e do texto que se apresenta no ecrã facilita a manipulação da informação

a) Na aplicação do computador

Concordo totalmente Concordo Sem opinião Discordo Discordo totalmente

b) Na aplicação do PocketPC

Concordo totalmente Concordo Sem opinião Discordo Discordo totalmente

3) Qual foi o grau de dificuldade que sentiu na introdução de dados:

a) No computador:

Muito difícil Difícil Mediano Fácil Muito fácil

b) No PocketPC

Muito difícil Difícil Mediano Fácil Muito fácil

4) A função dos botões das aplicações é fácil de identificar:

Concordo totalmente Concordo Sem opinião Discordo Discordo totalmente

5) A localização da informação nos ecrãs da aplicação é fácil de recordar:

a) Na aplicação do computador

Concordo totalmente Concordo Sem opinião Discordo Discordo totalmente

b) Na aplicação do PocketPC

Concordo totalmente Concordo Sem opinião Discordo Discordo totalmente

10) Qual é a sua opinião em relação às tarefas que realizou recorrendo às aplicações?

a) Julgo que se trata de uma forma simples de se publicar um anúncio/notícia:

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Concordo totalmente	Concordo	Sem opinião	Discordo	Discordo totalmente

b) Julgo que se trata de uma forma simples e eficiente de recolher informação:

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Concordo totalmente	Concordo	Sem opinião	Discordo	Discordo totalmente

Para as seguintes questões, agradecemos que indique a sua opinião escrevendo as razões que a justificam nas linhas correspondentes

11) O que acha da aplicação em termos de utilidade?

12) Na sua opinião que outras funcionalidades gostaria de ver implementadas no futuro?

13) Qual é a sua opinião em relação à futura utilização desta aplicação?

Muito agradecemos a atenção disponibilizada no preenchimento deste questionário.

Anexo III

CD-ROM