



Dissertação

Mestrado em Computação Móvel

***Modelo para a gestão de aplicações móveis assistida  
por eventos***

**Sílvia Bernardes**

Leiria, *Fevereiro* de 2010



*Esta página foi intencionalmente deixada em branco*





Dissertação

Mestrado em Computação Móvel

***Modelo para a gestão de aplicações móveis assistida  
por eventos***

**Sílvio Bernardes**

Dissertação de Mestrado realizada sob a orientação do Doutor Rui Rijo, Professor da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria

Leiria, *Fevereiro* de 2010



*Esta página foi intencionalmente deixada em branco*



*“640 Kb serão suficientes para qualquer pessoa.”*

*Autor Bill Gates, 1981*



*Esta página foi intencionalmente deixada em branco*



## ***À Minha Família***

À Filipa Seco que me acompanhou ao longo de todas as noites que passei sem dormir, transmitindo energia positiva para ultrapassar mais esta etapa da minha vida.

Aos meus pais, Manuel e Fátima que foram fonte encorajadora durante toda esta etapa.

Um especial agradecimento ao Eng.º José Miguel Barros pela disponibilização do dispositivo móvel utilizado ao longo do projecto, e por ter dado sentido de orientação ao tema da tese.

Ao Tiago Seco por todo o apoio e disponibilidade.

*Esta página foi intencionalmente deixada em branco*

## ***Agradecimentos***

---

Em primeiro lugar agradeço ao meu orientador Rui Pedro Charters Lopes Rijo, por toda a disponibilidade, compreensão e encorajamento dado durante a elaboração deste projecto.

Agradeço a toda a equipa da *SysDev*, e em particular ao Eng.º Arsénio Gil que disponibilizou de forma gratuita a plataforma *Kalipso*.

Aos meus amigos por toda a força acompanhamento e amizade.

O meu muito obrigado a todos.

*Esta página foi intencionalmente deixada em branco*

## **Resumo**

---

As Tecnologias de Informação e Comunicação são utilizadas praticamente em todos os sectores de actividade. Com a adopção, utilização e proliferação destas tecnologias nos dispositivos móveis, ocorre um significativo e complexo desenvolvimento ao nível das aplicações móveis.

Os fabricantes de *hardware* empenham-se no aperfeiçoamento dos seus componentes. Este aperfeiçoamento tem o desígnio de melhorar os índices de desempenho dos componentes e equipamentos.

A computação móvel tem vindo a sofrer constantes melhorias ao nível do processamento, armazenamento e autonomia, no entanto, existem vários desafios associados aos dispositivos móveis. Este documento foca essencialmente os desafios existentes ao nível do processamento e da autonomia nos dispositivos móveis.

Para minimizar os problemas de processamento e autonomia é apresentado um modelo que pretende minimizar o processamento e aumentar a autonomia dos dispositivos móveis. Pretende-se então abordar as vantagens da utilização de um modelo de notificação por eventos, utilizando processamento descentralizado em dispositivos móveis. Como parte do processamento é deslocado para outros computadores, é fundamental a presença de uma lógica eficaz de comunicação entres os dispositivos. Para esse fim, é utilizado um protocolo de comunicação baseado em texto. Com este modelo, as aplicações móveis apenas são executadas quando estritamente necessárias, evitando assim consumir os limitados recursos dos dispositivos móveis.

*Palavras-chave: computação móvel, smartphones, aplicações móveis, desempenho dispositivos móveis .*

*Esta página foi intencionalmente deixada em branco*

# ***Abstract***

---

Information and Communication Technologies are used in almost every activity sector. With the adoption and proliferation of such technologies in mobile devices, a complex and major growth of mobile applications is observed.

Hardware manufacturers are keen on perfecting their components. Such perfecting is done with the intent of improving component and equipment's performance indexes.

Mobile computing has been undergoing constant improvements as far as processing, storage and autonomy are concerned. However, there remain multiple challenges associated with mobile devices. This document will be focusing essentially on the challenges presented by the processing power and autonomy constraints.

In order to minimize the problems presented by the processing power and autonomy constraints, a model is presented which aims at minimizing the processing requirements and increasing the autonomy of mobile devices.

In more detail, the benefits of the utilization of a distributed event driven notification system will be studied. Since some of the processing will be shifting from mobile devices towards workstations, an efficient communication and control logic is mandatory. In order to achieve this, a text-based communication protocol is used. With this model, mobile applications are only executed when strictly necessary, avoiding the waste of mobile devices' limited resources.

*Key-Words: mobile computing, smartphones, mobile applications, mobile devices, performance.*

*Esta página foi intencionalmente deixada em branco*

## Índice de Figuras

---

Figura 1 Modelo <i>Wap</i> .....	13
Figura 2 Exemplo de ligações <i>Bluetooth</i> .....	14
Figura 3 Camadas de um sistema operativo móvel.....	21
Figura 4 Desempenho do dispositivo <i>versus</i> tamanho.....	30
Figura 5 Modelo de gestão de aplicações móveis assistido por eventos.....	40
Figura 6 Arquitectura para os serviços disponibilizados pelo servidor de <i>email</i> . ....	42
Figura 7 Arquitectura para o gestor aplicacional. ....	43
Figura 8 Arquitectura da aplicação cliente.....	44
Figura 9 Exemplo da arquitectura proposta para a consulta de <i>email</i> . ....	44
Figura 10 Fluxo informação e processamento para a consulta de <i>emails</i> . ....	45
Figura 11 Ecrã principal do gestor aplicacional. ....	48
Figura 12 Ecrã principal aplicação <i>Web</i> . ....	48
Figura 13 Ecrã principal da aplicação cliente genérica implementando o modelo proposto. ....	49
Figura 14 Aplicação móvel, cliente meteorológico implementando o modelo proposto.....	50
Figura 15 Ecrã da aplicação móvel cliente de <i>email</i> implementando o modelo proposto. ....	51
Figura 16 Ecrã principal da aplicação cliente de <i>email</i> implementando o modelo existente.....	52
Figura 17 Ecrã da aplicação cliente meteorológico implementando o modelo existente. ....	52
Figura 18 Aplicação móvel de registo dos sinistros rodoviários.....	53
Figura 19 Aplicação de registo dos sinistros rodoviários. ....	54
Figura 20 Arquitectura de uma aplicação cliente implementando o modelo de <i>email</i> existente. ...	59
Figura 21 Esquema da arquitectura proposta.....	60
Figura 22 Arquitectura de notificação similar à utilizada nas chamadas telefónicas e <i>SMSs</i> . ....	75
Figura 23 Ecrã principal do gestor de aplicações móveis assistido por eventos.....	83
Figura 24 Ecrã de configurações do gestor de aplicacional.....	84
Figura 25 Mensagens informativas do estado da execução do gestor. ....	84
Figura 26 Ecrã principal do gestor. ....	85
Figura 27 Mecanismos disponibilizados pelo gestor aplicacional.....	86
Figura 28 Aplicação de configuração do gestor de aplicações móveis assistido por eventos. ....	89
Figura 29 Registo na aplicação de configuração do gestor. ....	89
Figura 30 Formulário de configuração dos serviços do gestor aplicacional.....	90
Figura 31 Listagem dos serviços configurados no gestor aplicacional. ....	91
Figura 32 Aplicação móvel genérica consumidora de eventos. ....	95
Figura 33 Configuração da aplicação móvel genérica consumidora de eventos. ....	96

Figura 34 Autenticação na aplicação móvel genérica consumidora de eventos.....	96
Figura 35 Notificações na aplicação móvel genérica consumidora de eventos.....	97
Figura 36 Aplicação de <i>email</i> móvel implementando o modelo proposto. ....	101
Figura 37 Ecrã principal do cliente de <i>email</i> implementando o modelo existente.....	105
Figura 38 Ecrã de configurações do cliente de <i>email</i> implementando o modelo existente. ....	106
Figura 39 Aplicação meteorológica implementando o actual modelo existente.....	111
Figura 40 Aplicação meteorológica implementando o modelo existente. ....	112

# Índice de Tabelas

---

Tabela 1 Principais marcos da computação. ....	7
Tabela 2 Normas IEEE 802.1x. ....	15
Tabela 3 Protocolo para o comando “conn”. ....	41
Tabela 4 Protocolo para o comando “runApp”. ....	41
Tabela 5 Protocolo para o comando “runPopUp”. ....	41
Tabela 6 Protocolo para o comando “generic”. ....	42
Tabela 7 Tipo de configurações do gestor aplicacional.....	43
Tabela 8 Etapas da utilização da arquitectura proposta para consulta de <i>email</i> .....	45
Tabela 9 Serviços disponibilizados pelo <i>Webservice</i> .....	56
Tabela 10 Eventos provenientes do gestor aplicacional. ....	57
Tabela 11 Grupo de testes.....	58
Tabela 12 Etapas do fluxo de informação numa aplicação implementando o modelo existente. ..	59
Tabela 14 Etapas do fluxo de informação na arquitectura proposta.....	60
Tabela 15 Exemplo de um ficheiro <i>XML</i> de configuração do gestor aplicacional. ....	84
Tabela 16 Ficheiro <i>XML</i> da aplicação móvel genérica consumidora de eventos. ....	96
Tabela 17 Ficheiro <i>XML</i> da aplicação móvel de <i>email</i> implementando o modelo proposto. ....	102
Tabela 18 Ficheiro <i>XML</i> da aplicação móvel de <i>email</i> implementando o modelo existente. ....	107
Tabela 19 Ficheiro <i>XML</i> da aplicação de consulta das condições meteorológicas. ....	112

*Esta página foi intencionalmente deixada em branco*

## Índice de Gráficos

---

Gráfico 1 Evolução tecnológica. ....	8
Gráfico 2 Penetração de banda larga móvel activa na população da União Europeia. ....	18
Gráfico 3 Taxa de cobertura 3G na EU. ....	19
Gráfico 4 Sistemas operativos utilizados nos <i>smartphones</i> em 2008. ....	23
Gráfico 5 Percentagem de bateria consumida durante os testes. ....	63
Gráfico 6 Tempo de duração máximo da bateria do dispositivo móvel durante os testes da chegada de um <i>email</i> de vinte em vinte minutos. ....	65
Gráfico 7 Tempo de duração máximo da bateria do dispositivo móvel utilizando a aleatoriedade na recepção de <i>emails</i> . ....	67
Gráfico 8 Tempo de duração máximo da bateria do dispositivo móvel utilizando a aleatoriedade na recepção de <i>emails</i> . ....	69

*Esta página foi intencionalmente deixada em branco*

## ***Lista de Siglas***

---

<b>3G</b>	<i>Third Generation</i>
<b>AMPS</b>	<i>Advanced Mobile Phone System</i>
<b>ANACOM</b>	<i>Autoridade Nacional de Comunicações</i>
<b>API</b>	<i>Application Programming Interface</i>
<b>ARM</b>	<i>Advanced Risc Machine</i>
<b>CDMA</b>	<i>Code Division Multiple Access</i>
<b>CRM</b>	<i>Customer Relationship Management</i>
<b>D-AMPS</b>	<i>Digital Advanced Mobile Phone System</i>
<b>EMS</b>	<i>Enhanced Messaging Service</i>
<b>FDMA</b>	<i>Frequency Division Multiple Access</i>
<b>Gbps</b>	<i>Gigabits per second</i>
<b>GPRS</b>	<i>General Packet Radio Service</i>
<b>GPS</b>	<i>Global Positioning System</i>
<b>GSM</b>	<i>Global System for Mobile Communications</i>
<b>HDD</b>	<i>Hard Disk Drive</i>
<b>HiperLan</b>	<i>High Performance Radio Local Area Network</i>
<b>HTC</b>	<i>High Tech Computer Corporation</i>
<b>I/O</b>	<i>Input/Output</i>
<b>IEEE</b>	<i>Institute of Electrical and Electronics Engineers</i>
<b>INE</b>	<i>Instituto Nacional Estatística</i>
<b>IP</b>	<i>Internet Protocol</i>
<b>IPv4</b>	<i>Internet Protocol Version four</i>
<b>IPv6</b>	<i>Internet Protocol Version six</i>
<b>IrDA</b>	<i>Infra Red Data Association</i>
<b>ITU</b>	<i>International Telecommunication Union</i>
<b>Kb</b>	<i>Kilo byte</i>
<b>MB</b>	<i>Mega Byte</i>
<b>MMS</b>	<i>Multimedia Message Service</i>
<b>Mobile TV</b>	<i>Mobile Television</i>
<b>NMT</b>	<i>Nordic Mobile Telephony</i>
<b>OOP</b>	<i>Object Oriented Programming</i>

<b>OS</b>	<i>Operating System</i>
<b>PAN</b>	<i>Personal Area Network</i>
<b>PC</b>	<i>Personal Computer</i>
<b>PCS</b>	<i>Personal Communication Services</i>
<b>PDA</b>	<i>Personal Digital Assistants</i>
<b>PDC</b>	<i>Personal Digital Cellular (PDC)</i>
<b>PHS</b>	<i>Personal Handy Phone System</i>
<b>POO</b>	Programação Orientada aos Objectos
<b>RAD</b>	<i>Rapid Application Development</i>
<b>RAM</b>	<i>Random Access Memory</i>
<b>RIM</b>	<i>Research In Motion</i>
<b>SDK</b>	<i>Software Development Kit</i>
<b>SIG</b>	<i>Special Interest Group</i>
<b>SMS</b>	<i>Short Message Service</i>
<b>SOA</b>	<i>Service Oriented Architecture</i>
<b>SOAP</b>	<i>Simple Object Access Protocol</i>
<b>SRAM</b>	<i>Static Random Access Memory</i>
<b>SSL</b>	<i>Secure Socket Layer</i>
<b>STM</b>	Serviço Telefónico Móvel
<b>TACS</b>	<i>Total Communication System</i>
<b>TCP</b>	<i>Transmission Control Protocol</i>
<b>TCP/IP</b>	<i>Transmission Control Protocol / Internet Protocol</i>
<b>TDMA</b>	<i>Time Division Multiple Access</i>
<b>TV</b>	<i>Television</i>
<b>UE</b>	União Europeia
<b>UMTS</b>	<i>Universal Mobile Telecommunication Systems</i>
<b>USB</b>	<i>Universal Serial Bus</i>
<b>VCR</b>	<i>Video Cassette Recorder</i>
<b>WAP</b>	<i>Wireless Application Protocol</i>
<b>Wi-Fi</b>	<i>Wireless Fidelity</i>
<b>WWW</b>	<i>World Wide Web</i>
<b>XML</b>	<i>eXtensible Markup Language</i>

# Índice

---

DEDICATÓRIA.....	I
AGRADECIMENTOS .....	III
RESUMO.....	V
ABSTRACT .....	VII
ÍNDICE DE FIGURAS .....	IX
ÍNDICE DE TABELAS .....	XI
ÍNDICE DE GRÁFICOS .....	XIII
LISTA DE SIGLAS .....	XV
ÍNDICE .....	XVII
<b>1 INTRODUÇÃO.....</b>	<b>1</b>
1.1 Objectivos e contribuições .....	2
1.2 Estrutura da dissertação.....	4
<b>2 REVISÃO DA LITERATURA.....</b>	<b>5</b>
2.1 O Paradigma da computação móvel .....	5
2.2 Evolução tecnológica .....	6
2.2.1 Breve história da computação móvel.....	6
2.2.2 Comunicações móveis .....	8
2.2.3 <i>Internet Protocol (IP)</i> .....	11
2.2.4 Evolução tecnológica dos protocolos móveis.....	12
2.2.5 Tipos de dispositivos móveis .....	15
2.2.6 Mercado telefónico móvel .....	17
2.2.7 Sistemas operativos móveis .....	20
2.2.8 Plataformas.....	22
2.2.9 Análise comparativa dos sistemas operativos.....	27
2.2.10 Sistemas móveis <i>versus</i> sistemas tradicionais.....	29
2.2.11 Desafios inerentes à mobilidade dos sistemas.....	29

<b>3</b>	<b>METODOLOGIA</b> .....	<b>33</b>
3.1	Problema .....	33
3.2	Processo de investigação.....	34
3.3	Características do modelo .....	35
3.3.1	Equipamento .....	35
3.3.2	Plataformas de apoio ao desenvolvimento.....	35
3.4	Desenvolvimento.....	37
<b>4</b>	<b>MODELO DE NOTIFICAÇÃO POR EVENTOS</b> .....	<b>39</b>
4.1	Apresentação do modelo de notificação por eventos .....	39
4.2	Protocolo .....	40
4.3	Modelo .....	42
4.3.1	Serviços.....	42
4.3.2	Gestor aplicacional .....	42
4.3.3	Dispositivo móvel .....	43
4.4	Modelo funcional.....	46
4.5	Análise do modelo .....	46
4.6	Protótipos .....	46
4.6.1	Protótipos de suporte ao modelo proposto .....	47
4.6.2	Protótipos de suporte ao modelo existente.....	51
<b>5</b>	<b>TRABALHO EXPERIMENTAL</b> .....	<b>55</b>
5.1	Plano de avaliação dos testes.....	55
5.1.1	Testes de validação.....	56
5.1.2	Testes de avaliação.....	58
5.2	Grupo de testes avaliação ao modelo existente <i>versus</i> modelo tradicional durante 180 minutos aplicado à recepção periódica de <i>emails</i> [Grupo A].....	61
5.2.1	Teste ao modelo existente .....	61
5.2.2	Teste ao modelo proposto .....	61
5.2.3	Análise dos resultados.....	62
5.3	Grupo de testes de avaliação ao modelo existente <i>versus</i> modelo tradicional durante uma carga completa de bateria aplicado à recepção periódica de <i>emails</i> [Grupo B].....	63
5.3.1	Teste ao modelo existente .....	63
5.3.2	Teste ao modelo proposto .....	64
5.3.3	Avaliação dos resultados .....	64

5.4	Grupo de testes de avaliação do modelo existente <i>versus</i> modelo tradicional durante uma carga completa de bateria aplicado à recepção aleatória de <i>emails</i> [Grupo C] .....	65
5.4.1	Teste ao modelo existente .....	66
5.4.2	Teste ao modelo proposto .....	66
5.4.3	Avaliação dos resultados .....	67
5.5	Grupo de testes de avaliação do modelo existente <i>versus</i> modelo tradicional durante uma carga completa de bateria aplicado à consulta das condições meteorológicas [Grupo D] .	67
5.5.1	Teste ao modelo existente .....	68
5.5.2	Teste ao modelo proposto .....	68
5.5.3	Avaliação dos resultados .....	69
5.6	Grupo de testes de avaliação da utilização do modelo proposto no dispositivo de registo de sinistros rodoviários [Grupo E]. .....	69
5.6.1	Teste ao modelo proposto .....	69
5.7	Análise dos resultados.....	70
<b>6</b>	<b>CONCLUSÃO.....</b>	<b>71</b>
6.1	Problema .....	72
6.2	Metodologia .....	72
6.3	Testes.....	73
6.4	Principais contributos.....	74
6.5	Principais conclusões do estudo.....	74
6.6	Trabalho Futuro .....	75
	<b>BIBLIOGRAFIA .....</b>	<b>77</b>
	<b>APÊNDICES .....</b>	<b>79</b>
	<b>ANEXOS.....</b>	<b>113</b>

*Esta página foi intencionalmente deixada em branco*

# 1 Introdução

---

A proliferação a nível mundial dos computadores pessoais e a constante evolução da computação móvel proporciona aos utilizadores uma ampla gama de aplicações e serviços. Estas aplicações e serviços são independentes do local onde o dispositivo se encontra.

Com o crescimento do número de utilizadores de dispositivos móveis, estima-se que em poucos anos milhões de pessoas possuam um destes dispositivos (Ribeiro 2006).

Grande parte dos dispositivos vai ter a capacidade de efectuar comunicações sem fio. Podem interligar-se com outros dispositivos portáteis ou com equipamentos que façam parte integrante de uma rede fixa (Ribeiro 2006).

Com a evolução da computação móvel, emerge a necessidade de aperfeiçoar os serviços e aplicações disponibilizados por este tipo de dispositivos.

Verifica-se que os computadores portáteis estão gradualmente a substituir os computadores de secretária. O seu reduzido peso e a facilidade de transporte influenciam esta tendência.

No contexto das inovações em torno da tecnologia móvel, incluem-se as consolas portáteis, os *Personal Digital Assistante (PDA)* ou os *smartphones*. As consolas já disponibilizam ambientes acelerados a 3D e capacidade de navegação na *Internet*.

Os *PDA* e os *smartphones* são produtos com grande aceitação no mercado. Nestes, podem ser instaladas aplicações de processamento de cálculo, texto, áudio, vídeo, imagem ou mesmo aplicações de gestão empresarial.

A utilização generalizada dos telefones móveis veio facilitar o contacto entre as pessoas. Com a aceitação dos telefones móveis, as pessoas ficam facilmente contactáveis, independentemente do local onde se encontram. As empresas adoptam-nos como imprescindíveis instrumentos de trabalho, possibilitando um fluxo contínuo de informação.

Com a evolução destes dispositivos os utilizadores particulares e empresariais passam a usufruir nos dispositivos móveis de funcionalidades semelhantes às encontradas nos computadores pessoais.

Inicialmente, os telefones móveis eram apenas utilizados como um meio de comunicação de voz com suporte para pequenas aplicações simplistas e a 2D. Actualmente é bastante comum encontrar nos telefones móveis, aplicações de agenda, jogos 3D ou de acesso à *Internet*. Com o crescimento do mercado dos *smartphones* as aplicações têm sofrido constantes evoluções. Nestes dispositivos podemos encontrar aplicações de processamento de texto, jogos 3D, leitores de música, acessos à *Internet* de terceira geração (3G) ou aplicações de *Customer Relationship Management (CRM)*<sup>1</sup>.

Com os progressos ao nível da usabilidade dos dispositivos, das infra-estruturas e com o aumento da segurança nas redes móveis, espera-se aumentar o nível de confiança dos utilizadores neste tipo de computação.

Ao nível do *software* é cada vez mais evidente a tendência das empresas em disponibilizar aplicações compatíveis com os dispositivos móveis (Tavares e Ferreira 2009).

### 1.1 Objectivos e contribuições

Com a evolução tecnológica na área da computação móvel, aumenta a utilização deste tipo de dispositivos que são de fácil transporte e recarregamento. Com o desenvolvimento e aperfeiçoamento do *hardware* pretende-se aumentar o tempo útil de utilização efectiva do dispositivo.

A preocupação por parte dos fabricantes de *hardware* em melhorar as baterias e diminuir os consumos dos seus componentes electrónicos, contrapõe com o desenvolvimento de aplicações mais sofisticadas e complexas que consomem cada vez mais recursos.

A dissertação foca-se na especificação e concepção, de um modelo que permita minimizar os consumos energéticos dos dispositivos móveis.

---

<sup>1</sup> CRM – aplicação responsável por armazenar e analisar o relacionamento dos clientes com as organizações. Para este fim utiliza uma base de dados capaz de registar todas as transacções dos clientes.

Seguidamente são apresentados os principais contributos da dissertação.

### **Modelo de gestão de aplicações móveis assistida por eventos**

A primeira contribuição desta dissertação é a proposta de um modelo de suporte à gestão de aplicações móveis assistida por eventos. Com este modelo pretende-se melhorar o tempo de utilização dos dispositivos móveis.

### **Suporte ao modelo de gestão de aplicações móveis assistida por eventos**

O modelo apresentado funciona em servidores com uma aplicação de gestão aplicacional instalada, e com uma aplicação cliente móvel a ser executada num *smartphone*. Os servidores são responsáveis por efectuar parte do processamento que um dispositivo móvel habitualmente realiza.

Para além deste processamento o gestor aplicacional fica responsável por notificar os dispositivos móveis da ocorrência de eventos previamente configurados pelo utilizador. Consoante a notificação, é despoletado o processamento no dispositivo móvel.

### **Protocolo de comunicação**

Foi desenvolvido um protocolo de comunicação, que permite aos diversos intervenientes na arquitectura comunicarem entre si de forma transparente. Este é um protocolo simples e baseado em texto.

### **Desenvolvimento dos mecanismos propostos em ambiente de simulação**

Como resultado das contribuições anteriormente referenciadas, surge um ambiente de simulação desenvolvido sobre a plataforma *Microsoft Visual Studio .Net*.

Foi desenvolvida uma aplicação servidor que gere parte do processamento do dispositivo móvel e uma aplicação móvel responsável por processar os comandos provenientes do gestor aplicacional.

### **Sítio na *Internet***

Foi criado um sítio na *Internet* (<http://aplicacoesmoveis.golink.pt>) que disponibiliza todo material utilizado durante o projecto. Aqui podem ser consultados os testes, as referências bibliográficas, os documentos produzidos, efectuado o *download* da base de dados, dos protótipos, e do seu código fonte.

## **1.2 Estrutura da dissertação**

A presente dissertação começa por abordar no capítulo dois a revisão da literatura. Esta revisão ajudou a validar o estado da arte actual. Neste capítulo, estudou-se a evolução da computação móvel ao longo do tempo, e as vantagens e desvantagens desta vertente da computação.

O capítulo três foca a metodologia. Neste capítulo é apresentado o problema e a investigação efectuada ao longo da dissertação.

O capítulo quatro aborda o modelo de gestão por eventos. É apresentada a arquitectura, as características fundamentais para o desenvolvimento da mesma e um caso prático para a aplicação do modelo.

No capítulo cinco, é efectuada a avaliação da arquitectura com base nos testes realizados. Para os testes, foram utilizados diversos cenários distintos abrangendo situações habitualmente utilizadas pelo utilizador comum.

A conclusão da dissertação será apresentada no capítulo seis onde são analisados os resultados obtidos durante o corrente projecto.

## 2 Revisão da literatura

---

No capítulo anterior apresentou-se uma visão global da estrutura e do tema da dissertação.

Com o capítulo da revisão da literatura pretende-se identificar os principais marcos da história evolutiva da mobilidade e introduzir conceitos relevantes da área da computação móvel.

### 2.1 O Paradigma da computação móvel

As primeiras aplicações móveis focavam-se principalmente na organização de informação pessoal. Com o desenvolvimento tecnológico ocorrido na última década surgiu o paradigma da mobilidade computacional, impulsionando novos conceitos de computação. Assim, deixámos de estar dependentes de uma estrutura fixa de comunicações e passámos a usufruir de uma estrutura móvel de dados.

A computação móvel caracteriza-se como sendo a área da computação que permite a troca de dados entre terminais fixos ou móveis utilizando comunicações sem fio (*wireless*) (Junior e Fernandes 2006).

A computação móvel está presente num grande número de dispositivos. Destes dispositivos destacam-se os *PDA*, *smartphones* ou os *Global Positioning System (GPS)*, comumente utilizados no dia-a-dia. Estes dispositivos oferecem uma mais-valia ao mercado empresarial, pessoal e hospitalar.

O número de utilizadores que recorrem actualmente a dispositivos móveis aumenta a cada dia que passa, começando a fazer parte da rotina de cada cidadão (Junior e Fernandes 2006).

O paradigma da mobilidade computacional, apesar de recente, esteve sempre dependente da evolução e padronização das restantes áreas da computação. Com a padronização das redes e dos sistemas operativos foi possível evoluir de forma sustentável ao nível da mobilidade.

A evolução decorre procurando ultrapassar os diversos desafios existentes. Aspectos como a segurança, velocidade de processamento e capacidade de armazenamento continuam a sofrer fortes actualizações. Os constantes avanços tecnológicos advêm da necessidade de atingir níveis aceitáveis de confiança e satisfação dos utilizadores deste tipo de computação.

## **2.2 Evolução tecnológica**

A evolução tecnológica nos últimos anos apresenta sistemáticos avanços ao nível dos sistemas de informação e comunicação.

Com o aumento da eficiência e com a redução dos preços dos circuitos integrados, começou a ser viável a utilização de computadores pessoais. Esta redução dos preços impulsionou o desenvolvimento da tecnologia móvel, permitindo que rapidamente surgissem os primeiros terminais móveis, como por exemplo os *PDA*.

Assistimos a uma grande integração de dispositivos e redes móveis, possibilitando uma enorme diversidade de dispositivos e serviços aos utilizadores.

### **2.2.1 Breve história da computação móvel**

Com a evolução das redes sem fio, dos serviços disponibilizados por satélite e da computação móvel, tornou-se possível aceder à informação em qualquer altura, independentemente do lugar onde nos encontramos.

Verificando-se um elevado crescimento no sector da computação móvel, onde a cada dia aumenta o número de utilizadores, perspectiva-se que muito em breve este tipo de dispositivos faça parte da rotina de uma grande maioria da população a nível mundial (Junior e Fernandes 2006).

Com a computação móvel tornou-se inevitável a mobilidade dos sistemas de informação. Surgiu uma crescente vontade de diminuir as restrições existentes neste tipo de computação, quando comparada com a computação “imóvel” tradicional.

Na Tabela 1 são apresentadas as principais etapas evolutivas da computação ao longo de quatro décadas.

Década	Etapas
60	Processamento dados
70	Terminais
80	Redes
90	Computação Móvel

Tabela 1 Principais marcos da computação.

Os nossos antepassados chegaram a utilizar sinais de fumo para comunicar a longas distâncias. Desde essa altura que se tem verificado constantes evoluções para que esta se torne cómoda e eficiente.

Em **1820** *Hans Christian Oersted* descobre que a corrente eléctrica produz um campo magnético, tendo sido quantificada por *André Marie Ampère* como lei de *Ampère*.

*Michael Faraday* descobre em **1831**, que a variação do campo magnético induz uma corrente eléctrica.

Em **1864** *James Clark Maxwell* introduz melhorias à lei de *Ampère*, amplia a lei de *Faraday* e desenvolve quatro famosas equações sobre campos magnéticos “equações de *Maxwell*”.

O telefone surge como uma invenção de *Alexandre Bell* em **1876**.

Anos mais tarde surge o telégrafo sem fios por *Guglielmo Marconi*.

As primeiras comunicações sem fios surgiram em **1907**, com os primeiros sinais rádio que atravessavam o oceano Atlântico.

O início da primeira Guerra Mundial em **1914** impulsionou o desenvolvimento e aperfeiçoamento das comunicações. Em **1939**, com a segunda Guerra Mundial deu-se a expansão das comunicações via rádio.

Após o aparecimento dos sistemas de processamento de dados, dos terminais e das redes de computadores, começaram a surgir os primeiros computadores.

Nesta altura os sistemas sem fios eram bastante apelativos. No entanto, os elevados custos desta tecnologia eram um entrave ao desenvolvimento, sendo este bastante estimulado pelo crescente interesse dos consumidores.

Em **1992** surge o *Global Systems for Mobile Communications (GSM)*, seguido da *HiperLan* e da *Wireless Lan – Institute of Electrical and Electronics Engineers (IEEE) 802.11*.

No ano **2000** a rede *GSM* passa a suportar taxas de transferência mais elevadas e em **2001** surge o *3G*.

Desde o aparecimento do *3G* que a evolução tem sido muito acentuada quer com os actuais *Notebooks*, *iPhone* ou mesmo com a voz sobre o *Internet Protocol (IP)* tradicionalmente conhecida por *Voice Over Internet Protocol (VoIP)*.

Analisando o Gráfico 1 é possível perceber a evolução exponencial dos computadores e das tecnologias e serviços, quando comparados com outros sistemas tecnológicos que nos têm acompanhado ao longo dos anos.

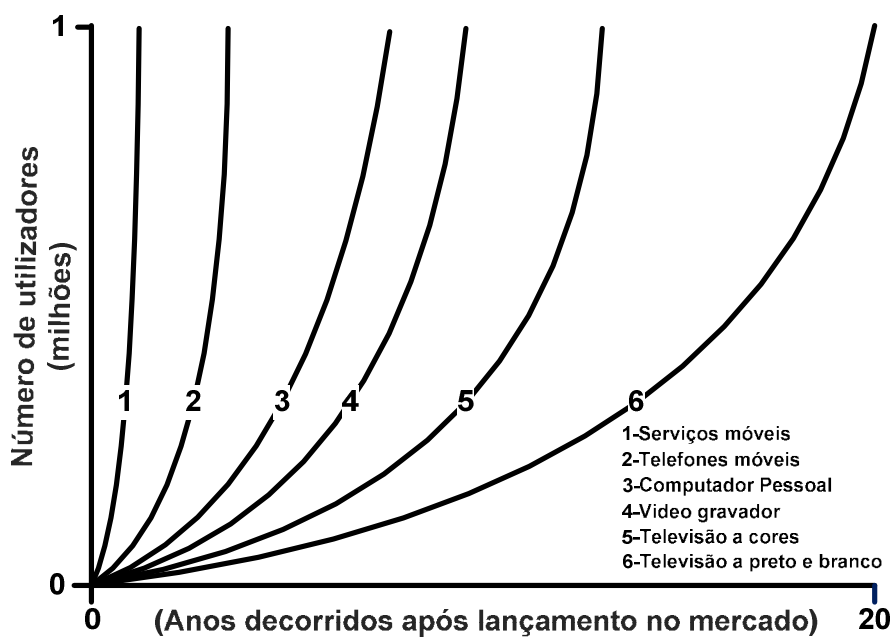


Gráfico 1 Evolução tecnológica.  
Adaptado de (Mateus e Loureiro 2004).

### 2.2.2 Comunicações móveis

As comunicações móveis de primeira geração (*1G*) eram analógicas, sucedendo-se a era digital com as comunicações de segunda (*2G*) e terceira geração (*3G*).

Com o aparecimento da segunda geração de comunicações móveis a *International Telecommunication Union (ITU)*, padronizou a identificação das diversas redes segundo uma terminologia de um número precedido da letra G (*1G, 2G, 3G, 4G, ...*) (Faísca 2009).

Apesar da tecnologia móvel estar-se a desenvolver desde 1950, apenas no final de 1970 início de 1980 é que surgiu o primeiro dispositivo móvel. (Mishra 2007)

Seguidamente são abordadas as diversas etapas evolutivas das redes móveis.

### **Primeira geração (1G)**

A primeira geração dos sistemas móveis (*1G*) era analógica, tendo a primeira rede móvel surgido em Tóquio. Anos depois surge na Europa a *Nordic Mobile Telephony (NMT)*, nos Estados Unidos os sistemas *Advanced Mobile Phone Service (AMPS)* baseados no *Frequency Division Multiple Access (FDMA)* e no Reino unido surge o *Total Communication System (TACS)* (Mishra 2007).

Apesar de todos estes serem analógicos e fazerem parte da mesma geração móvel, eram completamente incompatíveis entre si. Esta incompatibilidade devia-se à utilização de diferentes padrões, que também tornavam o *roaming*<sup>2</sup> bastante difícil (Faísca 2009).

Os equipamentos móveis associados à primeira geração eram bastante caros e de grandes dimensões, permitindo apenas comunicações de voz.

### **Segunda geração (2G)**

A segunda geração (*2G*) de comunicações móveis surge em **1990** sendo baseada nos sistemas digitais. Surge como necessidade de tornar os sistemas compatíveis entre si, eliminando os problemas de incompatibilidades existentes nos sistemas de primeira geração.

Com a segunda geração surgem sistemas como o *GSM*, *Time Division Multiple Access (TDMA)*, *Code Division Multiple Access (CDMA)*, *Personal Digital Cellular (PDC)* e o *Personal Handy Phone System (PHS)*. No entanto, o *GSM* é o protocolo adoptado pela maioria dos operadores móveis (Mishra 2007).

---

<sup>2</sup> *Roaming* – ocorre quando um utilizador usufrui dos serviços de outra rede que não é a sua.

A segunda geração possibilita um maior número de ligações em simultâneo, envio de *Small Messages Service (SMS)* e suporta a troca de dados entre dispositivos de fax e modem (Mishra 2007).

Apesar da evolução do analógico para o digital e da possibilidade de efectuar comunicações de *roaming* os sistemas focavam-se principalmente em serviços de voz, tornando os acessos à *Internet* lentos e ineficientes (Fáisca 2009).

### **Segunda geração e meia (2,5G)**

Em 2000, surge a segunda geração e meia (2,5G) que utiliza a tecnologia *General Packet Radio Service (GPRS)* que permite acessos mais rápidos e eficientes à *Internet* (Mishra 2007).

Com os sistemas *GPRS* que utilizam a transferência de dados por pacotes, é possível às operadoras facturarem aos utilizadores apenas o tráfego de dados efectuado (Mishra 2007).

### **Terceira geração (3G)**

Segundo as normas da *ITU*, surge em 2001 a terceira geração (3G). A tecnologia *3G* caracteriza-se pelos melhores desempenhos ao nível da transferência de dados, passando os utilizadores a poder usufruir de *Internet* móvel de alta velocidade.

O *3G* funciona nos mais diversificados ambientes. Funciona em ambientes interiores e exteriores, independentemente das condições climatéricas.

As infra-estruturas responsáveis pela disponibilização deste serviço podem ser terrestres ou através de satélite. Suportam grande diversidade de informação, sendo os jogos interactivos, vídeo-conferências, *Internet* de alta velocidade ou o *streaming* de áudio/vídeo as mais utilizadas (Karim e Mohsen 2002).

### **Quarta geração (4G)**

Começaram no Japão os primeiros testes à tecnologia de quarta geração (4G), que deverá ser lançada comercialmente no ano de **2010**. Esta permitirá atingir velocidades de cem *Mbps* quando nos encontrarmos em movimento e cinco *Giga bits* por segundo (*Gbps*) quando parados (Santana).

Assim, espera-se que na quarta geração os dispositivos tenham a capacidade de funcionar com variadas tecnologias sem fio. Espera-se também, que os dispositivos tenham a capacidade de se adaptarem automaticamente à rede e ao protocolo mais adequado (Mishra 2007).

### **2.2.3 Internet Protocol (IP)**

Nos anos noventa surgem os primeiros estudos sobre o *Transmission Control Protocol/Internet Protocol (TCP/IP)* que classificam o *Internet Protocol Version 4 (IPV4)* como obsoleto para os novos tipos de aplicações. Este protocolo é limitado ao nível da segurança e da capacidade de endereços que consegue disponibilizar para os serviços de *Internet* e de voz sobre *IP*.

Devido a estas limitações surge a necessidade de criar uma solução que possibilite aumentar o número de endereços que o protocolo consegue disponibilizar. Surge assim o *Internet Protocol Version 6 (IPV6)*.

### **Internet Protocol Version 4 (IPV4)**

A versão do *IPV4* sofreu poucas alterações desde o seu lançamento na década de setenta (Brandizzi, Santos et al. 2006).

Este protocolo utiliza endereçamento de 32 bits conferindo-lhe a capacidade de endereçar mais de quatro biliões de endereços. Com o crescimento das Tecnologias de Comunicação e Informação, com o crescimento das redes e com o aumento do número de telefones móveis este número de endereços torna-se insuficiente. Para além da desvantagem indicada, a segurança também é um dos seus problemas já que o protocolo não foi projectado para utilizar informação confidencial ou para suportar transacções financeiras (Brandizzi, Santos et al. 2006).

### ***Internet Protocol Version 6 (IPV6)***

A característica que mais se evidencia entre o *IPV6* e o *IPV4* é a capacidade de endereçamento. Este protocolo utiliza endereçamentos de 128 bits podendo suportando  $3,4 \times 10^{38}$  endereços distintos (Brandizzi, Santos et al. 2006).

A complexidade de migrar as redes de *IPV4* para *IPV6* influenciou o desenvolvimento de um protocolo compatível com o *IPV4* actualmente utilizado. Com o protocolo *IPV6* é possível representar um endereço *IPV4* permitindo que duas máquinas com protocolos diferentes possam comunicar entre si.

### **2.2.4 Evolução tecnológica dos protocolos móveis**

A primeira geração dos sistemas móveis era analógica, e baseada no acesso múltiplo por divisão de frequência (*FDMA*).

Com a era digital dez anos mais tarde surge a segunda geração de comunicações móveis.

Em 1990 padronizou-se o *TDMA* surgindo posteriormente o *Digital Advanced Mobile Phone System (D-AMPS)*, *CDMA*, *GSM*, *GPRS* e o *Universal Mobile Telecommunication Systems (UMTS)* (Mishra 2007). Das diversas infra-estruturas de comunicação existentes no mercado, padrões como o *TDMA*, *CDMA* e *GSM* ainda são dos mais utilizados, no entanto, prevê-se que as comunicações *3G* passem a dominar o mercado. Para além destes, podemos encontrar protocolos como o *Wireless Application Protocol (WAP)*, *Bluetooth* ou *Institute of Electrical and Electronics Engineers (IEEE) 802.11*, bastante importantes na evolução deste tipo de redes.

Seguidamente são abordados alguns dos protocolos utilizados nos sistemas de comunicação sem fio.

### **Wireless Application Protocol (WAP)**

O *WAP* é um protocolo que especifica o modo como a informação deve ser construída e apresentada nos dispositivos. A sua grande vantagem é ser independente do tipo de dispositivo utilizado (Freitas 2008).

Na Figura 1 está representado o esquema do modelo *WAP*.

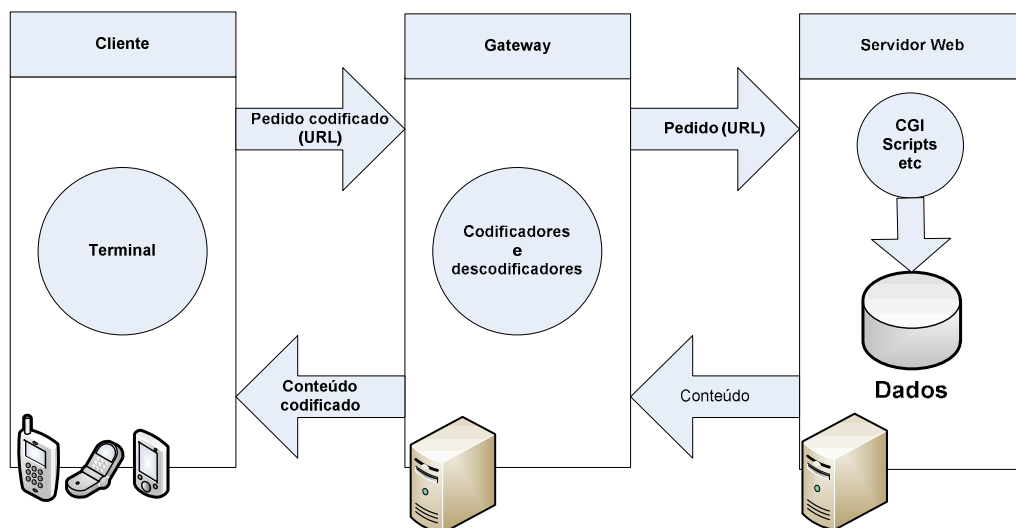


Figura 1 Modelo *Wap*.  
Adaptado de Forum, W. A. P. (Forum 2000).

### **Bluetooth**

O *Bluetooth* é um padrão proposto pelo *Bluetooth Sig* que é uma das maiores empresas de telecomunicações do mundo. Esta tecnologia surgiu com o intuito de substituir os habituais cabos que conectam os diversos periféricos de um sistema computacional.

Este padrão tem sofrido algumas evoluções, permitindo hoje em dia a comunicação entre diversos tipos de dispositivos em redes *ad-hoc*. Numa rede *ad-hoc* todos os dispositivos que fazem parte da rede funcionam como encaminhadores dos restantes terminais.

Devido ao seu reduzido alcance, costumam ser utilizados preferencialmente em pequenas redes pessoais, tradicionalmente conhecidas como *Personal Area Network (PAN)* (Haartsen 1998).

Na Figura 2 são apresentados exemplos de diversos tipos de ligações *Bluetooth*.

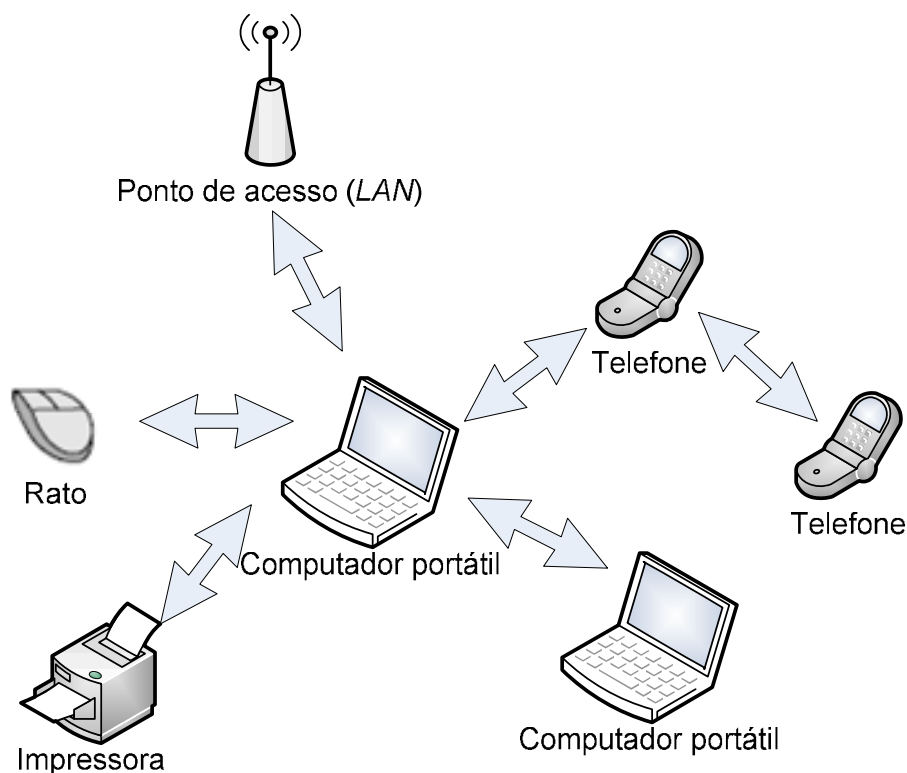


Figura 2 Exemplo de ligações *Bluetooth*.

Adaptado de Universal Radio Interface for *ad hoc*, wireless connectivity (Haartsen 1998).

### ***International Electrical and Electronics Engineers 802.11 (IEEE 802.11)***

Em **1997** após sete anos de pesquisas, o padrão *IEEE 802.11* foi aprovado, suportando comunicações sem fio – *Wireless Fidelity (Wi-Fi)*.

Este padrão tem sido alvo de constantes aperfeiçoamentos e foi adoptado por grande parte dos utilizadores de *Internet*. Estes aperfeiçoamentos e o crescente número de utilizadores proporcionaram um rápido crescimento das redes *Wireless* (Santana 2008).

O padrão *IEEE 802.11* tem-se destacado ao nível da cobertura e sobretudo pelo baixo custo de implementação. O desempenho deste padrão pode ser influenciado pela presença de obstáculos ou pela proximidade de fontes de interferência que reduzem a qualidade do sinal recepcionado (Nakahati, Oliveira et al. 2006).

A norma 802.1x tem vindo a ser melhorada ao longo dos anos. Na Tabela 2 pode ser observado as evoluções mais significativas deste modelo.

<b>Norma</b>	<b>Velocidade</b>
<i>IEEE 802.11b</i>	11 Mbps
<i>IEEE 802.11a</i>	54 Mbps
<i>IEEE 802.11g</i>	54 Mbps
<i>IEEE 802.11n</i>	600 Mbps

Tabela 2 Normas IEEE 802.1x.  
Adaptado de (BroadCom 2006).

Com excepção da norma *IEEE 802.11a*, todas as restantes vertentes da norma *IEEE 802.1x* apresentadas anteriormente são compatíveis entre si (Santos 2004).

Actualmente as normas *IEEE 802.11g* e *IEEE 802.11b* estão gradualmente a ser substituídas pela norma *IEEE 802.11n* que possibilita melhor propagação de sinal, maior segurança e maiores velocidades.

### **2.2.5 Tipos de dispositivos móveis**

Os dispositivos na área da computação móvel devem ter processamento autónomo, possibilitar troca de informação através da rede e acima de tudo serem de fácil transporte (Ribeiro 2006).

Estes dispositivos não devem estar dependentes de cabos. Devem possuir baterias para que o funcionamento seja independente da rede eléctrica, o que limitaria em muito, os espaços onde os dispositivos poderiam ser utilizados.

Dispositivos como *laptops*, *PDA*s ou telefones móveis, estão habitualmente associados à computação móvel. No entanto, aparelhos como o *GPS* ou consolas de jogos portáteis também são dispositivos desta mesma família.

### **Computadores portáteis**

São normalmente semelhantes aos computadores pessoais “*Personal Computers (PCs)*”, no entanto de menores dimensões e peso. O seu reduzido tamanho e peso, facilitam o transporte e permitem que os utilizadores usufruam das suas capacidades nos mais diversos pontos do planeta. A utilização de baterias permite que estes sejam autónomos, podendo ser utilizados em zonas onde a rede eléctrica não está acessível.

### ***Personal Digital Assistants (PDA)***

Os *PDA*s possuem capacidades de processamento e armazenamento inferiores às dos computadores pessoais. Permitem aos utilizadores aceder e consultar dados pessoais de forma simples e cómoda independentemente do local onde se encontrem.

O seu tamanho, peso reduzido, capacidade de programação e de execução de aplicações, a memória e a capacidade de se ligarem dispositivos de entrada e saída “*Input/Output (I/O)*” são as grandes vantagens destes dispositivos.

No que diz respeito às comunicações, encontram-se dotados com a tecnologia necessária para comunicarem segundo os mais recentes protocolos de comunicação sem fio.

Apesar das inúmeras vantagens anteriormente enunciadas, a falta de comodidade na introdução de textos, a reduzida dimensão do ecrã e a autonomia das suas baterias, são aspectos que ainda têm de ser aperfeiçoados.

Apesar dos problemas citados no parágrafo anterior, os *PDA*s são uma excelente opção para quem precisa da informação junto de si.

### **Telefones móveis**

Os primeiros telefones móveis serviam única e exclusivamente para efectuar chamadas. Com a evolução tecnológica começaram a adquirir capacidade de processamento, armazenamento e a disponibilizar interfaces para ligações a redes sem fio.

A maioria dos telefones móveis existentes no mercado possui ecrãs de dimensões reduzidas e teclados alfanuméricos.

Desde o lançamento do primeiro *smartphone* em 1994 que os dispositivos móveis passam a incorporar funções de *PDA* e telefone, transformando-se em dispositivos híbridos. Estes permitem a utilização de aplicações mais complexas e disponibilizam melhor suporte para expansão de dispositivos de entrada e saída (Jakl 2007).

Como aspectos menos positivos destaca-se a reduzida autonomia das baterias, aquando da utilização exaustiva das suas funcionalidades e os seus ecrãs reduzidos que dificultam a consulta e a introdução de dados.

### **2.2.6 Mercado telefónico móvel**

O Serviço Telefónico Móvel (*STM*) na União Europeia (*UE*) tem vindo a aumentar nos últimos anos. Em 2008 cresceu 7% relativamente ao ano transacto, tendo uma penetração de 119 por cada 100 habitantes (ANACOM 2009).

O serviço 3G também tem vindo a aumentar, ocupando em 2008 15,5% dos assinantes de serviços móveis e sendo já 3% das receitas anuais das operadoras.

As velocidades de transmissão da banda larga móvel são inferiores às disponibilizadas pela banda larga fixa. Apesar das velocidades inferiores da banda larga móvel, a Autoridade Nacional de Comunicações (*ANACOM*) considera que esta se está a tornar numa alternativa aos serviços fixos.

A banda larga móvel atingiu uma penetração de 13 acessos por cada 100 habitantes, dos quais 2.8% dos acessos são associados a dispositivos do tipo PC (ANACOM 2009).

O crescimento das redes móveis tem impulsionado o crescimento dos meios físicos utilizados pelos consumidores de serviços de comunicação electrónica. Desde 2000 os acessos móveis tiveram um crescimento de 6,6% ao ano, enquanto os acessos à rede fixa tradicional têm vindo a diminuir 4% ao ano.

O serviço telefónico fixo tem vindo a diminuir, sendo que em 2008 o tráfego móvel já representava cerca de 66% das comunicações de voz em Portugal.

No final de 2008 a penetração do serviço telefónico móvel em Portugal ascendeu a 240,4 por cada 100 habitantes.

No Gráfico 2 é perceptível a penetração da banda larga móvel activa na população da UE no segundo trimestre de 2008. Este estudo abrange os clientes residenciais que efectivamente utilizaram 3G nos últimos 90 dias por cada 100 habitantes.

Salienta-se que países como a Holanda, Reino Unido e França não são mencionados no gráfico, devido à falta de dados estatísticos a quando da realização do estudo (ANACOM 2009).

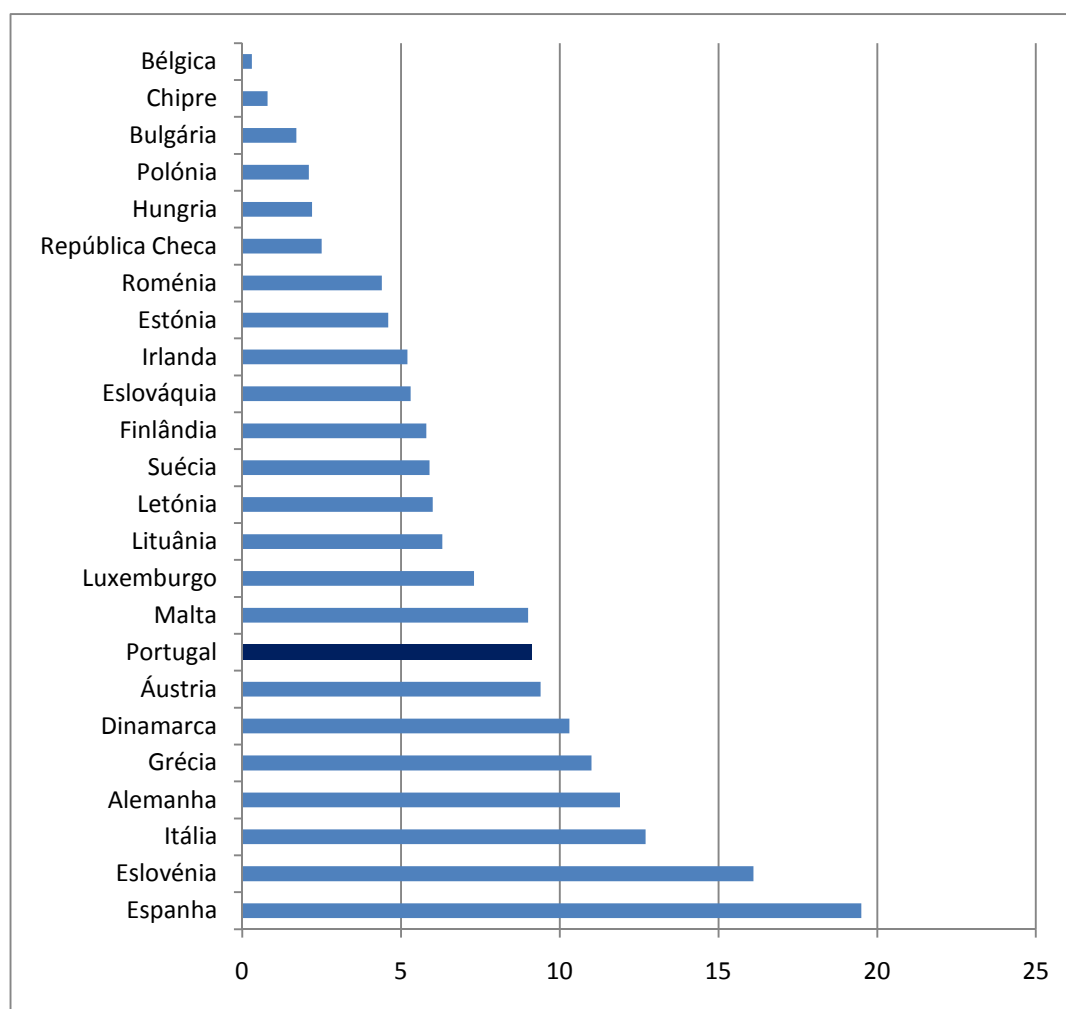


Gráfico 2 Penetração de banda larga móvel activa na população da União Europeia.  
Adaptado de (ANACOM 2009).

Considerando um rácio entre o número de utilizadores de serviços 3G e a população, Portugal encontra-se no nono lugar entre os demais países da UE.

Os novos serviços móveis como as *Multimedia Message Service (MMS)* ou o *Mobile Television (Mobile TV)* apresentam taxas de crescimento elevadas,

tendo o *Mobile TV* em Portugal cerca de 423 mil utilizadores no final do ano de 2008 (ANACOM 2009).

O número de assinantes do *STM* na *UE* também tem vindo aumentar nos últimos anos. Em Portugal, e de acordo com o Instituto Nacional de Estatística (*INE*), 15% dos agregados familiares com acesso à *Internet* possuem banda larga móvel através do telemóvel e 27% através de outro tipo de ligação sem fio (ANACOM 2009).

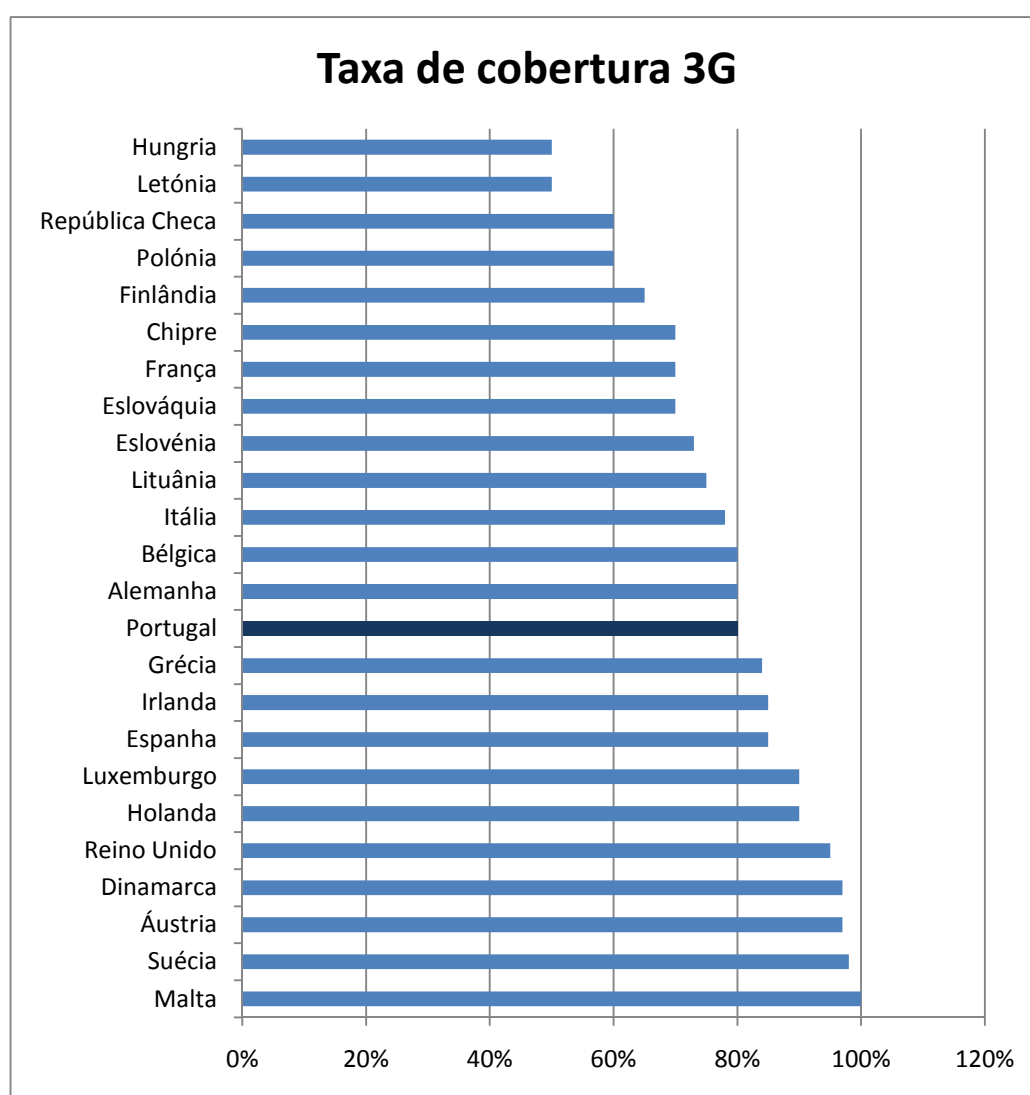


Gráfico 3 Taxa de cobertura 3G na EU.  
Adaptado de (ANACOM 2009).

As elevadas taxas de cobertura da rede 3G na Europa (Gráfico 3), poderá ser um indicador para a breve transição de muitos dos serviços fixos para serviços móveis. Segundo a ANACOM, o aumento da cobertura tem impulsionado o crescimento da computação móvel na Europa (ANACOM 2009).

Pela observação do Gráfico 3 percebe-se que em muitos dos países da *UE*, a taxa de cobertura é quase sempre superior a 50%, atingindo em muitos casos os 100% (ANACOM 2009).

Países como a Estónia, Bulgária e a Roménia, não estão representados no gráfico por não existirem dados estatísticos até à data do estudo.

### **2.2.7 Sistemas operativos móveis**

Um sistema operativo é um programa que permite estabelecer a “ponte” entre o *software* e o *hardware*. É responsável pela gestão da memória, processador, sistemas de ficheiros e periféricos (Gabriele e Ismail 2009).

Actualmente, quando adquirimos um telefone móvel, devemos tal como fazemos nos computadores pessoais, utilizar o sistema operativo que mais se adequa às nossas necessidades, procurando optar sempre por um sistema fiável rápido e seguro.

Usualmente, os sistemas operativos desenvolvidos para dispositivos móveis, à semelhança dos sistemas operativos desenvolvidos para os sistemas fixos, adoptam uma arquitectura de camadas (*Kernel, Middleware, Application Execution Environment, User Interface e Application Suite*).

A separação por camadas torna estes sistemas escaláveis, sendo possível evoluir numa das camadas sem existir necessidade de proceder a qualquer alteração nas restantes.

Seguidamente será efectuada uma abordagem detalhada a cada uma destas camadas apresentadas no diagrama da Figura 3.

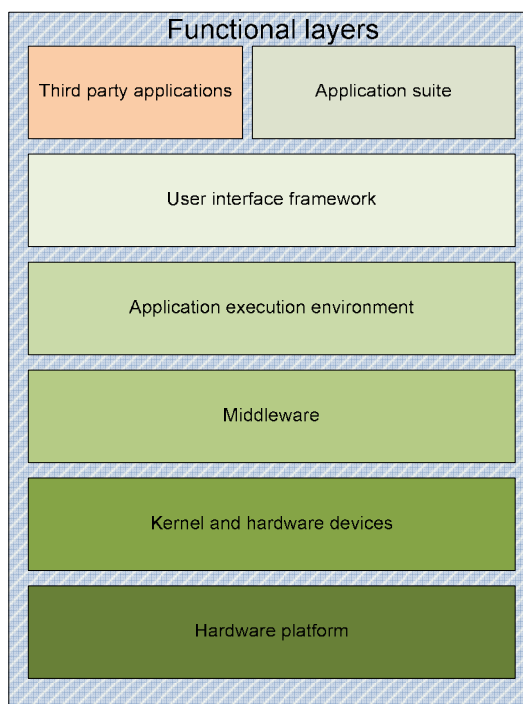


Figura 3 Camadas de um sistema operativo móvel.  
Adaptado de Mobile Operating Systems (Andreas 2006)

### ***Kernel and hardware devices***

É o cerne do sistema operativo. É a camada onde se encontram os *drivers* do *hardware* o sistema de ficheiros e onde é feita a gestão dos processos e memória (Andreas 2006).

### ***MiddleWare***

Camada transparente para o utilizador. É responsável por interligar os diversos tipos de periféricos através de bibliotecas de *software* (Andreas 2006).

### ***Application Execution Environment***

Responsável por disponibilizar as *Application Programming Interfaces (APIs)* de apoio aos programadores no desenvolvimento de novas aplicações.

### ***User Interface (UI)***

Camada onde é disponibilizado o ambiente gráfico característico de cada sistema operativo.

### ***Application suite***

É nesta camada que se encontram a maior parte das aplicações disponíveis no sistema. Entre outros encontram-se os *browsers*, menus de configuração, calendário e jogos.

### ***Third party applications***

Nesta camada podemos encontrar programas ou *plugins*<sup>3</sup> que permitem adicionar funcionalidades extra ao sistema. Estes programas ou *plugins* são usualmente desenvolvidos por entidades diferentes das que desenvolveram o sistema operativo. Nesta camada podemos encontrar os jogos, os *browsers* ou os clientes de *email*.

## **2.2.8 Plataformas**

Actualmente existem inúmeros fabricantes de sistemas operativos para dispositivos móveis. O estudo focará o *Symbian*, *Reasearch in motion (Rim)*, *Windows Mobile* e o *iPhone Operating System (OS)*, que foram os sistemas operativos mais utilizados durante o ano de 2008 (Jaskl 2009). Para além destes, será também estudado o novo sistema operativo proprietário da *Google*, o *Android OS*.

Apesar do elevado número de sistemas existentes no mercado todos eles têm funcionalidades em comum, no entanto divergem bastante na usabilidade, rapidez e fiabilidade.

Todos os sistemas operativos estudados disponibilizam suporte a leitores de música, vídeo, possibilidade de fotografar, agendas ou jogos.

---

<sup>3</sup> *Plugin* – ferramenta que confere novas funcionalidades a um determinado programa instalado num dispositivo.

Segundo o estudo publicado pela revista *PCWorld*, o *Apple iPhone OS* e o *Android OS* são os melhores sistemas operativos presentes no mercado (Harry 2009).

No Gráfico 4 pode-se observar os sistemas operativos mais utilizados durante o ano de 2008.

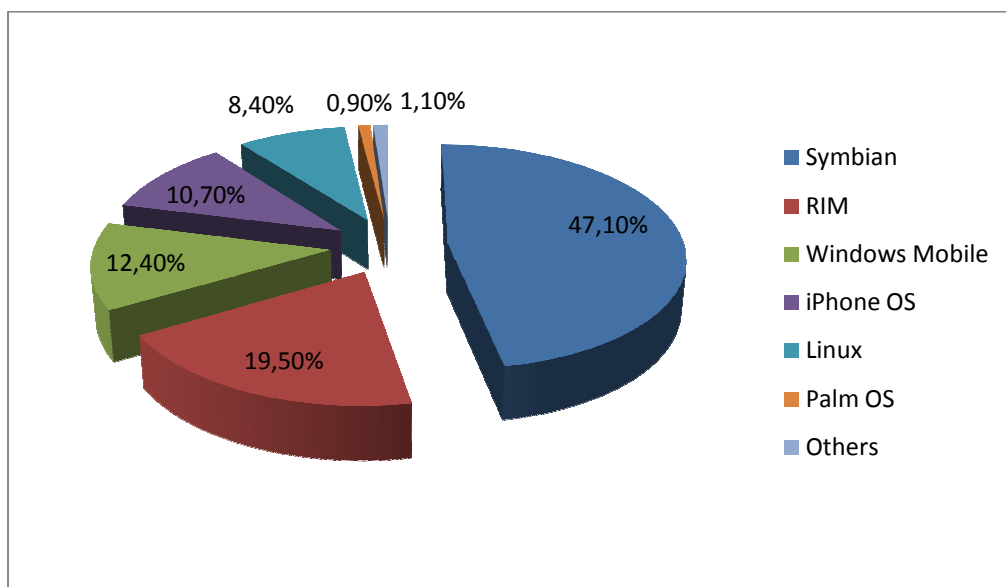


Gráfico 4 Sistemas operativos utilizados nos *smartphones* em 2008.  
Adaptado de *Mobile Operating Systems* (Jaskl 2009).

Seguidamente são apresentadas as características dos sistemas operativos que foram objecto de estudo.

### **Symbian**

A *Symbian Foundation* surgiu em 1998. Suportada por um conjunto de fabricantes com o desígnio de licenciar a plataforma de *software Symbian OS* para dispositivos móveis. Actualmente, os seus maiores accionistas são a *Nokia* (47,9%), *Ericsson* (15,6%), *Sony Ericsson* (13,1%), *Panasonic* (10,5%), *Samsung* (4,5%) e a *Siemens* (8,4%).

O *Symbian OS* é o sistema operativo mais comercializado. Encontra-se instalado em mais de 80 milhões de aparelhos e em mais de 100 modelos diferentes. A *Nokia* é a empresa detentora do maior número de dispositivos com este sistema.

Hoje em dia o *Symbian OS* é um sistema operativo reconhecido que possui um dos *kernels* mais avançados para dispositivos móveis.

Como requisitos mínimos, necessita de um processador *Advanced Risc Machine nine (ARM9)*<sup>4</sup>, sendo os restantes dependentes das *interfaces* adoptadas pelos fabricantes (Andreas 2006).

Este sistema operativo suporta tecnologia 2G e 3G, protocolos de comunicação *WAP*, *Transmission Control Protocol (TCP)*, *IPV4*, *IPV6*. Suporta, *Infra Red Data Association (IrDA)*, *Bluetooth*, *Universal Serial Bus (USB)*, *multi-task* e *multi-thread*.

O *Symbian* adapta-se aos mais diversos tipos de telefones, sejam estes de teclado numérico, alfa numérico ou os actuais de *touch-screen*.

Para além dos habituais serviços telefónicos suporta o envio e recepção de *SMS*, *Enhanced Messaging Service (EMS)*, *Multimédia Message Service (MMS)*, videoconferências sobre *IP*, *VoIP* e a capacidade de alternar entre redes de voz e *Wi-Fi*.

Usualmente vem equipado com aplicações de navegação por satélite, agenda, *email*, fax e um processador de texto simples.

A segurança não foi esquecida. O *Symbian* garante a confidencialidade e integridade da informação através de algoritmos de compressão e cifra. Estes algoritmos permitem utilizar diferentes tipos de cifras e certificados digitais consoante as diferentes necessidades aplicacionais (Gabriele e Ismail 2009).

### **Windows Mobile**

O *Windows Mobile* é uma variante do *Windows Compact Edition (Windows CE)*. Inicialmente foi desenvolvido para *Pocket PC* e surge em 2002 nos primeiros telefones da *High Tech Computer Corporation (HTC)*<sup>5</sup>, tendo a arquitectura sido desenvolvida para disponibilizar serviços de dados e multimédia.

Em meados de 2006, o código fonte do *Windows Mobile* foi disponibilizado aos programadores. Com esta abertura surgem inúmeras aplicações e serviços que permitiram ao *Windows Mobile* alcançar rapidamente o patamar dos sistemas operativos móveis mais utilizados (Andreas 2006).

---

<sup>4</sup> *ARM* – são processadores de 32 *bits*, versáteis, usualmente encontrados em *PDA*s, smartphones e calculadoras. Estes processadores possuem poucas instruções, têm dimensões reduzidas e consomem poucos recursos.

<sup>5</sup> *HTC* - fabricante de smartphones de Taiwan. Usualmente utiliza nos seus dispositivos móveis os sistemas operativos *Microsoft Windows Mobile* e *Google Android*.

O *Windows Mobile* disponibiliza três *APIs* de suporte ao desenvolvimento aplicativo. A *Win32-API* integra uma *interface* nativa que permite o desenvolvimento de aplicações com recurso à linguagem *C*. A *MFC-API* que é uma extensão da anterior, mas que possibilita o desenvolvimento de aplicações em *C* e *C++* suportando Programação Orientada a Objectos (*POO*). Disponibiliza ainda a *.NET Compact Framework*, que é uma versão compacta da *Framework* utilizada para o desenvolvimento de aplicações para *Desktop*.

Uma das estratégias da *Microsoft* é continuar a disponibilizar este tipo de *API*, estimulando o desenvolvimento de mais e melhores aplicações.

O *Windows Mobile* suporta *Bluetooth* através da interface *Winsock*, redes *902.11x*, *IPV4*, *IPV6*, *VoIP*, *GSM* e *CDMA*.

Ao nível aplicativo vem dotado do *Pocket Outlook*, versão adaptada do pacote do *Microsoft Office*, incluindo o *Outlook*, *Word* e *Excel*. Possui suporte para *Messenger*, *Browser* e *Remote Desktop* que possibilita de forma cómoda o acesso remoto a outros terminais fixos ou móveis.

Para facilitar a troca de informação entre dispositivos móveis e dispositivos *Desktop*, o *Windows Mobile* tem o *ActiveSync*, que é uma aplicação de sincronização de dados utilizada para sincronizar informação entre um dispositivo móvel com versões do *Windows Mobile* ou *CE* e um computador pessoal (Pedro 2003).

A vertente multimédia não foi esquecida, e o *Windows Mobile* suporta a reprodução de diversos formatos de música, vídeos e aplicações *3D*.

Ao nível da segurança, é dado suporte ao *Secure Socket Layer (SSL)*, *Kerberos*<sup>6</sup> e a possibilidade de utilizar diversos tipos de algoritmos de cifra de dados.

Uma das desvantagens deste sistema é que a sua *interface* está bastante dependente da utilização de um pequeno estilete que o torna pouco prático. Sistemas como o *iPhone OS*, *Google Android* ou mesmo o *RIM*, tem aproveitado esta desvantagem, visto os seus sistemas não necessitarem do estilete para funcionarem, tornando-se mais cómodos.

---

<sup>6</sup> *Kerberos* - protocolo que garante a integridade dos dados, permitindo que cliente e servidor se autenticuem mutuamente.

### ***Research in Motion (RIM)***

A *Research In Motion*® (*RIM*) é um fabricante Canadano que comercializa soluções para o mercado das comunicações móveis. Os seus principais produtos são os smartphones *BlackBerry*™. Os conceitos subjacentes ao *BlackBerry* têm vindo a mudar bastante na última década. Adoptam um rumo muito característico, incluindo nos seus dispositivos uma *trackball* de navegação que permite o acesso simplificado aos menus.

Os serviços de agenda e *email* disponibilizados pelo *RIM*, são bastante famosos no mundo empresarial. Estes são bastante eficientes e fáceis de utilizar tornando-se numa mais-valia no mercado empresarial que cada vez mais utiliza o *email* como meio de comunicação.

O acesso a páginas de *Internet* tem vindo a ser melhorado, fornecendo agora um serviço mais rápido e eficiente.

O *RIM* foi desenhado para um processador *Intel 32 bits i386* com *512 Kilo bytes (Kb)* de *Static Random Access Memory (SRAM)* e dependendo do modelo quatro ou cinco *Mega Bytes (MB)* de *Flash RAM* (Burnette 2002).

Uma das características que o distingue dos demais sistemas operativos é ser constituído apenas por um ficheiro (Burnette 2002).

Para apoiar o desenvolvimento das aplicações a *BlackBerry*, disponibiliza o “*BlackBerry Software Development Kit*”, constituído por diversas ferramentas de análise e desenvolvimento para o *RIM* (Burnette 2002).

### ***iPhone Operating System (iPhone OS)***

Sistema operativo proprietário da *Apple* que mune o *iPhone* e *iPod Touch*. É usualmente visto como a versão compacta do *Mac OS* para computadores pessoais e baseado na versão 10.5 do *Leopard*, que é uma das distribuições do sistema operativo da *Apple* para computadores pessoais.

A simplicidade e robustez que proporcionam na navegação entre menus e aplicações são duas das suas grandes potencialidades.

Alguns meses depois de a *Apple* disponibilizar o *Software Development Kit (SDK)* aos programadores, surgem inúmeras aplicações gratuitas ou de baixo

valor na *Apple Store*. Estas aplicações caracterizam-se pelo reduzido custo e pela simplicidade de aquisição e instalação.

Uma desvantagem deste *SDK* é ser restrito a utilizadores de sistemas operativos proprietários da *Apple*. Assim não é possível a programadores que utilizam sistemas *Microsoft* desenvolverem aplicações sobre este *SDK*.

Nativamente, o *iPhone OS* vem equipado com *software* diversificado, podendo o utilizador encontrar jogos *3D*, reprodutores de vídeo, música, ferramentas de processamento de texto e edição de imagem.

### **Google Android**

O *Android* é o novíssimo sistema operativo *open source* lançado pelo gigante Norte-Americano *Google*. É intuitivo, de fácil navegação e graficamente bastante semelhante ao *iPhone* e *BlackBerry*.

O facto de ser *open source*, poderá vir a ser sinónimo de aplicações mais baratas para o consumidor final.

O *kernel*, baseado no *Linux 2.6*, oferece suporte a redes *2G*, *3G*, ligações *Wi-Fi*, *IPV4* e *IPV6*.

Ao nível das aplicações multimédia oferece suporte *OpenGL*<sup>7</sup>, permitindo a execução de diversos tipos de formato de imagem, áudio e vídeo.

Como sistema de persistência de dados suporta o *SQLite*<sup>8</sup>.

No capítulo da segurança suporta *SSL* e uma grande variedade de algoritmos de cifra.

### **2.2.9 Análise comparativa dos sistemas operativos**

As principais tendências do mercado de *software* entre 2006 e 2009 focaram-se sobretudo na flexibilidade do *software*, na forte aposta no *MiddleWare*<sup>9</sup> e no aparecimento de sistemas *open source*.

---

<sup>7</sup> *OpenGL* - *API open source* usualmente utilizada em computação gráfica. Esta *API* suporta *2D* e *3D*, sendo bastante utilizada no desenvolvimento de aplicações multimédia.

<sup>8</sup> *SQLite* - biblioteca *open source* da linguagem C que implementa uma base de dados transaccional sem que esteja dependente de um servidor.

<sup>9</sup> *MiddleWare* - aplicação responsável por interligar duas ou mais aplicações independentemente da plataforma utilizada.

À semelhança dos sistemas operativos (*SO*) desenvolvidos para os computadores pessoais, também os *SO* dos dispositivos móveis são compatíveis com diversos tipos de dispositivos. Esta compatibilidade permite a personalização de *interfaces*, serviços de rede e multimédia consoante o dispositivo, conferindo-lhes assim um elevado potencial (Andreas 2006).

A capacidade de reprodução multimédia e a ligação à *Internet* recorrendo a redes sem fios, são características comuns nestes sistemas. Apesar de as características serem comuns, podem variar bastante ao nível do desempenho, segurança e número de utilizadores.

O *open source* surgiu nas plataformas móveis com o *Symbian* e o *Google Android*. O *open source* possibilita que os programadores facilmente adicionem mais-valias ao produto, contribuindo para a deflação dos preços dos equipamentos.

O *iPhone* tem uma plataforma parcialmente aberta aos programadores, no entanto tem como requisito ao desenvolvimento, um sistema operativo proprietário *Apple*. Para além deste requisito a *Apple* selecciona quais os produtos a serem publicados na *Apple Store*, não permitindo a existência de *software* concorrente a um produto original *Apple*.

É difícil prever qual dos sistemas operativos irá dominar o mercado. No entanto o *iPhone OS*, *Android OS* e o *Symbian* são excelentes candidatos. Têm uma boa aceitação por parte dos fabricantes e consumidores e acima de tudo possuem uma boa quota de mercado.

A *Apple*, ao disponibilizar aplicações como o *iTunes* ou o *Safari*, que correntemente são utilizados nos computadores pessoais, fomenta a rápida adaptação dos utilizadores ao seu produto. Esta redundância de aplicações evita a necessidade do utilizador se adaptar a novas *interfaces*.

Segundo informação disponibilizada no site do *Google*, o *Android*, pode ser executado em *Smart PCs*. Esta compatibilidade é uma mais-valia para os utilizadores que passam a poder usufruir do mesmo sistema operativo em ambientes distintos.

Toda a evolução que se tem verificado ao longo destes últimos anos ao nível da computação móvel e a disponibilização dos *SDK* por parte dos fabricantes de sistemas, impulsionou o crescimento exponencial das aplicações móveis (Pedro 2003).

Actualmente os *SDK* permitem a utilização de linguagens de programação orientadas a objectos. O *RIM* e *Android* com o *Java*, o *C*, *C++* e *C#* para o *Windows Mobile* e o *Object C* para o *iPhone* (Adolph 2009).

O futuro dos sistemas operativos móveis está bastante dependente do que é feito do lado dos servidores. Se passarmos parte do processamento dos dispositivos móveis para os servidores, conseguiremos otimizar os limitados recursos destes dispositivos (Timsater 2009).

### **2.2.10 Sistemas móveis versus sistemas tradicionais**

Actualmente os sistemas operativos móveis e os sistemas para computadores pessoais possuem características comuns. Apesar de grande parte destes aceitarem memórias *flash*<sup>10</sup>, encontram-se limitados ao nível da autonomia e dos recursos de entrada e saída.

Estas limitações são bastante perceptíveis nas aplicações desenvolvidos para sistemas móveis. Quase sempre, são apresentadas com limitações quando comparadas com as mesmas aplicações para computadores pessoais.

Ao comparar aplicações como o *Microsoft Office*, verifica-se que funcionalidades como tabelas, estilos, cabeçalhos e rodapés não se encontram disponíveis na *suite* para sistemas móveis.

Outra diferença significativa depara-se com as limitações das placas gráficas, som e de rede. Apesar de bastante evoluídas, não conseguem disponibilizar recursos suficientes para que possamos jogar em alta definição, em *streaming* ou com a qualidade sonora proporcionada por um computador pessoal e seus periféricos.

### **2.2.11 Desafios inerentes à mobilidade dos sistemas**

Apesar das vantagens provenientes da utilização dos sistemas móveis, existem alguns desafios a serem ultrapassados. Seguidamente são apresentados alguns dos desafios associados a este tipo de dispositivos.

---

<sup>10</sup> *Flash* - memória não volátil. Consome menos energia e é mais resistente ao choque do que os habituais discos rígidos *HDD*.

### Adaptação

A adaptação aos dispositivos móveis pode ser avaliada pela complexidade, rapidez e eficácia de um utilizador na introdução de textos. Devido ao seu reduzido tamanho, a adaptação na introdução de textos é a mais difícil de conseguir (Soukoreff 2002).

Tendo em conta o rápido crescimento do *software* para dispositivos móveis, a adaptação dos utilizadores ao *software* pode ser um dos factores preponderante para o seu sucesso. Os *softwares* que se destaquem nesta área ganham reconhecimento por parte dos utilizadores (Ribeiro 2006).

Uma das principais desvantagens destes dispositivos prende-se com a limitação de recursos de *hardware* e *software* que limita as funcionalidades do dispositivo (Ribeiro 2006).

### Desempenho

Tradicionalmente os dispositivos móveis têm desempenhos inferiores aos apresentados pelos computadores pessoais. O desempenho dos dispositivos móveis está directamente relacionado com as suas reduzidas dimensões.

Sendo o desempenho um aspecto crucial para a evolução da computação móvel, existem ainda muitas limitações de *hardware* que condicionam a perfeição.

Na Figura 4, é possível observar a relação existente entre o tamanho dos diversos tipos de dispositivos móveis e o seu desempenho.

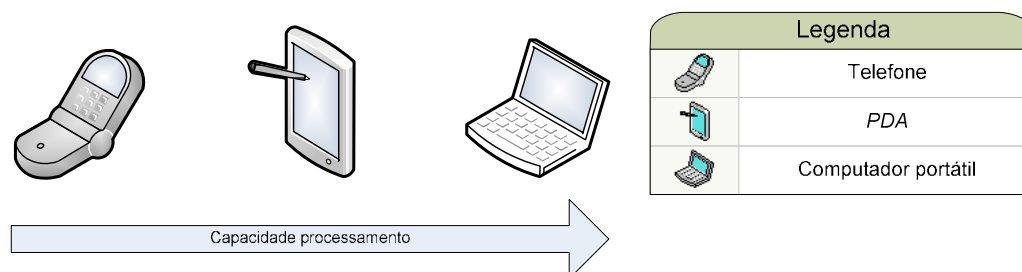


Figura 4 Desempenho do dispositivo *versus* tamanho.

### **Autonomia**

Os dispositivos móveis dependem directamente de baterias. Sem estas seria impossível usufruir dos dispositivos em locais onde não existisse rede eléctrica. As baterias são usualmente pesadas e disponibilizam autonomia para apenas algumas horas.

A autonomia tem sido um grande obstáculo da computação móvel. A produção de baterias não tem conseguido acompanhar o restante desenvolvimento tecnológico ao redor da computação móvel.

Com isto, tem quase sempre ficado do lado dos fabricantes de *hardware* a responsabilidade de desenvolverem componentes e aplicações que optimizem o consumo dos recursos energéticos (Mateus e Loureiro 2004).

Sendo a autonomia dos dispositivos uma das principais características associadas à mobilidade, é importante que a tecnologia não estagne e que esta seja tida em conta pelas empresas responsáveis pelo desenvolvimento de *hardware* e *software*.

### **Qualidade de serviço**

As inúmeras diferenças estruturais existentes nos sistemas móveis e as variações da qualidade de sinal torna-os instáveis em zonas com infra-estruturas comunicacionais fracas.

Para desfrutar das plenas capacidades aplicacionais destes sistemas é conveniente que se esteja numa zona de boa qualidade de sinal (Mateus e Loureiro 2004).

### **Confiabilidade**

Uma das grandes dificuldades que tende a persistir, e que deve ser ultrapassada é a forma como os dados são tratados. Este tratamento é bastante importante, sendo que uma análise ou tratamento incorrecto da informação pode causar perda de informação para o utilizador final.

Um dos grandes desafios surge quando é necessário permutar o dispositivo móvel entre o estado ligado e desligado. Quando isto sucede, é fundamental

perceber quais as actualizações fundamentais para se conseguir manter a persistência dos dados (Mateus e Loureiro 2004). Este processo é por vezes difícil de conseguir.

O estudo apresentado permite compreender ainda que de forma breve o processo evolutivo da computação móvel. Foram identificados os sistemas operativos de relevância mais significativa no mercado e enumeradas as vantagens e desafios da computação móvel.

O capítulo seguinte é responsável pela identificação e apresentação do problema em estudo. O problema centra-se na temática da fraca autonomia e capacidade de processamento. Será posteriormente, apresentado um modelo que permite melhorar o desempenho deste tipo de dispositivos.

## 3 Metodologia

---

A computação móvel provém da necessidade de usufruir das actuais funcionalidades dos computadores pessoais em equipamentos e redes móveis. Com a aceitação dos dispositivos e das redes sem fio prevê-se que este tipo de computação continue com o seu processo evolutivo.

Dos problemas inerentes à mobilidade dos sistemas destacam-se a adaptação, a autonomia, a qualidade de serviço, o desempenho e a confiabilidade.

Pretende-se com este capítulo apresentar o problema, o processo de investigação adoptado e as características necessárias à implementação do modelo a apresentar. É também efectuada uma abordagem às ferramentas utilizadas para o desenvolvimento do protótipo e ao dispositivo móvel utilizado durante na fase de testes.

### 3.1 Problema

A computação móvel tem sido alvo de significativas evoluções. Conhecem-se desafios e limites de investigação neste tipo de computação ao nível da interacção homem-máquina, autonomia, robustez, custos, capacidade de armazenamento, capacidade de processamento, entre outras.

Como anteriormente mencionado, os fabricantes de *hardware* são actualmente os grandes responsáveis pela evolução da computação móvel. Os fabricantes de *hardware* encontram-se empenhados em tornar os dispositivos móveis mais robustos, autónomos e eficientes. As empresas de *software*, focam-se em disponibilizar aos utilizadores uma vasta gama de aplicações e nem sempre têm em consideração os recursos que estes consomem.

A mobilidade dos dispositivos móveis está dependente da utilização de baterias. Preservar a carga da bateria é importante já que sem ela perde-se mobilidade.

É igualmente importante ter presente que quanto mais processos ou tarefas estão em execução no dispositivo, maiores os consumos de energia.

O desafio abordado neste projecto é o desenvolvimento de um modelo que implemente uma arquitectura que permita minimizar o número de processos e tarefas a serem executados pelo dispositivo, tentando assim minimizar o consumo energético do mesmo.

Para otimizar o consumo dos recursos energéticos do dispositivo móvel é apresentada uma arquitectura baseada em eventos. Com esta pretende-se colocar nos fabricantes de *software*, a responsabilidade de contribuir ainda mais para a evolução da computação móvel.

Com optimizações ao nível da autonomia, processamento e transferência de dados na rede conseguem-se dispositivos mais autónomos, robustos e interessantes.

### **3.2 Processo de investigação**

O processo de investigação iniciou-se com uma abordagem técnico documental com a consulta de bibliografia de diversas fontes, entre elas sítios na *Internet*, revistas, artigos e livros da especialidade. Esta abordagem contribuiu para perceber o estado actual e as tendências da computação móvel.

Após a abordagem inicial, efectuou-se o levantamento do estado da arte. Estudou-se a evolução histórica da computação móvel, as tecnologias imergentes e validou-se a importância do tema no actual contexto tecnológico. O levantamento do estado da arte contribuiu para fortalecer a base de conhecimentos na área.

O levantamento das características fundamentais para atingir a solução do problema, o desenho da arquitectura e o desenvolvimento do protocolo de comunicação, foram os passos que se seguiram. Por fim, foi implementado e testado o protótipo com o objectivo de validar e avaliar a arquitectura.

### 3.3 Características do modelo

O sistema de informação tem como objectivo, entre outros, o de tornar as tarefas mais eficientes.

Com o objectivo de contribuir para um aumento do conhecimento na área, seria interessante desenvolver um modelo com as seguintes características:

- Capacidade de comunicar com sistemas distintos;
- Transferir parte do processamento dos dispositivos móveis para outros dispositivos;
- Aplicação que efectue o processamento de apoio ao dispositivo móvel, notificando-o sempre que necessário;
- Desenvolver uma aplicação que permite interpretar as notificações da aplicação de apoio ao processamento;
- Servidor *Web* que disponibilize serviços.

#### 3.3.1 Equipamento

Existem inúmeros tipos de *smartphones* no mercado. No entanto, para o desenvolvimento e teste do protótipo foi utilizado um *HTC* com o sistema operativo *Microsoft Windows Mobile*.

A utilização deste *smartphone* deve-se ao facto de ter sido o único modelo que se encontrava disponível para a elaboração do projecto. No entanto é um *smartphone* recente de uma das maiores empresas no sector da computação móvel.

O *HTC* está equipado com as mais recentes tecnologias do mercado móvel. Possui *Bluetooth*, *Wi-Fi*, sistema operativo recente e ecrã táctil. Para além disto, é um dos mais utilizados na gama profissional de *smartphones*.

#### 3.3.2 Plataformas de apoio ao desenvolvimento

Actualmente, existe uma grande diversidade de plataformas de desenvolvimento para *smartphones*. A utilização do *Windows Mobile*, sistema operativo da *Microsoft*, condiciona de certo modo a escolha de uma ferramenta que faça *deployment* para este sistema operativo.

Optou-se por utilizar maioritariamente ferramentas proprietárias do gigante Americano, exceptuando a ferramenta *RAD*<sup>11</sup> *Kalipso* da *SysDev*. A *SysDev* é uma empresa Portuguesa, especializada no desenvolvimento de plataformas de apoio ao programador para aplicações móveis com sistemas operativos *Microsoft*.

Com a utilização de ferramentas *RAD* consegue-se otimizar tempo de desenvolvimento, traduzindo-se em menores custos de produção e manutenção (Piske 2006).

Para o desenvolvimento do protótipo foram utilizadas duas ferramentas *RAD* distintas o *Microsoft Visual Studio 2008* e o *Kalipso*.

### ***Microsoft Visual Studio 2008***

Ferramenta completa que permite ao programador desenvolver aplicações para diversos tipos de plataformas e dispositivos. Como linguagem de programação o programador pode optar pelo *C#*, *J#*, *Vb*, ou por outra compatível e do seu domínio.

Disponibiliza boa documentação, permitindo ao programador ter acesso a exemplos e descritivos das classes e métodos existentes. Possibilita ainda o desenvolvimento de diversificados tipos de aplicações para as mais distintas plataformas. É possível utilizando apenas uma única ferramenta desenvolver aplicações para *Tablet PCs*, *Pocket Pcs*, *SmartPhones* e *Desktops*. (Corporation 2007).

### ***Kalipso***

O *Kalipso* é uma ferramenta *RAD*, concebida para facilitar o desenvolvimento de aplicações para dispositivos móveis baseados em plataformas *Microsoft*.

Com os objectos do *Kalipso* é possível criar formulários, botões, listas, validar código, sincronizar dados de forma cómoda e rápida.

---

<sup>11</sup> *RAD* – *Rapid Application Development*, metodologia de desenvolvimento que surgiu na década de 70, oferecendo um ganho de produtividade para o programador no desenvolvimento de aplicações Piske, O. (2006). *Rapid Application Development*. F. Seidel. Piske, O. (2006). *Rapid Application Development*. F. Seidel. Piske, O. (2006). *Rapid Application Development*. F. Seidel. Piske, O. (2006). *Rapid Application Development*. F. Seidel. Piske, O. (2006). *Rapid Application Development*. F. Seidel..

O reconhecimento de voz é outro ponto forte da ferramenta que possibilita uma boa interacção homem-máquina.

### **Motor de base de dados**

Para armazenar as configurações do gestor aplicacional recorreu-se a um motor de base de dados da *Microsoft*.

### **Microsoft SQL Server 2008**

O *Microsoft Sql Server 2008*, possibilita integração directa com a ferramenta *Microsoft Visual Studio 2008*. Esta integração é uma mais-valia para o desenvolvimento, tornando mais fácil e eficiente a manipulação da base de dados e das aplicações.

### **Microsoft Windows Mobile SDK**

O *Microsoft Visual Studio* possui um emulador para dispositivos móveis. Com este é possível, desenvolver e testar aplicações móveis sem existir a necessidade de investir num dispositivo físico (Microsoft 2007).

A utilização do emulador é bastante útil na fase de desenvolvimento, permitindo que seja efectuado *debug*<sup>12</sup> no dispositivo emulado. Conjuntamente com o *Windows Mobile SDK*, a *Microsoft* disponibiliza diversas aplicações que permitem simular ambientes de testes distintos. O *CellEmu* permite simular ambientes de chamadas telefónicas e *SMS* e o *FakeGPS* permite efectuar testes com recurso a um *GPS* virtual (Microsoft 2007).

## **3.4 Desenvolvimento**

Nesta etapa foram desenvolvidas as aplicações que permitem validar e avaliar o modelo proposto. Foram desenvolvidas aplicações implementando o modelo proposto e o modelo actualmente existente nas tradicionais aplicações móveis.

---

<sup>12</sup> *Debug* – Programa ou componente de um programa que permite encontrar e minimizar os erros de programação por parte dos programadores.

## Metodologia

---

Estando limitados apenas a um dispositivo de testes, todas as aplicações foram desenvolvidas para a plataforma *Windows Mobile*.

Este capítulo apresentou o problema da autonomia nos dispositivos móveis como ponto central da tese. O *HTC Touch* é o dispositivo utilizado nos testes e as ferramentas da *Microsoft* e *SysDev* são as adoptadas para o desenvolvimento.

No capítulo seguinte, será apresentado o modelo de notificação por eventos proposto. Para uma fácil compreensão do modelo, é apresentado a título de exemplo um caso prático deste modelo aplicado à consulta de *emails*.

## 4 Modelo de notificação por eventos

---

Com a apresentação do problema, das ferramentas de desenvolvimento e do dispositivo adoptado para os testes no capítulo anterior, será seguidamente apresentado o modelo proposto que pretende otimizar o desempenho do dispositivo. O modelo é subdividido em três secções distintas. A primeira secção é dedicada aos serviços, a segunda ao gestor aplicacional e a terceira ao dispositivo móvel. Também neste capítulo será apresentado o protocolo comunicacional adoptado e um cenário exemplo para o modelo em estudo.

### 4.1 Apresentação do modelo de notificação por eventos

Com o intuito de minimizar o processamento efectuado pelos dispositivos móveis, parte deste é transferido para um ou mais servidores. Apesar dos dispositivos móveis terem sempre de efectuar processamento, podemos migrar tarefas que não estejam directamente dependentes do dispositivo. Nas tarefas migráveis incluem-se, as consultas de *email*, as cotações da bolsa ou a temperatura actual numa determinada cidade.

O gestor aplicacional deve estar instalado no servidor. É responsável por consultar periodicamente os serviços disponibilizados pelos *WebServices*<sup>13</sup>.

Com a descentralização do processamento é fundamental construir um sistema de comunicação eficaz que permita a troca de informação entre o gestor aplicacional

---

<sup>13</sup> *WebService* – serviço *Web*, baseado em *XML*, *SOAP* implementando arquitecturas *SOA*. Permite que aplicações distintas possam interagir entre si, sem que exista a necessidade de intervenção humana. Pode ser implementado em diversos tipos de arquitectura, sendo facilmente escaláveis.

## Modelo de notificação por eventos

e o dispositivo móvel. O protocolo de comunicação é baseado em texto e permite que as diferentes aplicações e dispositivos comuniquem eficientemente.

Na Figura 5 é apresentado o modelo da estrutura em estudo e as suas principais secções: serviços, gestor aplicacional e dispositivo móvel.

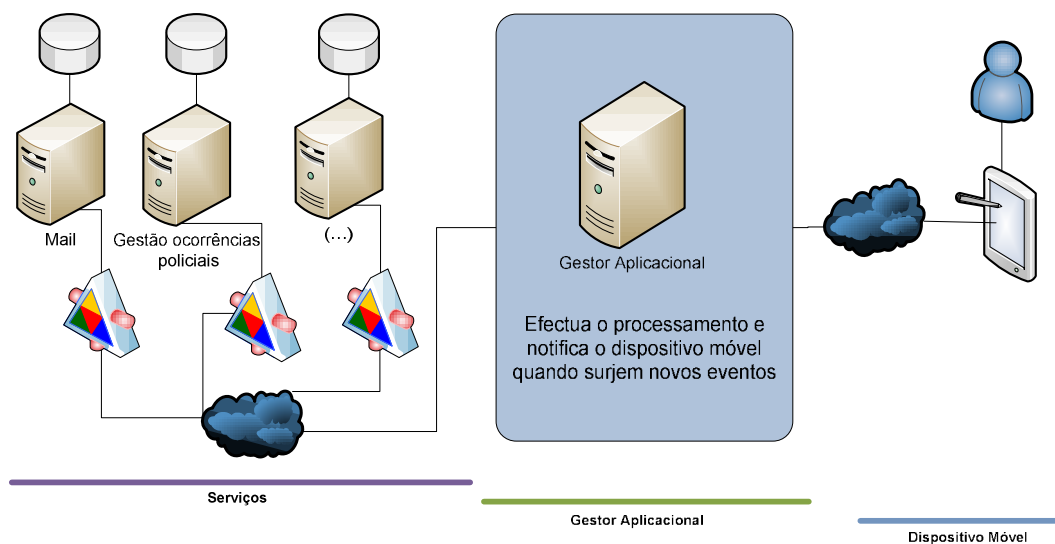


Figura 5 Modelo de gestão de aplicações móveis assistido por eventos.

Estas secções são, a seguir, apresentadas em mais detalhe.

## 4.2 Protocolo

Para que este modelo funcione é fundamental que exista um protocolo bem definido. O protocolo permite a comunicação entre o gestor aplicacional e o dispositivo móvel.

Este protocolo é textual e utiliza como carácter separador de comandos o cardinal “#”.

Seguidamente será apresentado nas Tabela 3, Tabela 4, Tabela 5 e Tabela 6 os comandos e sub-comandos que definem o protocolo.

## Modelo de notificação por eventos

Comando	Sub-Comando	Descrição
<b>conn</b>		Associado ao início, término e erros de conexão
	<b>start</b>	Estabelecimento de ligação
	<b>username</b>	Nome de utilizador
	<b>pwd</b>	Palavra chave do utilizador
conn#start#username#pwd		
conn#start#usera#123		
	<b>ok</b>	Cliente é informado que a conexão foi estabelecida com sucesso.
conn#ok		
	<b>close</b>	Término de ligação
conn#close		
	<b>err</b>	Mensagem de erro a ser enviada ao utilizador
conn#err		
conn#Conta bloqueada, por favor contacte o administrador!		

Tabela 3 Protocolo para o comando “conn”.

Comando	Sub-Comando	Descrição
<b>runApp</b>		Executar aplicação
	<b>desc</b>	Descritivo da aplicação
	<b>nomeApp</b>	Nome da aplicação a ser executada
runApp#desc#nomeApp		
runApp#Aplicação de <i>Email</i> #mail		

Tabela 4 Protocolo para o comando “runApp”.

Comando	Sub-Comando	Descrição
<b>runPopUp</b>		Exibir pop-up
	<b>desc</b>	Descritivo a ser apresentado ao utilizador
runPopUp#desc#nomeApp		
runApp#Chegou um novo <i>email</i> à sua caixa postal!# <i>emailClient</i>		

Tabela 5 Protocolo para o comando “runPopUp”.

Comando	Sub-Comando	Descrição
generic		Exibir pop-up
	desc	Descritivo a ser apresentado ao utilizador
generic#desc#nomeApp		
generic#Chegou um novo <i>email</i> à sua caixa postal!# <i>emailClient</i>		

Tabela 6 Protocolo para o comando “*generic*”.

### 4.3 Modelo

Para o sucesso deste modelo, é fundamental que as secções apresentadas no esquema da Figura 5 estejam correctamente definidas. Nos pontos seguintes são apresentadas cada uma destas secções.

#### 4.3.1 Serviços

Nesta secção, é fundamental existir um ou mais *WebServices*, que disponibilizem serviços ao gestor aplicacional. Na Figura 6 é apresentado um *WebService* que disponibiliza serviços para consulta de *email*.

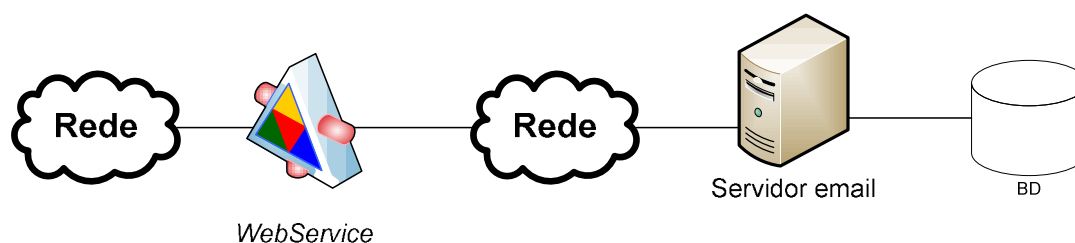


Figura 6 Arquitectura para os serviços disponibilizados pelo servidor de *email*.

#### 4.3.2 Gestor aplicacional

O gestor aplicacional apresentado na Figura 7 é responsável por efectuar processamento de apoio ao dispositivo móvel. Implementa uma arquitectura cliente/servidor, baseada na comunicação por *sockets*.

Para uma correcta comunicação entre a aplicação cliente e aplicação gestora é utilizado o protocolo apresentado na secção protocolo.

## Modelo de notificação por eventos

Com esta arquitectura, a aplicação gestora de eventos, apenas necessita de conhecer o endereço do *WebService*, que tem de consultar periodicamente, a fim de verificar a ocorrência de novos eventos.



Figura 7 Arquitectura para o gestor aplicacional.

Para que esta arquitectura seja viável é obrigatório que os dispositivos móveis estejam registados no gestor aplicacional. Como requisitos para o registo, é necessário um nome de utilizador e uma palavra-chave.

Após o registo, o utilizador pode configurar os serviços disponibilizados pelo gestor através de uma interface *Web*.

Configurações disponíveis	Descrição
Executar aplicação assim que receba notificações	Esta configuração permite que uma aplicação seja executada quando recebe determinado evento.
Notificar utilizador através pop-up	Com esta configuração, a existência de eventos é notificada através de <i>pop-up</i> .
Servidores destino	Endereços <i>IP</i> dos servidores que o cliente pretende obter informações.
Porto	Porto, nome do utilizador e palavra chave que habitualmente o utilizador usa para se autenticar perante o serviço.
Nome do utilizador	
Palavra chave	

Tabela 7 Tipo de configurações do gestor aplicacional.

### 4.3.3 Dispositivo móvel

Nesta secção encontra-se a aplicação cliente, ilustrada na Figura 8 que se liga por via de *socket* ao gestor aplicacional.

## Modelo de notificação por eventos

---

Para o correcto estabelecimento da ligação com o gestor aplicacional, o utilizador deve identificar as suas credenciais de acesso.

Após o estabelecimento da ligação, a aplicação cliente permanece em plano de fundo, aguardando notificações vindas do gestor aplicacional.

A aplicação cliente é configurável, adaptando-se às necessidades e ao perfil de cada utilizador.

Quando o servidor aplicacional notifica o cliente da chegada de um novo evento a aplicação cliente responde segundo as configurações pré-definidas pelo utilizador.

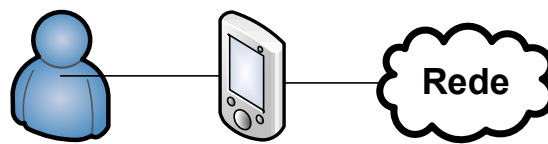


Figura 8 Arquitectura da aplicação cliente.

Para complementar o estudo do modelo apresentado anteriormente, é exposto de seguida um cenário exemplo de utilização efectiva. Na Figura 9 está esquematizado o modelo da arquitectura a ser utilizada. Na Figura 10 podemos observar o fluxo de informação e processamento que ocorre durante as diversas etapas do modelo.

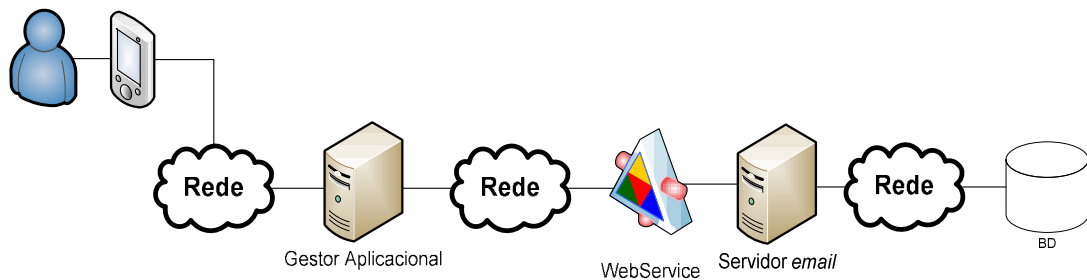


Figura 9 Exemplo da arquitectura proposta para a consulta de *email*.

O cenário, baseia-se na notificação do dispositivo móvel aquando da chegada de novas mensagens de *email* à conta de correio do cliente. Este exemplo foca-se apenas na consulta de *emails*.

Esta arquitectura é válida para qualquer tipo de aplicação que esteja dependente de informação proveniente de servidores externos ao dispositivo.

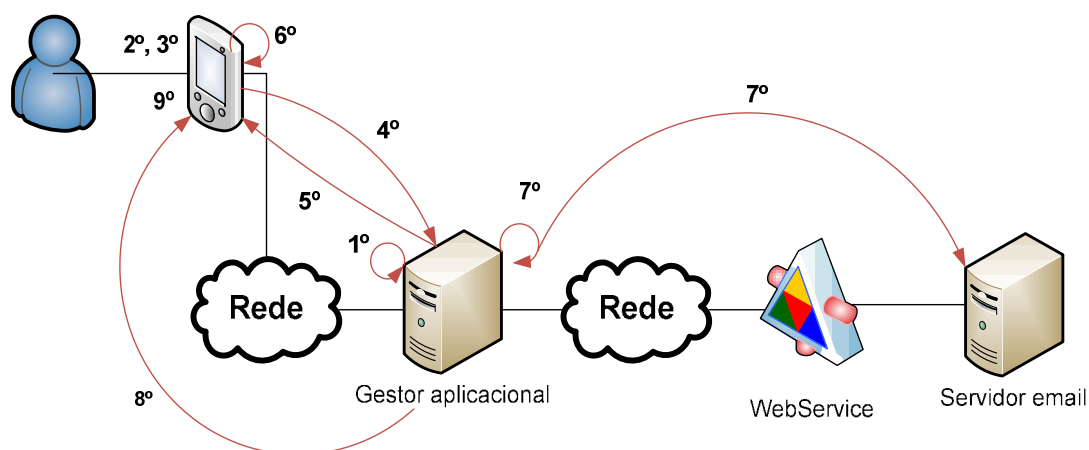


Figura 10 Fluxo informação e processamento para a consulta de *emails*.

Na Tabela 8 são apresentados os comandos e as descrições do funcionamento das etapas assinaladas no cenário da Figura 10.

#	Descrição	Comando
1	Iniciado o gestor aplicativo.	
2	Iniciado a aplicação cliente.	
3	Configuração e registo dos dados de acesso ao gestor e dos serviços que este deve processar.	
4	Efectuado um pedido de ligação ao gestor aplicativo	conn#usera#123
5	O gestor aplicativo informa a aplicação cliente que o seu pedido de autenticação foi aceite.	conn#ok
6	A aplicação cliente aguarda notificações do gestor aplicativo.	
7	A aplicação gestora fica a efectuar consultas periódicas de x em x segundos ao <i>WebService</i> , verificando a existência de novos <i>emails</i> .	
8	O gestor aplicativo apercebe-se da chegada de uma nova mensagem à caixa de correio do cliente e notifica-o.	runApp#Aplicação de Email#mail
9	A aplicação cliente recebe a notificação e executa a aplicação de <i>email</i> .	

Tabela 8 Etapas da utilização da arquitectura proposta para consulta de *email*.

### 4.4 Modelo funcional

Com este modelo, pretende-se transferir parte do processamento do dispositivo móvel para um servidor externo de processamento, minimizando os consumos de processador e aumentando assim o tempo útil de utilização efectiva do dispositivo.

É neste servidor externo, que fica instalada a aplicação gestora que se encarrega de notificar o dispositivo sempre que ocorram eventos previamente configurados pelo utilizador. Estes eventos podem ser: a chegada de um *email* à caixa de correio, a alteração do estado do tempo ou a oscilação das cotações da bolsa.

Enquanto a aplicação gestora efectua o processamento, o dispositivo móvel mantém a aplicação cliente em execução, aguardando notificações do gestor aplicacional. Sempre que recebe uma notificação a aplicação actua consoante a mensagem recebida.

Este tipo de implementação, permite diminuir o processamento usado no dispositivo móvel aumentando a autonomia dos dispositivos, face aos habituais modelos existentes no mercado.

### 4.5 Análise do modelo

A implementação do modelo proposto pode tornar-se dispendioso, com o aumento do número de utilizadores a usufruírem do serviço de processamento externo ao dispositivo móvel. Com este aumento de utilizadores, aumenta o consumo dos recursos, tornando-se inevitável investir em mais e melhores servidores.

Para o caso das operadoras venderem este modelo como um serviço, o problema citado anteriormente torna-se numa mais-valia para a empresa de telecomunicações mas num encargo para o cliente.

A comunicação por *sockets* pode condicionar a arquitectura quando as condições da rede são austeras.

### 4.6 Protótipos

Foram implementados uma série de aplicações no âmbito do presente trabalho, que facilitam o mecanismo de validação e avaliação do modelo proposto.

Os protótipos desenvolvidos foram agrupados em dois grupos distintos. O primeiro grupo abrange as aplicações de suporte ao modelo de gestão de aplicações móveis assistido por eventos. O segundo grupo engloba as aplicações desenvolvidas segundo a arquitectura tradicionalmente adoptada pelos fabricantes de *software* móvel.

Com o auxílio das ferramentas de desenvolvimento enunciadas anteriormente foram desenvolvidos os protótipos apresentados de seguida.

### 4.6.1 Protótipos de suporte ao modelo proposto

Todos os protótipos apresentados nesta secção são fundamentais à implementação do modelo de gestão de aplicações móveis assistido por eventos, estando disponível para download em <http://aplicacoesmoveis.golink.pt>.

Todas as aplicações desenvolvidas disponibilizam um menu de configuração que permite adaptá-las aos cenários pretendidos. Estas configurações são armazenadas sobre o padrão *eXtensible Markup Language (XML)*<sup>14</sup>.

#### **WebService**

O *WebService* disponibiliza serviços de consulta de *email*, que são consumidos pelo gestor aplicacional.

#### **Gestor aplicacional**

O gestor aplicacional apresentado na Figura 11 é uma aplicação cliente/servidor que implementa o modelo de gestão das aplicações móveis assistida por eventos. Consome serviços disponibilizados pelos *WebServices* e efectua sempre que se justifica notificações à aplicação móvel.

Para além destas funcionalidades foram desenvolvidos mecanismos automáticos de apoio aos testes. Entre estes, encontra-se a possibilidade de despoletar manualmente eventos e de simular os cenários de teste utilizados ao longo deste projecto.

No Apêndice “Gestor aplicacional” encontram-se descritas todas as funcionalidades da aplicação.

Para criar uma camada de abstracção entre o gestor aplicacional e a base de dados, foi desenvolvido um *middleware* que permite efectuar operações de forma transparente para o programador.

---

<sup>14</sup> *XML – eXtensible Markup Language* é uma linguagem baseada em *SGML* que facilita a partilha de dados entre diversos tipos de sistemas. Esta linguagem foi desenvolvida pelo *W3C*.

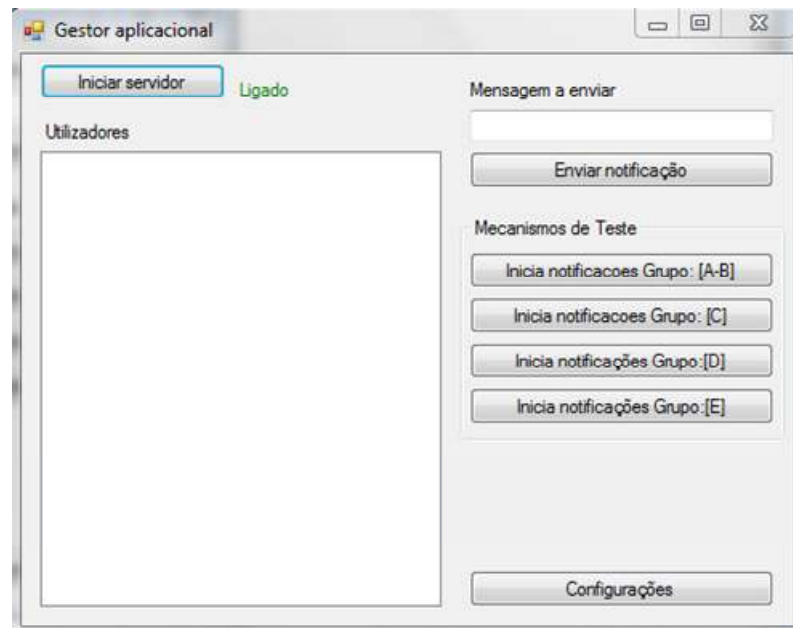


Figura 11 Ecrã principal do gestor aplicacional.

### Aplicação Web

Esta aplicação foi desenvolvida para facilitar a configuração do gestor aplicacional (ver Figura 12). A opção por uma aplicação *Web* advém da possibilidade desta ser configurada através de uma qualquer plataforma com acesso à *Internet*.

A aplicação *Web* possibilita aos utilizadores configurarem os serviços que pretendem ver processados pela aplicação gestora de eventos.



Figura 12 Ecrã principal aplicação *Web*.

### Aplicação móvel genérica assistida por eventos

É uma aplicação desenvolvida para *smartphones* e responsável por processar as mensagens provenientes do gestor aplicacional.

Esta aplicação liga-se ao gestor aplicacional através de *sockets* e aguarda por notificações vindas do gestor aplicacional. Assim que recebe uma notificação inicia o processamento actuando em conformidade com o conteúdo da mensagem recebida.

O funcionamento detalhado da aplicação da Figura 13 pode ser consultado no Apêndice “Aplicação móvel genérica consumidora de eventos implementada segundo o modelo proposto”.



Figura 13 Ecrã principal da aplicação cliente genérica implementando o modelo proposto.

### Aplicação meteorológica móvel assistida por eventos

A aplicação apresentada na Figura 14 funciona segundo o modelo de gestão de aplicações móveis assistido por eventos. Apesar de existir uma aplicação genérica que implementa o modelo em estudo, optou-se por desenvolver uma aplicação meteorológica específica para poder testar o modelo proposto num maior número de cenários distintos.

Com recurso aos *sockets*, a aplicação estabelece ligação ao gestor aplicacional que se encarrega segundo as configurações previamente definidas pelo utilizador de efectuar consultas periódicas ao *WebService* pré-definido.

O gestor fica responsável por notificar a aplicação, sempre que existirem alterações de temperatura. Como a temperatura de uma cidade não é um atributo que se

encontre em constante alteração, esta implementação elimina a necessidade de efectuar consultas periódicas ao servidor meteorológico a fim de obter a temperatura actual.

No Apêndice “Aplicação meteorológica implementada segundo o modelo existente” pode ser consultado mais detalhes sobre esta aplicação.



Figura 14 Aplicação móvel, cliente meteorológico implementando o modelo proposto.

### **Aplicação cliente de *email***

A aplicação da Figura 15 apenas consulta e transfere os *emails* existentes numa determinada conta de *email*.

Foi desenvolvida de forma a poder testar a opção do gestor aplicacional que permite notificar a aplicação cliente genérica do modelo proposto para a execução de uma determinada aplicação.

Sempre que esta aplicação é invocada, efectua a consulta e *download* dos novos *emails* existentes na respectiva caixa de correio de *email*.

No Apêndice “Aplicação cliente de *email* implementada segundo o modelo proposto”, encontram-se descritas todas as funcionalidades, assim como o modo operando da aplicação.



Figura 15 Ecrã da aplicação móvel cliente de *email* implementando o modelo proposto.

### 4.6.2 Protótipos de suporte ao modelo existente

Com esta secção pretende-se expor os protótipos desenvolvidos segundo o modelo tradicionalmente adoptado pela maioria dos fabricantes de aplicações móveis. Estes protótipos foram única e exclusivamente implementados para poder comparar os desempenhos destas quando comparadas com a arquitectura proposta.

#### Aplicação de *email* implementando o modelo existente

A aplicação cliente apresentada na Figura 16, tem um funcionamento semelhante ao apresentado pela maioria dos clientes de *email* tradicionais. Esta aplicação efectua consultas periódicas ao servidor de *email* verificando a existência de novas mensagens. Quando a aplicação detecta novos *emails* na caixa de correio, efectua a transferência do mesmo para o dispositivo.

A aplicação é bastante simples, podendo ser consultada no Apêndice “Aplicação de *email* implementada segundo o modelo existente”.



Figura 16 Ecrã principal da aplicação cliente de *email* implementando o modelo existente.

### **Aplicação meteorológica implementado o actual modelo existente**

A aplicação da Figura 17 possui um funcionamento similar à apresentada no ponto anterior. Efectua consultas periódicas a um servidor externo no entanto, ao invés de se ligar ao servidor de *email*, realiza consultas a um *WebService* que disponibiliza informação sobre as condições meteorológicas em determinada cidade.



Figura 17 Ecrã da aplicação cliente meteorológico implementando o modelo existente.

### **Aplicação móvel de registo dos sinistros rodoviários implementando o modelo proposto**

Este protótipo foi desenvolvido com recurso à plataforma *Kalipso*. Implementa o modelo proposto e permite simular o funcionamento deste, para o envio e recepção de informações referentes a sinistros rodoviários.

Após a aplicação móvel genérica receptora de eventos interceptar o evento do gestor aplicacional a notificar um sinistro, é executada a aplicação móvel de registo de sinistros. Na Figura 18 é possível observar um exemplo da chegada de uma notificação de sinistro.

Com o objectivo de minimizar a introdução de dados pelo agente de autoridade que utiliza a aplicação móvel é efectuado um pré-preenchimento dos campos. Os campos pré-preenchidos são os que foram possíveis apurar pelo operador da central policial que recebeu o alerta de sinistro.



Figura 18 Aplicação móvel de registo dos sinistros rodoviários.

### **Aplicação de registo dos sinistros rodoviários**

Para apoiar a aplicação móvel, foi desenvolvido uma aplicação para ser executada na central responsável por processar os alertas de sinistro. Esta aplicação permite que o agente registe os sinistros numa base de dados para o gestor aplicacional processar

## Modelo de notificação por eventos

---

(ver Figura 19). Quanto mais completo fôr o preenchimento do registo, menos informação terá de ser introduzida pelo agente que utiliza a aplicação móvel.

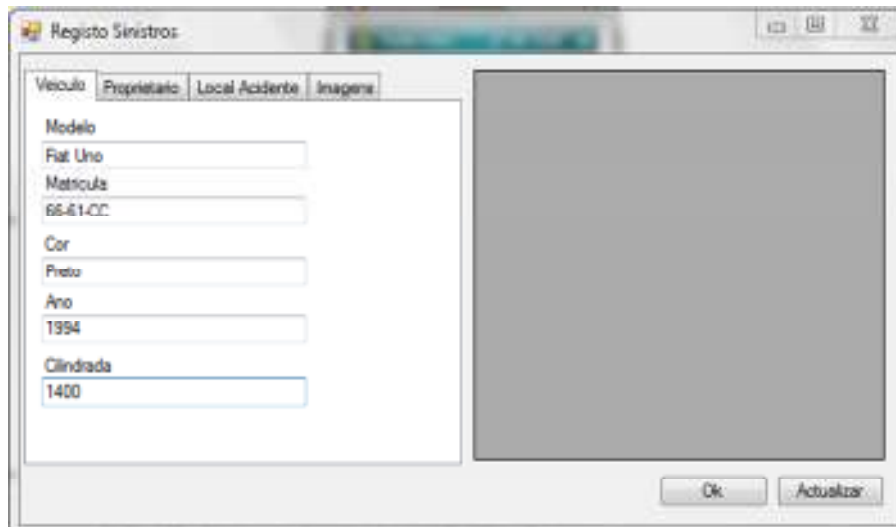


Figura 19 Aplicação de registo dos sinistros rodoviários.

O gestor efectua consultas regulares à base de dados policial com o intuito de notificar a aplicação móvel, sobre a ocorrência de novos sinistros rodoviários. Quando o gestor detecta um novo sinistro, notifica o dispositivo móvel do sucedido, enviando-lhe a informação relativa ao sinistro.

Com o modelo apresentado é possível transferir parte do processamento dos dispositivos móveis para um servidor externo.

No servidor, encontra-se instalado um gestor aplicacional que periodicamente verifica serviços pré-configurados pelo utilizador. Assim que algum dos serviços fornece informação relevante para o utilizador, o gestor notifica o dispositivo móvel.

No capítulo cinco é efectuado a avaliação do modelo apresentado anteriormente. Esta avaliação pretende validar o modelo recorrendo a cinco cenários distintos.

## 5 *Trabalho experimental*

---

O modelo da arquitectura apresentado no capítulo antecedente é dividido em três secções distintas. A primeira secção disponibiliza serviços para o gestor aplicacional da segunda secção. Na terceira secção, encontra-se a aplicação móvel responsável por ultimar o processamento das notificações vindas do gestor aplicacional.

Para que a comunicação entre o dispositivo móvel e o gestor aplicacional decorra de forma integrada é usado o protocolo de comunicação que se encontra descrito no capítulo Protocolo. No presente capítulo, é validado e avaliado o modelo de gestão de aplicações móveis assistido por eventos descrito no capítulo quatro.

Inicialmente são apresentados os testes de validação do modelo e posteriormente apresentada a sua avaliação. Ambos os testes foram efectuados em cenários próximos da realidade, tornando-os assim mais relevantes. Para finalizar o capítulo é exposto uma análise dos resultados obtidos.

### 5.1 **Plano de avaliação dos testes**

Esta secção descreve os testes de validação e avaliação efectuados ao modelo apresentado no capítulo quatro. Os primeiros testes possibilitaram validar os mecanismos que constituem o modelo proposto eliminando possíveis incorrecções de implementação.

Nos testes de avaliação foram utilizados cinco cenários distintos. Três deles apoiados na consulta da caixa de correio de um utilizador, para apurar a existência de novas mensagens de *email*. O quarto apoiado na consulta dos dados meteorológicos da cidade de Lisboa. O quinto, no comportamento de uma aplicação móvel de apoio ao registo de sinistros rodoviários. Exceptuando o grupo E, para todos os outros existe um cenário distinto que é testado em cada um dos modelos. A primeira é responsável por avaliar o

cenário proposto com uma ferramenta que implementa o modelo existente. A segunda avalia o cenário utilizando o modelo proposto. Com esta comparação torna-se mais fácil perceber as reais potencialidades e limitações deste tipo de modelo.

### 5.1.1 Testes de validação

Os testes de validação foram efectuados em quatro fases distintas. As primeiras três coincidem com as secções do modelo. A quarta, com o desenvolvimento de uma aplicação de consulta de *emails* e das condições meteorológicas implementando o modelo existente e o modelo proposto.

Este tipo de testes é indispensável. Permitem validar o perfeito funcionamento dos diferentes intervenientes no modelo, garantindo assim a fiabilidade dos testes.

#### Testes aos serviços

Os testes ao modelo iniciaram-se com a avaliação dos serviços disponibilizados pelo *WebService* apresentados na Tabela 9. Incidiram sobre a invocação dos métodos disponibilizados pelo *WebService*, esperando que este respondesse com a informação na íntegra.

O *WebService* utilizado nos testes é responsável por disponibilizar os seguintes serviços:

Serviços disponibilizados	Descrição
<i>getNumMails</i>	Devolve o número de <i>emails</i> existentes no servidor;
<i>getMessages</i>	Devolve todas as mensagens de <i>email</i> que se encontrem por ler no servidor de <i>email</i> ;
<i>getEmailMessage</i>	Devolve determinada mensagem de <i>email</i> ;
<i>haveNewMails</i>	Informa da existência de novas mensagens de <i>email</i> .

Tabela 9 Serviços disponibilizados pelo *WebService*

### Testes ao gestor aplicacional

Depois de efectuados os testes aos serviços, iniciaram-se os testes ao gestor aplicacional. Estes, tiveram como especial foco o mecanismo de consumo de informação proveniente do *WebService*.

Com o teste foi validado a eficácia do gestor em despoletar eventos aquando da existência de novos *emails* na caixa de correio do utilizador.

Conjuntamente com o protocolo comunicacional apresentado no capítulo quatro, foram efectuados testes de comunicação utilizando o gestor aplicacional e um terminal de *telnet*<sup>15</sup>. Com o teste foi possível validar a perfeita troca de mensagens entre cliente e gestor aplicacional.

### Testes à aplicação móvel

Os testes à aplicação móvel recaíram sobre o estabelecimento de ligações com o gestor aplicacional e com a interpretação das instruções provenientes do gestor aplicacional.

Os testes ao estabelecimento da conexão com o gestor aplicacional iniciam-se com um pedido de autenticação, mediante o protocolo adoptado.

Após os testes de conectividade, procederam-se os testes da interpretação dos eventos e mensagens vindas do gestor aplicacional.

As mensagens recebidas (ver Tabela 10) pela aplicação móvel devem gerar uma das seguintes operações:

Mensagem	Descritivo
Executar uma aplicação	A aplicação móvel executa a aplicação identificada na mensagem recebida;
Notificar o cliente através de um <i>pop-up</i> <sup>16</sup>	A aplicação apresenta no dispositivo móvel uma notificação em formato <i>pop-up</i> , com a mensagem recebida.

Tabela 10 Eventos provenientes do gestor aplicacional.

---

<sup>15</sup> *Telnet* – protocolo utilizado para permitir a comunicação entre dispositivos que estejam ligados numa rede.

<sup>16</sup> *Pop-Up* – é uma janela suplementar que abre para notificar o cliente sobre um determinado acontecimento que ocorre ou vai ocorrer no dispositivo móvel.

### Testes à aplicação de consulta de *email* implementando o modelo tradicional

A aplicação de consulta de *email* possui um funcionamento semelhante ao utilizado nos clientes de *email* tradicionais.

Os testes de validação da aplicação incidiram sobre o mecanismo de autenticação e sobre a correcta apresentação das mensagens de *email*.

#### 5.1.2 Testes de avaliação

Para avaliar o modelo apresentado no Capítulo quatro foram definidos quatro grupos de testes, expostos na Tabela 11.

Todos os testes efectuados sobrevêm da consulta da caixa de correio do utilizador, para verificar a existência de novas mensagens.

Exceptuando o grupo E, todos os restantes são constituídos por dois testes. Inicialmente, é testado o comportamento do dispositivo móvel implementando uma arquitectura semelhante à usualmente utilizada nas plataformas normais. Posteriormente, são efectuados testes sobre a arquitectura proposta.

Para garantir que todos os testes se iniciavam com características idênticas, foi sempre utilizado o mesmo dispositivo móvel com condições de carga idênticas. Todos os testes foram iniciados com a carga de bateria do dispositivo a 100%.

Grupo	Definição
Grupo A	Utilizando o modelo tradicional e o proposto, avaliar os recursos de bateria consumidos pelo dispositivo móvel durante um período de três horas utilizando uma aplicação de <i>email</i> .
Grupo B	Avaliar o tempo de duração de uma carga completa de bateria utilizando o modelo tradicional e o modelo proposto e utilizando uma aplicação de <i>email</i> . Os emails chegam à caixa de correio do utilizador com a periodicidade de 20 minutos.
Grupo C	Avaliar o tempo de duração de uma carga completa de bateria utilizando o modelo tradicional e o modelo proposto numa aplicação de <i>email</i> . Os emails chegam de forma aleatória à caixa postal do utilizador.
Grupo D	Avaliar o tempo de duração de uma carga completa de bateria utilizando o modelo existente e o modelo proposto utilizando uma aplicação para a consulta das condições meteorológicas.
Grupo E	Avaliar o comportamento da aplicação de registo de sinistros rodoviários e a fiabilidade do protocolo para a transmissão de dados.

Tabela 11 Grupo de testes

## Trabalho experimental

A arquitectura tradicional utilizada nos testes pode ser encontrada na esmagadora maioria das aplicações cliente tradicionais. Na Figura 20 pode ser consultado um esquema deste modelo.

Esta arquitectura tem por base uma aplicação móvel, configurada para aceder ao servidor de *email*, mediante um período temporal pré configurado pelo utilizador. Dentro do período temporal pré-definido pelo utilizador, a aplicação realiza consultas periódicas ao servidor de *email* para verificar a chegada de novas mensagens.

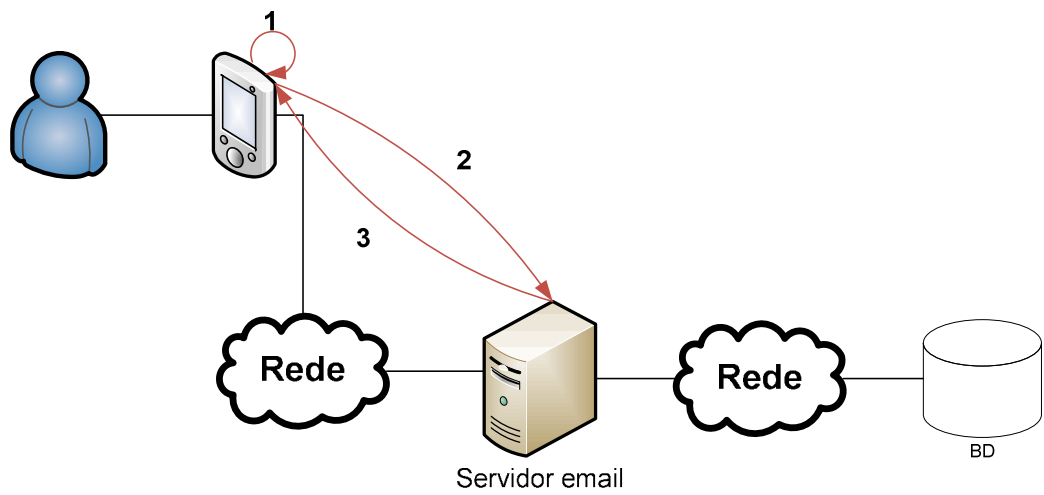


Figura 20 Arquitectura de uma aplicação cliente implementando o modelo de *email* existente.

Na Tabela 12 é apresentado a descrição de cada etapa enumerado na arquitectura da Figura 20.

#	Descrição
1	É iniciada a aplicação
2	De dez em dez minutos a aplicação móvel consulta o servidor de email verificando a chegada de novas mensagens
3	O servidor de email envia informação à aplicação móvel sobre a chegada de um novo email

Tabela 12 Etapas do fluxo de informação numa aplicação implementando o modelo existente.

## Trabalho experimental

O modelo gestor aplicacional móvel assistido por eventos utilizado durante os testes mantêm-se inalterável para os quatro grupos. Na Figura 21 pode ser analisada a referida arquitectura.

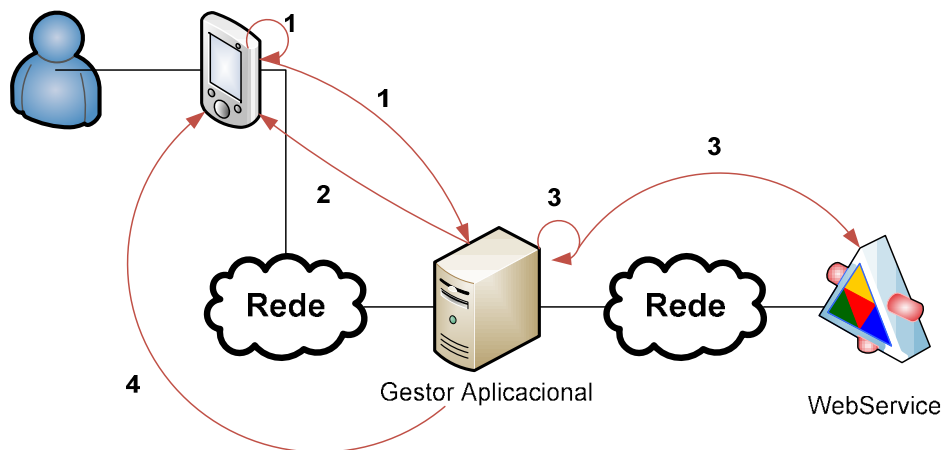


Figura 21 Esquema da arquitectura proposta.

O esquema apresentado enumera as etapas do fluxo de informação no modelo proposto. Estas etapas podem ser consultadas em detalhe na Tabela 13.

#	Descrição
1	A Aplicação móvel é colocada em execução e é estabelecido um pedido de autenticação perante o gestor aplicacional.
2	O gestor aplicacional confirma a ligação.
3	O gestor aplicacional consulta o <i>WebService</i> de dez em dez minutos.
4	O gestor aplicacional notifica o dispositivo móvel da chegada de novas mensagens.

Tabela 13Etapas do fluxo de informação na arquitectura proposta.

## **5.2 Grupo de testes avaliação ao modelo existente versus modelo tradicional durante 180 minutos aplicado à recepção periódica de *emails* [Grupo A]**

Com este grupo, pretende-se avaliar o comportamento do modelo proposto quando comparado com o modelo existente utilizado nas aplicações de *email* do mercado.

O comportamento é avaliado subsequentemente pela percentagem de bateria consumida pelo dispositivo móvel sob um período de tempo de três horas de utilização.

### **5.2.1 Teste ao modelo existente**

O teste, com duração de três horas utilizou uma aplicação de consulta de *email* semelhante às existentes no mercado das aplicações móveis.

Esta aplicação efectua consultas regulares ao servidor de *email* em períodos regulares de dez minutos e durante três horas. Estas consultas permitem verificar a existência de novas mensagens de *email*.

Apesar de a aplicação efectuar consultas regulares de dez em dez minutos, o utilizador apenas recebe novas mensagens de *email* de vinte em vinte minutos. Esta discrepância de tempo traduz-se em um acesso a mais ao servidor de *email* a cada vinte minutos.

### **Resultado do teste**

No final das três horas de teste, o dispositivo móvel tinha consumido 18% dos recursos da bateria.

### **5.2.2 Teste ao modelo proposto**

Para este teste, o gestor aplicacional foi configurado para efectuar consultas periódicas de dez em dez minutos ao *WebService*. No entanto, este processamento encontra-se a ser efectuado por um servidor externo ao invés de ser da responsabilidade do dispositivo móvel.

No dispositivo móvel apenas se encontra em execução a aplicação cliente. Esta aplicação fica a aguardar notificações provenientes do gestor aplicacional. Sempre que o gestor aplicacional se apercebe da chegada de uma nova mensagem de *email* à caixa de correio do utilizador, notifica a aplicação móvel do ocorrido.

Após a aplicação móvel ter sido notificada, executa a respectiva aplicação de *email*, de forma a que o utilizador possa ler a mensagem.

Com este modelo, garante-se que a aplicação móvel apenas consulta o servidor de *email* quando previamente informada pelo gestor. Assim, é garantido menos acessos e por sua vez menos processamento e recursos de rede por parte do dispositivo móvel.

### **Resultado do teste**

Os resultados do teste podem vir a ser considerados satisfatórios, visto ter apenas consumido 10% de bateria. Obteve-se um ganho de 8% quando comparado com o teste anterior.

Poderá vir a ser considerado satisfatório, uma vez que consumiu quase 50% menos de recursos que o teste utilizando o modelo existente.

### **5.2.3 Análise dos resultados**

Com os resultados obtidos e apresentados no Gráfico 5, é perceptível que o modelo de gestão de aplicações móveis assistido por eventos obteve um desempenho superior.

O dispositivo móvel consumiu mais recursos no modelo existente visto que efectua consultas regulares ao servidor de *email*.

O cenário da chegada de uma mensagem de *email* de vinte em vinte minutos durante três horas é uma aproximação à realidade, no entanto, a finalidade do teste é perceber o comportamento do modelo em períodos curtos de tempo. Estes períodos podem ser equiparados a pequenos picos de utilização por parte dos utilizadores dos dispositivos, que podem não querer usufruir do modelo durante todo o período da bateria.

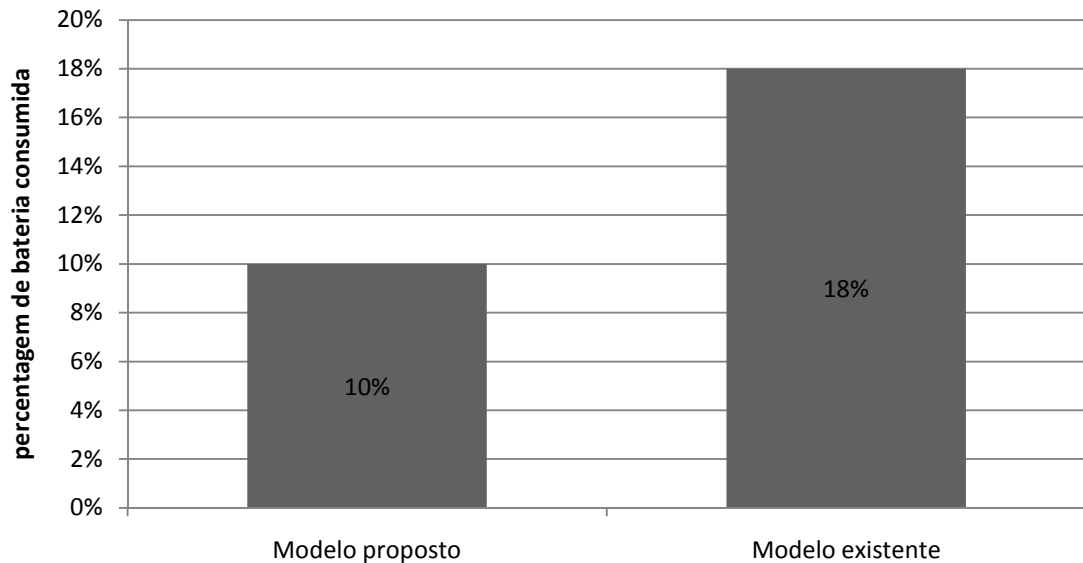


Gráfico 5 Percentagem de bateria consumida durante os testes.

### **5.3 Grupo de testes de avaliação ao modelo existente *versus* modelo tradicional durante uma carga completa de bateria aplicado à recepção periódica de *emails* [Grupo B]**

Utilizam os modelos e as aplicações apresentadas no ponto anterior. No entanto, os testes precedentes tinham uma duração de três horas, enquanto neste grupo será aproveitada a totalidade da carga do dispositivo móvel.

Neste teste irá ser contabilizado o tempo útil do telefone em cada um dos modelos sobre um cenário da chegada de uma mensagem de *email* a cada vinte minutos.

Como estes testes têm durações longas, foi criado um mecanismo de registo baseado em ficheiro *XML* que assinala a actividade das aplicações com o registo da hora de cada processamento efectuado pela aplicação. É com base neste ficheiro de registo que é efectuado a avaliação de desempenho dos dois modelos.

#### **5.3.1 Teste ao modelo existente**

Teste semelhante ao do grupo A, no entanto o período de teste é alargado ao período útil da bateria.

### **Resultado do teste**

Após a realização do teste obteve-se um tempo útil de bateria equivalente a 492 minutos.

### **5.3.2 Teste ao modelo proposto**

Este teste, tal como o apresentado no ponto anterior difere única e exclusivamente no tempo efectivo do teste. Neste cenário é igualmente utilizado a carga completa da bateria do dispositivo móvel.

### **Resultado do teste**

Com o teste foi conseguido um tempo útil de bateria de 1438 minutos. Este resultado é bastante mais díspar entre os dois modelos do que o apresentado no teste anterior. Consumindo menos recursos o dispositivo mantém-se utilizável durante mais tempo, confirmando as possíveis potenciais vantagens do modelo proposto.

### **5.3.3 Avaliação dos resultados**

Os resultados obtidos realçam as tendências dos testes apresentados no primeiro grupo de testes. No final dos testes do primeiro grupo, era perceptível a tendência do modelo proposto apresentar melhores desempenhos que o modelo existente.

No Gráfico 6 é possível observar que o modelo da arquitectura proposta obteve um desempenho bastante superior ao apresentado pelo modelo existente.

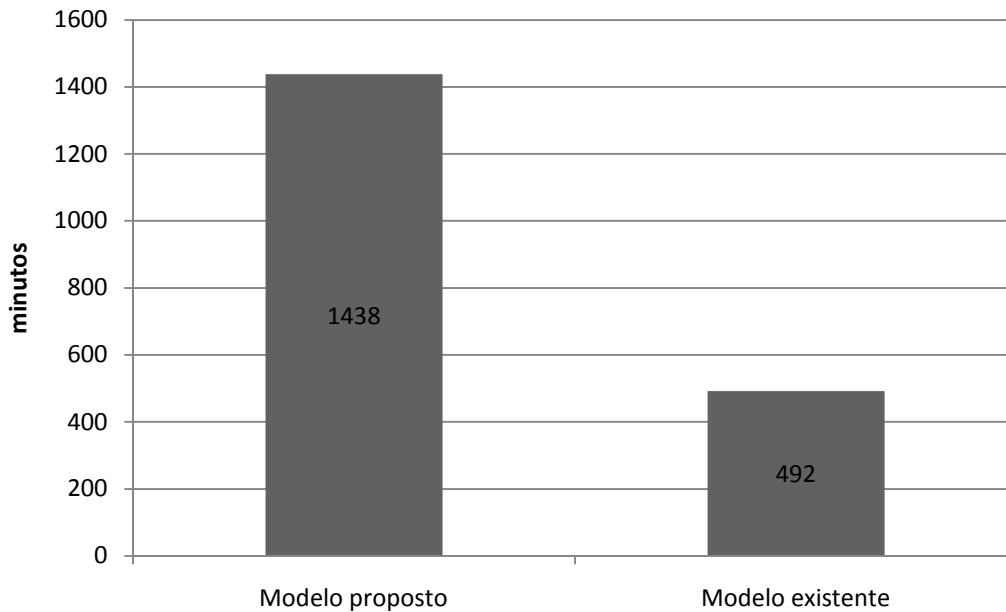


Gráfico 6 Tempo de duração máxima da bateria do dispositivo móvel durante os testes da chegada de um *email* de vinte em vinte minutos.

### **5.4 Grupo de testes de avaliação do modelo existente *versus* modelo tradicional durante uma carga completa de bateria aplicado à recepção aleatória de *emails* [Grupo C]**

Este grupo de testes difere do anterior uma vez que a recepção de *emails* é gerada de forma aleatória, enquanto nos testes dos grupos anteriores eram espaçadas por períodos de tempo fixo.

A aleatoriedade da chega de *emails* à caixa de correio utilizada no teste, é efectuada por um algoritmo que segundo determinado resultado matemático envia uma mensagem de *email* para a conta do utilizador.

Assim, a aleatoriedade da chegada de *emails* à caixa de correio do utilizador, torna o cenário mais próximo da realidade, tentando simulando um habitual tráfego de uma normal conta de *email*.

Estes dois últimos grupos de testes tiveram a duração de uma carga completa de bateria. Devido ao longo período dos testes monitorizou-se a actividade das aplicações em ficheiros *XML*.

Assim, para o teste da arquitectura existente, sempre que a aplicação efectuava uma leitura registava a hora de acesso num ficheiro *XML*.

Na arquitectura proposta, foi utilizado um ficheiro *XML* que armazena a hora das notificações provenientes do gestor aplicacional.

### **5.4.1 Teste ao modelo existente**

Para o teste ao modelo existente foi novamente utilizado uma aplicação de cliente de *email* que efectua consultas regulares ao servidor de *email* de dez em dez minutos.

Com este modelo são efectuadas inúmeras consultas desnecessárias ao servidor de *email*.

O cenário da chegada aleatória de mensagens e da aplicação efectuar consultas regulares ao servidor de *email*, mediante um período de tempo pré-estabelecido, enquadra-nos nas arquitecturas usualmente implementadas no mercado das aplicações móveis.

#### **Resultado do teste**

Os resultados obtidos com este teste, ficaram abaixo das expectativas já que a bateria apenas durou 400 minutos.

Este teste verificou-se mais penoso para o dispositivo que o apresentado no grupo anterior.

### **5.4.2 Teste ao modelo proposto**

Com este teste, os *emails* são enviados aleatoriamente para a caixa de correio do utilizador sendo a aplicação móvel notificada pelo gestor da chegada desses *emails*.

#### **Resultado do teste**

O modelo proposto obteve novamente os resultados mais satisfatórios. Ao utilizar o modelo proposto conseguiu-se manter o dispositivo ligado 990 minutos.

### 5.4.3 Avaliação dos resultados

Pretendeu-se validar o modelo orientado a eventos com a utilização de um sistema aleatório de envio de mensagens de *email*.

Com esse intuito foi criado um cenário de tráfego semelhante ao de uma conta de *email* comum, onde as mensagens chegam de forma não sequencial.

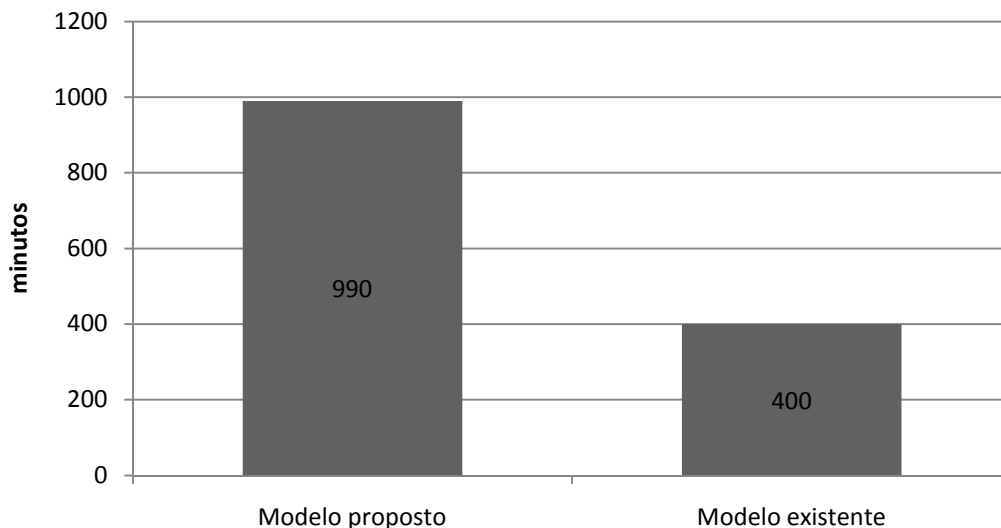


Gráfico 7 Tempo de duração máximo da bateria do dispositivo móvel utilizando a aleatoriedade na recepção de *emails*.

Os resultados deste teste (ver Gráfico 7) foram bastante satisfatórios. O dispositivo foi notificado quinze vezes sobre a chegada de *emails*, e permaneceu ligado durante 990 minutos. Estes resultados representam mais de 50% do que o conseguido pelo modelo usado no teste ao modelo existente.

## 5.5 Grupo de testes de avaliação do modelo existente *versus* modelo tradicional durante uma carga completa de bateria aplicado à consulta das condições meteorológicas [Grupo D]

Este grupo de testes incide sobre aplicações móveis desenvolvidas para proverem dados referentes às condições meteorológicas na cidade de Lisboa. Actualmente, este tipo de aplicações pode ser encontrado na maioria dos *smartphones* existentes no mercado.

Os testes consistem na avaliação do desempenho de duas aplicações móveis implementadas sobre duas metodologias diferentes.

A primeira aplicação utiliza o modelo usualmente adoptado pela maioria dos fabricantes de *software*. É testado no dispositivo móvel ao longo de uma carga completa de bateria, onde as temperaturas apresentadas serão actualizadas em períodos de dez minutos.

No segundo teste a aplicação implementa o modelo de gestão de aplicações móveis assistido por eventos, tendo também a duração de uma carga completa de bateria.

### **5.5.1 Teste ao modelo existente**

Para o teste ao modelo existente foi utilizado uma aplicação móvel que efectua consultas regulares a um *WebService* em períodos de vinte minutos.

Como as temperaturas de uma cidade não estão em constante alteração, esta aplicação realiza mais acessos ao *WebService* do que o necessário. Grande parte dos acessos efectuados indica que a temperatura se mantém inalterada, desde a última consulta.

#### **Resultado do teste**

Com o teste foi possível manter o dispositivo ligado durante 480 minutos, tendo ocorrido poucas oscilações de temperatura durante o tempo que o dispositivo se manteve ligado.

### **5.5.2 Teste ao modelo proposto**

Neste teste a aplicação gestora de eventos efectua consultas ao *WebService* em períodos regulares de vinte minutos, ficando responsável por notificar a aplicação móvel assim que exista uma variação de temperatura.

#### **Resultado do teste**

Com este teste foi possível manter o dispositivo ligado durante 1300 minutos, o que representa mais do dobro do tempo útil conseguido no teste ao modelo existente.

### 5.5.3 Avaliação dos resultados

Os resultados obtidos neste grupo de testes podem vir a ser bastante satisfatórios, visto ter-se conseguido perceber que quantos mais acessos efectuamos ao *WebService* mais recursos de processamento e energia são consumidos.

A diferença de tempo útil de utilização do *smartphone* é bastante evidente. Com a arquitectura proposta obtivemos 1300 minutos ao invés dos 480 conseguidos com a utilização de uma aplicação baseada no modelo habitualmente utilizado (ver Gráfico 8).

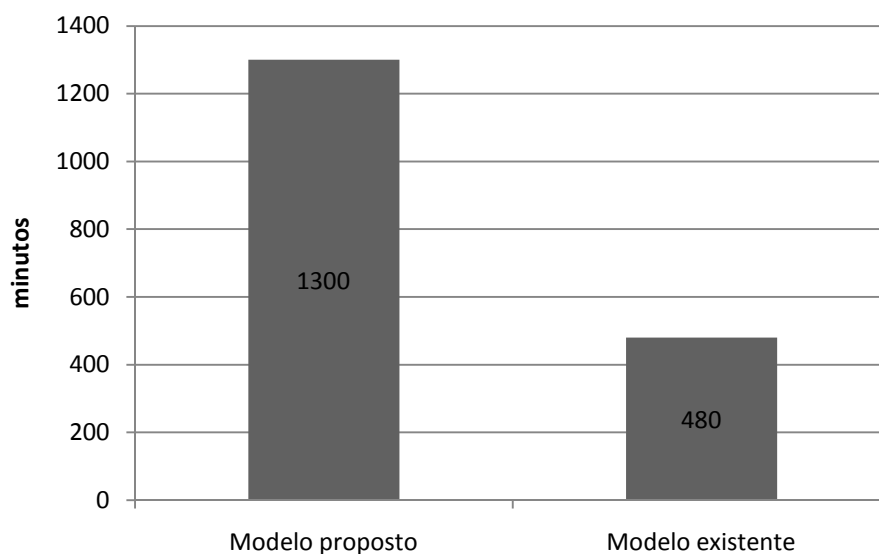


Gráfico 8 Tempo de duração máximo da bateria do dispositivo móvel utilizando a aleatoriedade na recepção de *emails*.

## 5.6 Grupo de testes de avaliação da utilização do modelo proposto no dispositivo de registo de sinistros rodoviários [Grupo E].

Este teste permitiu validar a utilização de uma aplicação de registo de sinistros rodoviários implementada segundo o modelo proposto.

### 5.6.1 Teste ao modelo proposto

Este teste inicia-se com a simulação da ocorrência de um sinistro com o prévio registo na aplicação de gestão de ocorrências. Após o registo a aplicação gestora notifica a aplicação genérica gestora de eventos. Esta dá ordem de execução da

aplicação móvel de apoio aos sinistros. Aquando da execução da aplicação móvel são pré-preenchidos os registos provenientes da notificação.

### **Resultado do teste**

O teste revelou-se eficiente, a aplicação móvel interpretou correctamente os dados provenientes do gestor aplicacional, conseguindo efectuar o pré-preenchimento dos dados do sinistro.

## **5.7 Análise dos resultados**

O modelo de gestão de aplicações móveis por eventos apresentou em todos os grupos de testes o melhor desempenho.

Com o primeiro grupo de testes, é possível perceber que o modelo é vantajoso mesmo para curtos períodos de utilização.

Ao comparar os resultados obtidos nos testes cuja duração é uma carga completa de bateria, percebe-se que o tempo útil de utilização do *smartphone* é superior quando se utiliza o modelo de gestão aplicacional assistido por eventos.

É de salientar que no grupo de testes de avaliação do modelo existente *versus* modelo tradicional durante uma carga completa de bateria aplicado à recepção periódica de emails foram obtidos os melhores resultados.

É possível que em cenários com elevado tráfego de mensagens de *email* ou com constantes oscilações de temperatura o tempo útil do *smartphone* seja mais reduzido.

Salienta-se também a condicionante de apenas se poder efectuar os testes num único dispositivo móvel podendo assim condicionar a avaliação dos resultados.

Os testes apresentados anteriormente permitem avaliar o modelo positivamente tendo sido escalonados em três grupos distintos. Cada um dos grupos efectua testes aos dois tipos de modelo referenciados, facilitando assim a leitura dos resultados obtidos.

No capítulo seis estão reunidos os aspectos mais relevantes do presente trabalho e será realizado a título de conclusão a análise dos testes efectuados no corrente capítulo.

Com as conclusões, são também apresentados estudos futuros em torno do modelo proposto.

## 6 Conclusão

---

O modelo de gestão de aplicações móveis assistido por eventos contorna a necessidade das aplicações precisarem de efectuar consultas regulares a serviços. Estas consultas permitem às aplicações móveis manterem a sua informação actualizada.

Ao serem comutados os acessos periódicos a serviços, por um sistema de notificação em tempo real, permitiu ao modelo proposto auferir os melhores desempenhos ao longo dos testes efectuados.

No presente capítulo são identificadas, analisadas e descritas as conclusões do estudo apresentado.

São ainda apresentadas linhas de investigação que podem vir a ser exequíveis num futuro próximo. Perspectiva-se assim, contribuir com cenários capazes de fomentar mais-valias para a investigação na área da computação móvel.

A computação móvel tem progredido bastante nos últimos anos. Actualmente com a redução de preços dos componentes móveis, surgem inovações diárias na área da mobilidade.

Os telefones móveis, em particular os *smartphones* disponibilizam inúmeros serviços importantes para uso pessoal e empresarial. Estas funcionalidades despertam cada vez mais o interesse dos consumidores, estimulando investimentos na área da mobilidade.

O interesse de fabricantes como a *Apple*, *HTC*, *BlackBerry* e a *Nokia*, foram preponderantes no desenvolvimento dos *smartphones*. As actuais arquitecturas resultam de constantes aperfeiçoamentos financiados por estas empresas.

Com o desenvolvimento dos *smartphones* surgem sistemas operativos móveis, que providenciam potencial análogo aos encontrados nos computadores pessoais.

O desenvolvimento das redes sem fio também impulsiona o crescimento da computação móvel. Actualmente, oferecem melhores níveis de cobertura e disponibilizam velocidades de acesso a dados bastante aceitáveis, quando comparadas com as providenciadas pelas redes fixas.

### 6.1 Problema

Em todas as tecnologias emergentes, existem áreas que se desenvolvem mais rapidamente que outras. A computação móvel necessita de evoluir ao nível da confiabilidade, qualidade de serviço, autonomia, desempenho e adaptação.

O problema do estudo pretendia conferir melhorias de desempenho e autonomia nos dispositivos móveis.

Os fabricantes de *hardware* são responsáveis por evoluírem os seus componentes de forma a apresentar os melhores índices de desempenho.

Ao ser disponibilizado melhor *hardware*, as empresas de *software* tentam imediatamente desenvolver novas funcionalidades para os utilizadores. Com esta complementaridade existe pouco espaço para o estudo de novos modelos que possibilitem melhorar a autonomia dos dispositivos móveis.

### 6.2 Metodologia

Com a abordagem do problema, foi possível definir as características essenciais à implementação de uma arquitectura que permite maximizar a autonomia dos dispositivos móveis.

O desenvolvimento estava limitado a aplicações compatíveis com o dispositivo *HTC Touch* que fora disponibilizado para os testes. No entanto, o desenvolvimento não estava dependente da utilização de ferramentas *Microsoft*, simplesmente porque o dispositivo utilizava um sistema operativo proprietário da marca.

Verificou-se vantajoso a aposta nas ferramentas do gigante americano e no *Kalipso* da *SysDev*. O *Microsoft Visual Studio* e o *Kalipso* permitiram desenvolver com eficiência e rapidez os protótipos utilizados nos cenários de teste.

A possibilidade de conjuntamente com o *Windows Mobile SDK* efectuar *debug* das aplicações em ambiente simulado tornou o desenvolvimento cómodo e independente do dispositivo.

### 6.3 Testes

Os testes efectuados indicam que o modelo proposto pode vir a ser vantajoso. Em todos eles, o modelo proposto obteve resultados superiores aos apresentados pelo modelo tradicional. (Bernardes, Rijo et al. 2010)

Com a leitura dos resultados dos testes efectuados, é possível concluir que para o *HTC* a migração do processamento do dispositivo móvel para servidores externos traduziu-se em melhorias de desempenho.

Os testes efectuados com a carga completa de bateria foram os mais morosos. A morosidade deve-se ao facto das falhas de rede que ocorriam e inviabilizavam o teste. Sempre que o teste era inviabilizado era necessário proceder ao restabelecimento da carga máxima do dispositivo.

Sendo crucial para o modelo usufruir dos recursos de rede, é importante que estes não sofram qualquer anomalia.

Com os testes efectuados ao cenário de testes que utilizava um sistema aleatório de mensagens, foi possível assimilar que mediante este tipo de cenários análogos aos do dia-a-dia, conseguem-se resultados satisfatórios, quando comparado com o modelo usualmente utilizado. Neste teste o dispositivo manteve-se ligado durante dezasseis horas e meia. Se for efectuada uma comparação com um mínimo de oito horas de trabalho diário, o dispositivo permanecia em funcionamento durante dois dias úteis. Obviamente que o período referido é uma mera leitura de números, já que para avaliar o tempo efectivo de utilização, teria de ser contabilizado o tráfego de chamadas, *SMSs* diários.

A aplicação de registo de sinistros rodoviários demonstra que é possível utilizar este modelo para diversos tipos de funcionalidades sem existir a necessidade de grandes implementações. Com o aumento do tempo útil dos dispositivos que implementam a arquitectura proposta, pode-se auferir que esta aplicação permite manter os dispositivos móveis mais tempo ao dispor das autoridades.

### 6.4 Principais contributos

O principal contributo da dissertação é a apresentação de uma alternativa ao modelo tradicionalmente utilizado nas aplicações móveis.

O modelo de gestão de aplicações móveis assistido por eventos oferece uma perspectiva diferente sobre a implementação e utilização das aplicações móveis em benefício da autonomia dos dispositivos.

Com a necessidade de efectuar testes ao modelo de gestão das aplicações móveis assistido por eventos, imergiu a necessidade de implementar protótipos de suporte ao modelo. Apesar de serem protótipos, são totalmente funcionais e adaptam-se a inúmeros cenários do quotidiano.

Por fim, contribuiu-se como o desenvolvimento de um protocolo de comunicação para auxiliar a correcta comunicação entre os diversos intervenientes do modelo.

### 6.5 Principais conclusões do estudo

Como principais conclusões, salienta-se o facto de ter sido apresentado um modelo que permite aumentar a autonomia de um dispositivo móvel que usufrua de informação proveniente de serviços na *Internet*.

A migração do processamento dos dispositivos móveis para servidores externos conferiu menores consumos nos dispositivos móveis. Este ganho de autonomia possibilita ao utilizador do dispositivo utilizá-los durante mais tempo, ou usufruir dos recursos que sobejam noutras aplicações.

A migração do processamento para servidores externos poderá despoletar custos adicionais para operadoras e/ou utilizadores.

É viável criar um pacote de *software* que permita a um utilizador com *Internet*, implementar facilmente o seu próprio servidor pessoal. No entanto, o sucesso do modelo será superior se os custos forem suportados total ou parcialmente pelas operadoras.

### 6.6 Trabalho Futuro

A arquitectura projectada indicia permitir a optimização do consumo dos recursos energéticos do dispositivo móvel.

Seria interessante estudar a possibilidade de desenvolver um gestor aplicacional onde os eventos cheguem ao dispositivo móvel através de sinalizações similares às utilizadas nos *SMSs* e chamadas telefónicas.

A implementação apresentada deverá diminuir ainda mais o processamento do dispositivo.

Com uma implementação semelhante à apresentada na Figura 22 o telefone ficaria completamente em *stand by* evitando o actual processamento da aplicação cliente.

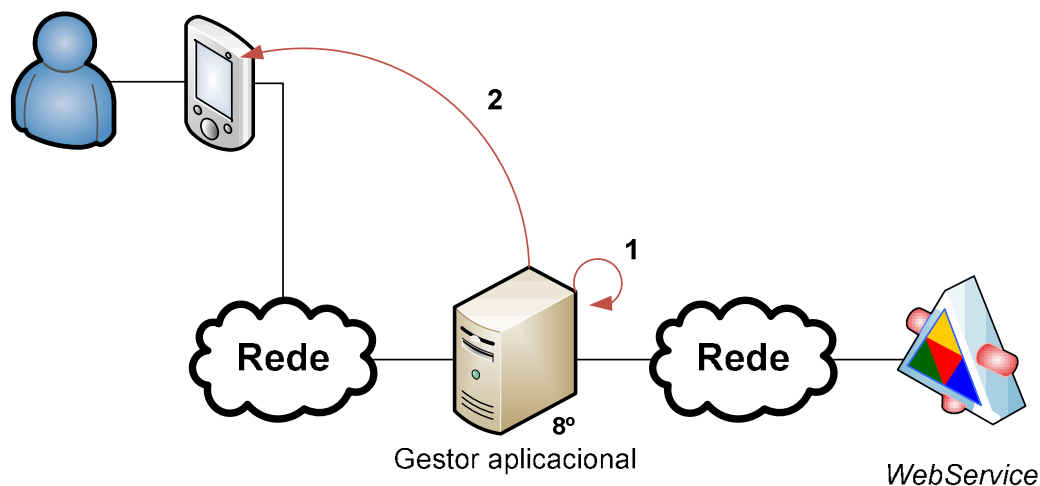


Figura 22 Arquitectura de notificação similar à utilizada nas chamadas telefónicas e *SMSs*.

Outra abordagem interessante seria a possibilidade de implementar modelos híbridos, que permitam comutar entre os dois modelos. Com este tipo de implementação seria evitado o processamento do gestor sempre que o dispositivo estivesse ligado a uma tomada eléctrica. Seria possível minimizar o processamento do servidor, libertando recursos e reduzindo os custos.

Para tornar segura a comunicação entre o servidor aplicacional e o dispositivo móvel, seria interessante desenvolver um mecanismo que permita utilizar cifras sobre os comandos trocados entre os diversos intervenientes. Utilizar comunicações cifradas exige um maior processamento, que pode ser traduzido num aumento ligeiro dos consumos do dispositivo.

## Conclusão

---

O estudo apresentado focou-se sobre o dispositivo da *HTC* com o sistema operativo *Microsoft Windows Mobile*. Seria útil analisar o desempenho do modelo em diferentes fabricantes de dispositivos e sistemas operativos móveis. Assim seria possível compreender o desempenho deste modelo noutros dispositivos móveis.

## **Bibliografia**

---

- APPLE INC. (2009). "iPhone OS Technology Overview." 48.
- Adolph, M. (2009). "Mobile Applications." (Jul, 2009): 5.
- ANACOM (2009). Situação das Comunicações, ICP-Autoridade Nacional de Comunicações.
- Andreas, C. (2006) "Mobile Operating Systems: The New Generation."
- Andreas Constantinou, P. D. V. L. (2006). "Mobile Operating Sstems: The New Generation." 56.
- Bernardes, S., R. Rijo, et al. (2010). Network mobility and mobile applications development. Handbook of Research on Mobility and Computing, Cunha, Maria e Moreira, Fernando
- Brandizzi, A., P. Santos, et al. (2006). Internet Protocol Version 6 - IPv6, Universidade de Brasília: 21.
- BroadCom (2006). 802.11n: Next-Generation Wireless LAN Technology.
- Burnette, M. W. (2002) "Forensic Examination of a RIM (BlackBerry) Wireless Device."
- Corporation, M. (2007). "An Overview of Microsoft Visual Studio 2008."
- Dean Jezard, J. M., David Holding (2008). "Google Android." (Oct, 2008): 8.
- Faísca, J. (2009). Tecnologia de Comunicações Móvel: Caso OniWay.
- Forum, W. A. P. (2000) "Wireless Application Protocol White Paper."
- Freitas, V. (2008). Protocolo WAP. Mato Grosso, [http://www.vitorgga.com/listas/arquivos/24/protocolo\\_wap.doc](http://www.vitorgga.com/listas/arquivos/24/protocolo_wap.doc): 1-10.
- Gabriele, K. e K. Ismail (2009). Mobile Computing: 102.
- Gabriele Kotsis, I. K. (2008/09). "Mobile Computing." 102.
- Haartsen, J. (1998). "Bluetooth - The Universal Radio Interface for ad hoc, wireless connectivity."
- Harry, M. (2009). Reviews & Ranking. PCWorld: 54-58.
- Jakl, A. (2007). Smartphones: 30.
- Jaskl, A. (2009). Mobile Operating Systems.
- Junior, C. e A. Fernandes (2006) "Análise das tendências tecnológicas para a computação móvel aplicada à área da saúde." 1-6.
- Karim, M. R. e S. Mohsen (2002). W-CDMA and cdma2000 for 3G Mobile Networks, McGraw-Hill TELECOM.
- Loureiro, A. (2003). Introdução à Computação Móvel.
- Mateus, G. e A. Loureiro (2004). Introdução à Computação Móvel. Second: 120.
- Microsoft (2007). Windows Mobile, Reference Guide, Microsoft Corporation: 61.

## Bibliografia

---

- Mishra, A. (2007). Advanced Cellular Network Planning and Optimisation. 2G/3.5G/3G... Evolution to 4G. England, John Wiley & Sons Ltd: 544.
- Nakahati, Y., F. Oliveira, et al. (2006). Padrões para redes sem fio (IEEE 802.11x WLAN). Brasil, Universidade de Brasília.
- Pedro, A. (2003). "Desenvolvimento de Aplicações Móveis com Tecnologia Microsoft." 13.
- Piske, O. (2006). Rapid Application Development. F. Seidel.
- Ramabhadran, A. "Forensic Investigation Process Model For Windows Mobile." 16.
- Ribeiro, D. (2006). Estudo de Interface Humano-Máquina em Dispositivos Móveis.
- Saif, U. "Opportunistic File-associations for Mobile Operating Systems." 6.
- Santana, R. (2008). Computação móvel histórico da evolução. São Paulo, <http://grenoble.ime.usp.br/~gold/cursos/2008/movel/mono/HistoricoComputacaoMovel.pdf>.
- Santos, J. (2004). Norma 802.11. Gaia, Instituto Superior Politécnico Gaya.
- Soukoreff, I. S. M. a. R. W. (2002). "Text Entry for Mobile Computing: Models and Methods, Theory and Practice." 17: 53.
- Tavares, P. e S. Ferreira (2009). M-Commerce e M-Business: um panorama da Computação Móvel e suas perspectivas. Relatórios Técnicos do Departamento de Informática Aplicada da UNIRIO.
- Timsater, M. (2009). The Battle of the platforms: What does it mean for operators?: 52-53.
- Trevett, N. (2008). "An Open Standard for Mobile Application Portability." (Feb, 2008): 16.

## ***Apêndices***

---



# **Gestor aplicativo**



## Apêndices – Gestor aplicacional

A aplicação gestora foi implementada segundo o modelo proposto para a gestão de aplicações móveis assistida por eventos. Foi concebida para simplificar o processo de validação das aplicações de teste e facultar mecanismos que facilitem os testes apresentados no capítulo “Avaliação”.

Na Figura 23 é possível observar o ecrã principal da aplicação, assim como, as suas funcionalidades.

Quando a aplicação é executada pela primeira vez é gerado um ficheiro *XML* com as configurações por omissão da aplicação. Este ficheiro pode ser editado de forma a adaptar-se ao cenário de teste pretendido.

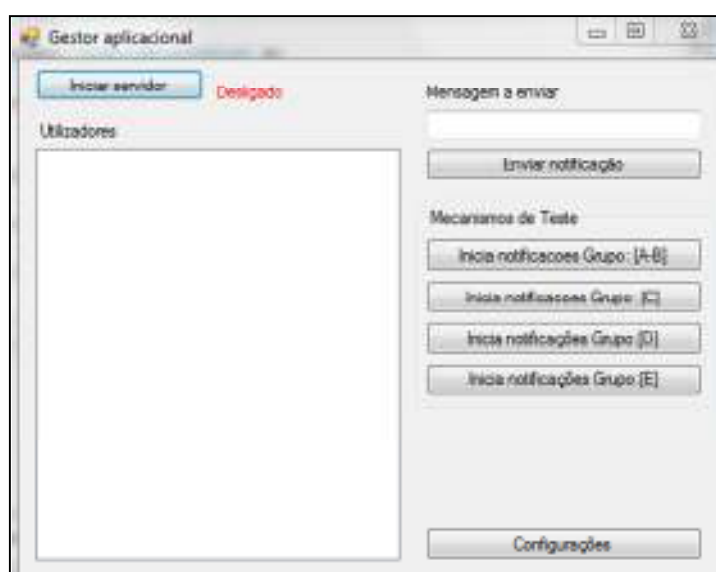


Figura 23 Ecrã principal do gestor de aplicações móveis assistido por eventos.

As configurações armazenadas no ficheiro *XML* são o endereço *IP* e o porto, onde o gestor irá ser executado. Pode ser consultado na Tabela 14 um exemplo do referido ficheiro.

```
<?xml version="1.0" encoding="utf-8" ?>

<confGestorAplicacional>

  <IPGestor xmlns="192.168.1.65"></IPGestor>

  <PortoGestor xmlns="8006"></PortoGestor>

</confGestorAplicacional>
```

Tabela 14 Exemplo de um ficheiro XML de configuração do gestor aplicacional.

Sempre que surge a necessidade de reformular os parâmetros do gestor, o utilizador deve pressionar o botão “Configurações”. Esta operação apresenta um novo ecrã de configuração semelhante ao apresentado na Figura 24.

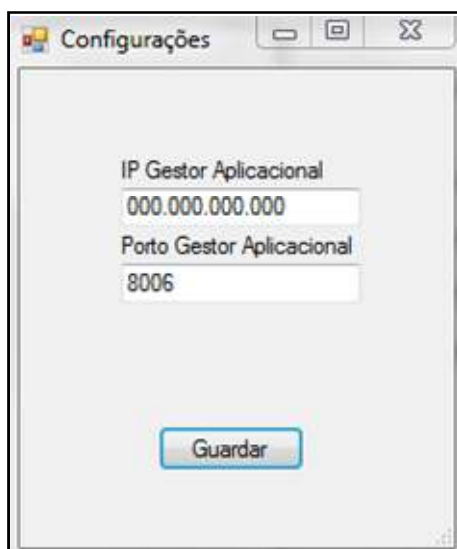


Figura 24 Ecrã de configurações do gestor de aplicacional.

Após configurado, é então possível iniciar o serviço pressionando a opção “Iniciar servidor”, que se encontra no ecrã principal. No caso de sucesso é apresentado no canto superior esquerdo a mensagem “Ligado”. Em caso de insucesso é apresentado a mensagem “Desligado” (ver Figura 25).



Figura 25 Mensagens informativas do estado da execução do gestor.

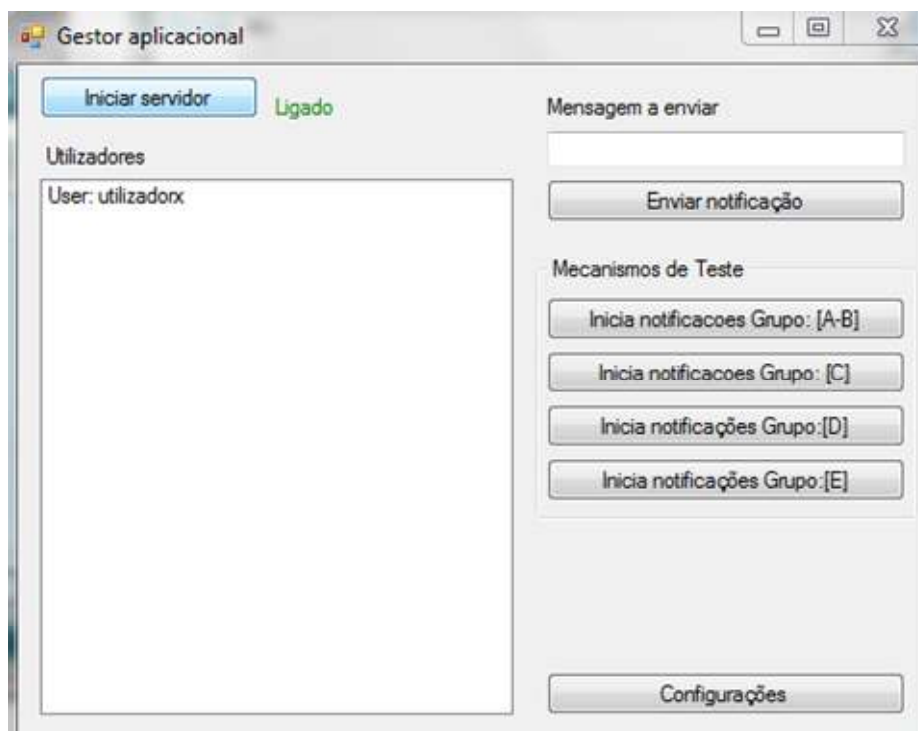


Figura 26 Ecrã principal do gestor.

Ao iniciar o serviço, a aplicação fica a aguardar a chegada de pedidos de ligação vindas das aplicações móveis que pretendem usufruir dos seus serviços. Sempre que um pedido de ligação é aceite, é apresentado o nome do utilizador na lista de utilizadores (ver Figura 26).

### **Mecanismos de testes**

Para tornar simples a validação e avaliação do modelo foi implementado um mecanismo de validação e três de avaliação.

O mecanismo de validação permite ao utilizador criar e enviar para a aplicação móvel uma mensagem segundo o protocolo adoptado e apresentado no capítulo Modelo de notificações por eventos. Assim, é possível validar o comportamento das aplicações móveis mediante as mensagens enviadas através do gestor.

Os restantes mecanismos permitem avaliar o desempenho dos diferentes cenários apresentados no capítulo de “Avaliação”. A cada botão está associado um grupo de testes (ver Figura 27).

## Apêndices – Gestor aplicativoal

---

Cada aplicação móvel pode ser configurada de acordo com as necessidades do utilizador. Estas configurações permitem ao gestor aplicativoal agir de forma a respeitar as regras ditadas pelo utilizador. Esta aplicação pode ser consultada no Apêndice “Aplicação de configuração do gestor”.

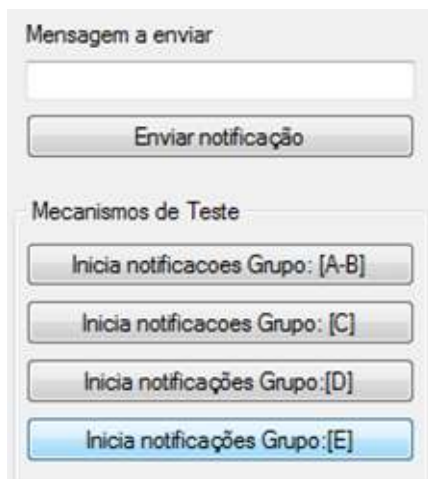


Figura 27 Mecanismos disponibilizados pelo gestor aplicativoal.

# **Aplicação de configuração do gestor**



A aplicação de configuração do gestor apresentada na Figura 28 foi desenvolvida com o intuito de possibilitar acessos provenientes de qualquer plataforma com acesso à *Internet*.



Figura 28 Aplicação de configuração do gestor de aplicações móveis assistido por eventos.

Com esta aplicação o utilizador tem a possibilidade de efectuar o seu registo, autenticar-se, configurar e alterar os parâmetros dos serviços que pretende ver monitorizados pelo gestor.

### Registar

Esta opção permite o registo de novos utilizadores. Apenas utilizadores previamente registados podem usufruir dos serviços do gestor aplicacional.

The image shows a registration form with three input fields: "Nome de utilizador:", "Palavra chave:", and "Confirmação palavra chave:". Below the fields are two buttons: "Registar" and "Limpar".

Figura 29 Registo na aplicação de configuração do gestor.

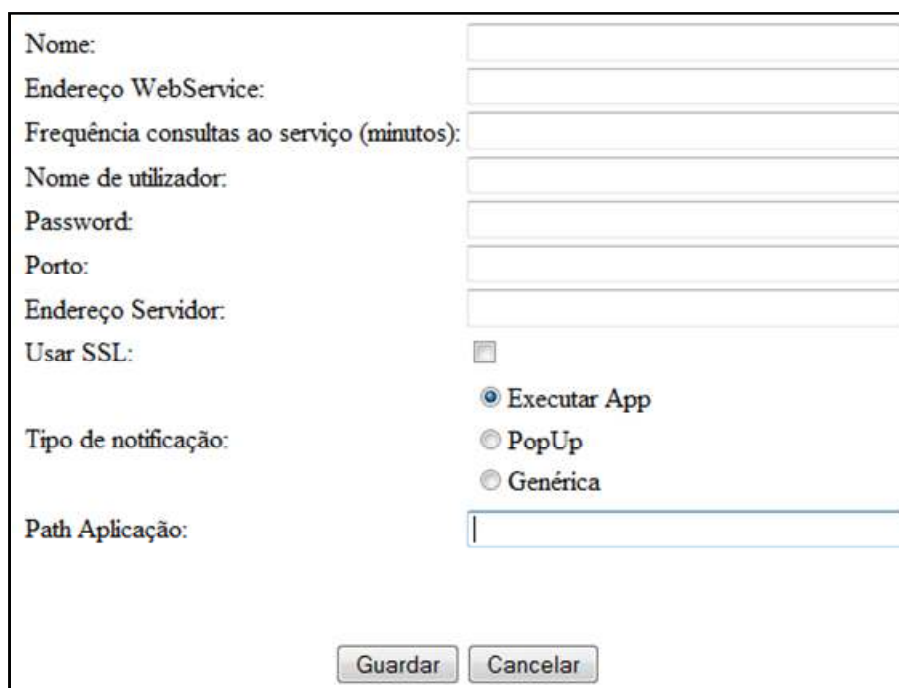
Para efectivar o registo, o utilizador necessita de preencher o formulário apresentado na Figura 29.

### Autenticação

Com a autenticação é possível garantir a integridade dos dados de cada utilizador. Assim qualquer operação associada ao gestor é precedida de uma validação de credenciais pessoais e intransmissíveis.

### Configuração

Sempre que o utilizador deseje associar um novo serviço deve preencher o formulário apresentado na Figura 30, que possibilita parametrizar o serviço a ser monitorizado.



Formulário de configuração dos serviços do gestor aplicacional. O formulário contém os seguintes campos e opções:

- Nome:
- Endereço Webservice:
- Frequência consultas ao serviço (minutos):
- Nome de utilizador:
- Password:
- Porto:
- Endereço Servidor:
- Usar SSL:
- Tipo de notificação:
  - Executar App
  - PopUp
  - Genérica
- Path Aplicação:

Botões: Guardar, Cancelar

Figura 30 Formulário de configuração dos serviços do gestor aplicacional.

### Lista de serviços

Esta funcionalidade permite ao utilizador consultar e alterar serviços previamente configurados no gestor aplicacional. Na Figura 31 Figura 31 pode ser observado um exemplo do menu de listagem de serviços.

	ativo	Identificador único	Descritivo	Endereço Webservice	Período(min)	Nome Utilizador	Porto	Endereço Servidor	Usa SSL	Utilizador
Ahora: estado:	False	00000000-0000-0000-0000-000000000000	1	2	3	4	6	7	True	a
Ahora: estado:	True	f94f44d0-99a4-473f-be55-72ccce088E58	Weather client	1	5		0		False	
Ahora: estado:	False	b90816c9-e507-4206-b5cb-c609bd1d147d	Mail Service	http://localhost	1200	ksomunazicpa@gmail.com	997	smtp.gmail.com	True	

Figura 31 Listagem dos serviços configurados no gestor aplicacional.



**Aplicação móvel genérica  
consumidora de eventos  
implementada segundo o modelo  
proposto**



## Apêndices – Aplicação móvel genérica consumidora de eventos

---

Esta aplicação está preparada para interpretar toda a diversidade de notificações previstas no protocolo do modelo de gestão de aplicações móveis assistido por eventos. É possível responder aos eventos através da execução de aplicações, por notificações via janela *pop-up* ou de simplesmente apresentar na aplicação a mensagem proveniente do gestor aplicacional.

No o ecrã principal apresentado na Figura 32, é possível ao utilizador efectuar a autenticação perante gestor, ou simplesmente configurar a aplicação acedendo ao formulário de configurações apresentado na Figura 33.



Figura 32 Aplicação móvel genérica consumidora de eventos.

Através do formulário de configuração apresentado na Figura 33 o utilizador tem a possibilidade de configurar o endereço *IP* e *Porto* onde o gestor aplicacional se encontra em funcionamento.



IP Gestor Apicacional  
192.168.1.64  
Porto Gestor Apicacional  
8006  
Guardar

Figura 33 Configuração da aplicação móvel genérica consumidora de eventos.

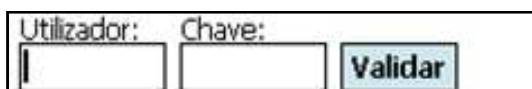
Os parâmetros apresentados no formulário de configuração são armazenados num ficheiro *XML* com a estrutura apresentada na Tabela 15.

```
<?xml version="1.0" encoding="utf-8" ?>  
<confGestorApicacional>  
  <IPGestor xmlns="192.168.1.64"></IPGestor>  
  <PortoGestor xmlns="8006"></PortoGestor>  
</confGestorApicacional>
```

Tabela 15 Ficheiro *XML* da aplicação móvel genérica consumidora de eventos.

Após ter procedido à configuração dos parâmetros de acesso ao servidor e efectuado a autenticação (ver Figura 34) com as credencias configuradas através da aplicação *Web*, é possível iniciar o serviço.

Com o início do serviço a aplicação fica a aguardar a chegada de notificações provenientes do gestor aplicacional a fim de as processar.



Utilizador: Chave:  
Validar

Figura 34 Autenticação na aplicação móvel genérica consumidora de eventos.

## Apêndices – Aplicação móvel genérica consumidora de eventos

---

As notificações provenientes do gestor aplicacional são apresentadas no ecrã principal (ver Figura 35).

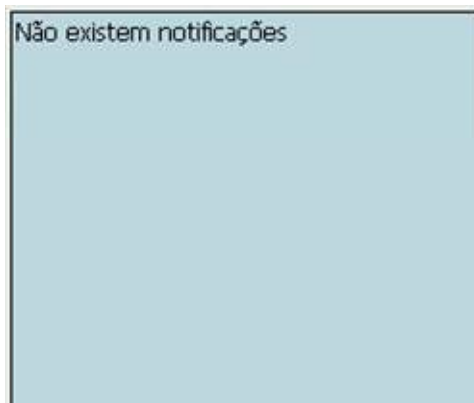


Figura 35 Notificações na aplicação móvel genérica consumidora de eventos.



**Aplicação cliente de *email*  
implementada segundo o modelo  
proposto**



O desenvolvimento desta aplicação prende-se com a necessidade de testar o evento proveniente do gestor aplicacional que notifica a aplicação móvel sobre a necessidade de executar a aplicação de *email*.

É uma aplicação simples, que ao ser invocada consulta a conta de *email* exibindo as mensagens existentes na respectiva caixa de correio. Após 30 segundos a aplicação é encerrada automaticamente. Na Figura 36 pode ser observado o ecrã principal da aplicação.



Figura 36 Aplicação de *email* móvel implementando o modelo proposto.

Para consultar as mensagens de *email* existentes na caixa de correio do utilizador é essencial definir os parâmetros de acesso ao servidor. Estes parâmetros são configuráveis através de um ficheiro *XML* que acompanha a aplicação e que apresenta a estrutura apresentada na Tabela 16.

```
<?xml version="1.0" encoding="utf-8" ?>

<conf>

  <UsernameServidorEmail
xmlns="computacaomovel@gmail.com"></UsernameServidorEmail>

  <PsswordServidorEmail
xmlns="testeteste"></PsswordServidorEmail>

  <IpServidorEmail xmlns="pop.gmail.com"></IpServidorEmail>

  <PortoServidorEmail xmlns="995"></PortoServidorEmail>

  <UsaSSL xmlns="True"></UsaSSL>

  <WebService
xmlns="http://localhost:49269/WsMail/Service.asmx"></WebService>

</conf>
```

Tabela 16 Ficheiro *XML* da aplicação móvel de *email* implementando o modelo proposto.

# **Aplicação de *email* implementada segundo o modelo actual**



A aplicação ilustrada na Figura 37 é uma versão simplificada dos tradicionais clientes de *email* utilizados em dispositivos móveis.

A aplicação implementa um sistema automático de verificação periódica de novas mensagens na caixa correio do utilizador. Sempre que detecta uma nova mensagem a aplicação apresenta-a no ecrã.



Figura 37 Ecrã principal do cliente de *email* implementando o modelo existente.

Para configurar os parâmetros da conta de *email* e a periodicidade com que a aplicação consulta o servidor o utilizador deve preencher o formulário presente na Figura 38.

Para aceder ao formulário o utilizador deve pressionar o botão “Configurações” presente no ecrã principal da aplicação apresentada na Figura 37.



Figura 38 Ecrã de configurações do cliente de *email* implementando o modelo existente.

Na Figura 38 é apresentado um exemplo dos parâmetros que são possíveis configurar para o acesso a uma conta de *email* do *Google*. Segundo estas, a aplicação irá efectuar acessos ao servidor em períodos regulares de dez minutos. Após o preenchimento do formulário os parâmetros são armazenados num ficheiro *XML* seguindo a estrutura apresentada na Tabela 17.

A necessidade de registar os acessos efectuados ao servidor de *email* num ficheiro sem influenciar o desempenho do dispositivo móvel, influenciou a decisão da consulta de *emails* ser efectuada por um *WebService* responsável por registar em ficheiro os pedidos recebidos.

```
<?xml version="1.0" encoding="utf-8" ?>

<conf>

  <PopServidorEmail xmlns="pop.gmail.com"></ PopServidorEmail>

  <PortoServidorEmail xmlns="995"></ PortoServidorEmail>

  <UsernameServidorEmail
xmlns="computacaomovel@gmail.com"></UsernameServidorEmail>

  <PsswordServidorEmail
xmlns="testeteste"></PsswordServidorEmail>

  <IpServidorEmail xmlns="pop.gmail.com"></IpServidorEmail>

  <PortoServidorEmail xmlns="995"></PortoServidorEmail>

  <UsaSSL xmlns="True"></UsaSSL>

  <WebService
xmlns="http://192.168.1.65:9090/WsMail/Service.asmx"></WebService
>

  <PeriodoConsultas xmlns="testeteste"></ PeriodoConsultas >

</conf>
```

Tabela 17 Ficheiro *XML* da aplicação móvel de *email* implementando o modelo existente.

Para iniciar o processo de consulta de mensagens de *email*, o utilizador deverá pressionar o botão “Iniciar” apresentado no ecrã principal da aplicação como apresentado na Figura 37.



**Aplicação meteorológica  
implementada segundo o modelo  
existente**



A aplicação meteorológica apresentada na Figura 39 funciona através de consultas regulares a um *WebService*. Estas consultas permitem obter a temperatura actual de uma determinada cidade apresentando-o no ecrã principal da aplicação.

Para que a aplicação possa ser utilizada é essencial configurar os parâmetros apresentados na Figura 40.



Figura 39 Aplicação meteorológica implementando o actual modelo existente.

Visto ser um protótipo, não foi implementado a funcionalidade que permite alterar o serviço a utilizar. Assim a aplicação encontra-se limitada à utilização do *WebService* “<http://www.websvcex.net/globalweather.asmx>” que se encontra definido por omissão.

Este *WebService* disponibiliza as condições climatéricas das principais cidades mundiais, o que o torna extremamente útil e personalizável.

Para o correcto funcionamento da aplicação é essencial identificar a cidade e o país do qual se pretende obter as informações climatéricas. A identificação dos parâmetros é efectuada através do formulário apresentado no ecrã de configurações ilustrado na Figura 40



Figura 40 Aplicação meteorológica implementando o modelo existente.

Após o preenchimento do formulário as configurações são guardadas num ficheiro *XML* com a estrutura apresentada na Tabela 18.

```
<?xml version="1.0" encoding="utf-8" ?>
<conf>
  <WebService
xmlns="http://www.webservices.net/globalweather.asmx"></
WebService >
  <Cidade xmlns="Lisboa"></ Cidade>
  <País xmlns="Portugal"></País>
  <PeriodoConsultas xmlns="10"></ PeriodoConsultas >
</conf>
```

Tabela 18 Ficheiro *XML* da aplicação de consulta das condições meteorológicas.

Após definidos os parâmetros de configuração é possível iniciar a aplicação através do botão “Iniciar” presente no ecrã principal.

## ***Anexos***

---



**Artigo publicado no *Handbook of  
Research on Mobility and Computing***



# Network mobility and mobile applications development

*Please do not include your names and affiliations, to allow a double blind review process*

## **ABSTRACT**

The use of mobile devices with possible connection to the Internet is increasing tremendously. This mobility poses new challenges at various levels, including hardware, network services, and the development of applications. The user searches small and lightweight devices, easy to use, and with vast autonomy in terms of energy. She/He seeks also to connect the Internet “every time, everywhere”, possibly using different access technologies. Given the interface limitations, and processing capabilities of small mobile devices, the software and the operating system used must be necessarily adapted. This chapter overviews the mobility area, provides deep insight in the field, and presents the main existing problems. Mobility and the development of mobile applications are closed related. The advances in network mobility lead to different approaches in the mobile applications development. The chapter proposes a model for developing mobile applications, based on our research.

## **INTRODUCTION**

In the last three years, there has been an impressive increase in multimedia content demand, stimulated by the increase of user-created video and Internet Protocol Television (IPTV) adoption. Video centric applications like Live Video, Video on Demand (VoD), Video Gaming, Conferencing and Surveillance, are becoming increasingly popular among users in general, and mobile users in particular, holding laptop computers or mobile handset devices. These applications stumble on a set of limitations in current networks. Delivering high quality streaming and interactive multimedia content with diverse QoS requirements, over a diverse set of access technologies (wired or wireless), launches new challenges, often specific to the underlying access technology, which may change under mobility. Furthermore, supporting these applications demands application specific techniques that dynamically adapt to the state of the network and, in the case of mobility, that adapt to the new access networks.

It is crucial to multimedia applications to provision QoS in these networks and to provide real-time assessment to multimedia QoS. Multimedia applications can be classified into three key areas: communications, video on demand, and live streaming. Each of these areas requires unique end-to-end treatment in order to ensure high-quality multimedia delivery to the end user.

Mobile multimedia delivery over diverse network technologies poses many challenges. However it is creating opportunities too. Mobile multimedia is gaining momentum as a revenue-generating opportunity. The software development environments for the mobile devices represent also a challenging issue for the mobility. It is important to identify and characterize the existing platforms to make the right development decisions in order to increase the devices autonomy.

The proposed chapter collects the most recent developments on the involved technologies (multimedia applications, QoS, multicast and IP mobility) and state how they could interact and be put together. It presents open research topics in this area. It also characterizes the existing platforms in the mobile devices and proposes, based in our research, a model for the mobile applications.

## **BACKGROUND**

This section introduces the main concepts about mobility and mobile operating systems.

The following section presents a brief definition of concepts and related work done in the mobility area. It also makes an overview of the mobile operating systems. It finishes with some conclusions and trends.

### **Transmission of multimedia content**

The importance of interactive audio and visual contents is increasing. Interest in multimedia applications is growing. But multimedia applications pose new demands to devices, to networks, and to communication protocols. When video and audio are being used, delays and jitter are not welcome. New protocols came to light in order to make possible multimedia transmission with the necessary quality. An example of such protocols is the RTP protocol family. RTP (Schulzrinne, Casner, Frederick, & Jacobson, 2003) provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data, over multicast or unicast network services. The data transport is augmented by the Real-time Control Protocol (RTCP) to allow monitoring of the data delivery in a manner scalable to large multicast networks, and to provide minimal control and identification functionality. The Secure Real-time Transport Protocol (SRTP) (Baugher, McGrew, Naslund, Carrara, & Norrman, 2004) is a profile of RTP, which can provide confidentiality, message authentication, and replay protection to the RTP traffic. The Real Time Streaming Protocol (RTSP) (Schulzrinne, Rao, & Lanphier, 1998) is an application-level protocol for control over the delivery of data with real-time properties.

In computer networks, the number of applications that need parameter guarantees like bandwidth, delay, jitter, and packet loss rate, is growing. Thus, it is necessary to use Quality of Service (QoS) in order to assure those parameters.

Video and voice real-time applications are being more used each day, posing new challenges to traffic management and congestion control.

To have QoS in video-broadcast means that the network is configured to assure certain traffic parameters, in order to allow the video to arrive with the requested quality. In this way, when video

is sent with a certain QoS, the receiver should smoothly obtain the traffic, independently of any network congestion.

The MPEG-4 [MPEG-4] standard specifies compression of audio-visual data into, for example an audio or video elementary stream. (der, Mackie, Swaminathan, Singer, & Gentric, 2003) defines a general and configurable payload structure to transport MPEG-4 elementary streams, in particular MPEG-4 audio (including speech) streams, MPEG-4 video streams and also MPEG-4 systems streams. Some types of MPEG-4 elementary streams include crucial information whose loss cannot be tolerated. However, RTP does not provide reliable transmission, so receipt of that crucial information is not assured. The standard specifies how stream state is conveyed so that the receiver can detect the loss of crucial information.

Wireless connections are quite different from the wired connections, mainly because of bandwidth limitations and mobile terminals restrictions. This way streaming in wireless networks poses a great challenge in achieving a good connection between sender and receiver and in choosing the most suitable compression method. MPEG-4 is becoming the main compression standard in audio and video transmission with mobile terminals, such as Personal Digital Assistants (PDAs). Video technologies are getting more and more important in wireless networks. The 3G mobile phone networks have been designed to take video streaming to the mobile equipments. The challenge is even bigger when real time transmission systems are involved. One solution that has been taken is to build transmission systems that are capable to adapt to de current connection type and to the current medium conditions.

## **IP Mobility**

IP mobility allows for a user to move between networks without disturbing its communications. The act of movement between networks has an implicit base process, the handover. This process occurs during the transition between networks and usually leads to the loss of several packets during the mobile terminal movement. This fact affects communications during the transition, with special emphasis in real-time communications. Nevertheless there are some mechanisms that can speed this process, minimizing the number of lost packets. An example of these mechanisms in IPv6 networks is Fast Handover for Mobile IPv6 (FMIPv6).

The success and increasing use of mobile IP devices in wireless networks (e.g., laptops, PDAs and cellular phones) allows a growing deployment of Personal Area Networks (PAN). This type of networks enables users to stay connected to the Internet, using different IP devices, without loss of service. As an example, a user with a PDA, a cellular phone, and a laptop can be “online” continuously using all devices, having each one its own Internet access and IP address.

In IPv6 networks it is possible to provide mobility to all the devices in these mobile networks using the MIPv6 protocol (Johnson, Perkins, & Arkko, 2004) (Soliman, 2004). However, the fact that all the network devices are mobile implies that each one must support IPv6 mobility, using MIPv6 protocol. This method is not efficient since all the devices have to implement mobility functions, even those that do not have enough resources to run such protocol. Moreover, usually it is less computationally costly to have several devices sharing the same Internet access and, as for instance in Vehicular Ad-Hoc Networks (VANETs), several devices may be continuously moving all together.

In order to solve such problems, the NEMO protocol (Devarapalli, Wakikawa, Petrescu, & Thubert, 2005) extends MIPv6 functionalities and provides mechanisms for network mobility management, enabling the networks to attach to different points in the Internet without loss of its current connections. In this protocol the mobile nodes connections to the Internet are done through a unique device, a mobile device, avoiding the need to have all the mobile devices supporting MIPv6.

The basic functioning of the network mobility management protocol NEMO uses a bidirectional tunnel between the MR and the HA. This tunnel is created when the MR moves and informs its HA of its current attachment point. The protocol does not describe any route optimization solution between the Mobile Network Nodes (MMN) and the CNs. Therefore, all the traffic between the MN and the HA is passed through the tunnel established between the MR and the HA. Moreover, this protocol does not present multi-homing solutions.

### **Multicast and security**

The deployment of IPv6 multicast services relies on Multicast Listener Discovery Protocol (MLD) and on Protocol Independent Multicast (PIM) for routing. MLD is identical to Internet Group Management Protocol (IGMP) in IPv4. ASM (Any Source Multicast) and SSM (Single Source Multicast) service models operate almost the same as in IPv4. Both have the same benefits and disadvantages as in IPv4. Nevertheless, the larger address space and the scoped address architecture provide major benefits for multicast IPv6 (Asadullah, Ahmed, Popoviciu, Savola, & Palet, 2007). Through (Haberman & Thaler, 2002), the large address space provides the means to assign global multicast group addresses to organizations or users that were assigned unicast prefixes. It is a significant improvement with respect to the IPv4 GLOP mechanism (Meyer & Lothberg, 2001). This facilitates the deployment of multicast services.

Within the context of computer networking, security is the science of protecting information and devices on a network from being misused by unauthorized users. This includes disclosure, modification, and destruction of information as well as unauthorized use of network resources, such as denying services to legitimate users.

The main aspects of security are integrity checking, confidentiality, data origin authentication, non repudiation, entity authentication and authorization (Soliman, 2004).

### **Mobile operating systems**

An operating system (OS) is a set of programs making the link between the hardware and the software. They manage the processor, the file system, the memory and the peripherals (Gabriele Kotsis 2008/09).

Today when someone buys a mobile phone must, as in any other computer, select the most appropriate OS, choosing always a liable, safe and with a good performance.

Most of these OS developed for the mobile devices adopt a layered architecture. Some of the common layers are: Kernel, Middleware, Application Execution Environment, User Interface e Application Suite (Andreas Constantinou 2006).

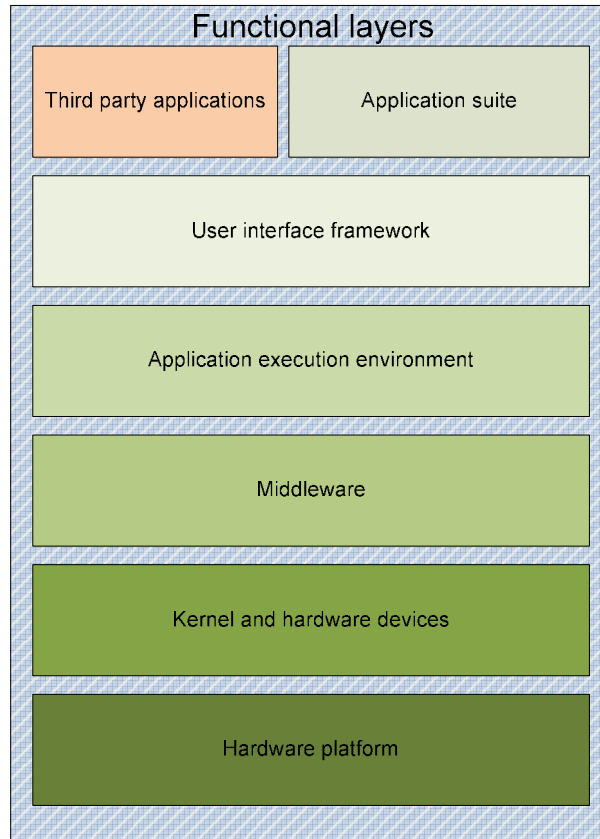


Figure18. Layers of a Mobile Operating System.

Adapted Kotsis (Andreas Constantinou 2006)

The Kernel is the core of the operating system where, among others, we can find the hardware, memory and file system drivers. It is also the responsible for the proper process management.

The Middleware is a transparent layer making the link to the peripherals through software libraries.

The Application Execution Environment offers application programming interfaces (APIs) for the development of new applications.

The User Interface layer furnishes the graphical environment of each system.

The Application Suite contains the majority of the applications available in the system. Here we can find browsers, configuration menus, the calendar, games, among others.

There are several mobile OS manufacturers. This chapter focuses in the most used mobile OSs during 2008, namely the Symbian, the Rim, the Windows Mobile, and iPhone OS. It is also considered Android, the new Google's system (Jaskl 2009).

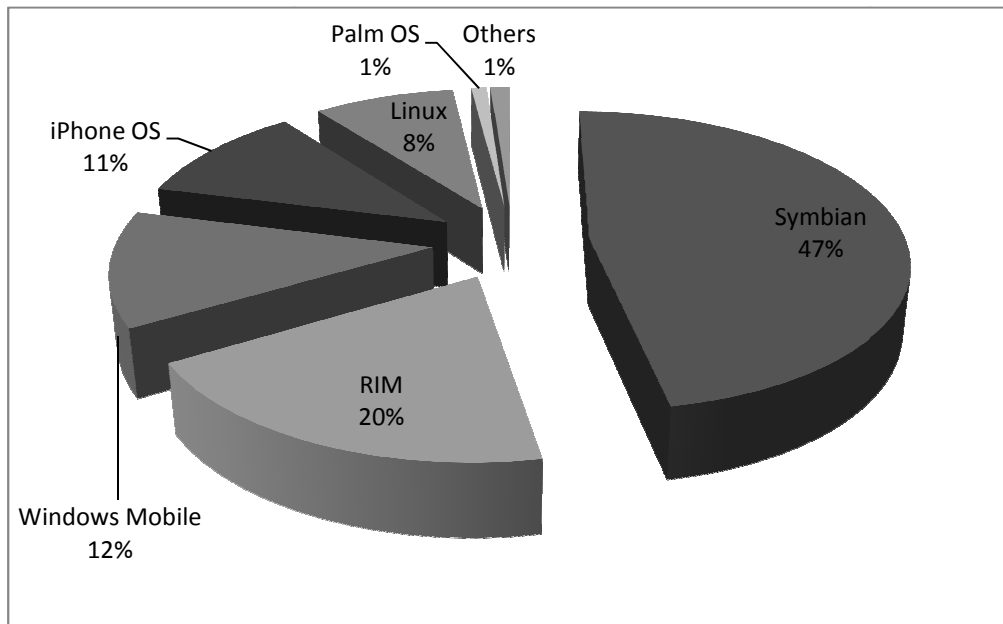


Figure 2. Operating Systems used on SmartPhones in 2008.

Adapted from Jaskl (Jaskl 2009)

Before we make a brief overview of these systems, we will make a short overview about the Wireless Private Network Technologies (WPAN) widely used in the mobile systems.

A wireless personal area network (WPAN) is a low-range wireless network which covers an area of only a few dozen metres. This sort of network is generally used for linking peripheral devices (like printers, cellphones, and home appliances) or a personal assistant (PDA) to a computer, or just two nearby computers, without using a hard-wired connection. There are several kinds of technology used for WPANs, namely Bluetooth, HomeRF, ZigBee, and IrDA.

The main WPAN technology is Bluetooth, launched by Ericsson in 1994, which offers a maximum throughput of 1 Mbps over a maximum range of about thirty metres. Bluetooth, also known as IEEE 802.15.1, has the advantage of being very energy-efficient, which makes it particularly well-suited to use in small devices.

HomeRF (for Home Radio Frequency), launched in 1998 by HomeRF Working Group (which includes the manufacturers Compaq, HP, Intel, Siemens, Motorola and Microsoft, among others) has a maximum throughput of 10 Mbps with a range of about 50 to 100 metres without an amplifier. The HomeRF standard, despite Intel's support, was abandoned in January 2003, largely because processor manufacturers had started to support on-board Wi-Fi (via Centrino technology, which included a microprocessor and a Wi-Fi adapter on a single component).

The technology ZigBee (also known as IEEE 802.15.4) can be used to connect devices wirelessly at a very low cost and with little energy consumption, which makes it particularly well-suited for being directly integrated into small electronic appliances (like home appliances, stereos, and toys). ZigBee, which operates on the frequency band of 2.4 GHz and on 16 channels, can reach transfer speeds of up to 250 Kbps with a maximum range of about 100 metres.

Finally, infrared connections can be used to create wireless connections over a few metres, with speeds that can reach a few megabits per second. This technology is widely used in home electronics (like remote controls), but light waves can interfere with the signal. IrDA (Infrared Data Association), formed in 1995, has more than 150 members.

## **Symbian**

The Symbian Foundation is a non-profit organisation that started its activity in 1998, supported by a set of manufacturers with the goal of licensing a software platform (which is based on Symbian OS) for mobile devices. These manufacturers are led by Nokia with 47,9% and followed by Ericsson with 15,6%, Sony Ericsson with 13,1%, Panasonic with 10,5%, Samsung with 4,5%, and Siemens with 8,4%.

The Symbian OS is the most commercialized system and it is present in more than 80 million of devices spread among more than 100 models. Nokia is the manufacturer with more devices with Symbian OS (Andreas Constantinou 2006).

In 2009 Symbian is a recognized operating system with one of the most advanced kernels for mobile devices. It needs an ARM9<sup>i</sup> processor. The other technical requisites differ according to the required interfaces (Andreas Constantinou 2006).

This OS supports 2G and 3G technology, communications protocols like WAP, TCP, IPv4, IPv6. At the personal area networks (PAN) level, Symbian OS supports IrDA, Bluetooth and USB. It also provides multi-task, multi-thread and the ability to work with the different types of phones, either they be numeric, alpha-numeric or touch screen. In addition to the telephony services, Symbian OS also supports others as short message service (SMS), enhanced messaging service (EMS) and multimedia messaging service (MMS), video conference, and the capability of switching between networks.

Navigation, agenda, e-mail, fax and a word processor are some of the applications developed for this OS.

It guarantees also the confidentiality and the integrity of the information, providing compression, cryptography, and digital certificates (Gabriele Kotsis 2008/09).

## **Windows Mobile**

Windows Mobile, variant of the Windows CE (also known officially as Windows Embedded Compact), was developed for the Pocket PCs at the beginning but arose by 2002 to the HTC<sup>ii</sup> mobile phones. This OS was engineered to offer data and multimedia services. By 2006, Windows Mobile becomes available for the developers community. Many new applications started using the system, turning Windows Mobile in one of the most used systems (Andreas Constantinou 2006).

Windows Mobile presents three APIs to support the development of applications, the Win32, the MFC, and the .NET Compact Framework.

The Win32-API uses a native interface allowing the development using the C language. The MFC-API is an extension of the Win32-API and permits the use of C and C++. Finally, the .NET Compact Framework uses some of the same class libraries as the full .NET Framework and also a few libraries

designed specifically for mobile devices. The libraries are not exact copies of the .NET Framework. The ones in the .NET Compact Framework are scaled down to take up less space.

As these APIs make possible the development of more and better applications, Microsoft® will continue support the APIs.

Windows Mobile permits Bluetooth connections through the interface Winsock. It also allows 902.11x, IPv4, IPv6, VoIP, GSM and CDMA connections. (Ramabhadran)

Some of the main applications available are the Pocket Outlook (adapted version of the Outlook for Desktops), Word and Excel. It provides also Messenger, Browser and remote desktop. The remote desktop is an easy way of access to other mobile or fixed terminals.

In order to facilitate the synchronization between the mobile devices and the desktops, the Windows mobile offers the ActiveSync application.

At the multimedia level, Windows Mobile reproduces music, video and 3D applications.

Security is also a concern, so secure socket layer (SSL), Kerberos and the use of encryption algorithms are available.

## **Research In Motion**

Research In Motion® (RIM) is a Canadian designer, manufacturer and marketer of wireless solutions for the worldwide mobile communications market. Products include the BlackBerry™ wireless email solution, wireless handhelds and wireless modems. RIM is the driving force behind BlackBerry smartphones and the BlackBerry solution.

RIM provides a proprietary multi-tasking operating system (OS) for the BlackBerry, which makes heavy use of the device's specialized input devices, particularly the scroll wheel or more recently the trackball.

The Blackberry OS is quite famous for its agenda and e-mail applications. It makes possible the content actualization in real-time and still it presents good performance and an easy World Wide Web navigation.

Blackberry OS was designed for an Intel processor with 32 bits i386, 512 Kb of static random access memory (SRAM) and, depending of the model, 4 or 5 Flash RAM(Burnette 2002).

This OS has only one file, the executable of the operating system.

The BlackBerry Software Development Kit (SDK) enables the development of applications (Burnette 2002).

## **IPhone OS**

IPhone OS is the Apple proprietary OS used in the Macintosh machines. An optimized version is used in the iPhone and iPod Touch. This version is seen as a compact version of the version 10.5 (the so-called "Leopard"). The simplicity and robustness provided either in the menus navigation or in the applications' navigation are two of the main potentialities of the OS. IPhone OS is also equipped with good quality multimedia software, including games, music and video players (2009). It has also a good set of tools including imaging editing and word processor.

Some months after the delivery of the SDK to the programmers' community, a countless number of free or low cost applications were available at AppleStore. These applications are easy to get and install.

The SDK only works in Apple's OSs. This may be a disadvantage comparing with other SDKs.

## **Google Android**

The Android is the open source mobile OS launched by Google (Dean Jezard 2008). It is intuitive, user-friendly and graphically similar to the iPhone and Blackberry. Being open source, the Android applications may be cheaper and the spread of the Android possibly will increase.

The Kernel is based on the Linux v 2.6 and supports 2G, 3G, Wifi, IPv4, and IPv6.

At the multimedia level, Android works with OpenGL<sup>iii</sup> and several images, audio, and video formats.

The persistence is assured with the support of the SQLite<sup>iv</sup>. Regarding security, the Android uses SSL and encryption algorithms.

## **Considerations and trends**

Currently mobile operating systems and personal computer operating systems share some common characteristics, however, mobile operating systems are limited as far as processing and storage capabilities go. Despite the fact that the vast majority of these systems support flash memories, they are inevitably limited when it comes to input resources and battery life.

These limitations are clearly perceptible when it comes to applications designed for mobile systems, since they are usually presented with a subset of their personal computers counterparts' functionalities. For instance, when we look at an application such as Microsoft Office, it can be observed that some functionalities, namely tables, styles, heading and footers are unavailable in the mobile version of the suite (Saif).

Another major difference presents itself when graphical, sound and network cards are compared. However good they may be, their performance does not allow for high definition streaming game play (Saif).

In the period of 2006 to 2009, software flexibility, focus on middleware development and open-source systems were software market's main trends. These systems are compatible with different kinds of hardware, allow interface customization and network and multimedia services, similarly to personal computers' operating systems, granting them a high potential (Andreas Constantinou 2006).

Multimedia and Internet connection through wireless technology are some of these system's standard characteristics, varying in performance, security and number of consumers (Trevett 2008).

Open-source was introduced in mobile platforms through Symbian and Google Android. Open-source allows programmers to easily add value to the product, therefore contributing to the deflation in the hardware price.

The iPhone is a partially open-source development platform but the installation of the SDK requires an Apple OS. Furthermore, Apple determines whether a certain product will or will not be published in Apple Store, since Apple verifies if the software purposed for publishing concurs with any Apple's existing software. This is a limitation that delays the growing of iPhone applications.

Despite this fact, we are assisting to an exponential growth of mobile applications (Pedro 2003) because of the SDKs available. These SDKs allow the use of programming languages like Java for the Blackberry and Android, C and C++ for Windows Mobile and Object C for the Apple (Adolph 2009).

Which operating system will dominate the market is an open question, however, Apple OS, Android and Symbian are likely candidates, and each owns a fare share of the market.

By giving access to applications such as Safari and iTunes which are currently used in personal computers, Apple eases the adoption of its products by the user.

Google's website indicates that Android can run in smart PC's. This becomes an asset, as it enables access by the user to the same kind of interfaces whichever the underlying hardware may be.

Mobile Operating System's future is highly dependent on server-side development (Timsater 2009), since one can only overcome the hardware limitations imposed by these handhelds if some of the processing is transferred to servers.

## **DEVELOPING MOBILE SOLUTIONS CONSIDERING THE DEVICE AUTONOMY**

The energy autonomy of a device is one of its most important characteristics. The higher the energy autonomy, the greater the user satisfaction. The batteries for the mobile devices are heavy and with low autonomy.

The energy autonomy represents one of the most difficult obstacles in the mobile computation. This problem has always been transferred for the hardware side along the time. It is considered that hardware manufacturers must develop components that reduce the energy consumption and produce batteries with an increasing higher capacity (Loureiro). Our research looks for solutions for the energy problem but starting from the software development side.

Due to the increased quality of the communications and the advances in the network mobility, it is possible to consider different approaches in the mobile application development.

The following sections suggest a model for developing mobile applications and the results of a set of tests to prove the feasibility of this proposal. Our research is still the beginning so these are just initial results.

## Event model for developing applications for mobile devices

In the general paradigm of the mobile OSs, an application is loaded into the memory, when the user explicitly selects and activates it.

Until the user closes the application it is loaded in the memory and increases the energy consumption.

Let us consider the e-mail application. If the user wants to see the e-mail, she/he must open (or activate) the email application. After checking the e-mail, if she/he wants to continue receiving e-mails, then the e-mail application must remain active. By remaining active, it continues to consume energy, increasing the total consumption of the system.

To improve this situation, the idea is to use an event based model to reduce the energy system consumption and increase, in this way, the availability of the mobile system. This is already used in the Web based applications allowing the use of a huge set of services based on a thin client / server architecture.

Applying this idea to the e-mail example, it would result in the use of the e-mail application only when needed. The e-mail would be deactivated all the time in the mobile device. On the server side an application is receiving the e-mail for that user. Every time an e-mail is received, a notification event is sent to the mobile device. The mobile device has a thin application that wakes up when an event is received. This thin application shows the event to the user and gives him the possibility of setting on the e-mail application.

Figure 3 presents a simplified architecture of the event model for developing applications for mobile devices.

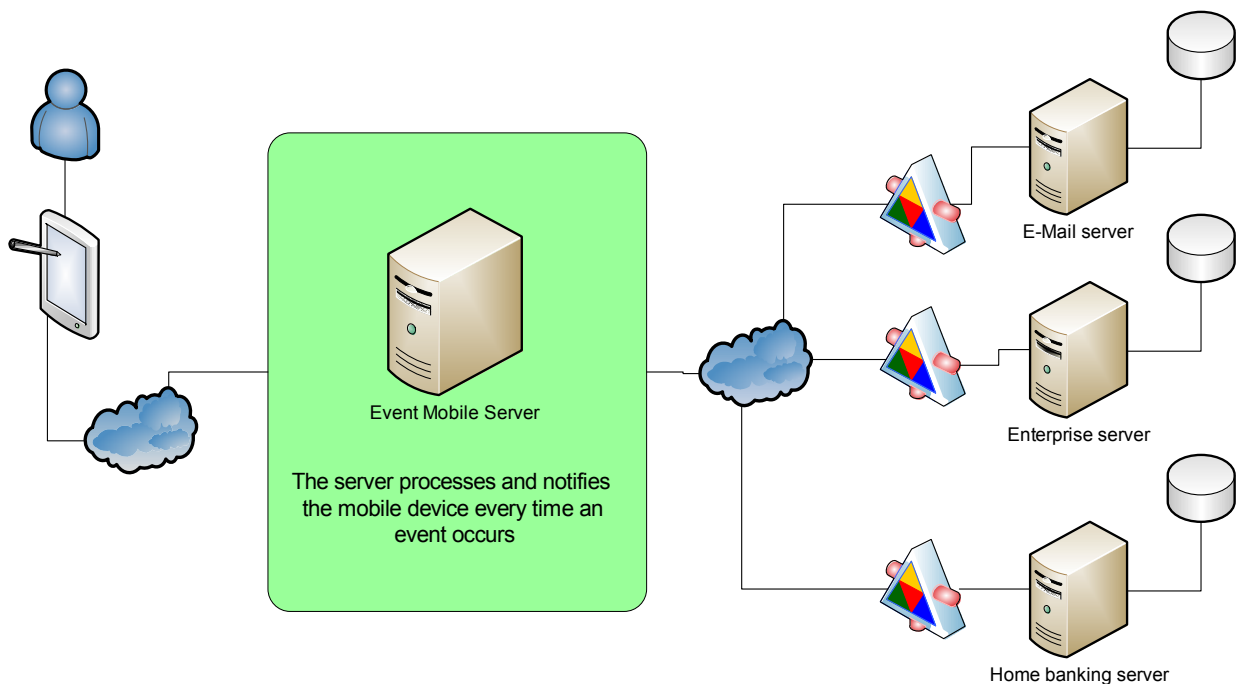


Figure 3. Event model for developing applications for mobile devices

The notification model may be extended to others services as, for example, an enterprise application or a home banking service. The Event Mobile Server receives, processes, and only then notifies the mobile application. The user is notified and decides if she/he wants to process (e.g. visualize, answer, receiving information) the event.

With this approach, the mobile device is always using the minimum necessary resources. This resource management enables the raise of the devices autonomy.

**Tests scenarios for the event model for developing applications for mobile devices**

In order to verify if the event model has a relevant impact in the autonomy of the devices, six initial tests were designed and executed. Based on these results, other tests are been prepared and constitute future work in the area. All these tests started with a device in its full energy capacity.

In the first test, the user receives an e-mail every twenty minutes. The e-mail application was always opened and working. The mobile device consumed 18% of the battery.

The second test ran in a similar way but using the event model. The user opens the e-mail application only when she/he receives the notification of the e-mail. After reading the mail message it closes the application. The mobile device only consumed 10% of the battery.

Two e-mail applications were developed for these tests. One application was developed using the event model approach (used in the second test) and the other using a fat client mail processing application (used in the first test). The figure 4 compares these two tests.

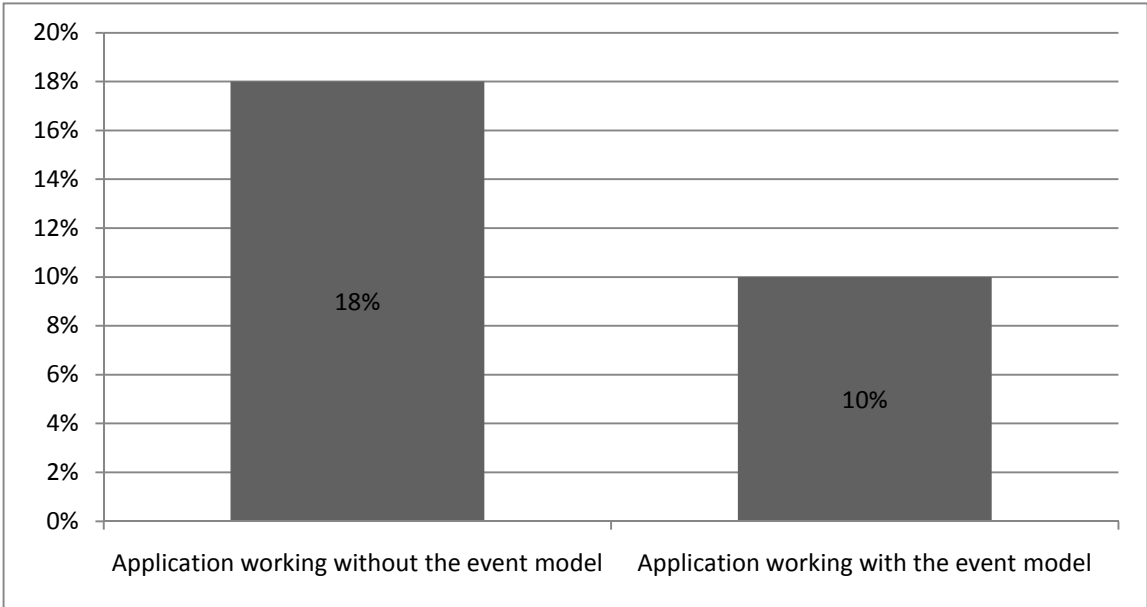


Figure 4. Percentage of battery used by two different e-mail applications, one using a fat client and other using the event model

The third test was done using an out-of-the box commercial e-mail application. The application was configured for checking the e-mail every ten minutes. An e-mail message was sent every twenty minutes. The battery lasted 492 minutes.

The fourth test used the application developed for the second test (event based e-mail application). An e-mail message was sent every twenty minutes. The battery lasted 1438 minutes.

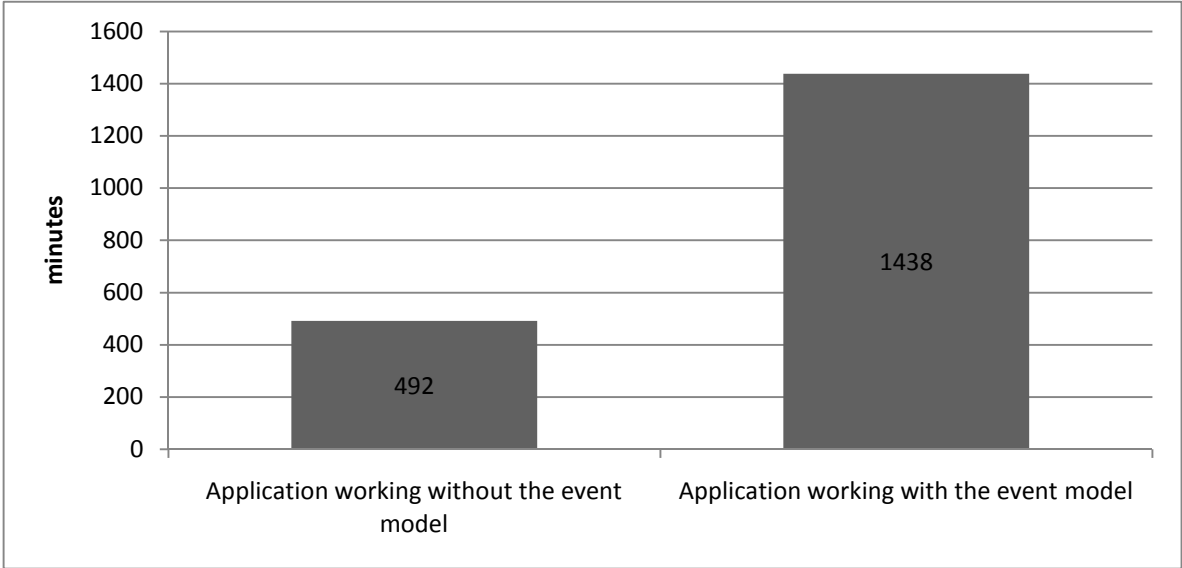
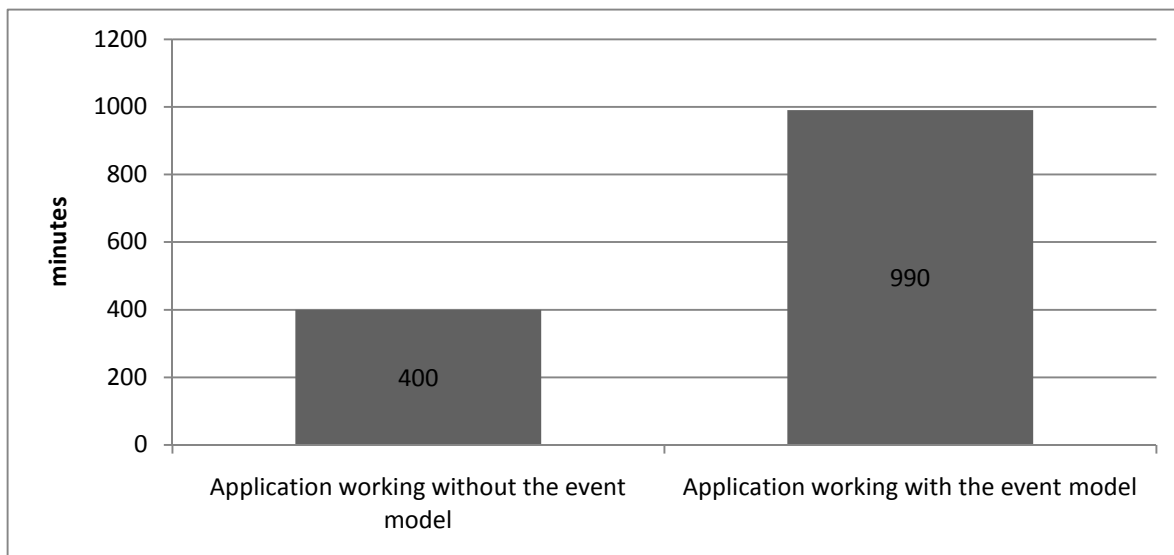


Figure 5. Battery duration using an e-mail application with the event model compared with an e-mail commercial application

In order to verify if the results would be similar when using a random email generation a fifth test was realized. The user checked the e-mail every 10 minutes using the commercial application. The e-mail delivery was controlled by an application using random number generation. The battery lasted 400 minutes.

The sixth test was similar to the previous one, but using the event model approach. The user opens the e-mail application only when she/he receives the notification of the e-mail. After reading the mail message it closes the application. The e-mail production, as before, was controlled by an application using random number generation. The battery lasted 990 minutes.



*Figure 6. Battery duration using an e-mail application with the event model compared with an e-mail commercial application, both using random e-mail generation*

These tests suggest that the event model may increase the autonomy of the devices. However more tests must be done, with different applications, operating systems and devices for a more sustained conclusion. We must find, for example, if the model works for all types of applications or if it has advantages on only some specific types.

## **FUTURE RESEARCH DIRECTIONS**

Network mobility advances lead to better communications. These improvements make possible the development of new models of building software applications. Despite the effort made in the research, many questions remain and claim a persistent effort.

Some of the major research subjects in computer communication networks and applications are undoubtedly QoS in mobile networks, security on multimedia contents access and IPv6 technologies in general. There has been a considerable amount of work in areas like Mobility, Multicast, Security, QoS and Multimedia Distribution, leading to some good solutions. But the integration of some subsets of these technologies, still poses big challenges, remaining as open research topics. When it comes to put these technologies in limited capacity mobile devices, the challenge is even greater.

The development of new approaches for mobile applications is close related with the network mobility topic. Good network mobility and communications lead to new possibilities in the way of building mobile applications. Still our research relates the development of the applications with the energy consumption there is the need to pursue the investigation. It is necessary to test the model in different types of applications, namely in the context ware ones and in those requiring enormous processing power, e.g., voice processing. It is also essential the test in multiple devices and all the operating systems. All these experiences will surely generate new questions and models.

## CONCLUSION

In our globalized world, users and organizations have a sense of need to be connected every time and everywhere. The expectation is the possibility to access the e-mail, the Internet, and all the work and leisure applications all the time. Researchers, telecommunications organizations, smart phone manufacturers, and other industry actors are discovering ways to turn the expectation a possible reality.

Some main challenges are the interface design, the mobility of the devices among networks, their energy autonomy, and the ubiquity characteristics of the applications.

This chapter focuses on the mobility and energy issues. It introduces a solid theoretical ground about the key concepts of the mobility, the existing protocols, and network mobility approaches. Mobility is one of the faces of the coin. The other important face is the energy autonomy. This aspect is, most of the times, considered from the hardware point of view. Although it is true that the hardware devices are becoming more and more powerful, it is also true that the way of developing applications influences the devices energy consumption. The chapter presents a set of experiences that show the relation between these two aspects. From this set of experiences, a model based on events is proposed. This model may improve the autonomy of the devices shifting operations from the devices to main servers. As the quality of communications and mobility increases this model may be explored.

## REFERENCES

- APPLE INC. (2009). "iPhone OS Technology Overview." 48.
- Adolph, M. (2009). "Mobile Applications." (Jul, 2009): 5.
- ANACOM (2009). Situação das Comunicações, ICP-Autoridade Nacional de Comunicações.
- Andreas, C. (2006) "Mobile Operating Systems: The New Generation."
- Andreas Constantinou, P. D. V. L. (2006). "Mobile Operating Sstems: The New Generation." 56.
- Bernardes, S., R. Rijo, et al. (2010). Network mobility and mobile applications development. Handbook of Research on Mobility and Computing, Cunha, Maria e Moreira, Fernando
- Brandizzi, A., P. Santos, et al. (2006). Internet Protocol Version 6 - IPv6, Universidade de Brasília: 21.
- BroadCom (2006). 802.11n: Next-Generation Wireless LAN Technology.
- Burnette, M. W. (2002) "Forensic Examination of a RIM (BlackBerry) Wireless Device."
- Corporation, M. (2007). "An Overview of Microsoft Visual Studio 2008."
- Dean Jezard, J. M., David Holding (2008). "Google Android." (Oct, 2008): 8.
- Faísca, J. (2009). Tecnologia de Comunicações Móvel: Caso OniWay.
- Forum, W. A. P. (2000) "Wireless Application Protocol White Paper."
- Freitas, V. (2008). Protocolo WAP. Mato Grosso, [http://www.vitorgga.com/listas/arquivos/24/protocolo\\_wap.doc](http://www.vitorgga.com/listas/arquivos/24/protocolo_wap.doc): 1-10.
- Gabriele, K. e K. Ismail (2009). Mobile Computing: 102.
- Gabriele Kotsis, I. K. (2008/09). "Mobile Computing." 102.
- Haartsen, J. (1998). "Bluetooth - The Universal Radio Interface for ad hoc, wireless connectivity."
- Harry, M. (2009). Reviews & Ranking. PCWorld: 54-58.
- Jakl, A. (2007). Smartphones: 30.
- Jaskl, A. (2009). Mobile Operating Systems.
- Junior, C. e A. Fernandes (2006) "Análise das tendências tecnológicas para a computação móvel aplicada à área da saúde." 1-6.
- Karim, M. R. e S. Mohsen (2002). W-CDMA and cdma2000 for 3G Mobile Networks, McGraw-Hill TELECOM.
- Loureiro, A. (2003). Introdução à Computação Móvel.
- Mateus, G. e A. Loureiro (2004). Introdução à Computação Móvel. Second: 120.
- Microsoft (2007). Windows Mobile, Reference Guide, Microsoft Corporation: 61.
- Mishra, A. (2007). Advanced Cellular Network Planning and Optimisation. 2G/3.5G/3G... Evolution to 4G. England, John Wiley & Sons Ltd: 544.

- Nakahati, Y., F. Oliveira, et al. (2006). Padrões para redes sem fio (IEEE 802.11x WLAN). Brasil, Universidade de Brasília.
- Pedro, A. (2003). "Desenvolvimento de Aplicações Móveis com Tecnologia Microsoft." 13.
- Piske, O. (2006). Rapid Application Development. F. Seidel.
- Ramabhadran, A. "Forensic Investigation Process Model For Windows Mobile." 16.
- Ribeiro, D. (2006). Estudo de Interface Humano-Máquina em Dispositivos Móveis.
- Saif, U. "Opportunistic File-associations for Mobile Operating Systems." 6.
- Santana, R. (2008). Computação móvel histórico da evolução. São Paulo, <http://grenoble.ime.usp.br/~gold/cursos/2008/movel/mono/HistoricoComputacaoMovel.pdf>.
- Santos, J. (2004). Norma 802.11. Gaia, Instituto Superior Politécnico Gaya.
- Soukoreff, I. S. M. a. R. W. (2002). "Text Entry for Mobile Computing: Models and Methods, Theory and Practice." 17: 53.
- Tavares, P. e S. Ferreira (2009). M-Commerce e M-Business: um panorama da Computação Móvel e suas perspectivas. Relatórios Técnicos do Departamento de Informática Aplicada da UNIRIO.
- Timsater, M. (2009). The Battle of the platforms: What does it mean for operators?: 52-53.
- Trevett, N. (2008). "An Open Standard for Mobile Application Portability." (Feb, 2008): 16.

## KEY TERMS & DEFINITIONS

### Devic

---

<sup>i</sup> The ARM is a 32-bit reduced instruction set computer (RISC) instruction set architecture (ISA) developed by ARM Limited. It was known as the Advanced RISC Machine. They were originally conceived as a processor for desktop personal computers by Acorn Computers, a market now dominated by the x86 family used by IBM PC compatible computers. But the relative simplicity of ARM processors made them suitable for low power applications. This has made them dominant in the mobile and embedded electronics market as relatively low cost and small microprocessors and microcontrollers.

<sup>ii</sup> HTC Corporation, formerly High Tech Computer Corporation, is a Taiwan-based manufacturer of primarily Microsoft Windows Mobile-based portable devices as well as several Google Android-based devices.

<sup>iii</sup> OpenGL is the environment for developing portable, interactive 2D and 3D graphics applications. Since its introduction in 1992, OpenGL has become the industry's most widely used and supported 2D and 3D graphics application programming interface (API), bringing thousands of applications to a wide variety of computer platforms. OpenGL incorporates a broad set of rendering, texture mapping, special effects, and other powerful visualization functions. Developers can leverage the power of OpenGL across all popular desktop and workstation platforms, ensuring wide application deployment.

<sup>iv</sup> SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private.