



Raquel Santos Soares

**Armazenamento de objetos digitais em dispositivos
móveis com *Android***



Raquel Santos Soares

**Armazenamento de objetos digitais em dispositivos
móveis com *Android***

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Professor Doutor Joaquim Arnaldo Carvalho Martins, Professor Catedrático do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

Dedico este trabalho à minha família.

o júri

presidente

Professor Doutor Joaquim Manuel Henriques de Sousa Pinto
Professor Auxiliar da Universidade de Aveiro

vogais

Professora Doutora Ana Alice Rodrigues Pereira Baptista
Professora Auxiliar do Departamento de Sistemas de Informação da Universidade do Minho

Professor Doutor Joaquim Arnaldo Carvalho Martins
Professor Catedrático da Universidade de Aveiro

agradecimentos

Ao meu orientador Professor Doutor Arnaldo Martins, o meu sincero agradecimento por toda a disponibilidade, apoio e orientação ao longo deste trabalho.

Ao Marco Pereira, aluno de doutoramento do IEETA, por toda a ajuda, conselhos, disponibilidade, motivação, amizade e por estares sempre presente nas horas mais complicadas de todo este projeto.

Um agradecimento muito especial à minha família. Pais (Rufino e Elizabeth), irmã (Filipa) e avós (Manuel e Armanda) um enorme obrigada não só pela força e compreensão que me deram ao longo desta árdua caminhada, mas também por toda a educação, dedicação, regalias e apoio incondicional que me foram proporcionando ao longo de toda a minha vida. Só graças a vocês tudo isto foi possível. Muito obrigada por tudo aquilo que representam para mim e por nunca me deixarem baixar os braços.

David, obrigada por estares sempre do meu lado nos momentos difíceis, pela amizade, inspiração, motivação, carinho, amizade e exemplo a seguir.

Finalmente, Ana Matos, Filipa Pereira, João Rodrigues e Susana Vasconcelos muito obrigada pela amizade sincera, motivação, ajuda, compreensão em momentos de alguma ausência e por estarem sempre prontos para umas boas gargalhadas.

palavras-chave

Objeto digital, Contexto, Preservação digital, Geo-localização, Dispositivo móvel, *Android*.

resumo

Motivação: Com a crescente expansão e utilização de dispositivos móveis, emergiu rapidamente a necessidade de recolha e preservação de objetos digitais. Cada vez mais, é importante tornar a nossa informação persistente e acessível em qualquer momento e em qualquer lugar. E se, para além do acesso, existisse a possibilidade de sujeitar cada objeto digital a uma contextualização adicional? Desta forma, o utilizador poderia relembrar a origem e fundamento dos respectivos dados ao longo de toda a sua vida.

Resultados: A aplicação CoBy surge com o intuito de permitir ao utilizador recolher, preservar e contextualizar todos os objetos digitais que vão surgindo no seu dispositivo móvel com sistema operativo *Android*. A cada um deles vai sendo associada a respectiva geo-localização de onde ocorreram. Estão disponíveis também funcionalidades de *backup* e *restore* a partir do próprio dispositivo móvel ou da *Cloud*. Esta aplicação dará um forte contributo no que toca à preservação e contextualização do património digital pessoal.

keywords

Digital object; Context; Digital preservation; Geo-location; Mobile device, Android.

abstract

Motivation: The increasing use and growth of mobile devices, spurred the need to collect and preserve digital objects. More and more, it is important to ensure that our information becomes persistent and accessible anytime and anywhere. What if, besides access, there was the possibility to contextualize each digital object? If so, the user could recall how, what and why it stored that data, throughout their lives.

Results: The CoBy application was created in order to allow the user to collect, preserve and contextualize all digital objects that will appear in mobile devices with Android as operating system. Each of them will be associated with place where they occurred (geo-location). The application includes other features, such as backup and restore from the mobile device or from the Cloud. The application provides a strong contribute to the preservation and contextualization of a person's digital heritage.

Conteúdo

Conteúdo	i
Lista de Figuras	iii
Lista de Tabelas	v
Lista de Acrónimos	vii
Introdução	1
1.1 Motivação	1
1.2 Objetivos	2
1.3 Estrutura	3
Enquadramento	5
2.1 Dispositivos móveis	5
2.2 Contextualização	7
2.3 Trabalho Relacionado	8
2.3.1 Descrição	8
2.3.2 Análise	12
Arquitetura	17
3.1 Visão Geral	17
3.2 Funcionalidades	20
3.2.1 <i>Account</i>	21
3.2.2 <i>Settings</i>	21
3.2.3 <i>Data</i>	22
3.2.4 <i>Backup</i> Manual	22
3.2.5 <i>Backup</i> Automático	23
3.2.6 <i>Restore</i>	23
3.3 Modelo Dominio	24
3.4 Modelo Genérico	25
Implementação	27
4.1 Visão Geral	27
4.2 Localização	28
4.3 Detecção de eventos	31
4.4 Base de dados	33

4.5	Repositório Digital Pessoal	36
4.6	Cartão memória	40
Protótipo		43
5.1	Interface gráfica	43
5.1.1	<i>Account</i>	44
5.1.2	<i>Main Activity</i>	44
5.1.3	<i>Settings</i>	46
5.1.4	<i>Data</i>	47
5.1.5	<i>Backup</i> Automático	51
5.1.6	<i>Backup</i> Manual	53
5.1.7	<i>Restore</i>	53
5.2	Ambiente	56
5.2.1	Desenvolvimento	56
5.2.2	Teste	56
Conclusões		59
6.1	Balanço Geral	59
6.2	Trabalho Futuro	60
Bibliografia		61
Anexos		65
7.1	Anexo A	65
7.2	Anexo B	66
7.3	Anexo C	67
7.4	Anexo D	68
7.5	Anexo E	69

Lista de Figuras

1.1	Visualização e recolha de objetos digitais do utilizador.	3
2.1	Surgimento das primeiras versões, dos vários sistemas operativos, que atualmente populam o mercado de dispositivos móveis.	5
2.2	Detalhes de uma fotografia tirada com um <i>smartphone</i> com suporte para localização por GPS e outra fotografia tirada com um telemóvel sem suporte a localização.	8
2.3	Estudo de alguns parâmetros do universo das vinte aplicações analisadas. . .	10
3.1	Procedimento usado quando é feito um <i>backup</i> manual.	18
3.2	Procedimento usado quando é feito um <i>retore</i>	19
3.3	Processo para localização de um objeto digital.	19
3.4	Diagrama de <i>use cases</i> relativo à conta do utilizador.	21
3.5	Diagrama de <i>use cases</i> relativo às definições configuráveis na aplicação. . . .	21
3.6	Diagrama de <i>use cases</i> relativo à visualização dos dados.	22
3.7	Diagrama de <i>use cases</i> relativo ao <i>backup</i> manual.	22
3.8	Diagrama de <i>use cases</i> relativo ao <i>backup</i> automático.	23
3.9	Diagrama de <i>use cases</i> relativo à funcionalidade de restauro.	23
3.10	Diagrama de domínio para suportar a entidade Mensagem e Contacto.	24
3.11	Diagrama de domínio para suportar a entidade utilizador e dados relativos ao agendamento de <i>backups</i>	24
3.12	Modelo genérico de recolha, preservação e contextualização de objetos digitais em repositórios pessoais e cartões de memória, baseado no modelo OAIS [1]. .	26
4.1	Logótipo da aplicação CoBy.	27
4.2	Parâmetros de localização obtidos para cada <i>provider</i> usado.	29
4.3	Permissões de localização presentes no ficheiro "AndroidManifest.xml".	30
4.4	Abordagens possíveis para detectar o envio e recepção de mensagens.	31
4.5	Estudo temporal para as duas soluções.	32
4.6	Permissões de acesso ao estado do dispositivo presentes no ficheiro "Android-Manifest.xml".	33
4.7	Fluxo de dados e bases de dados envolvidas.	34
4.8	Campos da tabela de mensagens da base de dados da aplicação CoBy.	34
4.9	Exemplo de uma tabela da base de dados de mensagens do <i>Android</i>	35
4.10	Permissões de acesso às mensagens e contactos presentes no ficheiro "Android-Manifest.xml".	36

4.11	Campos usados para criação de objetos JSON comuns aos objetos digitais sujeitos a geo-localização.	37
4.12	Campos usados na criação dos objetos JSON para as mensagens.	37
4.13	Campos usados na criação dos objetos JSON para os contactos.	37
4.14	Estrutura JSON para objetos digitais do tipo SMS e contactos.	38
4.15	Exemplo de um URL usado na aplicação CoBy.	38
4.16	Exemplo ilustrativo das duas abordagens distintas para envio de dados para o repositório.	39
4.17	Permissões de escrita das bases de dados presentes no ficheiro "AndroidManifest.xml".	40
4.18	Estrutura do conteúdo dos documentos XML, para mensagens e contactos.	41
4.19	Exemplo de tabelas da base de dados de contactos do <i>Android</i>	41
4.20	Permissões para escrita no cartão de memória do dispositivo móvel presentes no ficheiro "AndroidManifest.xml".	42
5.1	<i>Account Activity</i>	45
5.2	<i>Main Activity</i>	46
5.3	Definições suportadas.	48
5.4	<i>Tabs</i> usadas para diferenciar mensagens e contactos.	49
5.5	Visualização dos detalhes relativos às mensagens.	49
5.6	Visualização dos detalhes relativos aos contactos.	50
5.7	Notificação que reporta o sucesso ou insucesso do <i>backup</i> automático sem intervenção do utilizador.	51
5.8	<i>Backup</i> automático de Mensagens e Contactos.	52
5.9	<i>Backup</i> manual de Mensagens e Contactos.	54
5.10	<i>Tabs</i> usadas para diferenciar mensagens e contactos no repositório e cartão de memória.	55
5.11	<i>Restore</i> de Mensagens e Contactos a partir do repositório digital pessoal bem como do cartão SD do dispositivo móvel.	55
5.12	Logótipos dos elementos envolvidos na construção da aplicação.	56
5.13	Emulador de dispositivos móveis <i>Android</i>	56
5.14	Simulação da localização e da recepção de mensagens.	57
5.15	Dispositivo móvel real usado.	57
6.1	Análise SWOT.	59
7.1	Modelo da arquitetura do Sistema Operativo Android.	65
7.2	Estudo da venda de <i>smartphones</i> ao utilizador final, a nível mundial em função do Sistema Operativo.	66
7.3	Crescimento atual e previsão para os próximos anos.	66
7.4	Ciclo de vida de um serviço em <i>Android</i>	67
7.5	Ciclo vida de uma atividade em <i>Android</i>	68
7.6	Gráfico representativo das versões usadas do sistema operativo <i>Android</i>	69

Lista de Tabelas

2.1	Aplicações presentes no <i>Google Play</i> mais usadas para <i>backup</i> e restauro. . . .	9
2.2	Comparação entre a aplicação CoBy e as aplicações de <i>backup</i> e <i>restore</i> mais usadas no <i>Google Play</i>	14
2.3	Espaço ocupado por cada umas das aplicações analisadas.	16
3.1	Descrição das funcionalidades da aplicação.	20
4.1	Estudo efetuado sobre o consumo de bateria relativo às formas de localização usadas.	31
7.1	Tabela representativa das versões usadas do sistema operativo <i>Android</i>	69

Lista de Acrónimos

SMS *Short Message Service*

SD card *Secure Digital Card*

GPS *Global Positioning System*

JSON *JavaScript Object Notation*

GSM *Global System for Mobile Communications*

REST *Representational State Transfer*

RIM *Research In Motion Limited*

API *Application Programming Interface*

XML *Extensive Markup Language*

DOM *Document Object Model*

SAX *Simple API for XML*

UI *User Interface*

IDE *Integrated Development Environment*

ADT *Android Development Tools*

HTTP *Hypertext Transfer Protocol*

URL *Uniform Resource Locator*

MMS *Multimedia Messaging Service*

APK *Application Package File*

SDK *Software Development Kit*

ARM *Advanced RISC Machine*

MIPS *Microprocessor Without Interlocked Pipeline Stages*

UML *Unified Modeling Language*

DBMS *Database Management System*

RISTI *Revista Ibérica de Sistemas e Tecnologias de Informação*

Introdução

1.1 Motivação

A existência de objetos digitais no nosso dia a dia tem tido, ao longo dos últimos anos, um enorme crescimento. Estes podem ser concebidos como artefactos compostos tendo em conta três elementos essenciais: conteúdo, metadados e a sua relação com outros objetos [2]. Os objetos digitais podem assumir os mais variados tipos, retratando grande parte da nossa vida pessoal e profissional, e representam toda informação digital presente em dispositivos eletrónicos como: computadores, telemóveis, *smartphones*, *tablets*, entre outros. Esta informação digital é extremamente importante, para as pessoas que utilizam estes dispositivos, e ao mesmo tempo muito delicada e frágil. Se não tomarmos as devidas precauções, a qualquer momento pode-se perdê-la, e em alguns casos nada se pode fazer para contrariar a situação. É extremamente importante permitir aos detentores (utilizadores) de objetos digitais formas de os preservar. Assim, o utilizador consegue não só manter todas as suas memórias de sempre e para sempre, mas também dar o seu contributo para a passagem de conhecimento de geração em geração. É humanamente impossível recordar todos os acontecimentos que vão emergindo ao longo da vida, e como "Sem memórias não há história" [3], é crucial investir afincadamente nesta área. Com a sociedade maioritariamente digital em que vivemos hoje em dia, e a avançar a passos largos para se tornar inteiramente digital, é gritante a necessidade de preservar todo o material digital que dispomos, caso contrário, torna-se um pedaço da nossa história pessoal e universal que se perde.

Da necessidade de evitar a perda de informação digital surgiram os repositórios digitais [4]. Neles é possível armazenar coleções de objetos digitais, e que podem ser acedidos e recuperados para utilização imediata ou posterior. Um repositório digital suporta mecanismos de importação, exportação, identificação e armazenamento de objetos digitais. Para além disso, um repositório digital oferece ao utilizador uma interface permitindo-lhe aceder ao conteúdo que pretende. O uso de repositórios digitais traz benefícios a todos, até mesmo ao meio ambiente, evitando o consumo de alguns recursos naturais que tendem a escassear. A utilização vigorosa de repositórios digitais poderia contribuir para o atenuar deste problema, diminuindo a necessidade dos utilizadores preservarem os seus dados/conhecimentos em papel.

A necessidade de preservar a informação digital inerente à flexibilidade de armazenamento de qualquer tipo de objeto digital despoletou o aparecimento dos repositórios digitais pessoais. Estes permitem ao utilizador guardar os seus objetos digitais garantindo-lhe alguma capacidade de armazenamento e um controlo de acesso bastante prudente. Um utilizador quando recorre a este tipo de serviço pretende armazenar os seus dados para consulta e utilização a curto e longo prazo. Os repositórios digitais pessoais permitem a preservação controlada e persistente do nosso material digital, possibilitando o seu acesso em qualquer momento e em qualquer lugar, por quem de direito. A segurança dos dados é inteiramente assegurada, dado

que só terão acesso as pessoas que o utilizador permitir e os conteúdos que este definir.

Tendo a preservação dos nossos objetos digitais assegurada, levanta-se a questão: Será que conseguimos lembrar o seu contexto? O contexto, também designado como *Context Awareness* na literatura, [5–8], pode ser visto como uma análise ao contexto pessoal, social, histórico, físico, ambiental, posicional, etc., de um objeto digital. Resumidamente, o contexto de um objeto digital é tudo aquilo que fomenta o seu surgimento, a forma como é usado e interpretado, a que propósito foi concebido, o que influenciou a sua criação, local do evento, a sua relação com os outros objetos, entre outros. Segundo Arellano [9] uma aplicação cuja finalidade recaia sobre a preservação de informação deve ter sempre presente o seu conteúdo, estrutura e contexto. É muito útil termos os nossos dados preservados, mas também é indispensável saber o seu contexto, para ao longo dos anos ser possível a reconstrução da imagem mental do objeto digital.

1.2 Objetivos

As áreas de desenvolvimento envolvidas nesta temática são bastante diversas, exigindo a integração de tecnologias distintas. No entanto, a área móvel merece especial atenção, devido ao seu elevado crescimento e atual impacto na sociedade. O poder computacional dos dispositivos móveis é cada vez maior, e há muito que deixaram de ser meras ferramentas de comunicação. Neste caso, as atenções estão viradas para os *smartphones* e *tablets* que estão populados com grandes quantidades de informação, indispensável aos seus utilizadores, tais como: *Short Message Service* (SMS), contactos, histórico de chamadas, *bookmarks*, imagens, emails, páginas *Web* visitadas, lista de compromissos, entre muitos outros. Alguma desta informação só se encontra disponível no próprio dispositivo, e uma simples troca de equipamento pode proporcionar o seu desaparecimento.

Este projeto surge com o intuito de fornecer ao utilizador de dispositivos móveis com *Android* a possibilidade de recolher e preservar a sua informação digital. Para tal, é necessário criar uma aplicação que tenha em conta a delicadeza deste tipo de informação e ajude a preservar a curto e longo prazo os conteúdos presentes nestes dispositivos. Esta aplicação terá de ser desenvolvida a pensar nas pessoas e nos seus interesses, uma vez que, os dispositivos móveis contêm uma grande quantidade de informação pessoal e profissional, de alguma forma vulnerável.

Assim sendo, a aplicação a desenvolver deverá permitir a recolha para posterior preservação, de alguns objetos digitais. Numa primeira fase os objetos digitais a ter em conta serão as mensagens, também designadas por SMS e os contactos. A preservação destes objetos deverá ser garantida de duas formas distintas: usando um repositório digital pessoal presente na *Cloud* e um cartão de memória do dispositivo. Estas duas soluções são vantajosas dependendo, claro, da situação. Existem alturas em que nos é útil consultar os dados no repositório pessoal, e outras em que não. Se, por exemplo, no momento em que precisamos de determinada informação não dispomos de uma ligação à Internet, a possibilidade de ter essa informação no cartão de memória do dispositivo é benéfica. Para além disso, a facilidade com que se troca de cartão de memória entre dois dispositivos é de tal forma elevada que vale a pena ter esta opção em consideração. Assim, rapidamente se restaura o conteúdo de um dispositivo noutra. Como tal, a possibilidade de restauro dos dados no mesmo ou noutra dispositivo deverá estar presente nesta aplicação, quer a partir do cartão de memória, quer do repositório. Se a preservação dos dados apenas fosse garantida através do cartão de memória

haveria uma grande probabilidade de perda de parte do nosso património digital. O cartão de memória, regra geral, é trazido junto ao dispositivo. Se por alguma razão este dispositivo se perde, com ele vão os dados que tivemos tanto cuidado a preservar. Nesta situação o uso de um repositório digital pessoal é fulcral. O ideal é usar as duas abordagens simultaneamente, para que uma colmate as fraquezas da outra. O uso do repositório torna-se mais seguro e vê-se garantida a preservação do nosso espólio digital a longo prazo. O cartão de memória é uma solução menos segura e permite a preservação dos dados a curto prazo. O ideal é termos os dados que pretendemos preservar ao longo dos tempos no repositório, e aqueles que usamos regularmente ou foram obtidos recentemente no cartão de memória.

Tal como se encontra ilustrado na Figura 1.1 a recolha dos objetos digitais é apenas efetuada em dispositivos móveis com *Android*. Contrariamente ao que acontece com a visualização no repositório digital pessoal, que pode ser feita pelo utilizador sempre que pretender, independentemente do sítio, do tempo, do sistema operativo e do dispositivo de acesso. Usando um repositório pessoal existe ainda a liberdade de poder partilhar os seus conteúdos digitais com amigos, usando para isso as redes sociais (Facebook¹, Google+², YouTube³, Flickr⁴, LinkedIn⁵, entre outros), ou simplesmente permitir o acesso de determinada pessoa ao seu repositório digital pessoal.



Figura 1.1: Visualização e recolha de objetos digitais do utilizador.

Numa segunda fase, pretende-se estudar a possibilidade de melhorar os objetos digitais recolhidos com alguma informação de contextualização adicional, para mais facilmente permitir ao utilizador aperceber-se do contexto do seu surgimento.

1.3 Estrutura

Este documento encontra-se dividido em seis capítulos.

No presente capítulo, são traçados os principais objetivos da dissertação, bem como a motivação do seu surgimento.

No Enquadramento é efetuada uma breve referência histórica sobre os sistemas operativos para dispositivos móveis mais usados. Seguido de um levantamento do contexto em que esta aplicação foi projetada. São enumeradas algumas características das aplicação que de alguma forma se assemelham com a aplicação desenvolvida. Além disso, é feita uma análise comparativa entre a aplicação desenvolvida e as aplicações mais usadas no mercado com

¹www.facebook.com

²plus.google.com/

³www.youtube.com/

⁴www.flickr.com/

⁵www.linkedin.com/

características semelhantes. Neste capítulo é também abordado o tema contextualização de objetos digitais.

Na Arquitetura encontra-se descrito o estudo prévio necessário ao desenvolvimento de *software*. Está presente, também, o modelo genérico criado para solucionar o problema proposto.

Seguidamente, na Implementação são descritas as abordagens e comportamentos escolhidos para a construção da aplicação, justificando as opções tomadas.

No capítulo responsável pelo Protótipo é mostrada a interface gráfica da aplicação, bem como uma breve descrição do ambiente de desenvolvimento e teste, envolvidos.

Por fim, a Conclusão, onde é feito um balanço conclusivo deste projeto, bem como a especificação de algumas ideias para desenvolvimento futuro.

Enquadramento

2.1 Dispositivos móveis

O aparecimento do telefone em 1876 revolucionou impetuosamente a forma de comunicar das pessoas. Inicialmente, surgiram os telefones unicamente com suporte para chamadas de voz, mas ao longo dos tempos o suporte para dados foi despoletando, e estes foram convergindo para os telemóveis que hoje são usados, e indispensáveis à vida em sociedade.

Com a evolução da tecnologia, os telemóveis deixaram de ser meros telefones, cujo ponto forte estava no suporte para chamadas e mensagens. Estes, rapidamente deram o salto para os *smartphones* tão usados atualmente. O poder computacional dos *smartphones* está a intensificar-se de tal forma que as previsões apontam para que em breve, se tornem em autênticos computadores pessoais [10].

Atualmente, no mercado existem os mais variados tipos de dispositivos móveis, com características muito distintas de forma a ir de encontro com as exigências e necessidades dos consumidores. Os sistemas operativos que permitem dar vida aos dispositivos móveis mais usados atualmente são: *Android*, *iOS*, *Windows phone*, *Symbian* e *BlackBerry*. Alguns surgiram no mercado há algum tempo e ainda se mantêm nos sistemas operativos mais usados, outros emergiram mais recentemente e também deles fazem parte, Figura 2.1.

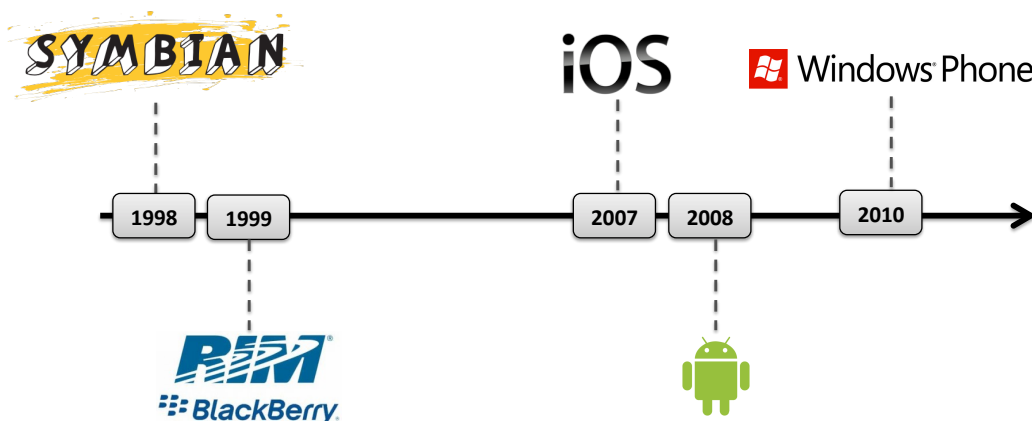


Figura 2.1: Surgimento das primeiras versões, dos vários sistemas operativos, que atualmente populam o mercado de dispositivos móveis.

Android

Android, foi inicialmente criado por um grupo de empresas, que faziam parte do consorcio designado por *Open Handset Alliance*, lideradas pela *Google*¹. Atualmente, algumas delas ainda fazem parte do projeto. Trata-se de um sistema operativo *open-source* baseado em *Linux*, desenhado especialmente para *smartphones* e *tablets*, mas também pode ser usado *net-books*, *laptops*, *ebook readers*, etc. Existe uma grande quantidade de dispositivos no mercado com este sistema operativo. Regra geral, as aplicações são desenvolvidas em Java e usam *Android Software Development Kit (SDK)*², apesar de existirem outras ferramentas de desenvolvimento como *Native development kit* e *Google App Inventor*. *Android* dispõe de várias camadas de *software* que inclui um *kernel* baseado em *Linux*, *middleware* (bibliotecas e *Application Programming Interface (API)*) e aplicações. No Anexo A encontra-se o diagrama da arquitetura do sistema operativo *Android*. Este foi nativamente desenvolvido para correr em processadores com arquitetura *Advanced RISC Machine (ARM)*, mas também pode ser usado em processadores *Microprocessor Without Interlocked Pipeline Stages (MIPS)* e x86 [11].

iOS

Trata-se de um sistema operativo proprietário criado pela *Apple*³ e originalmente desenvolvido para o *iphone*, mas também usado no *iPod Touch*, *iPad* e *Apple TV*. Este sistema operativo foi inicialmente chamado de *iPhone OS*, mas apenas manteve esse nome por quatro anos, sendo alterado para o iOS que hoje se conhece. Quando este sistema operativo foi lançado no mercado, apenas os criadores podiam desenvolver aplicações para ele. Mas um ano depois, foi aberto à comunidade de programadores com o lançamento de uma versão beta do SDK. As aplicações para iOS são desenvolvidas usando a linguagem de programação *Objective C* [12]. Este sistema operativo está desenhado para correr em dispositivos com processadores ARM.

Symbian

Symbian OS foi o nome inicialmente dado ao sistema operativo proprietário, desenvolvido por uma parceria entre várias empresas (*Pision*, *Nokia*, *Sony-Ericsson*, e *Motorola*), designadas por *Symbian Ltd*. Em 2008 a *Symbian Ltd* foi comprada pela *Nokia* e foi criada uma organização sem fins lucrativos designada *Symbian Foundation*. Esta, entretanto desenvolveu a plataforma *Symbian* completamente *open-source* como sucessora do antigo *Symbian OS*. A plataforma *Symbian* é atualmente mantida pela empresa de consultoria *Accenture plc*. Aplicações para este sistema operativo podem ser desenvolvidas em C++ usando a *cross-platform framework Qt*. Este é o ambiente de desenvolvimento padrão mas outras linguagens podem ser usadas. Foi projetado para correr em processadores ARM. [13, 14].

BlackBerry

BlackBerry OS inicialmente designado por *LeapFrog*, foi desenvolvido pela empresa canadiana *Research In Motion Limited (RIM)*⁴. Este sistema operativo foi concebido para

¹www.google.com

²<http://developer.android.com/sdk/index.html>

³<http://www.apple.com/>

⁴www.blackberry.com

dar vida aos dispositivos designados por *BlackBerries*. As aplicações podem ser criadas na linguagem Java usando *BlackBerry Widget SDK* [15].

Windows Phone

Windows Phone é um sistema operativo proprietário desenvolvido recentemente pela *Microsoft*⁵ e descendente do *Windows Mobile*. O *Windows Phone* sofreu uma remodelação completa relativamente ao *Windows Mobile*, chegando mesmo a haver alguma incompatibilidade entre ambos. As aplicações para este sistema operativo podem ser escritas em .NET [16]. Apresenta uma interface de utilizador designada por Metro.

A aplicação desenvolvida é direcionada para dispositivos móveis com *Android*. Esta escolha deveu-se, sobretudo, ao facto deste sistema operativo estar neste momento com a maior cota de mercado mundial [17]. E prevê-se que esse crescimento se acentue ainda mais [18]. No Anexo B encontra-se um levantamento desse crescimento e da conjectura para os próximos anos.

2.2 Contextualização

Quase toda a informação que dispomos hoje em dia, em dispositivos eletrónicos, possui algum tipo de contextualização. Trata-se de um tipo de contextualização transversal a quase todos os objetos digitais, e representa informação como:

- Nome;
- Data de criação;
- Tamanho;
- local onde se encontra gravado;
- Tipo ficheiro;
- Etc.

Em alguns casos, a contextualização que é obtida sobre um objeto digital, vai para além do contexto comum. Um bom exemplo disso, são as fotografias. Já existem máquinas fotográficas e dispositivos móveis que permitem associar a cada fotografia, obtida com o dispositivo, as coordenadas *Global Positioning System* (GPS) de onde foi tirada. Na Figura 2.2 está ilustrado os detalhes de uma fotografia tirada com um dispositivo móvel com suporte à localização e outro sem essa funcionalidade.

Um dos objetivos propostos neste trabalho recai sobre o melhoramento dos objetos digitais com algum tipo de informação adicional que permita uma melhor contextualização. A localização geográfica é uma forma eficaz de contextualização, uma vez que permite ao ser humano criar uma imagem mental do que o rodeava no momento em que determinado acontecimento surgiu. Para além disso, consegue facilmente conservar essa imagem mental a longo prazo.

⁵<http://www.microsoft.com/>

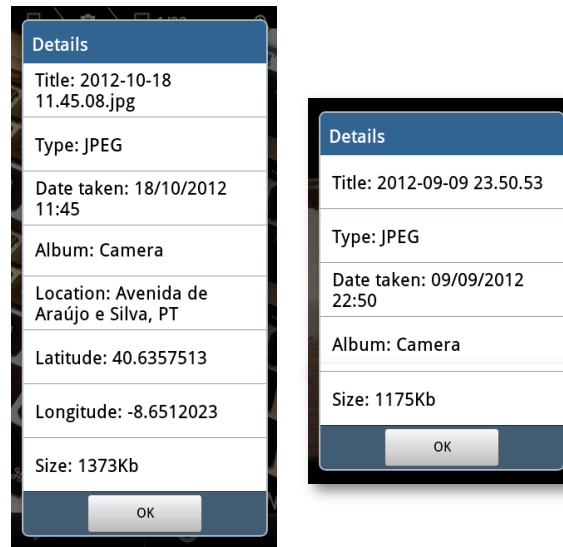


Figura 2.2: Detalhes de uma fotografia tirada com um *smartphone* com suporte para localização por GPS e outra fotografia tirada com um telemóvel sem suporte a localização.

A possibilidade de adicionar, a cada objeto digital que vai aparecendo nos dispositivos móveis, a sua localização, seria muito útil para o utilizador. Com suporte para esta funcionalidade, a aplicação a desenvolver tornar-se-ia mais completa e inovadora. Desta forma, o utilizador não só via os seus dados preservados como também o seu contexto geográfico. Será este o melhoramento a fazer no que toca à contextualização dos objetos.

2.3 Trabalho Relacionado

2.3.1 Descrição

Ao longo da história o homem foi desenvolvendo formas de preservar a sua informação, recorrendo a pinturas, fotografias, manuscritos, livros, entre muitas outras coisas. Mas, devido à evolução da tecnologia, foi surgindo outro tipo de informação, que contribuiu para a sociedade da informação em que se vive atualmente, amplamente rodeada de conhecimento digital. Foi igualmente necessário o desenvolvimento de ferramentas de salvaguarda deste tipo de dados, então, inúmeras aplicações foram desenvolvidas.

O sistema operativo *Android* disponibiliza uma aplicação designada *Play Store*, onde é possível obter todas as aplicações disponíveis no *Google Play* [19], a loja de aplicações do *Android*. Foi, então, efetuada uma pesquisa usando a *Play Store* de um dispositivo móvel em busca de aplicações que permitissem efetuar *backup* e restauro (também designado de *restore*) de dados. Foram filtradas as mais usadas, gratuitamente, e encontram-se ilustrada na Tabela 2.1, bem como as suas características mais importantes.

Do universo das vinte aplicações enumeradas na Tabela 2.1 foi efetuado um levantamento de quais os dados que os utilizadores preferencialmente querem ver preservados, bem como o local para onde o pretendem fazer, Figura 2.3. A preservação dos dados no cartão de memória, também designado por *Secure Digital Card* (SD card) ou cartão SD, do dispositivo móvel é quase o ponto comum a todas as aplicações. A maior parte delas preserva os seus dados

Aplicações [Fabricante]	Backup	Backups automáticos	Restore	Destino dos dados	Dados alvo
<i>App Backup & Restore</i> [INFOLIFE LLC]	✓	✓	✓	Cartão memória	Aplicações
<i>SMS Backup & Restore</i> [RITESH SAHU]	✓	✓	✓	Cartão memória	SMS
<i>AppMonster Free Backup Restore</i> [THINK ANDROID]	✓	✓	✓	Cartão memória	Aplicações, dados associados à aplicação
<i>Call logs Backup & Restore</i> [Ritesh Sahu]	✓	✓	✓	Cartão memória	Histórico chamadas
<i>SMS Backup & Restore</i> [INFOLIFE LLC]	✓	✓	✓	Cartão memória	SMS
<i>Titanium Backup root</i> [Titanium Track]	✓	✓	✓	Cartão memória	Aplicações, dados associados à aplicação
<i>SMS Backup +</i> [Jan Berkel]	✓	✓	✓	Email	SMS, MMS, Histórico chamadas
<i>Backup Assistant Plus</i> [VERIZON WIRELESS]	✓	✓	✓	Servidores próprios	Imagens, Musicas, Videos, Documentos, Contactos
<i>MyBackup</i> [Rerware, LLC]	✓	✓	✓	Servidores próprios, Cartão memória	Aplicações, SMS, MMS, Histórico chamadas, Contactos, Imagens, Músicas, <i>Bookmarks</i> , etc.
<i>JS Backup</i> [Johospace Co.,Ltd.]	✓	✓	✓	Cartão memória, <i>DropBox</i> , <i>SugarSync</i> , <i>GoogleDocs</i>	SMS, Contactos, MMS, <i>Bookmarks</i> , Histórico chamadas, Aplicações, Imagens, etc.
<i>Super Backup : SMS & Contactos</i> [Mobile Ideia Studio]	✓	✓	✓	Cartão memória, Email	SMS, Contactos, Histórico chamadas, <i>Bookmarks</i> , Calendários, Aplicações
<i>Go Backup Pro</i> [GO LAUNCHER EX]	✓	✓	✓	Cartão de memória, <i>DropBox</i>	Contactos, SMS, MMS, Histórico chamadas, Calendário, Aplicações
<i>Mobile backup II</i> [MobileHome Corp.]	✓	✗	✓	Cartão memória	SMS, Contactos, Histórico chamadas, Calendários
<i>Backup Contacts</i> [Red Rock AS]	✓	✗	✗	Cartão memória, <i>DropBox</i> , Email	Contactos
<i>Ultimate Backup</i> [JRummy Apps Inc.]	✓	✓	✓	Cartão memória, <i>DropBox</i> , Email	Aplicações
<i>App Backup & Reinstall</i> [NickyCho]	✓	✗	✓	Cartão memória	Aplicações
<i>Bookmark Sort & Backup</i> [happydroid]	✓	✗	✓	Cartão memória	Bookmarks
<i>Backup Master</i> [Funny Android Games]	✓	✓	✓	Cartão memória	SMS, MMS, <i>Bookmarks</i> , Histórico de chamadas, Aplicações, etc.
<i>BackupandShare Backup Solution</i> [Aress Software]	✓	✗	✓	Servidores próprios	Contactos, Imagens, Musicas
<i>Safe Backup</i> [Promwad Mobile]	✓	✗	✓	Cartão memória, <i>DropBox</i> , <i>Windows network</i> , <i>Box</i> , <i>FTP server</i>	SMS, Contactos, Histórico chamadas, <i>Bookmarks</i> , Aplicações, Imagens, etc.

Tabela 2.1: Aplicações presentes no *Google Play* mais usadas para *backup* e restauro.

neste local. Os dados que os utilizadores mais querem ver preservados são as SMS, aplicações e contactos. No desenvolvimento da aplicação relativa a este projeto, a escolha dos objetos digitais do tipo SMS e contactos e a preservação no cartão de memória, reflete este estudo. Segue uma breve descrição das funcionalidades da aplicação desenvolvida, bem como, de cada uma das aplicações cujos dados alvo são os contactos e/ou SMS.

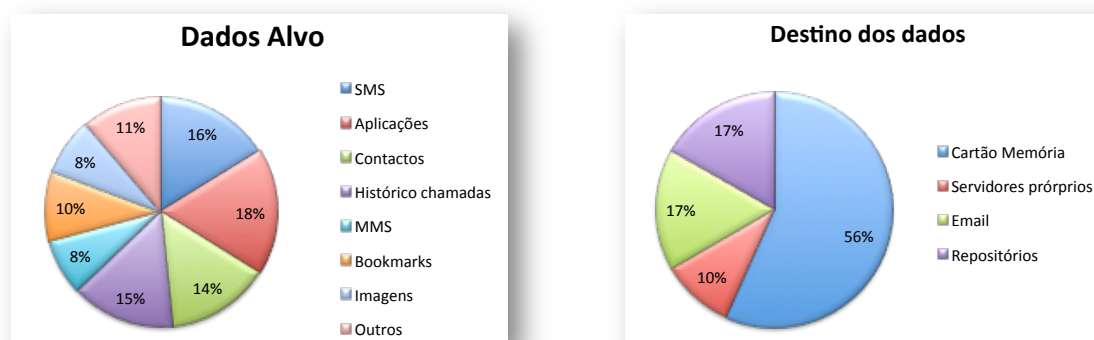


Figura 2.3: Estudo de alguns parâmetros do universo das vinte aplicações analisadas.

CoBy

A aplicação desenvolvida, foi designada por CoBy (*Contextualized Backup and Restore*), consegue englobar funcionalidades como: recolha, preservação e contextualização de vários tipos de objetos digitais (SMSs e contactos); geo-localização dos objetos; armazenamento da informação em locais distintos (cartão memória e repositório pessoal); sincronismo parcial dos dados com o repositório; *backups* automáticos e *restore* para/do cartão memória e repositório pessoal; *backups* automáticos com e sem intervenção do utilizador; preservar apenas objetos digitais do interesse do utilizador; interface gráfica de acesso aos dados no dispositivo móvel.

SMS Backup & Restore [Ritesh Sahu]

A aplicação *SMS Backup & Restore* desenvolvida por *Ritesh Sahu* permite efetuar *backups* manuais, *backups* automáticos e o restauro de SMS para o cartão de memória do dispositivo móvel. Os *backups* são guardados em documentos no formato *Extensive Markup Language* (XML) onde são colocadas todas as SMS que estiverem no dispositivo móvel, no momento. Nesta aplicação o utilizador pode consultar e pesquisar sobre o conteúdo dos ficheiros de *backup* [20]. Os ficheiros XML com os *backups* podem ser enviados por email.

SMS Backup & Restore [INFOLIFE LLC]

Trata-se de uma aplicação cujos objetos digitais alvos são as SMSs. Permite fazer *backup* para o cartão de memória onde os dados são guardados numa base de dados sqlite. É possível agendar os *backups*. A funcionalidade de restauro também se encontra presente nesta aplicação.

SMS Backup +

Esta aplicação abrange um maior número de tipos de objetos digitais para preservação comparativamente às anteriores. Possibilita o *backup* manual e automático de SMS, MMS e histórico de chamadas para a conta de email do *Gmail* ⁶, usando uma *label* para efetuar distinção dos restantes emails. A funcionalidade de restauro apenas é possível para as SMS e histórico de chamadas.

My Backup

A aplicação *My Backup* estipula dois grupos de objetos digitais: "*Applications & media*" e "*Data*". "*Applications & media*" engloba objetos como imagens, música, vídeos e aplicações. "*Data*" abrange objetos digitais como SMS, contactos, histórico chamadas, *bookmarks*, MMS, Alarmes, calendário, entre outros. O *backup* pode ser feito para o cartão de memória do dispositivo móvel ou para a *Cloud* onde estão os seus servidores ⁷. Num *backup* do grupo de dados designado "*Applications & media*" é possível escolher, individualmente, cada objeto de cada tipo que se pretende preservar, contrariamente ao que acontece com o grupo "*Data*" que possibilita um *backup* de todos os objetos de determinado tipo. Além disso, no mesmo dia, só o último *backup* feito para o cartão de memória é considerado. Isto porque, a aplicação usa a data como nome do ficheiro de *backup* e só é criado um por dia, e quando lá existem dados previamente, estes são apagados e escritos os do novo *backup*. O restauro pode ser efetuado de ambos os locais. Esta aplicação permite ver o conteúdo dos ficheiros de *backup*. Os *backups* automáticos são suportados e pode-se escolher o tipo de dados para esta funcionalidade.

JS Backup

A aplicação *JS Backup*, permite fazer *backup* de dados como: SMS, contactos, MMS, *bookmarks*, histórico chamadas, aplicações, imagens, entre outros. É possível fazê-lo usando a modalidade *full backup* onde engloba todos os objetos digitais de todos os tipos suportados. Como também existe outra opção, *selected backup*, onde é possível selecionar os tipos de objetos, fazendo *backup* de todos os objetos digitais daquele tipo. As funcionalidades de restauro e agendamento de *backups* também são suportadas. Quando se faz um *backup* pode-se escolher um dos quatro destinos: cartão memória, *DropBox* ⁸, *SugarSync* ⁹ e *Google Docs* ¹⁰.

Super Backup : SMS & Contacts

Trata-se de uma aplicação que faculta funcionalidades de *backup* e restauro de objetos digitais do tipo SMS, contactos, histórico chamadas, *bookmarks*, aplicações e calendários. É possível escolher o tipo de dados que se pretende preservar, mas dentro de determinado tipo são tidos em consideração todos os objetos digitais. Os *backups* são feitos para o cartão de memória ou podem ser enviados para por email. No cartão de memória são usados ficheiros no formato XML para guardar os dados. Há igualmente a possibilidade de agendar *backups* para todos os tipos de dados com excepção dos calendários e *bookmarks*.

⁶<https://mail.google.com/>

⁷<http://www.rerware.com/>

⁸<https://www.dropbox.com/>

⁹<https://www.sugarsync.com/>

¹⁰<http://www.google.com/google-d-s/documents/>

GO Backup Pro

A aplicação *GO Backup Pro*, emprega funcionalidades de *backup* e restauro de contactos, SMS, MMS, histórico chamadas, calendário e Aplicações. Para cada tipo de dados é feito um *backup* total, sendo considerados todos os objetos digitais desse tipo. O agendamento de *backups* e o restauro são, também, funcionalidades suportadas. O restauro e respectivo *backup* podem ser feitos para a *Cloud*, mais especificamente para a *DropBox*.

Mobile Backup II

Tata-se de uma aplicação, que permite o *backup* e o restauro de SMS, contactos, histórico de chamadas e calendário para o cartão de memória. Não dispõe de *backups* automáticos nem seleção dos objetos dentro de cada tipo.

Backup Contacts

A aplicação *Backup Contacts*, admite o *backup* de contactos para o cartão de memória, DropBox e email. É feito o *backup* de todos os contactos, não existe a possibilidade de apenas preservar alguns. Não é suportada a funcionalidades de restauro, nem é possível efetuar *backups* automáticos.

Backup Master

A aplicação *Backup Master* tolera o *backup* e restauro de objetos digitais do tipo SMS, MMS, Bookmarks, Histórico de chamadas, Aplicações, entre outros. Existe a possibilidade de escolher quais os tipos de objetos a preservar mas dentro de cada tipo, é feito *backup* de todos. É proporcionada a possibilidade de fazer *backups* automáticos. O destino dos dados é, unicamente, o cartão de memória.

BackupandShare Backup Solution

Esta aplicação, permite fazer *backup* e restauro de objetos do tipo: contactos, fotografias, vídeos, musicas e imagens. São disponibilizados servidores na *Cloud* que dão suporte ao *backup* e restauro dos dados.

Safe Backup

A aplicação *Safe Backup*, viabiliza o *backup* e restauro de objetos digitais do tipo SMS, contactos, histórico chamadas, *bookmarks*, aplicações, imagens, entre outros. O destino dos dados pode ser escolhido pelo utilizador, tendo à sua disposição as seguintes possibilidades: cartão memória, *dropBox*, *windows network*, *Box* e *FTP server*. Não é possível fazer o agendamento de *backups*.

2.3.2 Análise

Fazendo uma análise comparativa da amostra de aplicações que de alguma forma se assemelham com a CoBy, apenas focando alguns aspetos, obtêm-se a Tabela 2.2. Todas as aplicações referenciadas na Tabela 2.2 juntam funcionalidades de *backup* e restauro de dados, com a exceção da aplicação *Backup Contats*. Esta, apenas permite efetuar *backups*. A ausência da funcionalidade de *restore* torna esta aplicação de certa forma limitada.

Existem aplicações que apenas se focam num único tipo de objetos digitais, como é o caso das seguintes: *SMS Backup & Restore [Ritesh Sahu]*, *SMS Backup & Restore [INFOLIFE]* e *Backup Contacts*. Mas, à semelhança do que acontece com a aplicação CoBy, as restantes englobam vários tipos de dados. A capacidade de fazer *backup* e restauro de vários tipos de objetos digitais, numa só aplicação, confere-lhe alguma versatilidade, evitando a necessidade do utilizador instalar uma aplicação por cada tipo de dados.

A existência de aplicações que possibilitem *backup* e *restore* para e/ou a partir do cartão de memória do dispositivo móvel é uma realidade bastante clara, e é uma característica comum a quase todas as aplicações retratadas na Tabela 2.2, com a excepção das aplicações *SMS Backup+* e *BackupandShare Backup Solution*.

Já existem aplicações que de alguma forma permitem o armazenamento e a preservação de dados na *Cloud*. Algumas delas só disponibilizam essa funcionalidade para o email. Três bons exemplos disso são as aplicações *SMS Backup & Restore* desenvolvido por Ritesh Sahu, *Super Backup* e *SMS Backup+* que permitem a exportação dos dados para uma conta de email. Tornando-se algo confuso e pouco objectivo para o utilizador. As aplicações CoBy, *MyBackup*, *JS Backup*, *Go Backup Pro*, *Backup Contacts*, *BackupandShare Backup Solution* e *Safe Backup* são mais inovadoras neste aspecto, pois permitem a ligação a um repositório onde a informação se encontra armazenada e privada. Relativamente a esta funcionalidade, e numa abordagem ligeiramente diferente da usada nestas aplicações, a *Google* disponibiliza um serviço designado *Google Sync*¹¹, disponível para *Android*, que permite o sincronismo de alguns objetos (contactos, emails, calendários, entre outros) entre *smartphones*, *tablet* e computadores pessoais. É uma forma fácil de ter os dados em vários dispositivos sem ter de os inserir manualmente. Trata-se de uma solução de preservação de informação mas quando um novo dispositivo se sincroniza recebe todos os objetos digitais de determinado tipo, mesmo aqueles que o utilizador já não utiliza. Não havendo a possibilidade de fazer sincronismo apenas de parte dos objetos. Não existe a ideia de preservação de objetos a longo prazo.

Os repositórios usados por estas aplicações são servidores próprios, *DropBox*, *SugarSync* ou *Google Docs*. O que distingue estes repositórios, do repositório digital pessoal usado pela CoBy, é o processamento a que os dados estão sujeitos. A informação neste repositório encontra-se organizada e estruturada, sendo o acesso por parte do utilizador rápido e eficaz. A filtragem de informação mediante alguns parâmetros, disponível no repositório digital pessoal trata-se de uma mais valia e acelera o processo de pesquisa de um determinado objeto.

A utilização de um repositório digital pessoal como plataforma de armazenamento remoto é um dos traços distintivos da aplicação CoBy. Um repositório digital pessoal, mais do que uma plataforma intermédia onde os objetos digitais ficam armazenados à espera de serem restaurados para um dispositivo móvel é uma plataforma que permite organizar e relacionar diferentes tipos de objetos digitais provenientes de várias plataformas e aplicações [21]. O repositório digital pessoal é responsável por interpretar as estruturas JSON utilizadas pela aplicação CoBy de forma a criar uma representação semântica dos objetos digitais recebidos, e de forma inversa é responsável por recriar as estruturas JSON a partir da representação semântica do objeto digital quando comunica com a aplicação CoBy. A representação semântica de um objeto digital é armazenada numa *embedded triple-store* (baseada na base de dados Neo4J¹²), e é determinada por uma ontologia (que se encontra em desenvolvimento de forma a ser capaz de lidar com tipos adicionais de objetos digitais). A ontologia que está a

¹¹<http://www.google.com/sync/>

¹²<http://neo4j.org/>

Algumas funcionalidades	CoBy	SMS Backup & Restore [RITESH SAHU]	SMS Backup & Restore [INFOLIFE.LLC]	SMS Backup +	MyBackup	JS Backup	Super Backup : SMS & Contactos	Go Backup Pro	Mobile backup II	Backup Contacts	Backup Master	BackupandShare Backup Solution	Safe Backup
SMS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓
Contactos	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	✗	✓	✓
Outros Objetos digitais	✗	✗	✗	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓
Backup para SDCard	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓
Restore do SDCard	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗	✓	✗	✓
Backup para repositório digital pessoal ou Cloud	✓	✓ Email	✗	✓ Email	✓	✓	✓ Email	✓	✗	✓	✗	✓	✓
Restore da repositório digital pessoal ou Cloud	✓	✓ Email	✗	✓ Email	✓	✓	✓ Email	✓	✗	✗	✗	✓	✓
Backups automáticos	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗	✗
Capacidade de filtrar informação que pretende preservar dentro de cada tipo de objeto	✓	✗	✗	✗	✓ Parcialmente	✗	✗	✗	✗	✗	✗	✗	✗
Suporte repositório digital pessoal	✓	✗	✗	✗	✓	✓	✗	✓	✗	✓	✗	✓	✓
Geo-localização	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

Tabela 2.2: Comparação entre a aplicação CoBy e as aplicações de *backup* e *restore* mais usadas no *Google Play*.

ser desenvolvida é fortemente influenciada pela ontologia CIDOC CRM¹³ com extensões para descrever eventos associados ao ciclo de vida dos objetos digitais, ao próprio repositório digital pessoal, e para descrever os vários tipos de objetos digitais que um utilizador pode armazenar no repositório digital pessoal. O uso desta ontologia permite classificar de forma consistente os objetos digitais, estabelecer as suas propriedades, e descobrir relações entre diferentes objetos digitais, tornado assim possível que o utilizador do repositório digital pessoal tenha a noção da forma como, por exemplo, uma determinada SMS se encaixa com outras formas de conversação digital (mensagens instantâneas, emails, etc.) mantidas com um contacto. Nem todos os objetos digitais podem ser armazenados de forma eficiente somente com recurso à sua descrição semântica. Objetos digitais com conteúdos binários, como por exemplo imagens ou músicas, são armazenados no seu formato binário nativo, tendo a sua representação semântica uma propriedade (*Uniform Resource Locator* (URL) para a representação nativa) que permite o acesso ao objeto digital no formato original.

As capacidades de classificação e organização de objetos digitais individuais distinguem um repositório digital pessoal das soluções de *backup* e *restore* com funcionalidades de armazenamento na *Cloud*, que normalmente criam arquivos com conjuntos de objetos digitais que têm de ser restaurados para um dispositivo compatível de forma a ficarem novamente acessíveis ao utilizador.

Os *backups* podem ser efetuados manualmente ou automaticamente. No caso em que esta tarefa é feita manualmente, pode existir alguma perda de informação. É ineficaz para o utilizador ter de desencadear um *backup* sempre que recebe ou cria um objeto, ou quando achar que é necessário. Acabando por fazê-lo, raramente, havendo o risco de perder informação nesse período de tempo. Os *backups* automáticos são mais eficazes, libertando o utilizador da preocupação de salvaguardar os seus dados periodicamente, tendo neste caso, a possibilidade de definir na aplicação quando e de quanto em quanto tempo o pretende fazer. Esta funcionalidade está presente não só na aplicação CoBy, como também em todas as restantes, com a exceção das aplicações *Backup Contacts*, *Mobile Backup II*, *BackupandShare Backup Solution* e *Safe Backup*.

A capacidade de preservar informação é evidente em todas as aplicações da Tabela 2.2, mas nenhuma delas proporciona, na sua totalidade, ao utilizador a possibilidade de filtrar criteriosamente todos os dados que pretende preservar, à exceção da aplicação CoBy. Na aplicação *MyBackup* apenas alguns dos dados estão sujeitos a essa filtragem, que não são o caso das mensagens e contactos. Existem sempre objetos digitais em que a sua preservação pode não ser desejada. Tudo aquilo que é guardado com o objectivo de ser preservado deve sofrer um controlo integral por parte do utilizador. A aplicação CoBy fornece ao utilizador este controlo pormenorizado, permitindo-lhe preservar unicamente aquilo que pretender.

Nenhuma das aplicações referenciadas permitem qualquer tipo de contextualização adicional. O uso do contexto geo-espacial, trata-se de uma funcionalidade apenas da aplicação CoBy, permitindo ao utilizador preservar não só o conteúdo mas também o local em que o objeto digital surgiu.

Foi efetuado um levantamento relativo ao espaço que cada aplicação ocupa no dispositivo móvel, Tabela 2.3. Estes valores foram registados imediatamente após instalação da aplicação. Como se pode verificar, e tendo em conta as aplicações analisadas, a CoBy é a aplicação que ocupa menos espaço no dispositivo móvel.

Nenhuma das aplicações descritas conseguem integrar na sua totalidade as funcionalidades

¹³<http://www.cidoc-crm.org/>

	Espaço no dispositivo após instalação
CoBy	420KB
<i>SMS Backup & Restore [RITESH SAHU]</i>	1.4MB
<i>SMS Backup & Restore [INFOLIFE LLC]</i>	1008KB
<i>SMS Backup +</i>	1.5MB
<i>My Backup</i>	1.1MB
<i>JS Backup</i>	2.4MB
<i>Super Backup : SMS & Contactos</i>	656KB
<i>Go Backup Pro</i>	3.2MB
<i>Mobile backup II</i>	532KB
<i>Backup Contacts</i>	820KB
<i>BackupandShare Backup Solution</i>	944KB
<i>Safe Backup</i>	1.8MB

Tabela 2.3: Espaço ocupado por cada uma das aplicações analisadas.

presentes na aplicação CoBy. Mas aquilo que de facto, se pode considerar inovador, é a geo-contextualização de objetos, possibilitando ao utilizador legitimar o surgimento dos mesmos, bem como a possibilidade de garantir a sua preservação num repositório digital pessoal.

Este capítulo apresentou o contexto em que se encontra inserida a aplicação desenvolvida, mostrando algumas soluções que já existe no mercado. Também foi feita uma análise comparativa entre a aplicação desenvolvida e as aplicações com características semelhantes.

Arquitetura

3.1 Visão Geral

Antes de iniciar a elaboração de uma aplicação é necessário fazer um levantamento de todas as tarefas e funcionalidades a suportar, bem como perceber as necessidades do público alvo.

A aplicação a desenvolver deverá dar suporte a duas tarefas essenciais, o *backup* e restauro de objetos digitais do tipo mensagens e contactos, para o repositório pessoal presente na *Cloud* e para o cartão de memória do dispositivo móvel. De forma a ser mais perceptível, nas Figura 3.1 e 3.2 encontra-se ilustrado o procedimento usado para cada uma destas tarefas.

Quando o utilizador decide fazer um *backup*, deverá ter ao seu dispor a possibilidade de escolher, dentro dos objetos suportados pela aplicação, o tipo de dados. Dentro de cada tipo de dados deverá, igualmente, poder escolher os objetos digitais que tem intenção de salvar, tendo sempre a opção fácil de seleccioná-los a todos. Depois de escolhidos os objetos, o passo seguinte é desencadear o *backup* propriamente dito e escolher o destino dos seus dados. Terá a hipótese de seleccionar apenas um destino ou os dois abrangidos pela aplicação. Estas operações deverão ser feitas individualmente para cada tipo de dados. O utilizador terá de aguardar o tempo necessário à escrita dos dados no cartão de memória e/ou envio do repositório para saber se o *backup* que efetuou foi bem sucedido.

Se o utilizador necessitar de ver os seus dados, novamente, no dispositivo móvel, recorre à opção de *restore*. Dependendo do local de onde o pretende fazer e do tipo de objeto digital que necessita, o utilizador deve ter ao seu dispor as opções:

- Restauro de mensagens a partir do cartão de memória;
- Restauro de contactos a partir do cartão de memória;
- Restauro de mensagens a partir do repositório digital pessoal;
- Restauro de contactos a partir do repositório digital pessoal;

Depois de escolher uma das opções o utilizador terá de aguardar que o restauro seja feito para confirmar o sucesso da operação.

Para além dessas tarefas essenciais a aplicação deverá dar suporte a: agendamento de *backups*; obtenção de localização no momento do surgimento do objeto digital no dispositivo móvel, Figura 3.3; consulta dos dados já devidamente localizados; suporte a contas de utilizador.

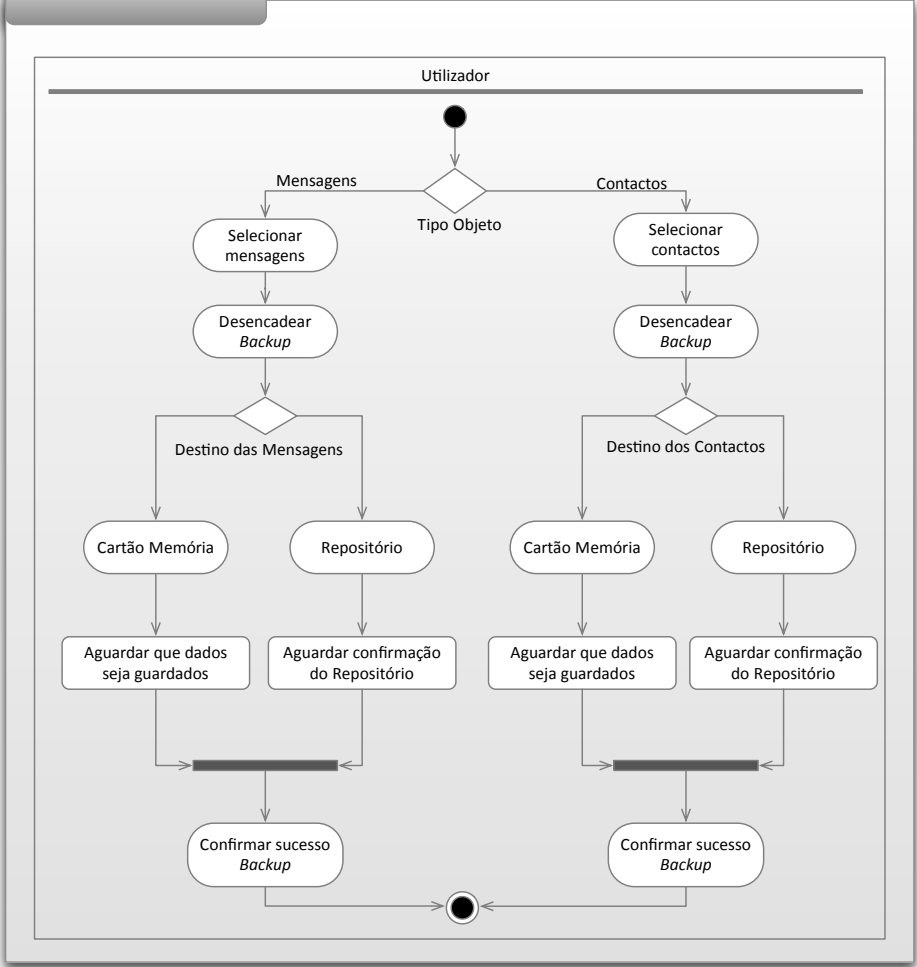


Figura 3.1: Procedimento usado quando é feito um *backup* manual.

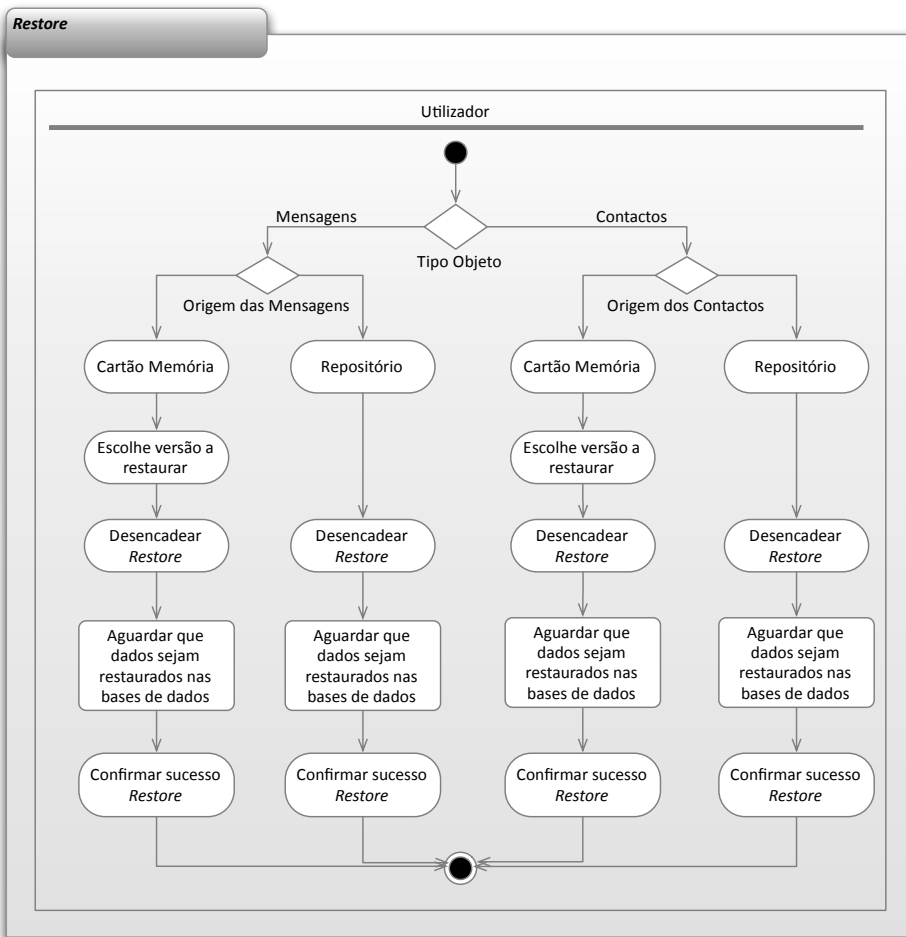


Figura 3.2: Procedimento usado quando é feito um *restore*.

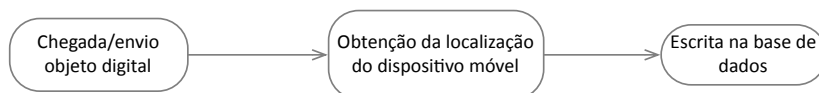


Figura 3.3: Processo para localização de um objeto digital.

3.2 Funcionalidades

Após o levantamento de todas as tarefas que englobam esta aplicação, é altura de descrever mais detalhadamente as suas funcionalidades. Para uma melhor percepção do funcionamento e da estrutura desta aplicação, as funcionalidades são agrupadas em função do papel que desempenham. Esse aglomerado é designado de bloco. Na Tabela 3.1 encontra-se elaborada uma descrição mais pormenorizada dessas funcionalidades proporcionando uma visão geral da estrutura da aplicação.

Bloco	Funcionalidade	Descrição
Conta	Inserir dados	Inserção dos dados necessários ao funcionamento da aplicação (Nome, telefone, email, url do repositório, <i>password</i>).
	Editar conta	Edição de todos os campos relativos à sua conta.
Backup manual	Escolher objeto digital	Optar por ver informação relativa às mensagens ou contactos.
	Ver lista mensagens	Visualização da lista das mensagens que foram enviadas ou recebidas após o uso da aplicação, devidamente localizadas.
	Selecionar mensagens	Seleção das mensagens que pretende preservar.
	Ver lista de contactos	Visualização de todos os contactos presentes no dispositivo móvel.
	Selecionar contactos	Seleção dos contactos que pretende preservar.
	Escolher destino do backup	Seleção do destino dos seus dados (cartão de memória e repositório digital pessoal).
	Backup	Envio dos dados para o destino.
Data	Escolher objeto digital	Optar por ver informação relativa às mensagens ou contactos.
	Ver lista de mensagens	Visualização da lista das mensagens, e alguns dos seus detalhes, que foram enviadas ou recebidas após o uso da aplicação, devidamente localizadas.
	Ver detalhes da mensagem	Consulta dos detalhes completos para mensagens escolhida (destinatário, remetente, corpo da mensagens, latitude, longitude, etc.).
	Ver localização no mapa	Visualização no mapa do local onde a mensagem foi enviada ou recebida
	Ver lista de contactos	Visualização de todos os contactos presentes no dispositivo móvel e alguns detalhes associados.
	Ver detalhes do contacto	Consulta dos detalhes do contacto escolhido (Nome, números de telefone, emails, morada)
Restore	Consultar o último backup feito para o repositório	Consulta da data do último backup feito para o repositório digital pessoal para as mensagens e contactos.
	Consultar a lista de backups feitos para o cartão de memória	Consulta da lista de backups para o cartão de memória, identificados pela data e hora a que foram feitos, para as mensagens e contactos.
	Escolher versão backup do cartão de memória	Escolha de uma versão de backup para restaurar.
	Restore	Restauração dos dados nas bases de dados correspondentes.
Settings	Escolher destino do backup automático	Seleção do destino dos seus dados (cartão de memória e repositório digital pessoal).
	Selecionar a frequência	Selecionar a frequência com que pretende que os backups sejam feitos (diariamente, semanalmente, mensalmente).
	Escolher data e hora	Escolher a data e hora a que pretende que o backup automático seja feito.
Backup Automático	Backup	Envio dos dados para o destino configurado.

Tabela 3.1: Descrição das funcionalidades da aplicação.

Segue uma breve descrição de cada bloco, recorrendo a casos de uso (*use cases*) baseados na noção gráfica *Unified Modeling Language* (UML) [22], que ajudam a definir o comportamento geral do aplicação. O público alvo será o utilizador comum de dispositivos móveis, como a maioria das aplicações que são desenvolvidas. O ator descritos nos casos de uso representa esse utilizador.

3.2.1 *Account*

Na Figura 3.4 encontram-se descritas as funcionalidades permitidas no bloco *Account*. Para o utilizador poder usar a aplicação será necessário a introdução de alguns dados pessoais, podendo posteriormente alterá-los caso se justifique.

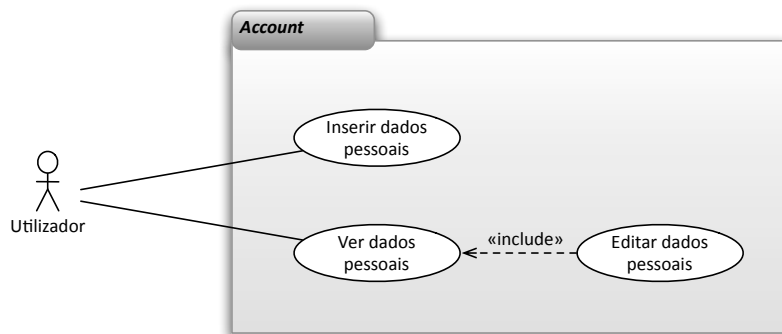


Figura 3.4: Diagrama de *use cases* relativo à conta do utilizador.

3.2.2 *Settings*

A possibilidade de agendar *backups* também é uma funcionalidade abrangida por esta aplicação. Essa funcionalidade está representada no bloco *settings*, Figura 3.5. Caso o utilizador deseje ativar a opção de *backups* automático, só precisa de definir o local para onde o pretende fazer bem como o espaçamento temporal que mais lhe agradar.

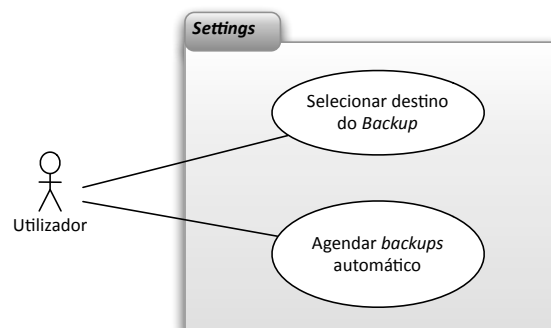


Figura 3.5: Diagrama de *use cases* relativo às definições configuráveis na aplicação.

3.2.3 Data

O utilizador deverá poder visualizar os detalhes dos contactos e das mensagens contextualizadas, Figura 3.6. Como a forma adicional de contextualização é a localização, a consulta no mapa do local do evento também deverá estar presente.

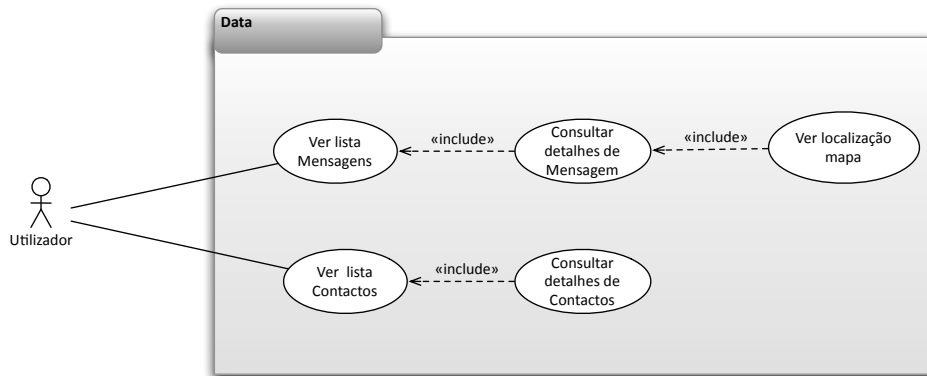


Figura 3.6: Diagrama de *use cases* relativo à visualização dos dados.

3.2.4 Backup Manual

Para fazer um *backup* manual o utilizador deverá ter acesso a uma listagem das suas mensagens e contactos por forma a conseguir seleccionar apenas aquilo que pretender ver preservado. Para fazer o *backup* propriamente dito é necessário a escolha do destino dos dados (repositório digital pessoal e/ou cartão de memória), Figura 3.7.

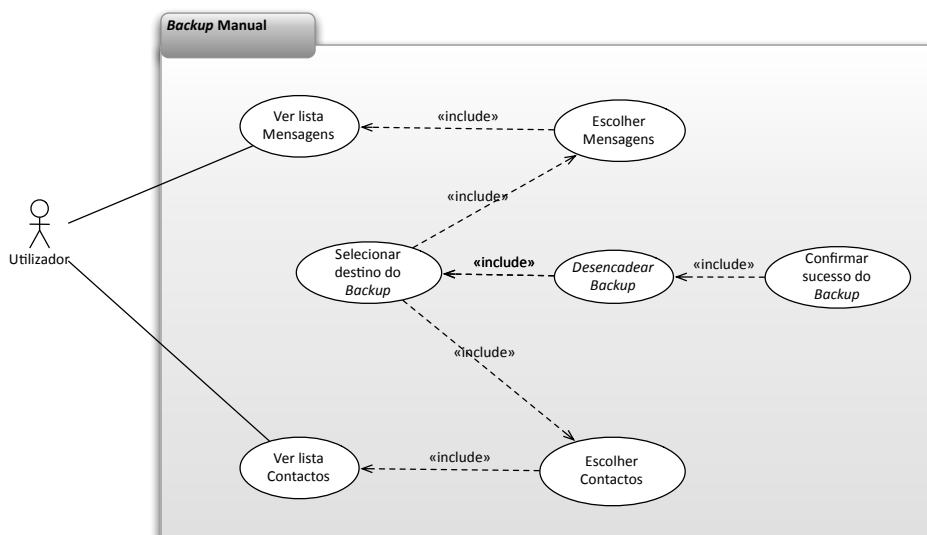


Figura 3.7: Diagrama de *use cases* relativo ao *backup* manual.

3.2.5 Backup Automático

Em função das definições colocadas no bloco *settings*, o *backup* automático será despoletado. Se a opção se encontrar ativa, o *backup* será desencadeado em função da frequência definida, Figura 3.8.

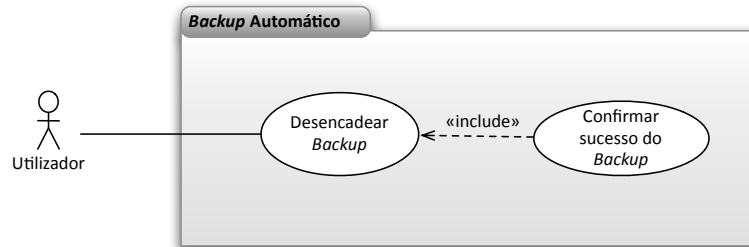


Figura 3.8: Diagrama de *use cases* relativo ao *backup* automático.

3.2.6 Restore

O restauro dos dados deverá ser feito de dois locais distintos: repositório digital pessoal e cartão de memória. Caso este seja feito a partir do cartão de memória ainda deverá estar presente a seleção da versão de *backup* que pretende restaurar, Figura 3.9.

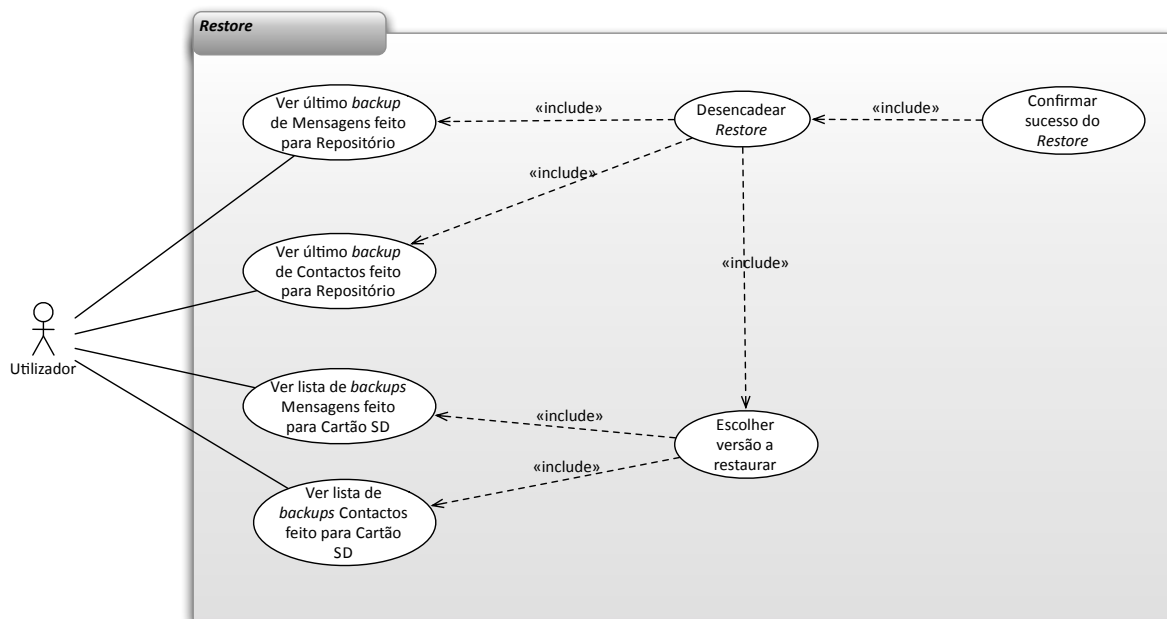


Figura 3.9: Diagrama de *use cases* relativo à funcionalidade de restauro.

3.3 Modelo Dominio

Após o levantamento de requisitos e funcionalidades a suportar foram elaborados dois diagramas que representam a base estrutural e fundamental deste trabalho. Para existência de *backup* e restauro de contactos e mensagens é necessário que haja uma classe que represente cada um destes tipos de dados, Figura 3.10. Neste caso, existem duas classes que representam a mesma mensagem, uma exibe informação proveniente da base de dados de mensagens do *Android* e a outra expõe os dados que são guardadas na base de dados da aplicação. O ponto comum, e que permite identificar uma mensagem, é o campo *timestamp*. Para fazer *backup* dos dados completos é apenas necessário ir buscar os dados da mensagem a cada fonte de dados correspondente. A cada mensagem pode existir um contacto, da lista de contactos, associado.

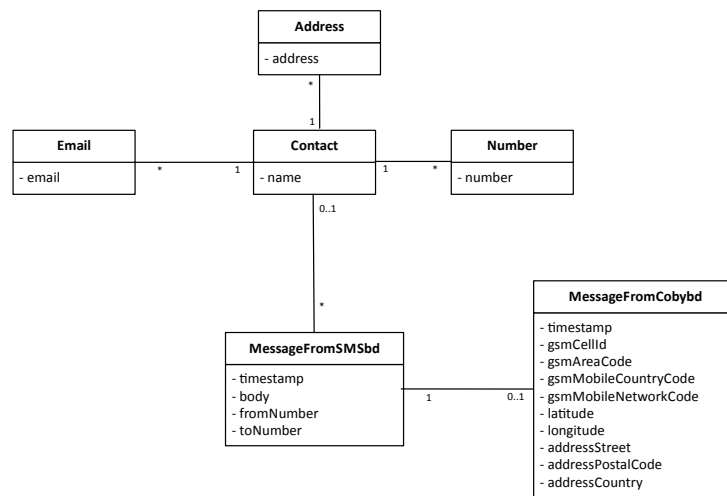


Figura 3.10: Diagrama de domínio para suportar a entidade Mensagem e Contacto.

Um vez que a aplicação não dará permissão para o utilizador a usar se não introduzir alguns dos seus dados pessoais, então é necessário a existência de uma classe que represente essa informação (*User*), Figura 3.11. Estes dados serão guardados nos detalhes associados à aplicação, para que o utilizador não necessite de os preencher sempre que a usar. Os dados relativos ao agendamento dos *backups* também serão guardados nas configurações da aplicação, logo também há necessidade de existir uma classe que os represente (*DataSettings*).

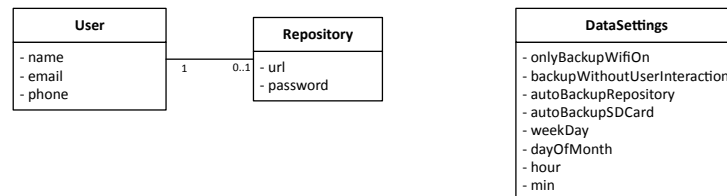


Figura 3.11: Diagrama de domínio para suportar a entidade utilizador e dados relativos ao agendamento de *backups*.

3.4 Modelo Genérico

Para dar suporte à recolha, preservação e contextualização do nosso património digital foi arquitetado um modelo. Na Figura 3.12 está representado esse modelo que pode ser usado para a criação de aplicações de recolha, preservação e contextualização de objetos digitais em repositórios digitais pessoais e cartões de memória, no qual a aplicação a desenvolver representa o seu protótipo.

Como já foi mencionado, o protótipo deste modelo será criado em *Android*, mas trata-se de um modelo completamente transversal ao nível do sistema operativo, podendo ser implementado em qualquer um.

Sempre que é enviado, recebido ou criado um objeto digital no dispositivo móvel é detetado esse evento. Sendo de imediato obtida a sua localização geográfica, caso esse tipo de contextualização faça sentido nesse objeto digital.

No modelo descrito é mantido um certo sincronismo, quer entre o repositório digital e o dispositivo móvel, quer entre o cartão de memória e o dispositivo. Se o utilizador enviar todos os seus dados para o repositório ou cartão de memória, vai permitir que naquele exato momento todo o conteúdo do seu dispositivo móvel se encontre no repositório ou no cartão de memória, tratando-se neste caso de um sincronismo total. Este tipo de sincronismo também existe no momento em que o utilizador efetua um restauro no dispositivo móvel, de todo o seu conteúdo informativo que dispõe no repositório ou cartão de memória. Caso o utilizador apenas permita o envio ou restauro de um fragmento dos seus dados para o repositório ou cartão de memória está implícito um sincronismo parcial.

Dentro do repositório os objetos digitais estão sujeitos a algum processamento e classificação para permitir ao utilizador aceder mais facilmente ao seu conteúdo. Este disponibiliza ao utilizador uma interface (página *Web*) para visualização dos dados. E estes poderão ser vistos sobre a forma de *timelíne* temporal, tendo capacidade de filtragem de informação usando alguns parâmetros, como por exemplo a geo-localização. O acesso ao repositório é totalmente independente do sistema operativo e dispositivo de acesso. No cartão de memória, os objetos digitais são apenas guardados, sem que haja processamento nem classificação. A classificação e processamento extra dos dados é deixado a cargo do repositório devido à escassez de recursos no dispositivo móvel, em oposição ao que acontece no repositório.

Neste modelo estão retratadas duas formas de preservar os dados: no repositório pessoal e no cartão de memória. O ideal é ter estas duas opções disponíveis. Mas caso não seja possível, ou o utilizador apenas opte por uma, o modelo continua a ser válido. O armazenamento da informação no cartão de memória e a preservação da mesma no repositório são duas operações completamente disjuntas. O uso de uma não implica nem invalida o uso da outra. Caso estejamos na presença de um dispositivo móvel sem cartão memória, o modelo é igualmente válido, apenas não é tido em consideração a parte relativa ao mesmo.

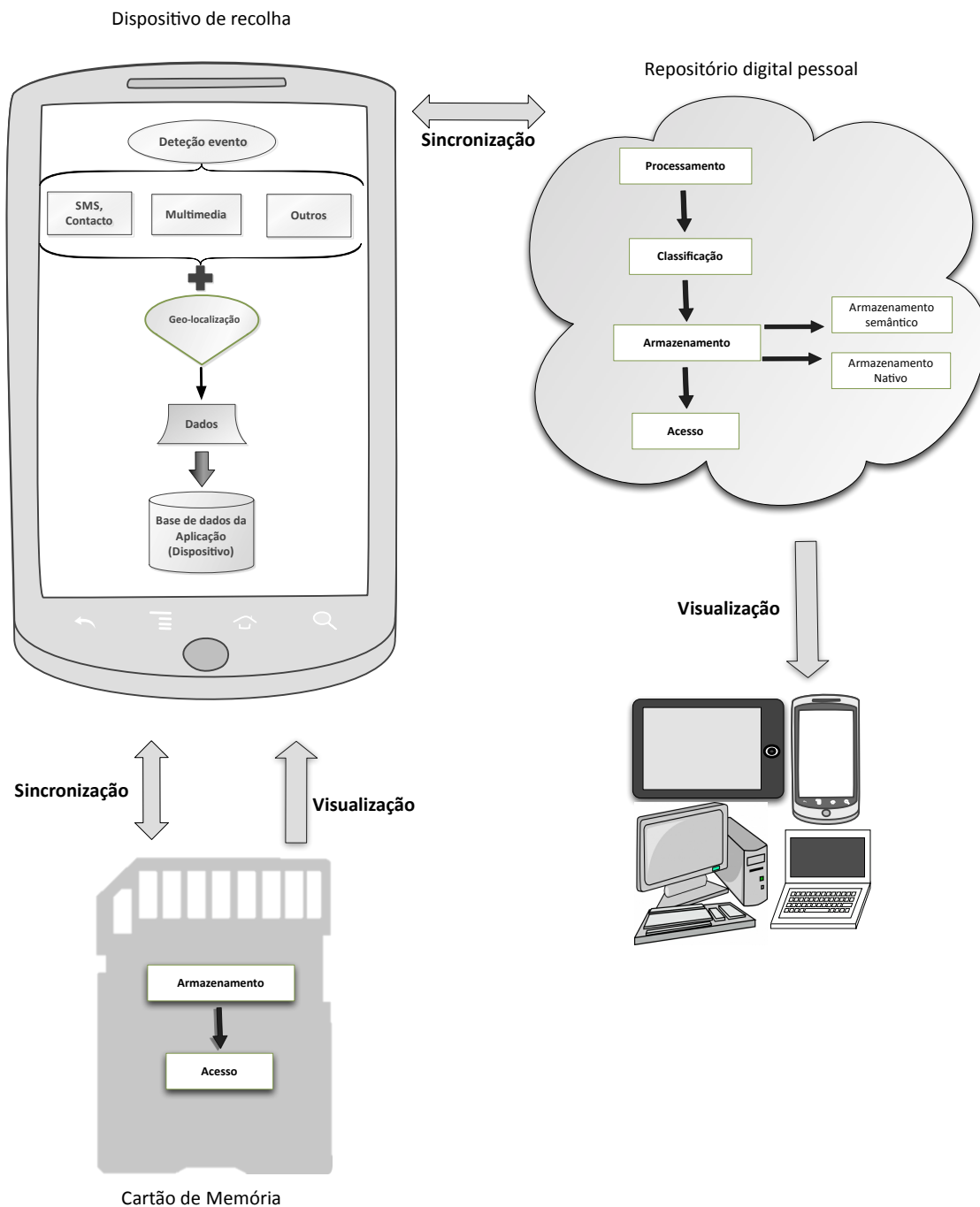


Figura 3.12: Modelo genérico de recolha, preservação e contextualização de objetos digitais em repositórios pessoais e cartões de memória, baseado no modelo OAIS [1].

Implementação

4.1 Visão Geral

A aplicação CoBy, tendo por base o modelo descrito no capítulo anterior, dá suporte à recolha e preservação de SMSs e contactos num dispositivo móvel *Android* e ao seu envio para um repositório digital pessoal e cartão de memória (Figura 4.1). No caso das mensagens, para além do contexto normal a que este objeto digital está sujeito, adicionou-se o contexto de geo-localização. A informação digital é guardada (*backup*) no cartão de memória do dispositivo móvel e/ou enviada para o repositório, já devidamente contextualizada. Também é dada a possibilidade de restaurar (*restore*) os dados do repositório ou do cartão de memória de volta para o dispositivo móvel [23]. Esta funcionalidade é extremamente importante caso haja uma mudança ou furto do *smartphone* ou *tablet*. Desta forma, o utilizador poderá voltar a ter todo o seu conteúdo digital num dispositivo móvel.



Figura 4.1: Logótipo da aplicação CoBy.

Nesta aplicação a adição de contactos não foi sujeita a localização, devido ao facto, desta operação, na maioria dos casos não se encontrar diretamente ligada ao local onde surgiu o novo contacto. Como se trata de uma operação que depende exclusivamente do utilizador, nem sempre o local onde o contacto é adicionado à lista de contactos do dispositivo móvel corresponde ao local onde este foi obtido. Um bom exemplo disso é quando o utilizador usa o serviço o *Google Sync* disponível para *Android* que possibilita o sincronismo dos seus contactos. Os contactos são todos repostos no dispositivo móvel, mas o local onde é feito não corresponde ao local do seu surgimento. Desta forma, não faz sentido guardar o contexto de geo-localização, pois pode não corresponder ao local do evento adicionar contacto. Mas trata-se de um dos objetos digitais que o utilizador mais interesse tem em preservar. Este problema não se coloca nas SMSs, daí a adição da sua geo-contextualização.

A aplicação CoBy permite também fazer *backups* automáticos para o cartão de memória do dispositivo móvel ou para o seu repositório digital pessoal presente na *Cloud*. O *backup* e o restauro a partir do repositório só precisa de uma ligação à Internet ativa. Esta aplicação para além das funcionalidades de *backup* e restauro, dá suporte à consulta dos detalhes relativos aos objetos digitais, bem como à sua localização no mapa. Integra funcionalidades como obtenção da localização do utilizador, persistência de dados no dispositivo e consumo de *Web-services*.

4.2 Localização

Para o sistema operativo *Android* existem algumas tecnologias que dão suporte à geo-localização, sendo designadas por *Location Providers* [24–26]. Nesta aplicação, a localização pode ser obtida de três formas distintas:

- *GPS*;
- *Network Location Provider*: engloba Cell-ID e Wi-Fi;
- Triangulação por Cell-ID.

GPS [27, 28] é um sistema de navegação por satélite que permite fornecer algumas informações a um receptor, nomeadamente a sua posição geográfica. Este representa o método mais preciso de obtenção da localização. Usando *Network Location Provider*, a localização é obtida, não só, através da informação vinda das estações-base da operadora móvel, mas também dos pontos de acesso *Wi-Fi* aos quais o dispositivo de encontra ligado. O Cell-ID [29] indica a posição da antena à qual o dispositivo se encontra ligado, e trata-se da forma de localização menos precisa, mas mais rápida.

Um objeto digital do tipo SMS por si só já tem agregado informações de contextualização, como é o caso da data, hora, destinatário, etc. Neste projeto, a contextualização adicional, consiste em associar informação de geo-localização ao objeto digital, no momento em que o utilizador o recebe, envia ou cria. Os parâmetros de informação de geo-localização usados para a contextualização das mensagens, podendo ser estendidos a quase todos os outros objetos digitais, são os seguintes:

- GSM Cell Id;
- GSM *Area Code*;
- GSM *Mobile Country Code*;
- GSM *Mobile Network Code*;
- Latitude;
- Longitude;
- *Address Street*;
- *Address Postal Code*;
- *Address Country*.

O parâmetro GSM Cell Id em conjunto com GSM *Area Code* representam um número único que permite identificar a antena à qual o dispositivo se encontra ligado e são obtidos usando a localização designada por Cell-ID. O GSM *Mobile Country Code* combinado com GSM *Mobile Network Code* identifica o operador de telemóvel usado naquele dispositivo móvel. Trata-se de informação irrelevante para a aplicação CoBy, mas é recolhida, pois permite ao repositório digital calcular uma localização aproximada para o objeto digital caso haja alguma falha na obtenção da localização no dispositivo móvel. Estes parâmetros são usados em conjunto com a localização por Cell-ID. A latitude e a longitude representam uma coordenada que permite

identificar um local na terra. Coordenada esta que pode ser obtida usando os sistemas de localização GPS, *Network Location Provider* ou Cell-ID.

Para a localização do utilizador no momento em que recebeu ou enviou uma SMS é apurado se este possui o GPS ativo. No caso afirmativo, é usado este sistema de geo-localização. No caso negativo, é investigado se tem o *Network Location Provider* ativo. Se tiver, usa esse *provider* para se localizar. Se o GPS e *Network Location Provider* se encontrarem ambos inativos é usada a localização por Cell-ID, que se encontra permanentemente ativa, e permite saber aproximadamente a localização do utilizador. Se dispuser de todos os *providers* ativos, é dado preferência ao GPS, seguidamente do *Network Location Provider* e por último ao Cell-Id, por questões de precisão [28]. Em algumas situações, apesar do GPS se encontrar ativo, não é possível obter uma localização, sobretudo se o dispositivo móvel se encontrar no interior de um edifício. Mas os campos GSM Cell Id, GSM Area Code, GSM Mobile Country Code, GSM Mobile Network Code, Latitude e Longitude são igualmente preenchidos graças à localização por Cell-ID, que se encontra sempre ativa e é sempre obtida. Na Figura 4.2 mostra o tipo de informação que é obtida em função do *provider* de localização. A geo-codificação (obtenção da morada a partir das coordenadas) só é obtida, imediatamente, se o utilizador tiver acesso à Internet (*Global System for Mobile Communications* (GSM) ou Wi-Fi). Caso o utilizador não tenha uma ligação à Internet no momento em que surgiu o objeto digital, é guardado na base de dados da aplicação CoBy apenas as coordenadas (GSM Cell Id, GSM Area Code, GSM Mobile Country Code, GSM Mobile Network Code, Latitude e Longitudes). A geo-codificação é conseguida posteriormente quando o utilizador já tiver oportunidade de se ligar a uma rede Wi-Fi ou GSM.

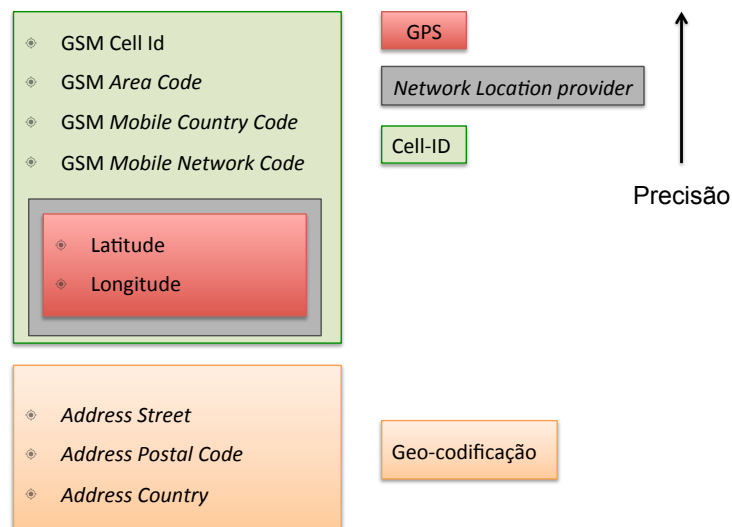


Figura 4.2: Parâmetros de localização obtidos para cada *provider* usado.

A obtenção da localização é conseguida através de um serviço que é ativado sempre que é recebida ou enviada uma mensagem [30]. O *Android* disponibiliza um pacote "*android.location*" que fornece uma API que permite obter a localização do dispositivo móvel. Neste pacote encontra-se uma classe designada por *LocationManager*, que autoriza a seleção dos *providers* de localização que se pretende usar. Nesta aplicação foram usados os seguintes: *GPS_PROVIDER* (corresponde à localização por GPS) e *NETWORK_PROVIDER*

(corresponde localização por *Network Location Provider*). Além disso, foi necessário o uso de um *LocationListener* que recebe periodicamente atualizações sobre a posição atual. Há a possibilidade de definir o período temporal e posicional em que se pretende atualizar a geo-localização do utilizador. Neste caso, foi considerado como tempo mínimo para obtenção de uma nova localização de 60000ms (1 minuto) e distância mínima entre atualizações de 5 metros. O dispositivo só irá obter nova localização caso tenha passado mais de 1 minuto ou o utilizador se tenha deslocado mais de 5 metros. Foram escolhidos valores relativamente espaçados, no tempo e no espaço, para minimizar o consumo de bateria nesta tarefa. Para a obtenção da localização por Cell-ID foi usada a classe *TelephonyManager* disponível no pacote "*android.telephony*" cedido pelo *Android*.

Nesta aplicação a funcionalidade de visualizar a localização de determinada mensagem no mapa está presente. Para tal, foi usado o pacote "*com.google.android.maps*" que permite a utilização do *Google Maps* ¹. A utilização desta API implica a obtenção de uma *Key*. Foi usada a classe *MapView* que permite a visualização propriamente dita do mapa sendo necessário ter uma ligação à Internet para o poder carregar.

Todos os projetos típicos para *Android* tem um ficheiro "*AndroidManifest.xml*" que define o comportamento e o conteúdo da aplicação. Para que todas estas funcionalidades inerentes à localização sejam possíveis é necessário dar permissão para tal. Isso é efetuado no ficheiro "*AndroidManifest.xml*", Figura 4.3.

```
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_MOCK_LOCATION" />
<uses-permission android:name="android.permission.INTERNET" />
```

Figura 4.3: Permissões de localização presentes no ficheiro "*AndroidManifest.xml*".

A duração da bateria dos dispositivos móveis é um dos grandes problemas com que os programadores e utilizadores se deparam [24]. É essencial que se desenvolvam aplicações que de alguma forma, consumam menos recursos energéticos possíveis. Foi realizado um estudo sobre o gasto de bateria para cada uma das formas de obtenção de localização. Num dispositivo móvel, foram enviadas 20 mensagens em 2 horas e tentou-se que todas as iterações reunissem as condições o mais semelhantes possíveis. Em todas elas o dispositivo móvel encontrava-se com 100% de bateria e este só foi usado para o envio das mensagens. Não é possível separar a localização por GPS ou *Network Location Provider* da localização por Cell-ID, porque esta última encontra-se sempre ativa. Como é possível constatar na Tabela 4.1 a localização por GPS é a que consome mais bateria, mas como já foi referido, trata-se da forma de localização mais precisa. Contrastando com a localização por Cell-ID, que consome significativamente menos bateria, mas não é tão precisa. No que toca à geo-localização é estabelecido um compromisso entre o consumo de bateria e a precisão de localização.

¹<https://maps.google.com/>

	Gasto Bateria			
	iteração 1	iteração 2	Iteração 3	Média
GPS + Cell-ID	18%	16%	21%	18%
Network Location Provider + Cell-ID	12%	13%	10%	12%
Cell-ID	7%	6%	4%	6%

Tabela 4.1: Estudo efetuado sobre o consumo de bateria relativo às formas de localização usadas.

4.3 Detecção de eventos

Outro aspecto que implica algum consumo de recursos energéticos do dispositivo móvel são os serviços. Mas muitas vezes não existe nenhuma solução alternativa, e têm mesmo de ser usados.

Para detectar eventos de chegada e envio de mensagens no dispositivo móvel pode ter-se em conta duas abordagens distintas, que se encontram descritas na Figura 4.4.

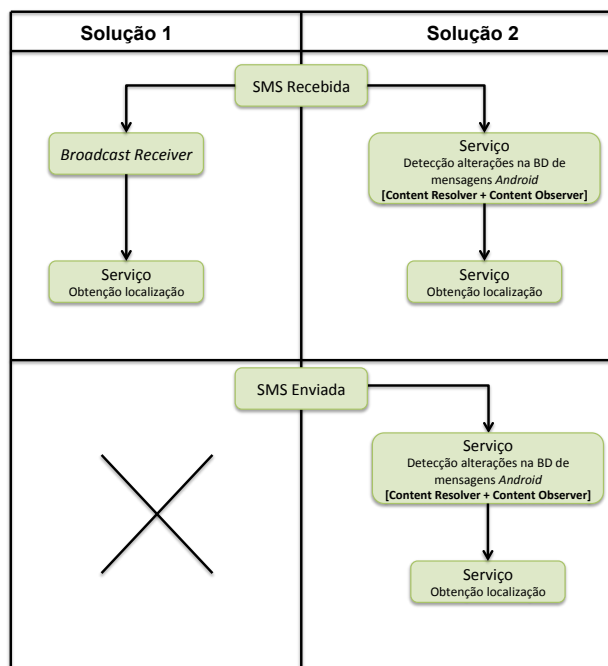


Figura 4.4: Abordagens possíveis para detectar o envio e recepção de mensagens.

Na Solução 1 existe um *Broadcast Receiver* que deteta a chegada de uma mensagem e despoleta o serviço que irá obter a localização do utilizador usando os sistemas de geolocalização disponíveis no dispositivo móvel, no momento. Este ativa o serviço apenas quando deteta a recepção de uma SMS, evitando a necessidade de ter um serviço permanentemente a correr.

Na Solução 2 tem-se um serviço sempre ativo que possibilita a detecção de alterações na base de dados de mensagens do *Android*. Isto é, permite detectar a recepção e o envio das mensagens. Este serviço dispõe de um *Content Resolver* e de um *Content Observer* que juntos permitem ver qual foi especificamente a alteração que ocorreu. O *Content Resolver* faculta o acesso à base de dados de mensagens do *Android*. O *Content Observer* permite, por *polling* à base de dados, ver qual foi a alteração detetada. Em função da alteração registada, é despoletado, ou não, o serviço responsável pela obtenção da geo-localização, e o utilizador é localizado. Este serviço é terminado assim que a localização do utilizador for obtida.

Foi efetuado um estudo temporal para cada uma das abordagens com o objetivo de optar por uma delas. Esse estudo encontra-se ilustrado na Figura 4.5.

	Solução 1	Solução 2
Início contagem	Ativação do <i>Broadcast Receiver</i>	Assim que detecta alteração na base de dados de mensagens do <i>Android</i>
Fim contagem	Invocação do serviço de localização	
Amostras recolhidas (ms)	29	270
	7	275
	4	251
	8	281
	10	287
	5	290
	7	272
	12	277
	15	282
7	263	
Média	10ms	275ms

Figura 4.5: Estudo temporal para as duas soluções.

Analisando os valores registados, e apesar das amostras recolhidas não terem sido muitas, já permite constatar que existe alguma discrepância entre as duas soluções. Os tempos obtidos diferem numa ordem de grandeza. À primeira vista, estes dados levar-nos-iam à primeira solução. Mas esta possui uma grande desvantagem: não é possível configurar o *Broadcast Receiver* para detectar o envio de mensagens, apenas recepção. Poder-se-ia implementar as duas soluções. A solução 1 seria unicamente usada para as mensagens recebidas, e a solução 2 para as enviadas. Dado que seria necessário haver sempre um serviço a correr para detectar todas as mudanças na base de dados de mensagens do *Android*, ter as duas abordagens implementadas seria desnecessário. Uma vez que uma delas cobre o envio e recepção de mensagens. Então, optou-se por usar a solução 2 quer para as mensagens enviadas, quer para as recebidas.

O serviço que deteta as alterações na base de dados de mensagens do *Android* é lançado quando a aplicação é criada ou iniciada. Em casos em que o dispositivo móvel é desligado existe um *Broadcast Receiver* que deteta que o sistema operativo acabou de iniciar, e lança o serviço em questão novamente. A utilização deste *Broadcast Receiver* implica uma autorização prévia no ficheiro "AndroidManifest.xml", Figura 4.6. Um problema que se coloca quando se usa um serviço está no facto deste poder ser terminado pelo sistema operativo quando este

fica com falta de memória. Esta questão foi tida em consideração, e foi retornada a constante `START_STICKY` no método `onStartCommand()` para indicar ao sistema operativo que assim que tiver memória disponível, reiniciar o serviço (Anexo C).

```
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
```

Figura 4.6: Permissões de acesso ao estado do dispositivo presentes no ficheiro "AndroidManifest.xml".

Outro evento imprescindível a esta aplicação é o disparo de um alarme para executar os *backups* automáticos. Caso exista algum *backup* agendado na aplicação, é necessário ativar um serviço que realize essa tarefa. Para tal foi usada a classe designada por *AlarmManager*, disponível no pacote "*android.app.AlarmManager*" do *Android* SDK, que permite o acesso aos serviços de alarme do sistema operativo. O alarme dispara quando tiver um *backup* para realizar, comunica esse evento ao sistema operativo e este lança o serviço que desencadeia o *backup*.

4.4 Base de dados

De forma a persistir os dados localmente no dispositivo móvel, a plataforma *Android* disponibiliza uma abstração designada *Content Provider* [10], acedido através de cursores de uma forma espectável e independentemente da fonte de dados utilizada. As *queries* feitas a uma base de dados, em *Android*, em vez de devolverem uma cópia dos valores pedidos, devolvem um cursor que permite o acesso à informação. Um cursor mais não é do que um ponteiro para os resultados. Para usar os padrões nativos deste sistema operativo, foi utilizada uma base de dados relacional SQLite ² para guardar os dados localmente, construindo-se um *Content Provider* para facilitar o acesso aos mesmos [31]. É guardada nesta base de dados toda a informação de localização que se conseguir apurar.

No caso das SMSs estão envolvidas duas bases de dados, a da aplicação CoBy para guardar a informação de localização e a base de dados de mensagens do *Android*. No caso dos contactos, é apenas necessária a base de dados de contactos do *Android*, devido ao facto de não haver necessidade de guardar a localização dos mesmos. Nas Figura 4.7 encontra-se ilustrado o fluxo de dados existente entre o dispositivo móvel, o repositório digital e o cartão de memória relativo às mensagens e contactos a preservar, bem como as bases de dados envolvidas.

Para cada mensagens que chega ao dispositivo móvel é criada uma entrada na tabela da base de dados da aplicação CoBy, cujos campos se encontram descritos na Figura 4.8. Para que tal seja possível, foi necessário recorrer à classe *ContentValues* disponível pelo *Android*, que permite inserir novas linhas na tabela.

O *timestamp* que representa a data e o tempo de determinado evento, funciona como chave primária em todas as tabelas da base de dados da aplicação CoBy. Dois objetos do mesmo tipo terão sempre *timestamps* diferentes. Seguidamente ao *timestamp* encontra-se a informação de localização, propriamente dita.

A forma de localização mais rápida corresponde à localização por Cell-ID, que é sempre

²<http://www.sqlite.org/>

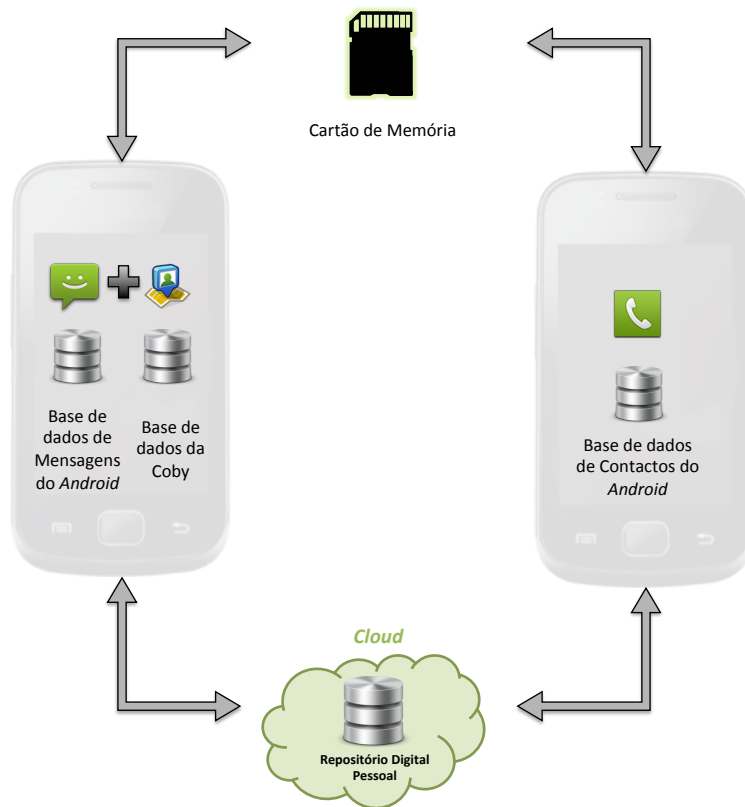


Figura 4.7: Fluxo de dados e bases de dados envolvidas.

Timestamp	Gsm Cell Id	Gsm Area Code	Gsm Mobile Country Code	Gsm Mobile Network Code	Latitude	Longitude	Address Street	Address Postal Code	Address Country	Provider	In Repository
-----------	-------------	---------------	-------------------------	-------------------------	----------	-----------	----------------	---------------------	-----------------	----------	---------------

Figura 4.8: Campos da tabela de mensagens da base de dados da aplicação CoBy.

obtida para garantir algum tipo de localização, por menos precisa que seja, caso haja alguma falha nas outras duas formas de localização. Então, assim que é conseguida tem de ser guardada na base de dados da aplicação. Posteriormente, se for obtida uma localização, do mesmo evento, mas mais precisa usando GPS ou *Network Location Provider* é necessário atualizar os dados previamente fornecidos por Cell-ID. O campo *provider* existente na base de dados da aplicação CoBy serve para saber qual foi o *provider* usado para localizar determinada mensagem, que numa fase inicial é sempre o Cell-ID, por ser o mais rápido a obter. Mas, posteriormente, será substituído por uma localização mais precisa, caso seja possível obter. O campo *provider* garante que os dados de localização guardados correspondem ao *provider* com maior precisão.

O campo *In Repository* indica para cada mensagem se esta foi enviada com sucesso para o repositório. Isto garante que o utilizador não tem informação replicada no repositório, apenas são enviados objetos digitais que ainda lá não estejam presentes. Este comportamento só se verifica no caso dos *backups* automáticos. O utilizador pode, por alguma razão, apagar determinado objeto digital do repositório e querer tê-lo de novo lá presente, para tal só precisa de fazer um *backup* manual desse objeto. Para o cartão de memória não foi usada esta estratégia, porque como se trata de uma forma de armazenamento em que não há necessidade de recorrer à Internet, mais facilmente se usa, sem haver grandes limitações em termos de espaço ou demora no envio.

Na base de dados da aplicação CoBy é maioritariamente guardada informação relativa à geo-localização. Todo o restante conteúdo das mensagens (remetente, destinatário, corpo da mensagem, etc.) não está presente nesta base de dados, devido ao facto de já se encontrar acessível a partir da base de dados de mensagens do *Android* (Figura 4.9). Evitando desta forma redundância de informação. A base de dados da aplicação CoBy encontra-se guardada no sistema de ficheiros do próprio dispositivo.

_id	thread_id	address	person	date	protocol	read	status	type	reply_path_present	subject	body	service_center	locked	error_code	seen
1	1	965349...	1	130090...	0	1	-1	1	0		Eu sou ...		0	0	1
2	2	965349...	2	130090...	0	1	-1	1	0		Eu sou ...		0	0	1
3	3	965349...	3	130090...	0	1	-1	1	0		Eu sou ...		0	0	1
4	4	965349...	4	130090...	0	1	-1	1	0		Maria....		0	0	1
5	5	965349...		130090...	0	1	-1	1	0		Eu sou ...		0	0	1
6	4	965349...		130090...		1	-1	2			Sim...Ra...		0	0	1
7	5	965349...		130090...		1	-1	2			Ola des...		0	0	1
8	3	965349...		130090...		1	-1	2			Ola Filipa		0	0	1
9	3	965349...		130090...		1	-1	2			Como e...		0	0	1
10	2	965349...		130090...		1	-1	2			Ola Eliz...		0	0	1
11	1	965349...		130090...		1	-1	2			Ola David		0	0	1
12	4	965349...	4	130090...	0	0	-1	1	0		Esta na...		0	0	1

Figura 4.9: Exemplo de uma tabela da base de dados de mensagens do *Android*

O *timestamp* estabelece a ligação entre a base de dados da aplicação CoBy e a de mensagens do *Android*. Este permite identificar, inequivocamente, um objeto nas duas bases de dados. Na base de dados de mensagens do *Android* o campo correspondente ao *timestamp* é o campo *Date*.

Caso o utilizador apague as suas mensagens, estas deixam de estar visíveis na aplicação CoBy. Apenas são listadas as mensagens que existem na base de dados de mensagens do *Android*. O ato de apagar uma mensagem é um ato consciente por parte do utilizador e significa que não pretende preservar aquele objeto digital, então também não deverá existir qualquer referência ao mesmo na CoBy.

De forma a permitir o acesso às bases de dados do *Android*, para leitura de mensagens

e contactos foi necessário requisitar autorização para tal no ficheiro "AndroidManifest.xml", Figura 4.10.

```
<uses-permission android:name="android.permission.READ_SMS" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
```

Figura 4.10: Permissões de acesso às mensagens e contactos presentes no ficheiro "Android-Manifest.xml".

Depois de criada a base de dados que dá suporte à aplicação CoBy, e a permissão de acesso garantido nas restantes bases de dados envolvidas, é necessário dar suporte às funcionalidades de *backup* e *restore*. A salvaguarda e o restauro dos objetos digitais pode ser feita para dois locais distintos: repositório digital pessoal e para o cartão de memória.

4.5 Repositório Digital Pessoal

Nesta aplicação o utilizador tem total controlo sobre os seus dados. Existem opções que permitem uma vigilância afincada sobre aquilo que o utilizador pretende ou não enviar para o repositório digital pessoal, assim como aquilo que pretende ou não preservar no cartão de memória do dispositivo móvel, como *backup*. O utilizador tem duas opções de *backup* para o repositório e cartão SD: manual e automático. O *backup* manual permite ao utilizador seleccionar exactamente aquilo que pretende ou não preservar no repositório. Caso o utilizador ative a opção *backup* automático, serão preservados todos os dados (contactos e mensagens). No caso em que esta tarefa é feita manualmente, pode existir alguma perda de informação. É ineficaz para o utilizador ter de desencadear um *backup* sempre que recebe ou cria um objeto, ou quando achar que é necessário. Acabando por fazê-lo, mas com intervalos temporais muito espaçados, correndo o risco de perder informação nesse período de tempo. Esta opção faz mais sentido quando o utilizador pretende fazer uma seriação mais pormenorizada dos seus dados. Os *backups* automáticos permitem libertar o utilizador da preocupação de salvaguardar os seus dados periodicamente, tendo neste caso, a possibilidade de definir na aplicação quando e de quanto em quanto tempo o pretende fazer.

A aplicação CoBy sincroniza-se com o repositório digital pessoal usando para isso um serviço *Representational State Transfer* (REST) [32, 33], que consome e produz dados no formato *JavaScript Object Notation* (JSON)³. Esta escolha deveu-se à sua simplicidade de codificação e leitura de dados. Na figura 4.11 encontram-se representados os campos a enviar para o *Web-service* (repositório digital pessoal), através de objetos JSON, comum aos vários tipos de objetos digital, com exceção daqueles que não estão sujeitos à contextualização adicional.

Consoante o tipo de objeto digital, existem alguns campos comuns e outros específicos de cada tipo de dados. Os campos a enviar relativos às mensagens seguem o estereótipo anterior. Na Figura 4.12 está ilustrado o formato completo dos campos de um objeto JSON do tipo SMS, que é constituído por:

- *Created Time*: identifica a data e hora da criação;

³<http://www.json.org/>

Created Date	Gsm Cell Id	Gsm Area Code	Gsm Mobile Country Code	Gsm Mobile Network Code	Latitude	Longitude	Address Street	Address Postal Code	Address Country	...
--------------	-------------	---------------	-------------------------	-------------------------	----------	-----------	----------------	---------------------	-----------------	-----

Figura 4.11: Campos usados para criação de objetos JSON comuns aos objetos digitais sujeitos a geo-localização.

- GSM cell Id, GSM Area Code, GSM Mobile Country Code, GSM Mobile Network Code, Longitude, Latitude, Address: representa informação de localização;
- Body: indica o corpo da SMS propriamente dito;
- Person Contact: representa o número da pessoa com a qual se está a estabelecer contacto;
- Own Contact: identifica o número do telemóvel do utilizador da aplicação.

Created Date	Gsm Cell Id	Gsm Area Code	Gsm Mobile Country Code	Gsm Mobile Network Code	Latitude	Longitude	Address Street	Address Postal Code	Address Country	Body	Person Contact	Own Contact
--------------	-------------	---------------	-------------------------	-------------------------	----------	-----------	----------------	---------------------	-----------------	------	----------------	-------------

Figura 4.12: Campos usados na criação dos objetos JSON para as mensagens.

Os contactos, contrariamente ao que acontece com as SMSs, não segue o estereótipo definido anteriormente, devido ao facto da geo-contextualização não se adequar a este tipo de objeto digital. Na Figura 4.13 encontra-se ilustrado o formato dos campos de um objeto JSON do tipo contactos.

Name	Numbers	Emails	Address
------	---------	--------	---------

Figura 4.13: Campos usados na criação dos objetos JSON para os contactos.

Para cada tipo de objeto digital (SMSs e contactos) são criados objetos JSON distintos. As SMSs e os contactos são enviados para o repositório separadamente. A Figura 4.14 mostra um exemplo de uma estrutura JSON enviada para o repositório digital pessoal.

Os objetos JSON foram criados manipulando classes como *JSONObject* e *JSONArray* disponíveis na API JSON suportada pela linguagem JAVA e incluída no *Android*. Depois de construídos os objetos JSON é necessário enviá-los para o repositório digital pessoal. Este envio é feito usando o protocolo *Hypertext Transfer Protocol* (HTTP). Quando é necessário enviar os dados para o repositório é feito um HTTP POST [34]. Neste tipo de pedidos é necessário indicar o URL do repositório, Figura 4.15.

Assim que o envio dos objetos digitais do tipo SMS, para o repositório esteja concluído é recebido um HTTP Response indicando o sucesso da operação. Caso tenha sido bem sucedido é ativado um serviço que escreve no campo *InRepository* presente na base de dados da aplicação. Desta forma, para cada mensagem que foi enviada é colocado o este campo a "1", indicando a sua presença no repositório.

Mensagens

```

{
  "SMS": [
    {
      "Received": [
        {
          "body": "...",
          "gsmAreaCode": "51",
          "personContact": "...",
          "gsmMobileNetworkCode": "01",
          "gsmCellId": "20456737",
          "ownContact": "...",
          "addressPostalCode": "3840-...",
          "longitude": "-8....",
          "latitude": "40....",
          "addressCountry": "Portugal",
          "addressStreet": "...",
          "gsmMobileCountryCode": "268",
          "createdDate": "2012-09-27T22:06:48+0100"
        },
        {
          "body": "Ok",
          "gsmAreaCode": "51",
          "personContact": "...",
          "gsmMobileNetworkCode": "01",
          "gsmCellId": "20456737",
          "ownContact": "...",
          "addressPostalCode": "3840-...",
          "longitude": "-8....",
          "latitude": "40....",
          "addressCountry": "Portugal",
          "addressStreet": "Rua Principal ...",
          "gsmMobileCountryCode": "268",
          "createdDate": "2012-09-27T22:16:41+0100"
        }
      ]
    },
    {
      "Sent": [
        {
          "body": "...",
          "gsmAreaCode": "51",
          "personContact": "...",
          "gsmMobileNetworkCode": "01",
          "gsmCellId": "20456737",
          "ownContact": "...",
          "addressPostalCode": "3840-...",
          "longitude": "-8....",
          "latitude": "40....",
          "addressCountry": "Portugal",
          "addressStreet": "Rua Principal ...",
          "gsmMobileCountryCode": "268",
          "createdDate": "2012-09-27T22:05:33+0100"
        },
        {
          "body": "...",
          "gsmAreaCode": "51",
          "personContact": "...",
          "gsmMobileNetworkCode": "01",
          "gsmCellId": "20456737",
          "ownContact": "...",
          "addressPostalCode": "3840-...",
          "longitude": "-8....",
          "latitude": "40....",
          "addressCountry": "Portugal",
          "addressStreet": "Rua Principal ...",
          "gsmMobileCountryCode": "268",
          "createdDate": "2012-09-27T22:08:33+0100"
        }
      ]
    }
  ]
}

```

Contactos

```

{
  "Contact": [
    {
      "Name": "...",
      "Emails": [
        "...",
        "..."
      ],
      "Address": "",
      "Numbers": [
        "..."
      ]
    },
    {
      "Name": "...",
      "Emails": [
        "...",
        "..."
      ],
      "Address": "",
      "Numbers": [
        "..."
      ]
    },
    {
      "Name": "Beatriz Lopes",
      "Emails": [
        "beatriz@hotmail.com",
        "beatriz@gmail.com"
      ],
      "Address": "Rua Principal\nAveiro 3840-357\nPortugal",
      "Numbers": [
        "915634090",
        "965349382"
      ]
    }
  ]
}

```

Figura 4.14: Estrutura JSON para objetos digitais do tipo SMS e contactos.

http://192.168.127.1:8080/
PersonalDigitalRepository/dataManagement/storeContent/Mobile/

Domínio a introduzir pelo utilizador na aplicação

Figura 4.15: Exemplo de um URL usado na aplicação CoBy.

Como o envio de dados para o repositório é feito a partir de um dispositivo móvel, que está em constante movimento, é fácil ocorrerem falhas na ligação à Internet. Um exemplo comum acontece quando o utilizador se esquece de que está a fazer *upload* de dados via Wi-Fi e sai do alcance da rede onde estava a fazê-lo. O envio dos dados pode ser feito de duas formas distintas: enviar os dados todos de uma só vez (usando o conceito de *Multipart*) ou enviar blocos de dados com tamanho máximo de 2Mb (limite padrão, configurável imposto pelos servidores usados na recepção dos dados). Tomando como exemplo a Figura 4.16 onde é mostrado o comportamento destas duas abordagens, sendo simulada uma falha na ligação no envio da SMS número 402, considerando um universo de 500 SMSs a enviar para o repositório. Na solução 1, são enviadas 401 SMS para o repositório, mas como a falha ocorreu antes do repositório responder, ou receber todos os dados, do ponto de vista da aplicação CoBy nenhuma das 500 SMSs se encontram no repositório. Se a falha ocorrer no envio dos dados, então eles não vão, de facto, estar no repositório, mas se a falha ocorrer ao receber a resposta de confirmação, estes encontram-se lá, mas não serão marcados como tal na aplicação. No *backup* automático seguinte irá para o repositório as 500 SMSs em questão juntamente com as que entretanto surgiram no dispositivo móvel. Na solução 2, a aplicação CoBy tem a indicação de que 400 SMSs já se encontram no repositório, falhando apenas 100. Assim, no *backup* automático seguinte serão enviadas 100 SMSs em conjunto com as que surgiram entretanto. O corpo de cada SMS tem 140 bytes de tamanho. Se forem enviadas 200 SMS prefaz um total de 28000 bytes. Então as 200 SMSs cabem perfeitamente num pacote HTTP POST de forma a respeitar um tamanho máximo de 2097152 bytes. Tendo em conta estes factos, foi escolhida a solução 2. Para os contactos foi usada a mesma abordagem.

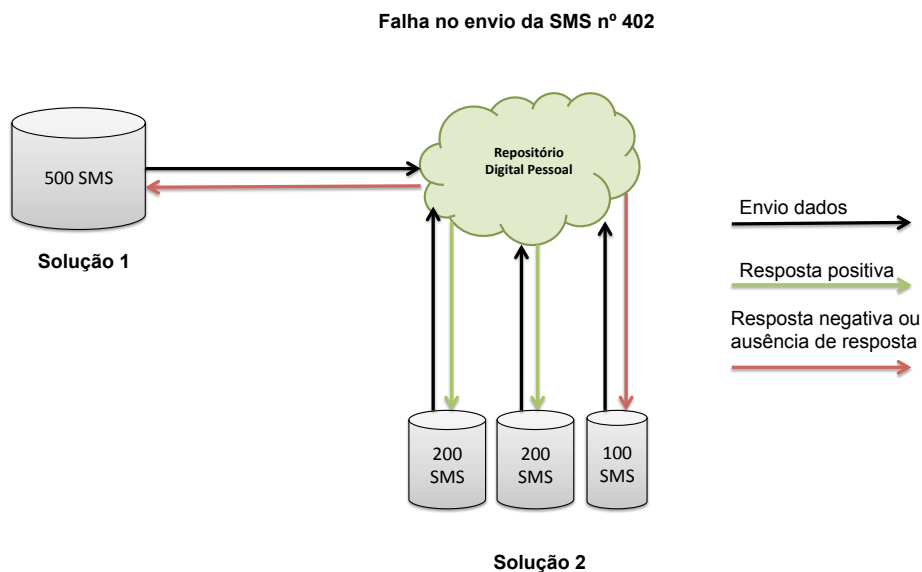


Figura 4.16: Exemplo ilustrativo das duas abordagens distintas para envio de dados para o repositório.

Quando o utilizador solicita um restauro a partir do repositório é feito um pedido através de HTTP *Request*. Os dados são enviados para o dispositivo móvel usando objetos JSON. Para extrair a informação relativa aos objetos digitais é necessário fazer *parsing* dos objetos JSON, que serão colocados nas bases de dados respetivas (da aplicação CoBy, mensagens e

contactos do *Android*). O conteúdo dessas bases de dados é previamente apagado antes do restauro propriamente dito. Isto, porque a verificação da existência das mensagens e contactos a restaurar iria implicar um consumo exaustivo dos recursos energéticos do dispositivo móvel. E caso não existisse essa verificação a possibilidade de ter informação replicada no dispositivo era muito grande, o que à partida não é desejável. Trata-se também da abordagem usada pelas aplicações de *backup* e restauro analisadas no capítulo Trabalho Relacionado.

Para dar suporte à funcionalidade de restauro, foi necessário requisitar permissões para escrita de mensagens e contactos no ficheiro "AndroidManifest.xml", Figura 4.17.

```
<uses-permission android:name="android.permission.WRITE_SMS" />
<uses-permission android:name="android.permission.WRITE_CONTACTS" />
```

Figura 4.17: Permissões de escrita das bases de dados presentes no ficheiro "AndroidManifest.xml".

Caso o utilizador achar que não precisa de preservar os seus objetos no repositório digital pessoal, pode simplesmente usar esta aplicação para fazer *backup* e *restore* a partir do cartão SD do seu dispositivo móvel. Mas se houver uma perda ou furto do mesmo, toda a informação perder-se-á. A não ser que o utilizador tenha o cuidado de copiar, regularmente, os documentos XML para outro dispositivo.

4.6 Cartão memória

As mensagens e os contactos são guardados no cartão de memória através de documentos no formato XML. Esta escolha deveu-se à elevada flexibilidade, portabilidade e facilidade de visualização dos dados, inerente a esta linguagem. Em alternativa ao formato XML, os dados poderiam ter sido guardados recorrendo ao formato JSON. Mas, a pensar no futuro desta aplicação a escolha recaiu sobre o formato XML porque este permite exportação e importação diretamente para múltiplos sistemas de bases de dados, *Database Management System* (DBMS) [35]. Quando esta aplicação for extendida a outros sistemas operativos, a troca de informação entre eles pode ser facilitada desta forma. Dada a utilização de ficheiros XML para preservar os dados no cartão de memória, é necessário o suporte a operações como: criação e *parsing* de documentos XML. Para tal, teve-se em conta duas APIs distintas: *Document Object Model* (DOM) e *Simple API for XML* (SAX) [36].

A API DOM suporta a leitura (*parsing*) e escrita de ficheiros XML. Esta API carrega o documento XML todo em memória e cria uma árvore do seu conteúdo. Para documentos grandes, tem a desvantagem de consumir mais recursos, nomeadamente memória. Contrariamente ao que acontece com a API SAX, esta tem um consumo muito baixo de recursos, o que a torna ideal para dispositivos móveis. Mas apenas suporta a operação de leitura (*parsing*) de documentos XML.

Na aplicação CoBy, dada a enorme vantagem que a API SAX apresenta, esta é usada na operação de leitura e a API DOM cobre a operação de escrita. A operação de escrita surge com a necessidade de fazer *backup* dos dados para o cartão de memória. A operação de leitura advém da funcionalidade de restauro.

Na Figura 4.18 encontra-se ilustrada a estrutura do conteúdo dos documentos XML criados para guardar as mensagens e os contactos. Os campos a guardar são exatamente os mesmos

que são enviados para o repositório. No caso dos contactos os dados são obtidos através de uma pesquisa simples na base de dados de contactos do *Android* (Figura 4.19) obtendo a informação que se pretende preservar. Os campos que estejam vazios na base de dados dos contactos, mensagens ou da CoBy são omitidos na criação do documento XML. No que toca às mensagens, a fonte de informação para a construção do documento XML provem de dois locais distintos (base de dados de mensagens do *Android* e da base de dados da aplicação). A base de dados da aplicação CoBy apenas tem informação de localização, é necessário obter os restantes dados de cada mensagem. Estes são obtidos por pesquisa na base de dados de mensagens do *Android* através do *timestamp*.

Mensagens

```
<?xml version="1.0" encoding="UTF-8"?>
<SMSset>
  <SMS>
    <Sent>
      <timestamp>134877933176</timestamp>
      <gsmCellId>20456737</gsmCellId>
      <gsmAreaCode>51</gsmAreaCode>
      <gsmMobileCountryCode>268</gsmMobileCountryCode>
      <gsmMobileNetworkCode>01</gsmMobileNetworkCode>
      <latitude>40. </latitude>
      <longitude>-8. </longitude>
      <addressStreet>Rua Principal </addressStreet>
      <addressPostalCode>3840 </addressPostalCode>
      <addressCountry>Portugal</addressCountry>
      <body> </body>
      <personContact> </personContact>
      <OwnContact> </OwnContact>
    </Sent>
  </SMS>
  <SMS>
    <Received>
      <timestamp>134878008230</timestamp>
      <gsmCellId>20456737</gsmCellId>
      <gsmAreaCode>51</gsmAreaCode>
      <gsmMobileCountryCode>268</gsmMobileCountryCode>
      <gsmMobileNetworkCode>01</gsmMobileNetworkCode>
      <latitude>40. </latitude>
      <longitude>-8. </longitude>
      <addressStreet>Rua Principal </addressStreet>
      <addressPostalCode>3840 </addressPostalCode>
      <addressCountry>Portugal</addressCountry>
      <body> </body>
      <personContact> </personContact>
      <OwnContact> </OwnContact>
    </Received>
  </SMS>
</SMSset>
```

Contactos

```
<?xml version="1.0" encoding="UTF-8"?>
<ContactsSet>
  <Contact>
    <name> </name>
    <numbers> </numbers>
    <emails> </emails>
  </Contact>
  <Contact>
    <name> </name>
    <numbers> </numbers>
    <emails> </emails>
  </Contact>
  <Contact>
    <name>Beatriz Lopes</name>
    <numbers>
      <number>915634090</number>
      <number>965349382</number>
    </numbers>
    <emails>
      <email>beatriz@hotmail.com</email>
      <email>beatriz@gmail.com</email>
    </emails>
    <address>Rua Principal Aveiro 3840-357 Portugal</address>
  </Contact>
</ContactsSet>
```

Figura 4.18: Estrutura do conteúdo dos documentos XML, para mensagens e contactos.

_id	is_restricted	account_name	account_type	sourceid	version	dirty	deleted	contact_id	aggregation_mode	aggregation_needed	custom_rington
1	0				2	1	0	1	0	0	
2	0				2	1	0	2	0	0	
send_to_voicemail	times_contacted	last_time_contacted	starred	display_name	display_name_alt	display_name_source	phonetic_name	phonetic_name_style			
0	0		0	Ana Maria	Maria, Ana	40		0			
0	0		0	Raquel Soares	Soares, Raquel	40		0			
sort_key	sort_key_alt	name_verified	contact_in_visible_group	sync1	sync2	sync3	sync4				
Ana Maria	Maria, Ana	0	1								
Raquel Soares	Soares, Raquel	0	1								

_id	package_id	mimetype_id	raw_contact_id	is_primary	is_super_primary	data_version	data1	data2	data3	data4	data5
1		5	1	0	0	0	915-364-852	1		258463...	
2		1	1	0	0	0	maria@gmail.com	1			
3		6	1	0	0	0	Ana Maria	Ana	Maria		
4		5	2	0	0	0	915-368-963	1		369863...	
5		6	2	0	0	0	Raquel Soares	Raquel	Soares		
6		1	2	0	0	0	raquel@gmail.com	1			

Figura 4.19: Exemplo de tabelas da base de dados de contactos do *Android*.

Caso o utilizador faça um *backup* manual, são guardados todos os dados que este selecionar. Caso tenha a opção de *backups* automáticos para o cartão de memória ativa, serão guardadas

todas as mensagens com localização e todos os contactos.

Os ficheiros de *backup* são guardados nas pastas: `/sdcard/Coby/Backups/Messages` e `/sdcard/Coby/Backups/Contacts` dependendo do tipo de objeto digital.

Para dar suporte à operação de restauro é necessário fazer *parsing* do documento XML previamente criado na opção *backup*, para extrair toda a informação lá presente, que posteriormente será colocada nas bases de dados correspondentes. O *parsing* do documento é feito usando a API SAX. Assim que o utilizador selecionar a opção de restauro, os dados das bases de dados envolvidas serão apagados, tal como acontece no restauro a partir do repositório, e substituídos pelos que se encontram no documento XML correspondente. Tanto nas mensagens como nos contactos é possível verificar que apenas parte dos campos são guardados no documento XML relativos às bases de dados do *Android*. Mas tratam-se de campos que aquando da substituição são preenchidos pelo sistema operativo. A título de exemplo pode-se considerar o campo *thread_id* da base de dados de mensagens do *Android* que permite agrupar sob a forma de conversa todas as mensagens trocadas com determinado contacto (uma por cada número de telemóvel). As mensagens trocadas com o mesmo número de telemóvel tem o mesmo *thread_id*. Trata-se de um campo de preenchimento necessário, mas tal tarefa é delegada ao sistema operativo.

A escrita no cartão de memória só é possível mediante requisição para tal no "Android-Manifest.xml", Figura 4.20.

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Figura 4.20: Permissões para escrita no cartão de memória do dispositivo móvel presentes no ficheiro "AndroidManifest.xml".

Protótipo

5.1 Interface gráfica

O *Android* SDK disponibiliza uma variedade de ferramentas que permitem construir a interface gráfica da aplicação. Esta é construída usando componentes como: *views*, *view groups* e *activities*. *Activity* corresponde à representação visual da aplicação que permite a interação com o utilizador (Anexo D). Uma aplicação pode ser construída recorrendo a várias atividades (*activities*). Uma atividade é composta por *views* que dão suporte aos mais variados tipos de *widgets* (*button*, *TextView*, *TextEdit*, *CheckBox*, etc.). O elemento *View Groups* permite agregar vários objetos do tipo *view* e são também conhecidos como gestores de *layouts*. Um *layout* representa a estrutura visual de uma atividade. Todos estes componentes foram construídos usando a linguagem XML. Para o lançamento de uma atividade é necessário recorrer à declaração abstrata que define uma intenção (*Intent*). Um *Intent* representa uma mensagem indicando aquilo que se pretende efetuar. A aplicação ou o sistema operativo reagem a esse *Intent*. Nesta aplicação também foram usados *Intents* para o lançamento do *Broadcast* e dos serviços.

O facto desta aplicação ser destinada a dispositivos móveis faz com que os desafios apresentados e as metáforas de utilização sejam diferentes das empregadas noutras plataformas. Aquando da construção da interface gráfica, *User Interface* (UI), há determinados aspetos inalteráveis e comuns a todos os dispositivos, que tem de ser tomados em conta [37]:

- Ecrã considerado reduzido;
- Recursos hardware limitados;
- Uso do dedo como apontador para interação com o dispositivo;
- Existência de mais do que um apontador, com tamanhos muito diferentes;
- Ausência de rato;
- Inexistência de teclado físico;
- Existência teclado construído por software, ocupando espaço no ecrã, restando menos espaço para a aplicação;
- Dificuldade acrescida na introdução de dados.

Quando se desenvolve uma aplicação deste tipo, é importante que a sua interface gráfica reflita alguns dos seguintes requisitos:

- Simplicidade, para não confundir o utilizador;

- Usar *widgets* com tamanho suficiente para serem sensíveis ao toque e ao mesmo tempo não ocupem demasiado espaço no ecrã;
- Não mostrar demasiada informação ao utilizador de uma só vez;
- Fácil aprendizagem;
- Minimizar o número de toques;
- Evitar erros da parte do utilizador
- Manter a consistência com os padrões usados pelo sistema operativo;
- Dar *feedback* ao utilizador;

Tendo em consideração as limitações e especificações inerentes aos dispositivos móveis, construiu-se a interface gráfica da aplicação CoBy, que se encontra descrita a seguir.

5.1.1 *Account*

Para uma primeira utilização desta aplicação, o utilizador necessita de preencher alguns campos necessários ao seu bom funcionamento. Na Figura 5.1 estão presentes esses campos, e todos eles são sujeitos a validação. Caso sejam colocados dados num campo que não correspondam ao tipo de dados a que está destinado, é mostrada uma mensagem com essa indicação. O utilizador só poderá avançar para a *Main Activity* da aplicação se preencher os campos corretamente. A mensagem de alerta que é mostrada recorre ao mecanismo *Toast Notifications* disponível pelo *Android*, que mostra uma mensagem sobre a forma de *pop-up*. Dado o seu destaque é quase certo que o utilizador a vê, sendo esse o objetivo. Só há obrigatoriedade de preencher este formulário quando o utilizador instala ou usa a aplicação pela primeira vez. Ficando depois disponível para modificação no menu da *Main Activity*.

O nome e número de telemóvel, essenciais à aplicação, são os únicos campos obrigatórios pedidos assim que o utilizador usa a aplicação pela primeira vez. Os restantes não são obrigatórios, numa primeira fase, com o objetivo do utilizador poder usar esta aplicação escolhendo unicamente um dos destinos disponíveis para os seus dados (cartão de memória ou repositório digital). Os campos *Email*, *Repository Url*, *Password* são obrigatórios aos *backups* com destino ao repositório digital. Se no momento de criar um *backup*, para este destino, estes campos não se encontrarem previamente preenchidos, o utilizador é informado através de uma notificação e o *backup* é impedido de prosseguir. O campo *Email* é obtido automaticamente de uma das contas configuradas no dispositivo móvel, podendo ser alterado se pretender utilizar outro. O *backup* para o cartão de memória é garantido apenas com os campos pedidos quando a aplicação é usada pela primeira vez.

Os dados pedidos nesta *Activity* são guardados nas preferências da aplicação, usando a interface *SharedPreferences*. Esta interface é usada para guardar rapidamente e eficientemente os dados num documento XML no dispositivo móvel.

5.1.2 *Main Activity*

Depois dos dados pessoais devidamente preenchidos, o utilizador é encaminhado para a *Main Activity* da aplicação, Figura 5.2. Aqui estão disponíveis três botões com as seguintes funcionalidades:

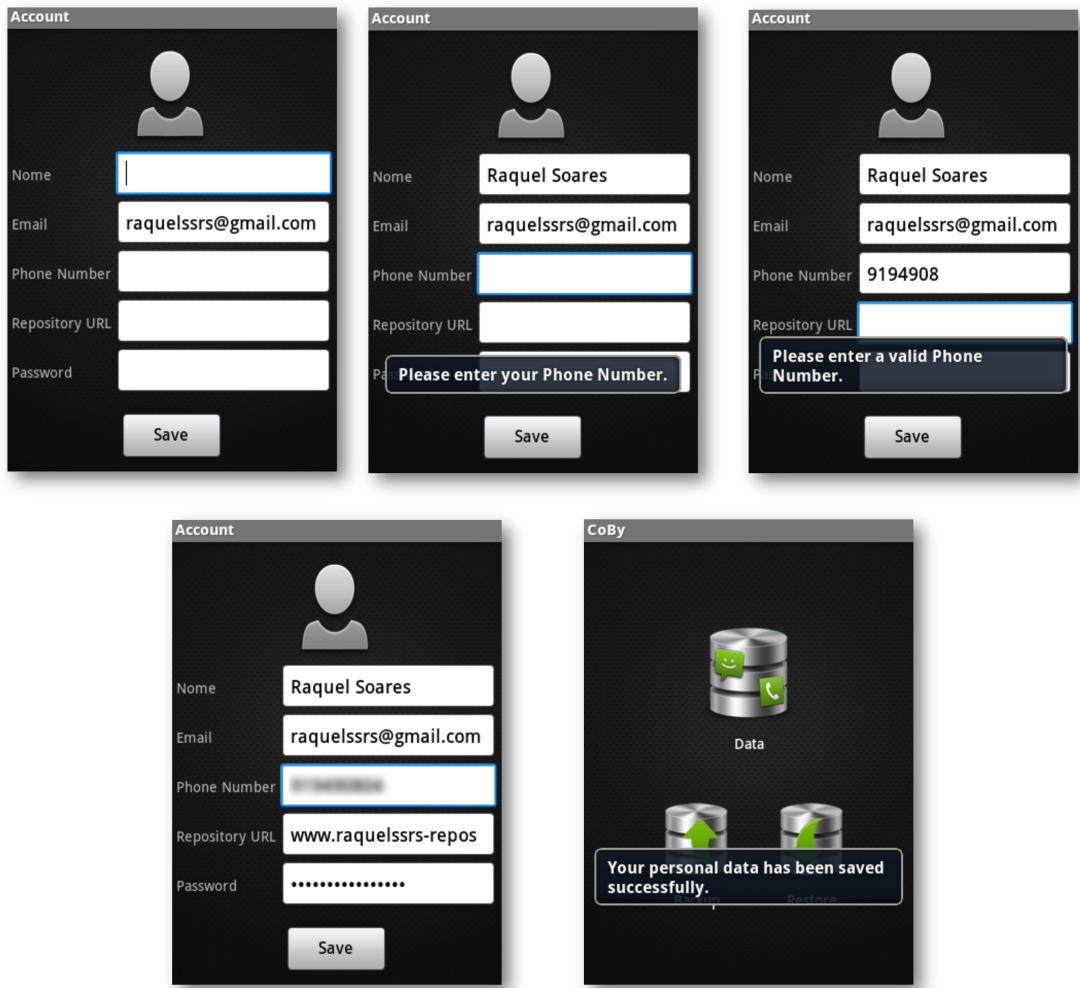


Figura 5.1: *Account Activity.*

- *Data*: admite o acesso à informação detalhada dos contactos e das SMSs;
- *Backup*: permite ao utilizador efetuar o *backup* manual de SMSs e contactos;
- *Restore*: possibilita o restauro dos dados (contactos e SMSs) a partir do cartão de memória e do repositório digital pessoal.

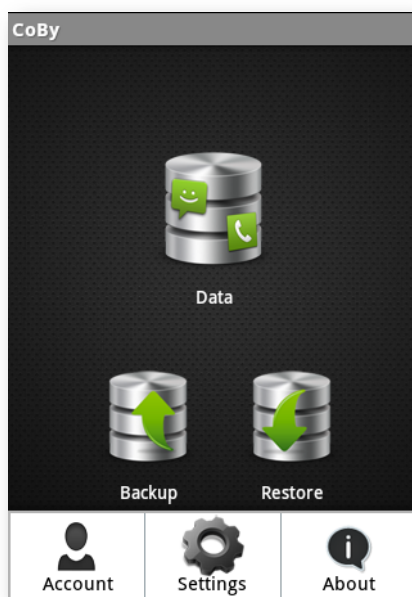


Figura 5.2: *Main Activity*.

Adicionalmente existe um menu, apenas visível mediante clique no botão físico "Menu" do dispositivo móvel, e permite o acesso às seguintes atividades:

- *Account*: onde o utilizador pode editar os seus dados pessoais;
- *Settings*: disponibiliza funcionalidades relativas ao *backup* automático;
- *About*: contém informação relativa à aplicação.

5.1.3 *Settings*

Através do menu acessível a partir da *Main Activity*, o utilizador tem a possibilidade de indicar alguns parâmetros que definem o comportamento da aplicação, Figura 5.3.

Uma das funcionalidades implementadas nesta aplicação são os *backups* automáticos. A frequência com que são efetuados pode aqui ser definida. O utilizador tem a possibilidade de o fazer diariamente, semanalmente ou mensalmente, mediante a sua preferência. Caso opte por efetuar *backups* diários, apenas necessita de definir a hora, que mais lhe convier. Se for selecionada a opção semanal, o utilizador só carece de indicar a que dia da semana pretende, assim como a respetiva hora do dia. Caso escolha o *backup* mensal, há necessidade de especificar o dia do mês e a hora a que tenciona fazê-lo. Quando o utilizador não precisa

de seleccionar determinado parâmetro, em função do tipo de agendamento que está a fazer, tem as restantes opções desativadas, para desta forma minimizar os possíveis erros cometidos. Por exemplo, quando o utilizador escolhe *backups* diários, apenas necessita de definir a hora, tendo as restantes opções (dia da semana, dia do mês) desativadas. A escolha da data e da hora é feita usando um *DatePickerDialog* e um *TimePickerDialog* respetivamente.

O local para onde o utilizador pretende efetuar o *backup* automático, também pode ser escolhido. Tem a opção de envio para o repositório e/ou cartão de memória. Ativando esta opção todas as mensagens e todos os contactos serão tidos em conta, e enviados para o destino. Para desativar os *backups* automáticos basta remover a seleção das opções *SDCard* e *Repository*. O utilizador tem ainda a possibilidade de escolher se pretende fazer um *backup* com ou sem sua interação, que será explicado mais à frente em que consiste.

Os *backups* automáticos só são de facto ativados depois da escolha do local para onde o pretende fazer bem como a sua frequência.

Existe ainda a possibilidade de só efetuar *backups* automáticos caso o dispositivo móvel tenha acesso a uma ligação à Internet via Wi-Fi.

5.1.4 *Data*

Na atividade que dá suporte à visualização dos dados, estão presentes duas *Tabs* que permitem o acesso à informação relativa às SMSs e aos contactos, 5.4.

Mensagens

Na Figura 5.5 estão presentes as várias atividades que permitem a visualização das mensagens e dos detalhes associados.

Caso a aplicação esteja a ser usada pela primeira vez, ou o utilizador ainda não tenha trocado mensagens depois de a instalar, tem acesso à atividade a) representada na Figura 5.5, que indica que ainda não existe nenhuma mensagem recebida ou enviada entretanto. Caso o utilizador tenha trocado mensagens depois da instalação da aplicação terá acesso à mesma atividade, mas com um *layout* diferente, Figura 5.5 b), que lista todas as mensagens recebidas ou enviadas. Nessa lista encontram-se, também, visíveis alguns detalhes como: o remetente, destinatário, coordenadas de geo-localização e corpo associados a cada mensagem. Se pretender obter mais detalhes sobre determinada SMS, basta clicar sobre a mesma e é encaminhado para a atividade Figura 5.5 c), responsável por mostrar informação como: rua, código postal, data, etc. Existe ainda a possibilidade do utilizador consultar no mapa, o local onde a mensagem foi recebida ou enviada, consoante o caso. Se o utilizador não tiver nenhuma ligação à Internet disponível, é-lhe dado essa indicado através da mensagem "*There is no Internet connection.*". Esta é necessária para carregar os mapas.

Contactos

Como os contactos não estão sujeitos à geo-contextualização, serão sempre listados todos os que o utilizador apresentar no dispositivo móvel, Figura 5.6. Na lista de contactos apenas é mostrado o respetivo nome, bem como até dois números de telefone a ele associados. Se o utilizador pretender saber mais detalhes sobre determinado contacto, nomeadamente um email associado, basta clicar sobre ele.

A aplicação CoBy foi implementada de forma a suportar os dois tipos de orientação do ecrã do dispositivo móvel: *Portrait* (Figura 5.6 a) e b)) e *Landscape* (Figura 5.6 c)).



Figura 5.3: Definições suportadas.



Figura 5.4: *Tabs* usadas para diferenciar mensagens e contactos.

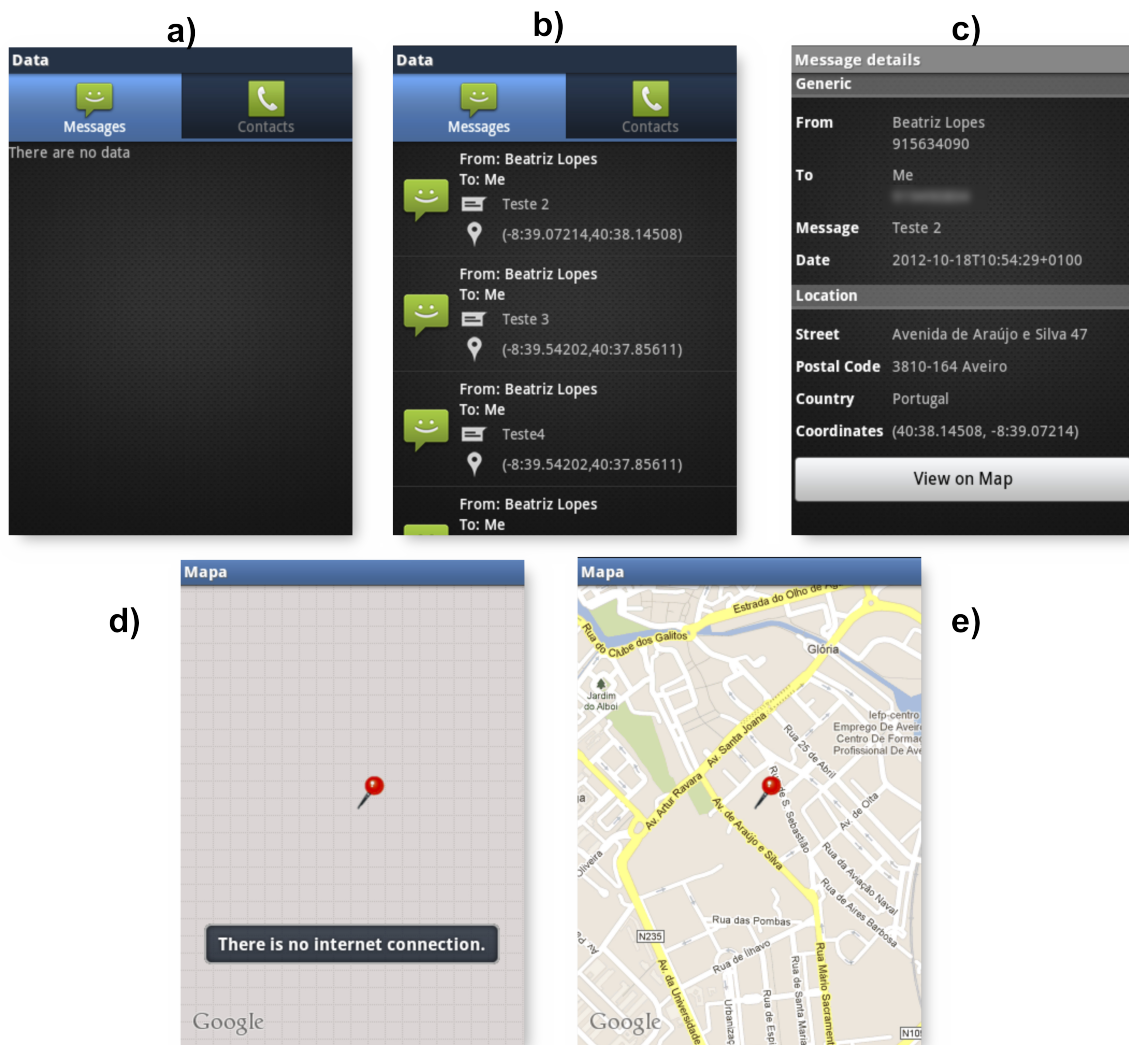


Figura 5.5: Visualização dos detalhes relativos às mensagens.

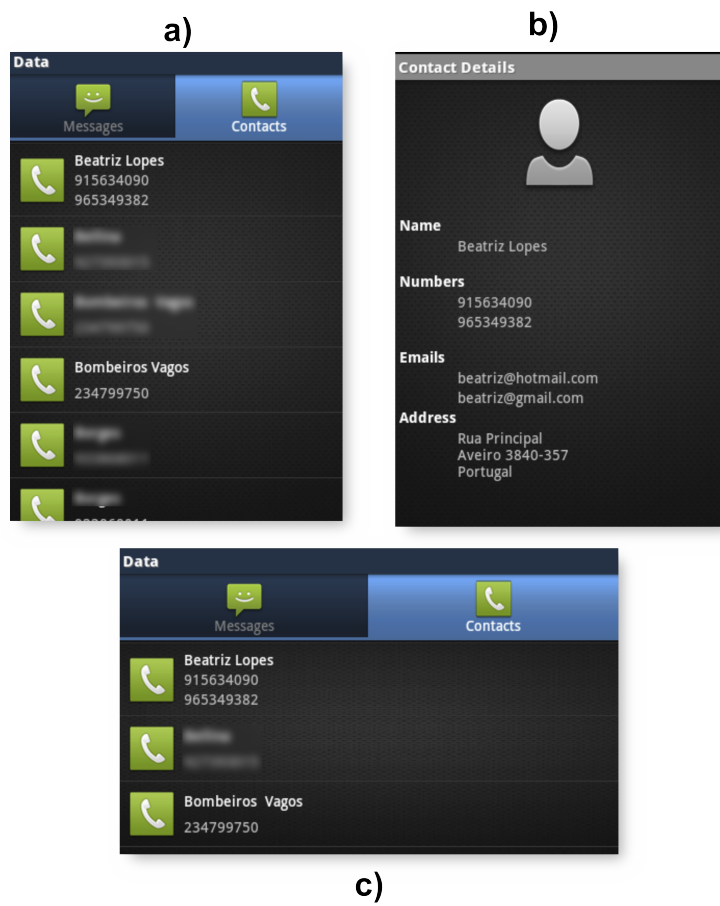


Figura 5.6: Visualização dos detalhes relativos aos contactos.

5.1.5 Backup Automático

Os *backups* automáticos podem ser feitos com e sem intervenção do utilizador, dependendo da opção selecionada nas definições da aplicação. O *backup* automático sem intervenção do utilizador é desencadeado, conforme as configurações de agendamento indicadas, sem que o utilizador precise de interagir com a aplicação. Este opção é útil para *backups* noturnos. Se o utilizador não tiver nenhuma ligação à Internet, e tiver a opção de *backup* para o repositório ativa, este não é efetuado. Se o utilizador assume o compromisso de não intervir na realização desta tarefa tem de garantir as condições necessárias à sua realização. A aplicação podia ativar automaticamente a Internet (GSM ou Wi-Fi caso estivesse abrangido por alguma rede), mas considera-se uma medida muito invasiva para o utilizador. Concluído o *backup* o utilizador é notificado, usando para isso *NotificationManager*, 5.7.



Figura 5.7: Notificação que reporta o sucesso ou insucesso do *backup* automático sem intervenção do utilizador.

Sempre que um *backup* se encontrar agendado, e o utilizador pretender a sua intervenção nesta tarefa é mostrada a atividade representada na Figura 5.8. Permite ao utilizador escolher se de facto quer fazer aquele *backup*, naquele preciso momento. Caso não pretenda, é descartado esse *backup* e agendado o seguinte em função das definições predefinidas pelo utilizador na opção *settings*. Caso opte por fazer o *backup* naquele momento, e se o utilizador tiver a opção de *backup* para o repositório selecionada, é averiguado se existe alguma ligação à Internet ativa. Se não existir é indicado ao utilizador, e a aplicação aguarda até que seja estabelecida essa ligação. Assim que tal aconteça os dados são enviados, e depois de concluído o envio, é mostrada a mensagem "Your data was successful saved." com essa indicação, usando um *AlertDialog*. Caso ocorra algum problema durante o *backup* é mostrado ao utilizador a mensagem "Your data was NOT successful saved. Please try again."

Nesta aplicação existiu a preocupação de dar algum *feedback* ao utilizador sobre as operações que por ele vão sendo efetuadas.

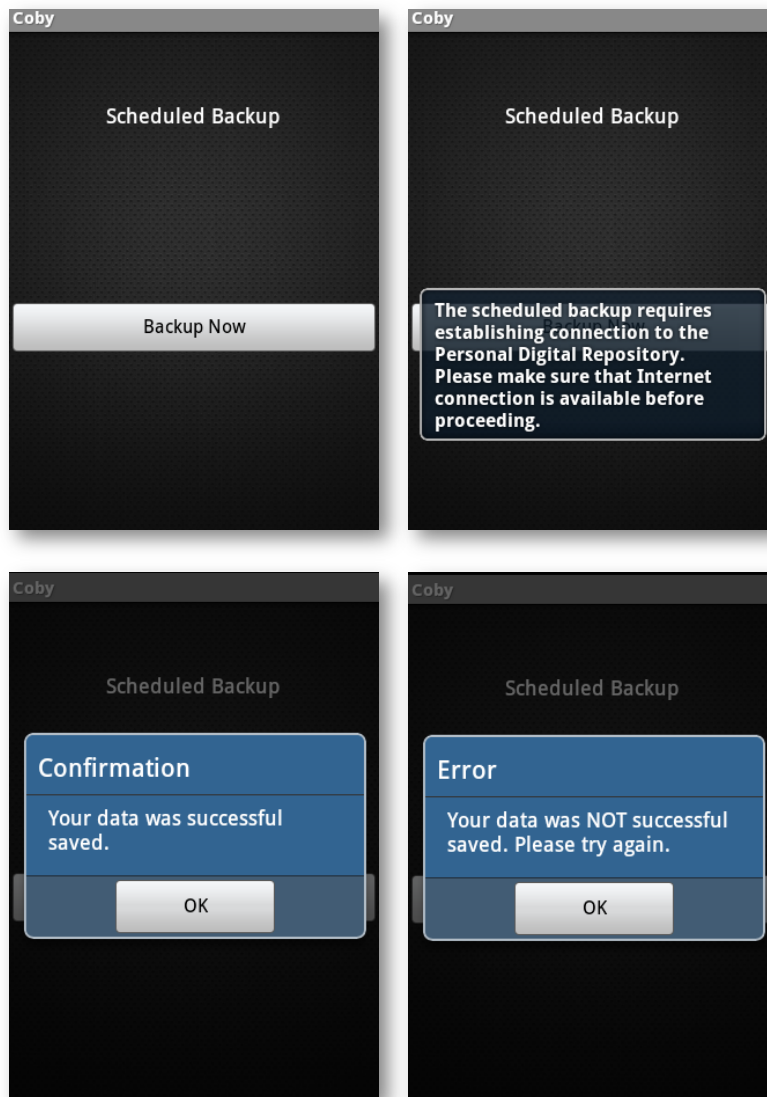


Figura 5.8: *Backup* automático de Mensagens e Contactos.

5.1.6 Backup Manual

O utilizador, através do *backup* manual, tem a possibilidade de selecionar exatamente aquilo que pretende ou não preservar, tanto para as mensagens como contactos, Figura 5.9. Está disponível uma *CheckBox* para cada mensagens e contacto, permitindo uma seleção individual. É essencial ter pelo menos um objeto digital selecionado para poder fazer um *backup*. Este é feito individualmente para cada tipo de dados. Caso pretenda fazer um *backup* manual de todo o conteúdo, pode sempre usar a opção *select all*, que permite selecionar todas as mensagens ou contactos, dependendo da atividade onde se encontrar. No *backup* manual o utilizador pode igualmente escolher o destino dos seus dados (repositório e/ou cartão de memória, Figura 5.9 e)). Caso tenha a opção de *backup* para o repositório ativa, é averiguado se dispõe de uma ligação à Internet disponível. No caso negativo, o *backup* é cancelado, Figura 5.9 f). No caso positivo o *backup* é efetuado, e logo que concluído, é indicado ao utilizador, Figura 5.9 g). Se este ambicionar fazer um *backup* para o cartão de memória é investigado a sua existência. Caso não exista o utilizador é informado, e o *backup* para esse destino é cancelado.

5.1.7 Restore

Na atividade responsável pelo restauro dos dados, existem quatro *Tabs* que permitem diferenciar a origem e o tipo de dados em questão, Figura 5.10.

O restauro de mensagens e contactos é feito individualmente a partir do repositório digital pessoal ou do cartão de memória, Figura 5.11. No caso do repositório pessoal o utilizador tem a possibilidade de restaurar o conteúdo do último *backup* efetuado (independentemente de ter sido executado automaticamente ou manualmente), Figura 5.11 b) e c). Se o utilizador ainda não tiver feito nenhum *backup* para o repositório ou cartão de memória, Figura 5.11 a), é-lhe indicado através da mensagens "A previous backup is not available at the moment. To restore your data you must first perform a backup to either the repository or the SD card." usando *Toast Notifications*. Quando o utilizador deseja fazer um restauro a partir do cartão de memória, tem ao seu dispor todas as versões de *backup* que efetuou, Figura 5.11 d) e f). Estas são listadas, e o nome representa a data de criação do *backup*. Desta forma, o utilizador mais facilmente poder selecionar a versão do *backup* que pretende restaurar. Após clicar sobre a versão que pretende é mostrado um *AlertDialog*, Figura 5.11 e), para confirmar que tal ato foi intencional. Desta forma previne que o utilizador, cometa o erro de substituir os seu dados (contactos ou mensagens), caso não pretenda. Quando o faz, está completamente ciente da opção que tomou.

Tendo em conta os requisitos e funcionalidades da aplicação bem como as limitações dos dispositivos móveis construiu-se uma aplicação com o objetivo de obter uma interface gráfica simples, funcional, de fácil utilização e intuitiva. Teve-se em conta os padrões usados pelo sistema operativo *Android*, e tentou-se refletir isso na aplicação CoBy. Sempre que possível, foram empregados os mesmos *icons* usados pelo sistema operativo, de forma a não causar estranheza no utilizador, e mais facilmente perceber o seu significado.

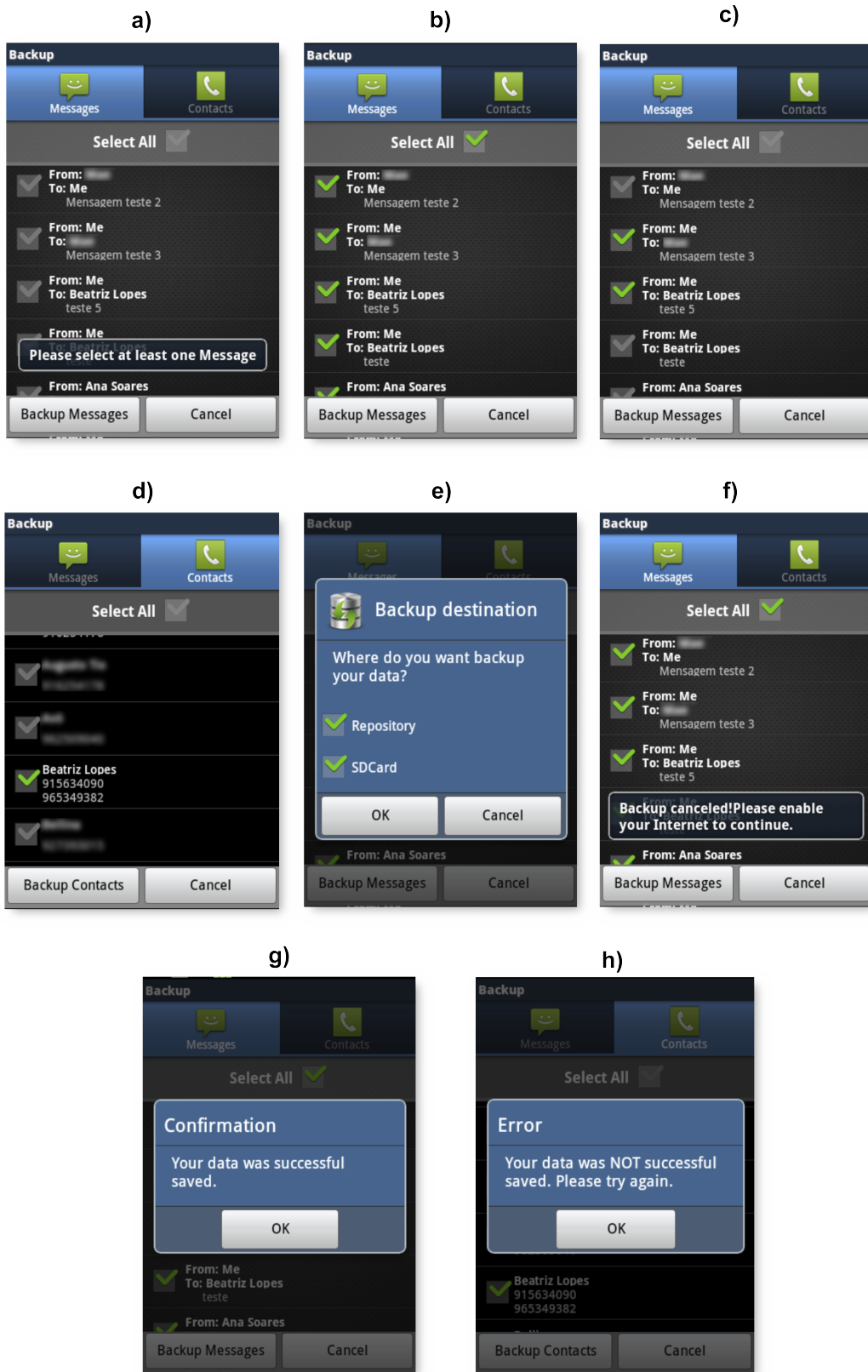


Figura 5.9: Backup manual de Mensagens e Contactos.

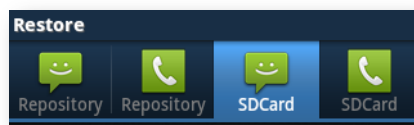


Figura 5.10: *Tabs* usadas para diferenciar mensagens e contactos no repositório e cartão de memória.



Figura 5.11: *Restore* de Mensagens e Contactos a partir do repositório digital pessoal bem como do cartão SD do dispositivo móvel.

5.2 Ambiente

5.2.1 Desenvolvimento

A aplicação CoBy foi desenvolvida usando linguagem Java, aproveitando as ferramentas disponíveis no pacote *Android SDK*, Figura 5.12. Foi usado o *Eclipse Integrated Development Environment* (IDE) como plataforma de desenvolvimento juntamente com o *plugin Android Development Tools* (ADT). Esta aplicação foi construída com destino a dispositivos móveis com sistema operativo *Android* versão 2.3.3, também designada por *Gingerbread*. Esta escolha deveu-se ao facto desta versão ser a mais usada pelos utilizadores, atualmente, em dispositivos móveis reais com sistema operativo *Android*, tal como é ilustrado no Anexo E.



Figura 5.12: Logótipos dos elementos envolvidos na construção da aplicação.

5.2.2 Teste

O ambiente de criação e teste desta aplicação passou por dois locais distintos. Inicialmente, apenas se dispunha de um emulador de dispositivos móveis presente no *Android SDK*. Posteriormente, já foi possível utilizar um dispositivo real para correr a aplicação.

O emulador, Figura 5.13, é muito útil para desenvolver aplicações para *Android*, quando não se dispõe de um dispositivo real. Este, permite simular os recursos de *software* e *hardware* disponíveis num dispositivo móvel.

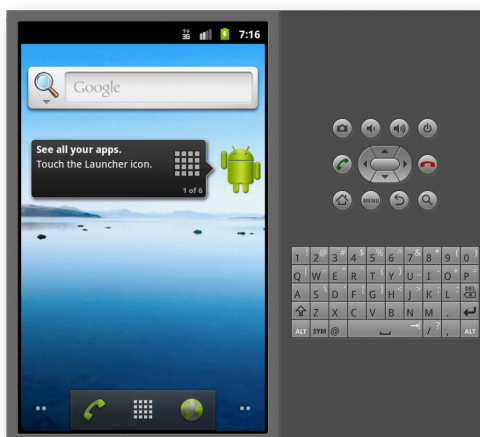


Figura 5.13: Emulador de dispositivos móveis *Android*.

Sendo um emulador e não um dispositivo real, é normal que apresente algumas limitações e que algumas funcionalidades e comportamentos não sejam suportados. Olhando apenas

para as limitações que abrangeram a aplicação CoBy, existiram dois comportamentos que tiveram de ser simulados: a recepção de mensagens e a obtenção localização, Figura 5.14.



Figura 5.14: Simulação da localização e da recepção de mensagens.

Para criação e teste da aplicação, numa segunda fase, foi usado um *Samsung Galaxy Gio* como dispositivo móvel real, Figura 5.15. Desta forma é possível obter resultados mais fidedignos, comparativamente ao emulador. Deixando de ser necessário a simulação de alguns comportamentos ou parâmetros, como é o caso da localização.



Figura 5.15: Dispositivo móvel real usado.

Conclusões

6.1 Balanço Geral

Para fazer um balanço geral de todo o cenário que envolve a aplicação CoBy, foi elaborada uma análise de SWOT (*Strengths, Weaknesses, Opportunities, Threats*), fornecendo um apanhado das características que favorecem ou não esta aplicação, Figura 6.1.



Figura 6.1: Análise SWOT.

Os objetivos traçados para este projeto foram cumpridos. Apresentou-se a aplicação CoBy como protótipo de um modelo de recolha, preservação e contextualização de informação. Esta aplicação demonstra uma elevada utilidade para utilizadores de dispositivos móveis. Trata-se de uma aplicação com capacidade de *backup* automático e *restore* de contactos e SMSs devidamente contextualizadas. O utilizador pode fazer uma cópia de segurança e restauro não só para o seu repositório digital com também para cartão de memória do seu dispositivo. A sua ligação a um repositório digital pessoal permite colmatar o problema da acelerada obsolescência inerente às diversas tecnologias e softwares que vão surgindo. Assim, vê-se garantida a preservação persistente e continuada do material digital, sem receio de perda, deterioração ou corrupção dos mesmos. A contextualização permite ao utilizador a criação de uma imagem mental inteligível de tudo aquilo que o rodeava quando recebeu, criou ou enviou o objeto digital. Permitindo, desta forma não só preservar o conteúdo propriamente dito, mas também a imagem mental circundante. Com esta aplicação, o utilizador consegue salvaguardar as SMSs e contactos que achar que tem relevância para tal e dar um grande contributo para a escrita da sua própria história, não só pessoal mas também profissional.

Adicionalmente, foi publicado um artigo científico [38] que retrata a fase inicial deste projeto, na a revista Revista Ibérica de Sistemas e Tecnologias de Informação (RISTI).

6.2 Trabalho Futuro

Com um método semelhante ao descrito para as SMSs e contactos, a aplicação CoBy poderá permitir não só a recolha, preservação e contextualização destes objetos, mas também de imagens, *Multimedia Messaging Service* (MMS), emails, fotos, *bookmarks*, músicas, calendário, etc. Em suma, todo o material digital disponível no dispositivo móvel e que seja do interesse do utilizador preservar. Outra funcionalidade que poderá ser implementada na aplicação consiste na possibilidade de indicar ao repositório digital pessoal que apenas pretende fazer restauro dos dados compreendidos em determinado período temporal. Poderão, também, ser efetuados testes de usabilidade de forma a tornar a aplicação mais robusta. Em dispositivos móveis sem suporte para cartões de memória poderá, alternativamente, ser implementado o *backup* com destino à memória interna. Futuramente, seguindo o modelo apresentado neste documento, poderá ser possível criar versões da aplicação CoBy para outros sistemas operativos móveis, o que permitirá uma maior cobertura dos dispositivos existentes.

Bibliografia

- [1] CCSDS, “Reference model for an open archival information system (oais),” June 2012.
- [2] K. Saidis and A. Delis, “Type-consistent digital objects,” *D-Lib Magazine*, vol. 13, no. 5, p. 4, 2007.
- [3] R. Da Costa, “História e memória: a importância da preservação e da recordação do passado,” *SINAIS - Revista Eletrônica - Ciências Sociais*, vol. 1, no. 2, pp. 02–15, 2007.
- [4] R. Heery and S. Anderson, “Digital repositories review,” 2005.
- [5] P. Mehra, “Context-aware computing: Beyond search and location-based services,” *Internet Computing, IEEE*, vol. 16, no. 2, pp. 12–16, 2012.
- [6] G. Abowd, M. Ebling, G. Hung, H. Lei, and H. Gellersen, “Context-aware computing,” *Pervasive Computing, IEEE*, vol. 99, no. 3, pp. 22–23, 2002.
- [7] A. Dey, “Understanding and using context,” *Personal and ubiquitous computing*, vol. 5, no. 1, pp. 4–7, 2001.
- [8] J. Pascoe, “Adding generic contextual capabilities to wearable computers,” in *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*. IEEE, 1998, pp. 92–99.
- [9] M. Arellano, “Preservation of digital documents,” *Ciência da Informação*, vol. 33, no. 2, pp. 15–27, 2004.
- [10] S. Hashimi, S. Komatineni, and D. MacLean, *Pro Android 2*. Apress, 2010, no. 2.
- [11] S. Brahler, “Analysis of the android architecture,” *Karlsruhe institute for technology*, 2010.
- [12] V. Nahavandipoor, *iOS 5 Programming Cookbook: Solutions & Examples for iPhone, iPad, and iPod touch Apps*. O’Reilly Media, Incorporated, 2012.
- [13] F. Ancarani and V. Shankar, “Symbian: Customer interaction through collaboration and competition in a convergent industry,” *Journal of interactive marketing*, vol. 17, no. 1, pp. 56–76, 2003.
- [14] S. Babin, *Developing Software for Symbian OS 2nd Edition: A Beginner’s Guide to Creating Symbian OS v9 Smartphone Applications in C++*. Wiley, 2008.
- [15] Q. Mahmoud, “A mobile web-based approach to introductory programming,” in *Pro-*

- ceedings of the 16th annual joint conference on Innovation and technology in computer science education.* ACM, 2011, pp. 334–334.
- [16] C. Petzold, *Programming Windows Phone 7*, D. Musgrave, Ed. Microsoft Press, 2010.
- [17] Gartner. (2012, 09) Worldwide mobile device sales to end users. [Online]. Available: <http://www.gartner.com/technology/home.jsp>
- [18] I. Darwin, *Android Cookbook*. O’Reilly Media, 2012.
- [19] Google. (2012, 10) Google play. [Online]. Available: <https://play.google.com/store>
- [20] R. Sahu. (2012, 9) Sms backup & restore. [Online]. Available: <http://android.riteshsahu.com/apps/sms-backup-restore>
- [21] J. Sousa, M. Pereira, and J. A. Martins, “Improving browser history using semantic information,” *ICEIS 2012 - 14th International Conference on Enterprise Information Systems.*, 2012.
- [22] I. Neustadt and J. ARLOW, *UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design*. Addison-Wesley Professional, 2005.
- [23] V. Ottaviani, A. Lentini, A. Grillo, S. Di Cesare, and G. Italiano, “Shared backup & restore: Save, recover and share personal information into closed groups of smartphones,” in *New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on*. IEEE, 2011, pp. 1–5.
- [24] Z. Zhuang, K. Kim, and J. Singh, “Improving energy efficiency of location sensing on smartphones,” in *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 2010, pp. 315–330.
- [25] R. Meier, *Professional Android 2 application development*. wrox, 2010.
- [26] S. Kumar, M. Qadeer, and A. Gupta, “Location based services using android (lbsoid),” in *Internet Multimedia Services Architecture and Applications (IMSAA), 2009 IEEE International Conference on*. Ieee, 2009, pp. 1–5.
- [27] E. Kaplan and C. Hegarty, *Understanding GPS: principles and applications*. Artech House Publishers, 2006.
- [28] G. Djuknic and R. Richton, “Geolocation and assisted gps,” *Computer*, vol. 34, no. 2, pp. 123–125, 2001.
- [29] E. Trevisani and A. Vitaletti, “Cell-id location technique, limits and benefits: an experimental study,” in *Mobile Computing Systems and Applications, 2004. WMCSA 2004. Sixth IEEE Workshop on*. IEEE, 2004, pp. 51–60.
- [30] X. Shu, Z. Du, and R. Chen, “Research on mobile location service design based on android,” in *Wireless Communications, Networking and Mobile Computing, 2009. Wi-Com’09. 5th International Conference on*. IEEE, 2009, pp. 1–4.
- [31] M. Owens, *The definitive guide to SQLite*. Apress, 2006.

- [32] A. Cobârzan, “Consuming web services on mobile platforms,” *Informatica Economică*, vol. 14, no. 3, pp. 1453–1305, 2010.
- [33] H. He, “What is service-oriented architecture,” *Publicação eletrônica em*, vol. 30, 2003.
- [34] (2012, 9) Hypertext transfer protocol. [Online]. Available: <http://www.w3.org/Protocols/rfc2616/rfc2616.html>
- [35] M. Nicola and B. Van der Linden, “Native xml support in db2 universal database,” in *Proceedings of the 31st international conference on Very large data bases*. VLDB Endowment, 2005, pp. 1164–1174.
- [36] E. Harold, *Processing XML with Java: a guide to SAX, DOM, JDOM, JAXP, and TrAX*. Addison-Wesley Professional, 2003.
- [37] M. Rauch, “Mobile documentation: Usability guidelines, and considerations for providing documentation on kindle, tablets, and smartphones,” in *Professional Communication Conference (IPCC), 2011 IEEE International*. IEEE, 2011, pp. 1–13.
- [38] R. Soares, M. Pereira, and J. Martins, “Recolha, preservação e contextualização de objectos digitais para dispositivos móveis com android,” *Iberian Journal of Information Systems and Technologies*, vol. 0, no. 9, pp. 75–89, 2012. [Online]. Available: <http://ojs.academypublisher.com/index.php/risti/article/view/risti097589>
- [39] A. Developers. Platform versions. [Online]. Available: <http://developer.android.com/about/dashboards/index.html>

Anexos

7.1 Anexo A

Na Figura 7.1 está ilustrado o modelo da arquitetura usado pelo sistema operativo *Android*.

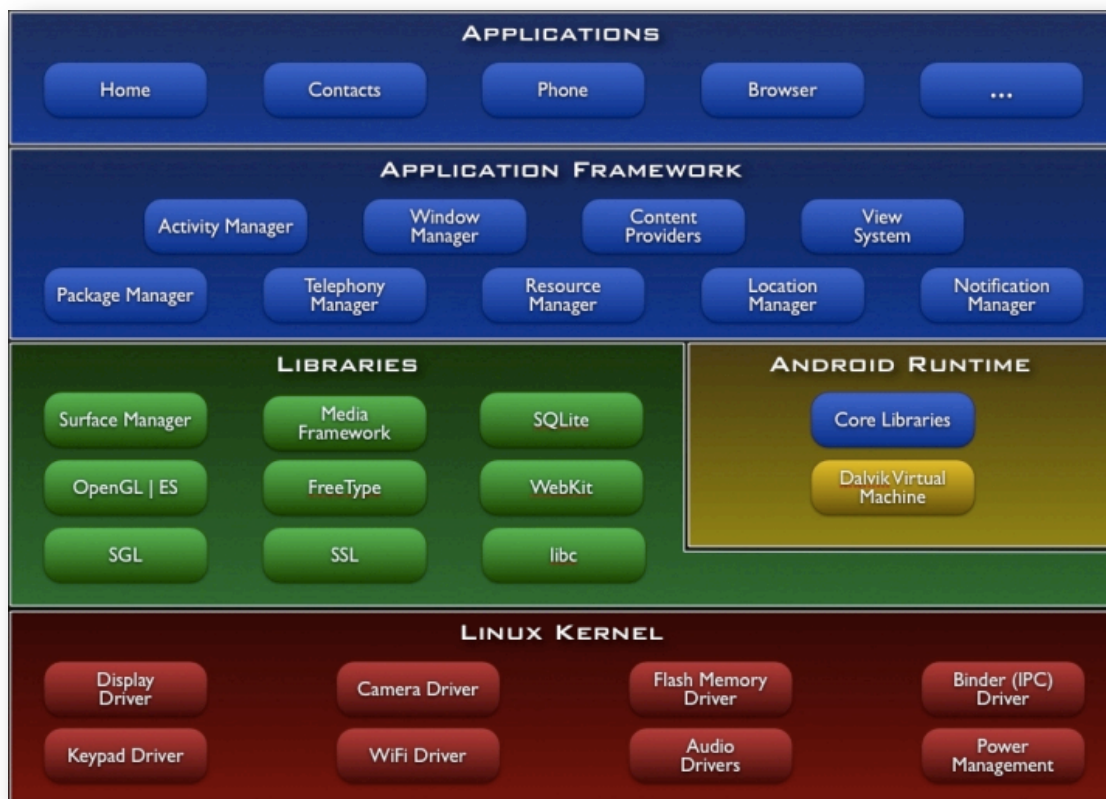


Figura 7.1: Modelo da arquitetura do Sistema Operativo Android.

7.2 Anexo B

No Gráfico 7.2 está ilustrado um estudo realizado sobre a venda de *smartphones* ao utilizador final, tendo em conta o sistema operativo [17].

No Gráfico 7.3 pode ser consultada a utilização atual bem como as previsões de uso para os próximos anos, dos vários sistemas operativos para dispositivos móveis [18].

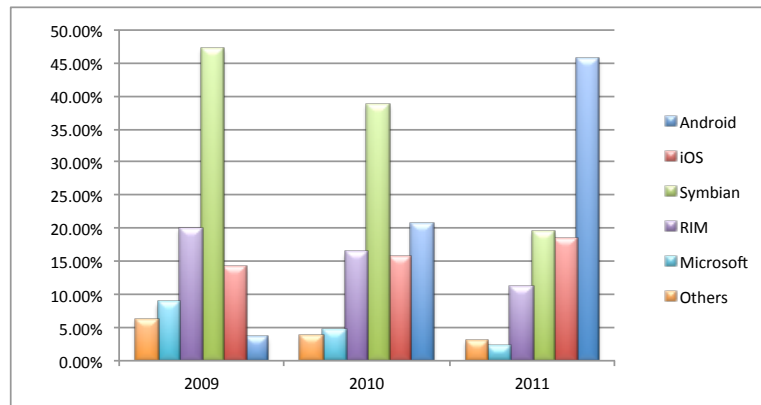


Figura 7.2: Estudo da venda de *smartphones* ao utilizador final, a nível mundial em função do Sistema Operativo.

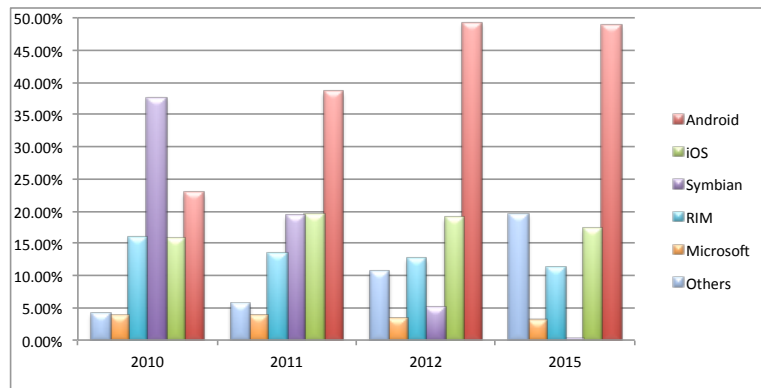


Figura 7.3: Crescimento atual e previsão para os próximos anos.

7.3 Anexo C

Na Figura 7.4 está representado o ciclo de vida de um serviço em *Android* ¹.

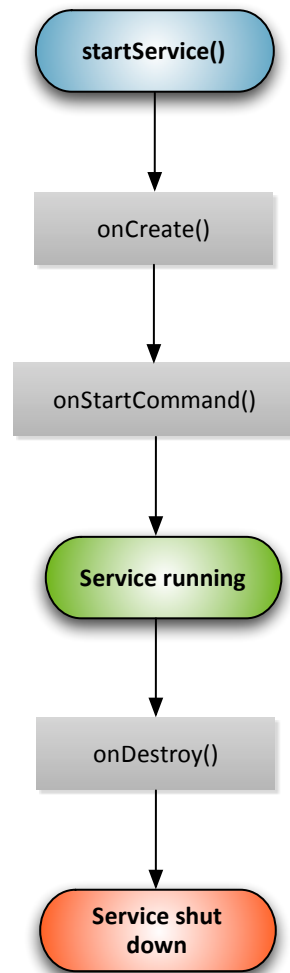


Figura 7.4: Ciclo de vida de um serviço em *Android*.

¹<http://developer.android.com/guide/components/services.html>

7.4 Anexo D

Na Figura 7.5 encontra-se ilustrado o ciclo de vida de uma atividade em *Android* ².

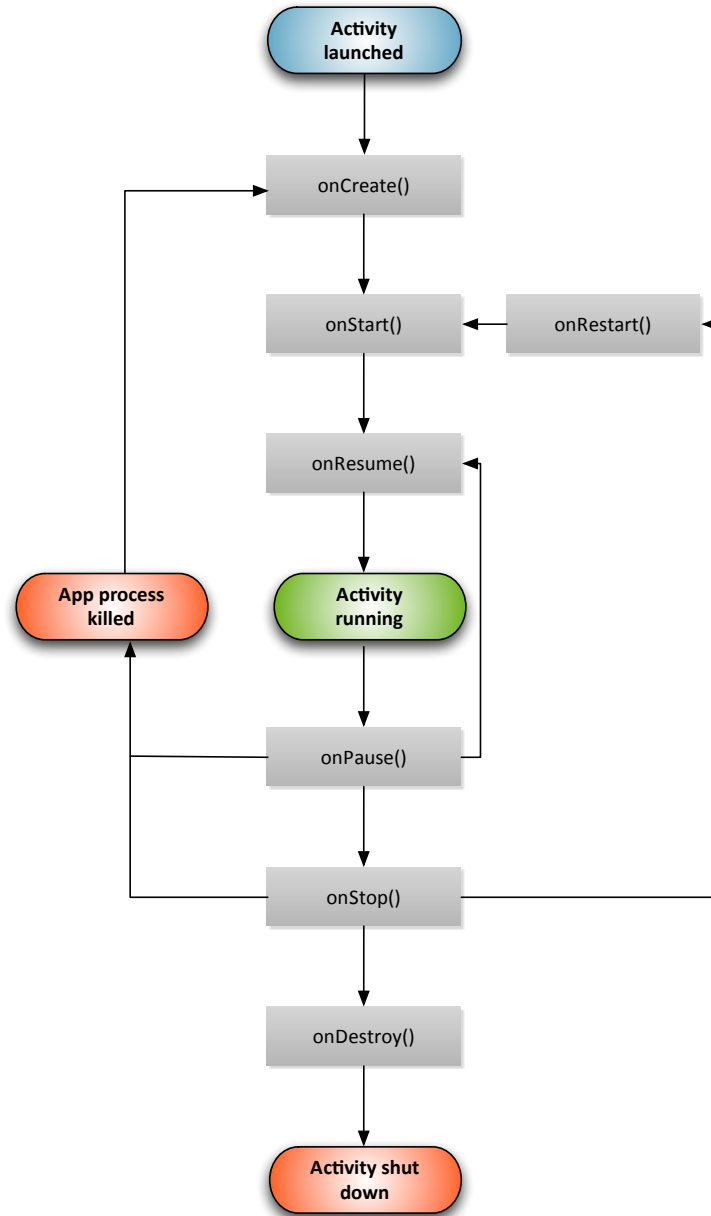


Figura 7.5: Ciclo vida de uma atividade em *Android*.

²<http://developer.android.com/reference/android/app/Activity.html>

7.5 Anexo E

Na Figura 7.6 e no Gráfico 7.1 encontra-se retratado a percentagem de utilização das várias versões do sistema operativo *Android* [39].

Versão	Nome	Percentagem uso
1.5	Cupcake	0.10%
1.6	Donut	0.40%
2.1	Eclair	3.40%
2.2	Froyo	12.90%
2.3 a 2.3.2	Gingerbread	0.30%
2.3.3 a 2.3.7		55.50%
3.1	Honeycomb	0.40%
3.2		1.50%
4.0.3 a 4.0.4	Ice Cream Sandwich	23.70%
4.1	Jelly Bean	1.80%

Tabela 7.1: Tabela representativa das versões usadas do sistema operativo *Android*.

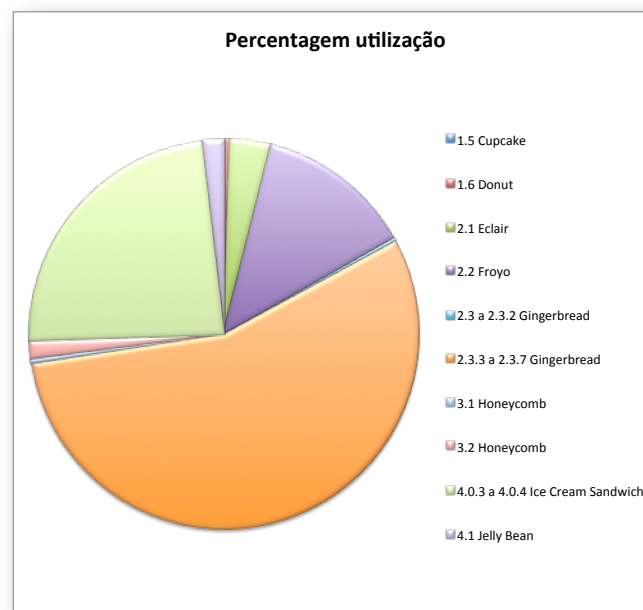


Figura 7.6: Gráfico representativo das versões usadas do sistema operativo *Android*.