

Universidade Federal do Piauí  
Campus Senador Helvídio Nunes de Barros  
Curso Bacharelado em Sistemas de Informação

Francisco Rafael Melo Silva

***BackShare - um Aplicativo de Backup e Compartilhamento de  
Contatos para Android***

Picos  
2014

Francisco Rafael Melo Silva

*BackShare - um Aplicativo de Backup e Compartilhamento de Contatos para Android*

Trabalho de Conclusão de Curso apresentado ao Curso de Sistemas de Informação Campus Senador Helvídio Nunes de Barros da Universidade Federal do Piauí como parte dos requisitos para obtenção do Grau de Bacharelado, sob orientação do Prof. Esp. Ismael de Holanda Leal.

Picos  
2014

Eu, **Francisco Rafael Melo Silva**, abaixo identificado(a) como autor(a), autorizo a biblioteca da Universidade Federal do Piauí a divulgar, gratuitamente, sem ressarcimento de direitos autorais, o texto integral da publicação abaixo discriminada, de minha autoria, em seu site, em formato PDF, para fins de leitura e/ou impressão, a partir da data de hoje.

Picos-PI 10 de março de 2014.

Francisco Rafael Melo Silva  
Assinatura

**FICHA CATALOGRÁFICA**

**Serviço de Processamento Técnico da Universidade Federal do Piauí  
Biblioteca José Albano de Macêdo**

**S586b** Silva, Francisco Rafael Melo.  
Backshare – um aplicativo de backup e  
compartilhamento de contatos para android / Francisco  
Rafael Melo Silva. – 2013.  
CD-ROM : il. ; 4 ¾ pol. (58 p.)

Monografia(Bacharelado em Sistemas de Informação) –  
Universidade Federal do Piauí. Picos-PI, 2013.  
Orientador(A): Prof. Esp. Ismael de Holanda Leal

1. Backup. 2. Contatos. 3. Android. I. Título.


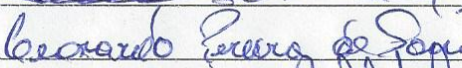
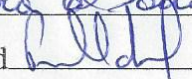
**CDD 005.133**

Francisco Rafael Melo Silva

*BackShare* - um Aplicativo de *Backup* e Compartilhamento de Contatos para *Android*

Trabalho de Conclusão de Curso apresentado ao Curso de Sistemas de Informação Campus Senador Helvídio Nunes de Barros da Universidade Federal do Piauí como parte dos requisitos para obtenção do Grau de Bacharelado, sob orientação do Prof. Esp. Ismael de Holanda Leal.

Data de Aprovação: 27/02/2014

Ismael de Holanda Leal  UFPI - CSHNB  
Leonardo Pereira de Sousa  UFPI - CSHNB  
Patricia Medyna Lauritzen de Lucena Drumond  UFPI - CSHNB

Dedico este trabalho a Deus, antes de tudo.  
Dedico também a minha mãe, Francisca Lindalva Teixeira Melo Silva e ao meu pai, José Norberto da Silva, e meus irmãos F<sup>o</sup> Lucas Melo Silva, F<sup>o</sup> Gabriel Melo Silva e F<sup>a</sup> Mariana Melo Silva que com simplicidade e esforço me ajudaram e apoiaram durante toda essa caminhada.  
Diante de tantos obstáculos e dificuldades que muitas vezes nos fazem pensar em desistir, aprendi que tudo na vida sempre tem um sentido e nele pude colocar minha esperança.

Ao concluir este sonho, lembro-me de muitas pessoas a quem ressalto reconhecimento, pois, esta conquista concretiza-se com a contribuição de cada uma delas, seja direta ou indiretamente. No decorrer dos dias, vocês colocaram uma pitada de amor e esperança para que neste momento findasse essa etapa tão significativa para mim.

Em primeiro lugar agradeço a Deus, fonte de vida e libertação, que me embebeda todos os dias no seu amor e me faz acreditar em um mundo mais justo, mais humano e mais fraterno, crença essa que me mantém em pé todos os dias da minha vida. Sem Ele, não estaria aqui.

A todos da minha família que, de alguma forma, incentivaram-me na constante busca pelo conhecimento. Em especial aos meus pais José Norberto da Silva e Francisca Lindalva Teixeira Melo Silva, aos meus irmãos Lucas Melo, Gabriel Melo e Mariana Melo, por me apresentar a simplicidade e o gosto da e pela vida, inculcando valores sem os quais jamais teria me tornado pessoa, buscando de fato todos os dias, ser mais humano e sensível às necessidades dos outros.

A todos os professores que me acompanharam durante a graduação, pelos ensinamentos e conselhos que foram essenciais para minha formação.

Agradeço ao meu orientador Ismael Holanda por toda paciência e conhecimento adquirido neste trabalho.

Gostaria de agradecer a Danila Feitosa por toda paciência e ajuda neste caminho árduo, gostaria também de agradecer aos meus amigos Melksedek Amorim, Jonison Rebelatto, Gleyson, Thiago José, Itamar Egídio, Auricélio, Israel, João Mariano, Weder, Afonso Melo, Robson, Jaílson Joaquim, Laíse, Francilene, que fiz durante toda esta caminhada.

A todos aqueles que de alguma forma estiveram e estão próximos de mim, fazendo esta vida valer cada vez mais a pena.

"Construí amigos, enfrentei derrotas, Venci obstáculos, bati na porta da vida e disse-lhe: Não tenho medo de vivê-la!".

(Augusto Cury)

“É muito melhor lançar-se em busca de conquistas grandiosas, mesmo expondo-se ao fracasso, do que alinhar-se com os pobres de espírito, que nem gozam muito nem sofrem muito, porque vivem numa penumbra cinzenta, onde não conhecem nem vitória, nem derrota.”

(Theodore Roosevelt)

“É preciso força pra sonhar e perceber que a estrada vai além do que se vê”.

(Los Hermanos)

# Resumo

Com a expansão tecnológica e a necessidade das pessoas se manterem conectadas houve um aumento considerável nas vendas de telefonia móvel, e conseqüentemente o número de celulares que são roubados/furtados aumenta consideravelmente. Este trabalho tem por objetivo principal mostrar o desenvolvimento de um aplicativo que visa ajudar as pessoas que têm celulares furtados, perdendo, assim, seus contatos. Objetiva discutir a evolução da comunicação móvel, furto de celulares, bem como abordar as características da Plataforma *Android*. Para a construção deste trabalho realizou-se uma pesquisa bibliográfica e o desenvolvimento de um aplicativo para *Android* utilizando a linguagem de programação *Java* e técnicas de estruturação de arquivos. Após o desenvolvimento e testes do aplicativo *Backshare*, pode-se concluir que o mesmo é funcional no *backup* e compartilhamento de contatos em diversos modelos diferentes de celulares e versões do SO *Android*.

**Palavras-chave:** *Backup. Android. Contatos.*



# Abstract

With the expanding technology and the need for people to stay connected, there was a considerable increase in mobile phone sales, and consequently the number of phones that are stolen / pilfered increases considerably. This work has as main objective to show the development of an application that aims to help people who have stolen mobiles, thus losing your contacts. Aims to discuss the evolution of mobile communication, cellular theft, as well as addressing the characteristics of the Android Platform. For the construction of this work we carried out a literature review and the development of an Android app using the programming language java and techniques for structuring file. After developing and testing the application Backshare, can conclude that it is functional in backup and share contacts in many different models of phones and Android OS versions.

**Keywords:** Backup. Android. Contacts.

# Lista de Figuras

Figura 1 -	<i>Exemplo da estrutura de um arquivo vCard 3.0</i>	30
Figura 2 -	<i>Wireframes das Três Telas Iniciais</i>	32
Figura 3 -	<i>Wireframes das Telas Finais</i>	33
Figura 4 -	<i>Tela de Apresentação</i>	34
Figura 5 -	<i>Telas para Selecionar Diretório e Nome do Arquivo</i>	34
Figura 6 -	<i>Telas de Processo</i>	35
Figura 7 -	<i>Tela de Compartilhamento</i>	36
Figura 8 -	<i>Compartilhamento com DropBox</i>	36
Figura 9 -	<i>Código Tela Apresentação</i>	43
Figura 10 -	<i>Código - Selecionar Diretório e Nome do Arquivo (Parte 1)</i>	44
Figura 11 -	<i>Código - Selecionar Diretório e Nome do Arquivo (Parte 2)</i>	45
Figura 12 -	<i>Código - Selecionar Diretório e Nome do Arquivo (Parte 3)</i>	46
Figura 13 -	<i>Código - Selecionar Diretório e Nome do Arquivo (Parte 4)</i>	47
Figura 14 -	<i>Código - Armazenamento de Contatos em Arquivo (Parte 1)</i>	48
Figura 15 -	<i>Código - Armazenamento de Contatos em Arquivo (Parte 2)</i>	49
Figura 16 -	<i>Código - Armazenamento de Contatos em Arquivo (Parte 3)</i>	50
Figura 17 -	<i>Código - Compartilhamento do Backup Gerado</i>	51
Figura 18 -	<i>Galaxy Pocket Plus S5301</i>	52
Figura 19 -	<i>Galaxy Win I8550</i>	53
Figura 20 -	<i>Optimus L3 II Dual E435</i>	54
Figura 21 -	<i>Emulador AVD</i>	55

# Lista de Tabelas

Tabela 1 - Resultados Obtidos . . . . .	37
---	----

# Lista de abreviaturas e siglas

AAC	<i>Advanced Audio Coding</i>
ADT	<i>Android Development Tools</i>
AMPS	<i>Advanced Mobile Phone Service</i>
AMPS	<i>Advanced Mobile Phone System</i>
AMR	<i>Adaptive Multi-Rate</i>
ANATEL	Agência Nacional de Telecomunicações
API	<i>Application Programming Interface</i>
AVD	<i>Android Virtual Device</i>
CSV	<i>Comma Separated Values</i>
FCC	<i>Federal Communication Commission</i>
GIF	<i>Graphics Interchange Format</i>
GPS	<i>Global Positioning System</i>
GSM	<i>Group Special Mobile</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
IDE	<i>Integrated Development Environment</i>
IMC	<i>Internet Mail Consortium</i>
JDK	<i>Java Development Kit</i>
JPG	<i>Joint Photographic Group</i>
JVM	<i>Java Virtual Machine</i>
LED	<i>Light Emitting Diodes</i>
MP3	<i>MPEG-1/2 Audio Layer 3</i>
MPEG	<i>Moving Picture Experts Group</i>
OHA	<i>Open Handset Alliance</i>
PNG	<i>Portable Network Graphics</i>
SDC	<i>Secure Digital Card</i>
SDK	<i>Software Development Kit</i>
SO	Sistema Operacional
UIT	União Internacional de Telecomunicações

# Sumário

<b>1</b>	<b>Introdução</b>	<b>12</b>
<b>2</b>	<b>Comunicação Móvel: Uma Breve Abordagem a Cerca do Celular</b>	<b>14</b>
2.1	Celular: História e Evolução . . . . .	14
2.2	As Gerações de Sistemas Móveis Celulares . . . . .	15
2.3	O Celular no Contexto Brasileiro . . . . .	17
2.4	Furto de Celulares: Uma Realidade da Sociedade Atual . . . . .	18
<b>3</b>	<b><i>Android</i></b>	<b>20</b>
3.1	Conceituado a Plataforma <i>Android</i> . . . . .	20
3.2	Breve Histórico da Plataforma <i>Android</i> . . . . .	21
3.3	A Arquitetura da Plataforma <i>Android</i> . . . . .	22
3.3.1	Máquina Virtual <i>Dalvik</i> . . . . .	23
3.4	Linguagem <i>Java</i> . . . . .	24
3.4.1	Principais Características da Linguagem <i>Java</i> . . . . .	25
3.4.2	<i>Comma Separated Values</i> (CSV) . . . . .	26
3.5	<i>Contact Provider</i> . . . . .	28
3.6	<i>vCard</i> . . . . .	28
3.7	<i>Android Development Tools</i> (ADT) . . . . .	29
3.8	<i>Java Development Kit</i> (JDK) . . . . .	29
<b>4</b>	<b><i>BackShare</i>: Uma Aplicação de <i>Backup</i> e Compartilhamento de Contatos para <i>Android</i></b>	<b>30</b>
4.1	Instalação e Configuração dos <i>Softwares</i> . . . . .	31

4.2	Funcionamento do Sistema <i>BackShare</i> . . . . .	32
4.3	Resultados obtidos dos testes realizados em Dispositivos Móveis com SO <i>Android</i>	37
<b>5</b>	<b>Conclusão e Trabalhos Futuros</b>	<b>38</b>
5.1	Conclusão . . . . .	38
5.2	Trabalhos Futuros . . . . .	39
	<b>Referências</b>	<b>40</b>
	<b>Apêndice A – Classe Apresentação</b>	<b>43</b>
	<b>Apêndice B – Algoritmo para Selecionar Diretório e Nome do Arquivo</b>	<b>44</b>
	<b>Apêndice C – Algoritmo de Processos</b>	<b>48</b>
	<b>Apêndice D – Algoritmo para Compartilhamento do Backup Gerado</b>	<b>51</b>
	<b>Apêndice E – Teste Realizados</b>	<b>52</b>

# 1 Introdução

No Brasil, o número de celulares já é maior do que o número de habitantes. Com o aumento do número de celulares, cresce cada vez mais o número de roubos. O maior problema de ter um celular roubado é quando não existe uma cópia salva de todo conteúdo do mesmo, isso traz grandes prejuízos às pessoas, pois o celular contém informações importantes para as mesmas como os dados do contato, o número do celular, endereço e *e-mail*. Com essa observação foi verificada a necessidade de desenvolver um aplicativo para *Android* para que essas informações de contato possam ser armazenadas no dispositivo móvel e também podendo ser compartilhado em um ambiente online para serem acessadas por meio da internet. Por exemplo, podendo ser armazenado online nos seguintes serviços: *DropBox*, *Google Drive*, *Hotmail*.

Foi percebido que quanto maior a quantidade de celulares existentes maior será a quantidade de problemas devido a indivíduos mal intencionados. Observa-se a necessidade de que essas pessoas possam ter uma aplicação no *smartphone* que faça um *backup* dos contatos (número, *e-mail*, endereço) em um ambiente online para que possam ser acessados através da internet, sendo o *Android* o ambiente de desenvolvimento escolhido devido ao seu crescimento contínuo e a tecnologia inovadora que o mesmo oferece.

Outros motivos para utilização do *Android* como plataforma de desenvolvimento segundo Ogliari (2012), envolve os seguintes quesitos:

**Interfaces gráficas** - A construção das *interfaces* gráficas em aplicativos *Android* é facilitada por diversos motivos. O primeiro deles é a possibilidade e a liberdade dado ao desenvolvedor para escolher: utilizar a linguagem Java ou a linguagem de marcação XML.

**Versão de imagens específicas para determinadas resoluções** - O *Android* trabalha com separação de imagens por resolução de tela. Em tempo de execução, o sistema operacional *Android* verificará a imagem que está sendo solicitada, vai checar qual sua resolução e vai devolver ao aplicativo a versão da imagem mais apropriada àquela tela. Simples, claro e rápido.

Assim o capítulo 2 deste trabalho apresenta um breve histórico sobre a evolução do celular, bem como sua importância na sociedade atual. Também aborda sobre o furto de dispositivos móveis, problema que vem ocorrendo devido ao crescimento constante destes aparelhos. No capítulo 3 aborda aspectos técnicos, bem como históricos sobre a Plataforma *Android* e o desenvolvimento para dispositivos móveis que possuem este Sistema Operacional. Ainda é comentado sobre as ferramentas que foram usadas para desenvolvimento da aplicação deste trabalho, tais como, Linguagem *Java*, *vCard* e *ContactProvider*.

O capítulo 4 faz uma explanação sobre o sistema desenvolvido neste trabalho, explicando o funcionamento, detalhes técnicos, objetivo desta aplicação e testes.

Por fim o capítulo 5 mostra os objetivos atingidos e sugestões para trabalhos futuros.



## 2 Comunicação Móvel: Uma Breve Abordagem a Cerca do Celular

Este capítulo busca discutir o advento do celular e a evolução pelo qual o mesmo passou, e vem passando ao decorrer dos anos, deixando de ser utilizado simplesmente para realizar chamadas, ganhando assim muitas outras utilidades que fizeram do mesmo uma ferramenta que grande parte da sociedade contemporânea considera indispensável durante o dia-a-dia.

No Brasil o número de celulares já chega a ser superior ao número de habitantes, possibilitando a comunicação em massa da sociedade atual. No entanto, com o aumento dos aparelhos celulares surgem problemas decorrentes desse progresso, como, por exemplo, os roubos dos mesmos, tão frequentes nos dias de hoje. Desse modo, abordasse-a alguns aspectos referentes ao aparelho celular e suas peculiaridades na época em que vivemos.

### 2.1 Celular: História e Evolução

No ano de 1947, uma empresa americana chamada *Bell Company* desenvolveu um sistema que utilizava-se do conceito de células e permitia a utilização de telefonia móvel dentro de uma área, de tal sistema derivou-se o nome celular. De acordo com Guimaraes (1998), o conceito de telefone celular com um sistema de comunicação móvel usava modulação AM<sup>1</sup>.

Segundo Abreu (2005) foi ainda em 1947, nos Estados Unidos da América, que a AT&T e a Bell levaram a *Federal Communication Commission* (FCC), a proposta de alocar um número de frequência de rádio que era específica para comunicação móvel, “mas a FCC disponibilizou apenas poucas frequências, possibilitando que somente 23 pessoas se conectassem simultaneamente ao sistema de uma determinada área de cobertura. Isto, na época, tornou a tecnologia inviável comercialmente” (ABREU, 2005, p. 22).

Corroborando Abreu (2005) diz que vinte anos se passaram até que a FCC aumentasse o número de frequências que se destinavam a telefonia móvel no intuito de suportar mais usuários. O autor afirma, ainda, que em 1968 a utilização de torres foi definida pela Bell e AT&T para que fossem atendidos os usuários por áreas de acordo com seu deslocamento, a partir de então este sistema apenas cresceu e prosperou até conseguir a cobertura que oferece nos dias de hoje.

---

<sup>1</sup> Modulação em Amplitude

Os anos de 1973 e 1979 trouxeram consigo novos caminhos trilhados pelo o celular, através de incorporações e testes, mas foi, somente, em 1983 que surgiu o primeiro celular aprovado pelo FCC. Todo esse percurso da telefonia móvel é descrito com presteza por Abreu (2005, p. 22):

A Bell, em 1973, já possuía um sistema de comunicação instalado em carros de polícia, mas foi a *Motorola*, naquele mesmo ano, a primeira a incorporar essa tecnologia a um dispositivo móvel de comunicação fora de um veículo, para uso pessoal. Contudo, em janeiro de 1979 o sistema foi realmente testado com 200 pessoas em Chicago. Mas, foi 10 anos depois, em 1983, que surgiu o primeiro celular aprovado pelo FCC, o *DynaTAC 8000X*, da *Motorola* - que junto com a empresa *Ameritech* iniciou o uso comercial da telefonia celular no Estados Unidos e no mundo. Nesse momento a *Motorola* já havia investido cerca de US\$ 100 milhões em 15 anos de pesquisas em tecnologia móvel celular.

Ainda de acordo com Abreu (2005), o primeiro aparelho móvel pesava cerca de um quilo, sua capacidade de conversação era apenas de uma hora, de *standy-by* contava com oito horas, quanto a contatos, números de telefone, o celular suportava apenas trinta números, possuía *display Light Emitting Diodes*, LED.

Apesar de todas as limitações que o primeiro celular trazia consigo e que hoje é uma realidade difícil de ser imaginada, ele trazia também a impressionante ideia de estar sempre conectado, o que deixava as pessoas eufóricas e dispostas a pagar um preço alto pelo aparelho.

Para Abreu (2005) os anos que se passaram após o início da telefonia móvel comercial, mais de vinte anos, trouxeram consigo mudanças ao conceito do uso de celulares, sendo que o autor enfatiza que hoje os consumidores de telefonia móvel buscam por formas de reforçar, melhorar seus estilos de vida. Para Abreu (2005, p. 23):

O celular tornou-se, neste contexto, uma extensão da personalidade do usuário, uma peça capaz de enriquecer relacionamentos, divertir, aumentar a produtividade e expressar individualidade. Isso significa comunicar, dividir, criar e divertir com voz, textos, imagens, músicas e vídeos.

Ao longo dos anos a tecnologia tornou-se um tanto mais barata, fazendo com que o número de celulares aumentasse consideravelmente. Para Abreu (2005, p.23) este passou de cerca de 300 mil, em 1984, para mais de 1,2 bilhão, atualmente.

Com o crescimento das indústrias e os avanços tecnológicos, as empresas competem entre si por tecnologias inovadoras que revolucionem, sendo que o acesso à *internet* sem fio, imagens e músicas digitais, bem como jogos estão no topo das preferências dispostas pelas empresas para atenderem as necessidades de seus consumidores.

## 2.2 As Gerações de Sistemas Móveis Celulares

A primeira geração de celulares, já mencionada brevemente no tópico anterior, refere-se aos primeiros celulares comercializados, estes celulares possuíam características não muito

atraentes, mas devido a grande novidade que apontavam tornaram-se cobiçados.

De acordo com Rocha (2013) os primeiros celulares tinham um tamanho relativamente grande, eram pesados e bastante caros, com baterias que duravam apenas trinta minutos, levavam entre quatro e oito horas para serem carregados. Implantado no Sistema *Advance Mobile Phone System* (AMPS), um sistema americano em que cada usuário ocupa um canal analógico de voz 30 Khz e que acarretava inúmeras interferências e apresentava baixa segurança, sendo possível ouvir ligações por meio de um sintonizador de rádio, bem como usurpar frequência e creditar ligações na conta de outro. Esse sistema corresponde a geração Analógica, a 1G, sigla utilizada para se referir a primeira geração de telefonia móvel, introduzidas no mercado nos anos de 1980 e 1990, estes celulares permitiam apenas falar.

De acordo com Pereira e Guedes (2004) a primeira geração de celulares atendeu a cerca de 49 milhões de usuários no mundo, isso ocorreu de 1981 a 1994. Sobre essa geração de celulares, Pereira e Guedes (2004, p. 26) comentam que:

Os diversos sistemas celulares analógicos operavam basicamente em três faixas de frequências distintas: 450, 800 e 900 MHz. Alguns sistemas ofereciam serviço de *roaming* nacional e internacional, mas, em muitos casos, não havia padronização entre as diversas *interfaces* dos sistemas, ou seja, os sistemas utilizam *interfaces* proprietárias. Os sistemas analógicos de primeira geração que mais se destacaram, a cerca do nível de penetração junto ao público, qualidade, facilidades e variedade de tipos de terminais foram: AMPS, TACS, NMT e NTT.

Pereira e Guedes (2004) corroboram que nos anos de 1980 ficou claro que a G1, com base em técnicas de sinalização analógica, se tornara arcaica e atingiu sua capacidade máxima, à medida que a tecnologia avançava com circuitos integrados se tornou interessante economicamente e mais viável.

Surge então a G2, segunda geração de celulares, com padrões que baseavam-se em tecnologias digitais de comunicação, apresentavam-se mais eficientes e utilizavam-se de novos serviços, não só de voz, mas transmitiam de dados.

Sobre a segunda geração de celulares reflete Rocha (2013, p. 1):

Por volta de 1990, as duas primeiras centrais telefônicas orientadas para o serviço móvel celular foram implantadas no Brasil, uma no Rio de Janeiro e outra em Brasília. Os telefones eram feios, enormes, pesados e caríssimos. Sendo que, além da interferência e queda de chamadas, as baterias duravam 30 minutos e levavam de 4 a 8 horas para ser recarregadas. Foi um produto restrito demais, naquela época o sistema implantando era o AMPS (*Advanced Mobile Phone System*) americano, cada usuário ocupava um canal analógico de voz de 30 Khz, o espectro de frequência era poluído e sua reutilização muitas vezes dava “colisão” de frequência, a famosa interferência ou queda de chamada. Este sistema corresponde a Primeira Geração de Celulares – a Analógica.

As mensagens de texto, SMS, era a grande novidade apresentada pela G2, a ideia central era usar a capacidade dos canais para enviar além de Voz, pequenas mensagens de texto, uma revolução, os famosos torpedos, fizeram com que os celulares ficassem ainda mais populares (ROCHA, 2013, p.1).

A inclusão dos primeiros *browsers* e a possibilidade da utilização dos celulares como

*modem* permitindo o acesso a *internet* foram outras novidades advindas com a segunda geração de celulares.

Em 1988, a telefonia celular era um verdadeiro sucesso e preparava-se a futura terceira geração, isso acontecia em meio às disputas entre os padrões GSM e CDMA no intuito de saber qual padrão seria utilizado na 3G, foi lançada então as redes intermediárias, conforme Rocha (2013, p.1):

Neste ínterim se lançou as redes intermediárias, a 2,5 G, do lado GSM o *Edge* (velocidade final 64kbits) e a do CDMA, o *1Xrtt* (velocidade final de 144bits). Ambas permitiram que os aparelhos acessassem a *internet* com *browser* próprio sem mais servir como *modem*, era a preparação para a nova geração, além de educar os bilhões de usuários no mundo nesta nova revolução tecnológica. Os celulares são cada vez menores e mais leves, com telas mais nítidas, trazendo recursos como tocadores de músicas, no formato MP3.

O sistema móvel de terceira geração, o 3G, foi projetado para garantir acesso, por meio de *interface* aérea, e diversos serviços de telecomunicações que as redes fixas suportam. Para Pereira e Guedes (2004) os sistemas 3G devem permitir taxas de transmissão mais altas e focar no tráfego de dados e também de multimídia, a geração 3G dispõe, assim, de chamadas telefônicas, fax, caixa postal, serviços de localização, videoconferência, Tv por assinatura, SMS e *email*, *roaming* global e também oferece acesso a *internet* em alta velocidade. Desse modo os celulares da terceira geração oferecem serviços pessoais, de multimídia, mobilidade, *internet* e uma tecnologia inovadora que atende aos padrões globais.

Assim, os novos aparelhos tem acesso pleno à *internet* e são verdadeiras plataformas que oferecem múltiplos serviços, os *smartphones* são tidos como uma ferramenta de trabalho, na qual o *Android*, que atualmente é o sistema da maioria de celulares, lança vários aparelhos e modelos.

## 2.3 O Celular no Contexto Brasileiro

Foi no ano de 1990 que a telefonia celular chegou ao Brasil. Segundo Machado (2013) o Rio de Janeiro foi a primeira cidade brasileira a utilizar-se da telefonia móvel, conforme o autor no ano de 1991 surgiria o sistema Telebrásília e dois anos depois seria inaugurada a Telefonia Móvel Celular no Estado de São Paulo. Segundo Peixoto (2007, p. 17):

As operações do serviço móvel celular (SMC) no Brasil tiveram início com a utilização de sistemas analógicos e seguindo a faixa de frequências dos sistemas móveis de acordo com o padrão norte-americano. Esta faixa, subdividida entre bandas A e B, foi atribuída para cada operadora, mas, por determinação da ANATEL (Agência Nacional de Telecomunicações), uma mesma operadora não poderia ocupar duas bandas em uma mesma região geográfica. A tecnologia analógica empregada foi o AMPS (*Advanced Mobile Phone Service*).

Sobre a entrada da telefonia móvel no Brasil:

A história da telefonia celular no Brasil começou em 1990. Na época, segundo dados da Anatel, o país contava com 667 aparelhos, número que passou para 6.700 unidades no ano seguinte, ultrapassando os 30 mil em 1992 e chegaram a 47.865.593 telefones celulares em fevereiro de 2004. A trajetória de crescimento teve como fator determinante a privatização da telefonia móvel no Brasil (XAVIER et al., 2006, p. 318).

Salienta Abreu (2005) que já no ano de 2003 o número de telefones móveis já era superior aos de telefones fixos, que eram tidos como de péssima qualidade, o autor comenta ainda que no princípio da telefonia móvel era muito caro utilizar-se deste sistema, mas com o passar do tempo, o que é típico na área tecnológica aconteceu e o preço do sistema acabou caindo, à medida que a tecnologia evoluía e a produção aumentava.

Com o passar do tempo, o celular deixou de ser luxo e tornou-se algo extremamente necessário, não era mais um simples objeto de desejo, mas um item básico e foi assim que “a telefonia móvel atingiu um patamar que permeia todo o tecido da sociedade brasileira moderna e que, assim como os computadores, criou um forte vínculo de dependência com essa tecnologia” (ABREU, 2005, p. 24).

Para Pellanda (2009) mais de 80% dos celulares no Brasil comercializam-se nos planos pré-pagos, e é exatamente este ponto o responsável pela comunicação *wireless* no país, para o autor a comunicação móvel esta transformando as atividades sociais e também econômicas de forma bastante profunda, pois possibilita diversos serviços e é utilizada em vários trabalhos.

Outra questão importante no ponto de vista de Pellanda (2009) no uso de telefone móvel é que grande parte da população não tem acesso a telefones fixos. Assim, discorre Pellanda (2009, p. 16):

Isso se deve ao fato de áreas populosas, mas informais, como as favelas ou áreas rurais, não terem infraestrutura para as ligações. Em certas áreas, há também um desinteresse econômico das empresas, que deste modo, ignoram os locais. Mas a tecnologia sem fio transpõe este problema por não precisar de ligações diretas com as residências. Uma única base de telefonia celular pode ser responsável pela existência de diversas linhas. Essa flexibilidade, aliada a uma expansibilidade, é um dos principais fatores de inclusão digital da tecnologia.

Assim, ressalta-se que a tecnologia 3G, ou seja, a terceira geração de celulares, que trazem consigo a possibilidade de acesso a *internet* banda larga pelos dispositivos móveis foi um lançamento em massa no Brasil no ano de 2008, já que todas as capitais e principais centros urbanos do país dispõem dessa tecnologia, desta forma, a população brasileira inclui-se na comunicação digital, que apenas cresce ao longo dos anos e atualmente é marcada pelo sucesso do versátil *Android*.

## **2.4 Furto de Celulares: Uma Realidade da Sociedade Atual**

Segundo a Tavares (2010), o IBGE (Instituto Brasileiro de Geografia e Estatística) afirma que pela primeira vez o número de celulares já é maior do que de habitantes no Brasil sendo 194,4 milhões de celulares para 185,7 milhões de habitantes.

O número de celulares, não só no Brasil, mas no mundo é impressionante, as pessoas a cada dia que passa sentem a necessidade de estarem conectadas o tempo inteiro, sendo possível tal necessidade graças a gama de serviços que os aparelhos móveis oferecem hoje em dia, desse

modo o celular se torna indispensável à vida social e econômica das pessoas e, assim, elas estão sempre com este aparelho junto de si, mas toda essa indispensabilidade garantida ao celular vem trazendo como consequência o número de roubos cada vez maior.

O Ministério da Justiça (2011) afirma que cerca de um milhão de celulares são roubados no Brasil por ano, sendo que esta afirmação refere-se apenas a furtos que são registrados nas delegacias, sendo que relevando os furtos que não tem queixa registrada em delegacia a estimativa seja muito maior. Com isso é possível perceber que quanto maior a quantidade de celulares existentes maior será a quantidade de problemas devido a indivíduos mal intencionados.

O celular é desse modo, um dos principais alvos dos criminosos. Segundo Pankiewicz (2009, p. 1):

Isso ocorre porque muitos dos aparelhos ficam expostos, têm um custo elevado (alguns chegam a ultrapassar a barreira de dois mil reais), seguem os padrões GSM e são destravados (ou serão, com relativa facilidade), aceitando qualquer chip de operadora que seja colocado em seguida. Depois de retirar o cartão original, é só o assaltante revender a um preço bem abaixo do mercado para obter lucro fácil e rápido.

O furto do celular trás consigo o perigo, a perda do aparelho e acarreta na perda de todas as informações contidas no mesmo, como imagens, lista de contatos, endereços de amigos e familiares.

Observa-se a necessidade de que as pessoas vitimas do furto de celulares possam ter para sua usabilidade uma aplicação no *smartphone* que faça um *backup* dos contatos (número, *e-mail*, endereço) em um ambiente *online* para que possam ser acessados através da *internet*, sendo o *Android* o ambiente de desenvolvimento escolhido devido ao crescimento contínuo dessa plataforma.

O intuito do projeto contido neste trabalho refere-se à necessidade de desenvolver um aplicativo para *Android* para que as informações de contato possam ser armazenadas e compartilhadas em um ambiente *online* para serem acessadas por meio da *internet*. Por exemplo, podendo ser armazenado *online* nos seguintes serviços: *DropBox*, *Google Drive*, *Hotmail*, sendo que o serviço de armazenamento *online* escolhido para este projeto é o *DropBox*, pois o mesmo trabalha com o conceito computação em nuvens, oferece 4gb de armazenamento *online* grátis, pacote *DropBox* API para desenvolvedores e também pode ser acessado em diversos sistemas operacionais.

Segundo Amoroso (2012, p. 1), o *DropBox* trabalha com o conceito de armazenamento na nuvem que segundo o mesmo, uma vez devidamente conectado ao serviço *online*, é possível desfrutar suas ferramentas e salvar todo o trabalho que for feito para acessá-lo depois de qualquer lugar, é justamente por isso que o computador estará nas nuvens, pois será possível acessar os arquivos a partir de qualquer computador ou dispositivo móvel que tenha acesso à *internet*.

## 3 *Android*

Este capítulo procura discutir aspectos técnicos, bem como históricos sobre a Plataforma *Android*. A relevância em se construir um trabalho que tenha a Plataforma *Android* como base, está no fato de que ocorre uma evolução na telefonia móvel. É relevante também destacar que a sociedade atual apresenta uma necessidade cada vez maior de comunicar-se e que tal sociedade pretende muito mais de um telefone móvel, o que difere de antigamente, em que o telefone móvel tinha como função executar somente ligações.

A evolução pela qual a telefonia móvel passou trouxe consigo o emprego de convergência da informação e não mais a simples utilidade de realizar comunicação telefônica.

É desse modo que o *Android* vem assumindo maior espaço em meio aos dispositivos móveis. Ao constatar-se o alcance da plataforma *Android* mediante ao *Google* no mundo *Wireless* é que se pretende discutir neste capítulo a história do *Android* com sua aquisição realizada em 2005 e o conseqüente anúncio oficial em 2007, bem como se busca, neste capítulo, conhecer os recursos que o *Android* disponibiliza para desenvolvimento de sistemas, como o proposto neste trabalho.

### 3.1 **Conceituado a Plataforma *Android***

*Android* é uma plataforma de ampliação para dispositivos móveis, cujo sistema funcional é o *Linux*, mais precisamente o *Kernel Linux*, constituindo-se em um ambiente operacional completo, que oferece serviços de informática e suporte funcional. O objetivo de tal plataforma consiste em prover um sistema singular o qual permite que o mesmo seja instalado em dispositivos fornecidos por qualquer um dos fabricantes que façam parte da aliança.

Com as variedades oferecidas pela plataforma *Android* pode-se atribuir, a mesma, conceitos como flexibilidade e integração, visto que os aplicativos que o *Android* fornece podem ser substituídos por aplicações customizadas. O *Android* permite, ainda, uma interação com demais aplicações que não são próprias do seu sistema.

É relevante mencionar que o *Android* vem cada dia adquirindo mais adeptos entre desenvolvedores para dispositivos móveis, isso se deve a fatores que o destacam dos demais aplicativos, bem como ao sistema de desenvolvimento flexível ao programador. Também contribui para sua popularidade o fato de o mesmo possuir Código aberto e de ser acessível aos desenvol-

vedores.

A plataforma *Android* ganha espaço no cenário de comunicação móvel, pelo fato de garantir suporte *hardwares* e variados serviços, para tanto o *Android* utiliza-se de uma máquina virtual chamada *Dalvik* que através da máquina virtual *Java* converte *bytecode* compilado em *bytecode* da *Dalvik*.

As aplicações que a plataforma *Android* possibilita são de fáceis instalações nos dispositivos e sua arquitetura funda-se em componentes e integra-se com IDEs, como *Netbeans* e *Eclipse*, que são bastante populares.

## 3.2 Breve Histórico da Plataforma *Android*

Segundo Lemos (2011), a história do *Android* teve início em meados de 2005, momento em que a *Google*, empresa multinacional de serviços *online* e *software* dos Estados Unidos, comprou a *Android Inc*, uma empresa de pequeno porte localizada em Palo Alto na Califórnia, Estados Unidos. Com o objetivo de se ter uma plataforma aberta, com facilidade de manuseio para aqueles que a fabricavam foi que se criou uma plataforma de telefonia móvel que tinha por base *Software Linux*.

No ano que se seguiu, segundo Lemos (2011), ouviram-se os primeiros rumores de que a *Google* entrava para o mercado de telefonia móvel, o que era assunto fortemente reforçado pela mídia norte americana.

Em novembro do ano de 2007 foi lançada a OHA (*Open Handlest Alliance*), um grupo de empresas que se uniram no intuito de colaborar na construção de um telefone móvel melhor do que os já existentes. O *Google* liderava a OHA e o grupo somava mais de trinta empresas, entre operadoras de telefonia móvel, provedores de plataformas, empresas de *marketing* e fabricantes de aparelhos. Segundo Teles (2010, p. 4) entre as empresas que formam a OHA estão:

- Fabricantes de Dispositivos Móveis – *Samsung, LG, HTC, Motorola, etc.*
- Fabricantes de Componentes – *Texas Instruments, Intel, Nvidia, etc.*
- Fabricantes de *Software* – *PV, Ebay, Esmertec, etc.*
- Operadoras Móveis de todo mundo – *T-Mobile. Italia Telecom, China Mobile, etc.*
- Empresas de Comercialização – *Aplix, Noser, Systems, etc.*

No ano de 2008 foi lançado o *Android Market*, de acordo com Lemos (2011, p. 8), o mesmo “consistia em um canal dentro do aparelho onde os desenvolvedores podiam vender os



aplicativos e os usuários podiam comprar, baixar e atualizar os aplicativos já adquiridos”.

Perboni (2013, p. 1) assim comenta a história do *Android*:

O primeiro telefone comercialmente disponível a rodar no sistema *Android* foi o HTC *Dream*, lançado a 22 de outubro de 2008. Em 27 de maio de 2010 Matias Duarte, antigo diretor da *interface* do *WebOS*, junta-se à equipe do *Android* após a compra da *Palm* pela HP. O primeiro *tablet* comercialmente disponível a rodar o sistema operacional *Android* 3.0 foi o *Motorola Xoom*, lançado a 24 de fevereiro de 2011 nos Estados Unidos.

O *Android* ganhou novas versões nos anos seguintes, ambas com as mesmas funções do *Android Market*, porém possuem opções mais modernas e atualizadas.

### 3.3 A Arquitetura da Plataforma *Android*

Várias camadas compõem a arquitetura *Android*, entre elas a *Applications*, *Application Framework*, *Android Runtime*, *Linux Kernel* e *Libraries*. Para Andrade et al. (2011) a arquitetura da Plataforma *Android* incorpora a concepção de flexibilidade e integração, de forma que os aplicativos fornecidos juntamente com o sistema podem interagir ou até mesmo serem substituídos por aplicações diversas que não são nativas do sistema.

Assim corrobora Lemos (2011, p. 10) sobre a arquitetura da Plataforma *Android*:

A arquitetura da plataforma *Android* é constituída por cinco módulos: *Kernel Linux*, Biblioteca, *Runtime* (ambiente de execução), *Framework* de Aplicações e Aplicações. A arquitetura da plataforma é baseada no *kernel* do *Linux*, ele funciona como uma camada de abstração entre o *hardware* e o restante dos *softwares* da plataforma. Com isso, ele tem várias formas de execução das aplicações. Por exemplo: gerenciamento de memória, gerenciamento de processos, pilhas de protocolos, módulo de segurança, entre outros.

Rabello (2013) explica que na camada *Applications* localiza-se uma lista de aplicações padrões que incluem gerenciador de contatos, navegador, cliente de *e-mail*, entre outros, sendo que é possível a comunidade desenvolver outras aplicações, todas elas escritas em linguagem *Java*. Já Lemos (2011, p. 10) tece comentários sobre o módulo Biblioteca:

Acima do *Kernel Linux*, temos o módulo Biblioteca, que são escritas em C/C++, compostas por uma série de bibliotecas utilizadas pelo *Android*, algumas delas são: biblioteca de sistema C e de mídias, gerenciador de superfície, *LibWebCore*, SGL, 3D *libraries*, *FreeType*, *SQLite*, entre outras.

O módulo *Runtime* é o ambiente de execução que é composta pela máquina virtual *Dalvik*, que foi desenvolvida com a intenção de que os dispositivos possam suportar máquinas virtuais múltiplas adequadamente, dessa forma, Lemos (2011, p. 10) explica que “*Dalvik* roda classes compiladas por um compilador de linguagem *Java*, além de usar o *kernel* para obter um bom funcionamento de múltiplos tópicos e gerenciamento de memória de baixo nível”.

A camada *Framework*, por sua vez contém os componentes que permitirão utilizar novas estruturas para que haja futuras aplicações, sendo reutilizado o código. A camada *Framework* é desenvolvida para o uso de *Application Programming Interface*, os APIS, que são aplicações chave do *Android*. Pacheco e Castro (2011, p. 7) salientam que “a arquitetura deste *Framework*

foi desenvolvida para simplificar a reutilização dos componentes. Desta forma qualquer desenvolvedor pode construir um aplicativo e disponibilizar suas “capacidades”, permitindo que elas sejam utilizadas por outros programas”.

*Activity, Service, BroadcastReceiver, Intent, ContentProvider* e *View* são componentes tidos como aplicações indispensáveis do *Android*.

A *Activity* constitui-se em uma tela do *Android*, funcionando como *interface* visual da aplicação, por meio do qual o usuário poderá interagir, ressalta-se que cada *Activity* independe uma da outra. A *Service* é mais uma aplicação e se responsabiliza pelas tarefas executadas em segundo plano e permite criar um serviço, a mesma não possui *interface* visual para o usuário. Já a *BroadcastReceiver* tem a função de receber e reagir a uma mensagem do SO, funciona como um mecanismo de alerta. A *Intent* contém as mensagens que são utilizadas no intuito de facilitar a ligação entre os componentes da aplicação, ela descreve a ação que se quer executar. Quanto ao *View* pode-se afirmar que é todo o componente da *interface* gráfica do SO, o mesmo tem a utilidade para definir os objetos visuais que se exibem na tela.

O *Kernel Linux* é responsável pelos principais serviços do sistema, como, por exemplo, gerenciamento de processos, gerenciamento de memória e segurança. Pacheco e Castro (2011, p. 8) reforçam:

A versão do *kernel Linux 2.6* possui serviços essenciais ao sistema, como segurança, gerenciamento de memória, gerenciamento de processos, rede e *drivers*. O *kernel* também funciona como uma camada de abstração entre o *hardware* do dispositivo e o resto do conjunto de *softwares* que são desenvolvidos em paralelo.

A importância de uma plataforma completamente aberta levou a necessidade da criação do *Android*, pois conforme o exposto pode-se observar que todas as camadas da Arquitetura desta Plataforma reutilizam componentes existentes e de código aberto, o que possibilita a realização de melhorias, quando necessário. O fato de os usuários poderem ter acesso ao código e poderem contribuir com esse projeto foi o que levou a comunidade científica a aderir à ideia do *Android* e assim contribuir para o desenvolvimento deste Sistema Operacional.

### 3.3.1 Máquina Virtual *Dalvik*

A *Dalvik* é um procedimento de máquina virtual que foi desenvolvida pela *Google*, escrita pelo engenheiro Dan Bornstein em colaboração com demais engenheiros da empresa, a mesma veio acoplada com o lançamento da JDK do *Android* em fins de 2007 e foi concebida com base na especificação POSIX (*Portable Operating System Interface*, podendo ser traduzido como *Interface* Portável entre Sistemas Operacionais) para sistema operacional *UNIX* (Sistema Operativo Portátil Multiusuário) com a finalidade de aperfeiçoar o gerenciamento de processo e isolamento *thread* (Linha de Execução).

Esta máquina virtual é um dos componentes do módulo *Runtime*, ambiente de execução,

ela foi otimizada especialmente para dispositivos móveis e, como já mencionado anteriormente, foi desenvolvida na intenção de que os dispositivos possam suportar múltiplas máquinas virtuais de modo adequado. De acordo com Rabello toda aplicação em *Android* roda dentro seu próprio processo, o que significa dizer que roda no contexto da sua instância de máquina virtual.

A *Dalvik* executa arquivos no formato *Dalvik Executable* com extensão *.dex*. Rabello (2013, p. 3) explica que “um arquivo *.dex* nada mais é do que uma espécie de *bytecodes* do *Java* (os famosos arquivos compilados *.class*) otimizados para o SO.”

Lemos (2011) corrobora que a *Dalvik* roda classes compiladas por um compilador de linguagem *Java* e usa o *Kernel* na obtenção de um apropriado funcionamento de diversos tópicos e com gerenciamento de memória de baixo grau.

Passos (2009, p. 16) reforça:

A máquina virtual *Dalvik* roda aplicações no formato *.dex* (do acrônimo em inglês *Dalvik Executable*). As aplicações do *Android* irão rodar no ambiente *virtual Dalvik* sempre. Isso significa que qualquer que seja o dispositivo, o ambiente em que o programa roda será sempre o mesmo (mesmo sendo celulares completamente diferentes ou, ainda, mesmo não sendo celulares), dando portabilidade às aplicações escritas para *Android*. [...] Cada aplicação no *Android* roda no seu próprio processo, com a sua própria instância da Máquina *Virtual Dalvik*. Em outras palavras um dispositivo rodando *Android* pode rodar múltiplas máquinas virtuais de maneira eficiente, e cada processo roda em cima da sua própria máquina virtual.

Quanto ao gerenciamento de memória de baixo-nível e outras funcionalidades da Máquina *Virtual Dalvik* dependem do *Kernel Linux* e melhoram significativamente o desempenho das aplicações.

O objetivo da *Google* ao criar a Máquina Virtual foi melhorar pontos em que a Máquina Virtual *Java* (JVM) falhava em relação à comunicação móvel. Dessa maneira a *Dalvik* apresenta como seus principais pontos a segurança, o desempenho e a possibilidade de evoluir o *Android*.

É possível compreender que toda a estrutura da Plataforma *Android* foi pensada no objetivo de se tornar mais atrativa do que qualquer outra desenvolvida para comunicação móvel. Outras características, além da arquitetura até aqui discutida fazem parte dos atrativos do *Android*, é o que veremos a seguir.

### 3.4 Linguagem *Java*

Segundo Indrusiak (1996) o desenvolvimento desta linguagem ocorreu em meados dos anos de 1990 nos laboratórios da *Sun Microsystems*, o intuito da Linguagem *Java* era ser eficiente e muito mais simples do que as linguagens computacionais que a antecederam. Inicialmente a Linguagem *Java* procurava produzir *software* para produtos eletrônicos de consumo, como agendas eletrônicas, por exemplo, sendo que Indrusiak (1996, p. 1) afirma que “um dos requisitos para esse tipo de *software* é ter código compacto e de arquitetura neutra”.

Ainda segundo Indrusiak (1996) a Linguagem não conseguiu sucesso comercial, este

só viria a partir da adaptação da *Java* para ser utilizada em microcomputadores conectados a *internet*.

Assim, Indrusiak (1996, p. 1) ilustra:

A linguagem obteve sucesso em cumprir os requisitos de sua especificação, mas apesar de sua eficiência não conseguiu sucesso comercial. Com a popularização da rede *Internet*, os pesquisadores da *Sun Microsystems* perceberam que aquele seria um nicho ideal para aplicar a recém criada linguagem de programação. A partir disso, adaptaram o código *Java* para que pudesse ser utilizado em microcomputadores conectados a rede *Internet*, mais especificamente no ambiente da *World Wide Web*. *Java* permitiu a criação de programas batizados *applets*, que trafegam e trocam dados através da *Internet* e se utilizam da *interface* gráfica de um *web browser*. Implementaram também o primeiro *browser* compatível com a linguagem, o *HotJava*, que fazia a *interface* entre as aplicações *Java* e o sistema operacional dos computadores. Com isso, a linguagem conseguiu uma popularização fora de série, passando a ser usada amplamente na construção de documentos *web* que permitam maior interatividade. Os principais *web browsers* disponíveis comercialmente passaram a dar suporte aos programas *Java*, e outras tecnologias em áreas como computação gráfica e banco de dados também buscaram integrar-se com o novo paradigma proposto pela linguagem: aplicações voltadas para o uso de redes de computadores.

Comenta Sobral e Claro (2008) que a Linguagem *Java* é capaz de criar aplicativos para *desktop*, *softwares* robustos, completos, bem como independentes, aplicativos para a *web* e ainda aplicações comerciais. A Linguagem *Java* dispõe de mecanismos que garantem a segurança dos aplicativos.

Sobral e Claro (2008) explanam que na Linguagem *Java* ao escrever programas podem ser criados *Appletes*, aplicativos e *Servlets*. Em relação à execução de aplicativos é necessário ter um interpretador instalado na máquina. O *Java* pode ser usado para a programação de aplicativos para dispositivos móveis, assim pode ser utilizado no desenvolvimento de aplicativos para *Android*, ressalva-se que o *Android* utiliza um *Kit* próprio chamado de SDK, sendo que o *Java* possui o “JDK é um kit de desenvolvimento *Java* fornecido livremente pela *Sun*. Constitui de um conjunto de programas que engloba compilador, interpretador e utilitários”. (SOBRAL; CLARO, 2008, p. 13)

*Java* é bastante versátil, o que permite ao programador utilizar-se muito mais do que de seus ambientes distribuídos e complexos, oferece uma linguagem de programação de usos geral, com diversos recursos que possibilitam uma variedade de aplicativos, tanto dependentes de conectividade como outros que são independentes da mesma. A linguagem *Java* apresenta algumas características principais que se procura resumir a seguir.

### 3.4.1 Principais Características da Linguagem *Java*

A simplicidade e eficiência da Linguagem *Java* são características que se destacam, pois *Java* é simples e de fácil aprendizado de migração, isso se deve ao fato de o mesmo possuir um reduzido número de construções. Sobre isso explica Indrusiak (1996, p. 3):

A diminuição das construções mais suscetíveis a erros de programação, tais como ponteiros e gerenciamento de memória via código de programação também faz com que a programação em *Java* seja mais eficiente. Contém um conjunto de bibliotecas que fornecem grande parte da funcionalidade básica da linguagem, incluindo rotinas de acesso à rede e criação de *interface* gráfica. Baseada no paradigma da Orientação a Objetos - encapsulamento em um bloco de *software* dos dados (variáveis) e métodos de manipulação desses dados - a linguagem permite a modularização das aplicações, reuso e manutenção simples do código já implementado.

Entre as características da *Java*, sobressai também o fato de a mesma ser orientada a objetos, paradigma que na contemporaneidade é o mais utilizado na construção de *softwares*, explicam Sobral e Claro (2008) que isso permite que se focalize um dado, ou seja, o objeto. Em *Java* há os tipos primitivos de dados que não são objetos “mas foram criados e incorporados ao *Java* para permitir uma melhor forma de utilização da linguagem pelos programadores. Outra característica importante da linguagem *Java* em relação à linguagem C++, é que *Java* não suporta herança múltipla” (SOBRAL; CLARO, 2008, p. 15).

Compilada e interpretada, são características apontadas por Sobral e Claro (2008, p. 15) no texto que se segue:

Um programa desenvolvido em *Java* necessita ser compilado, gerando um *bytecode*. Para executá-lo é necessário então, que um interpretador leia o código binário, o *bytecode* e repasse as instruções ao processador da máquina específica. Esse interpretador é conhecido como JVM (*Java Virtual Machine*). Os *bytecodes* são conjuntos de instruções, parecidas com código de máquina. É um formato próprio do *Java* para a representação das instruções no código compilado.

A Linguagem *Java* também é pronta para redes, o que significa dizer que as funcionalidades oferecidas por ela para o desenvolvimento de programas que manipulam as redes através das APIs são de grande potencial e simples. A Linguagem *Java* apresenta, ainda, como uma característica principal o fato de ser distribuído, conforme Sobral e Claro (2008, p. 15) “Programas *Java* são “*linkados*” em tempo de execução. Os *bytecodes* gerados durante a compilação só serão integrados na execução”.

Confirma Sobral e Claro (2008) que *Java* também apresenta multiprocessamento, pois suporta a utilização de *threads*, linhas de execução. Portabilidade é mais uma característica da Linguagem *Java* e permite executar qualquer arquitetura de *hardware* e sistema operacional, sem a necessidade de recompilar-se. Por fim, dentre as características da Linguagem *Java*, não se pode deixar de destacar sua segurança, pois fornece uma série de mecanismos a fim de manter a segurança dos aplicativos, sendo que um programa em *Java* não tem contato algum com o computador real, reconhecendo apenas a máquina virtual.

### 3.4.2 *Comma Separated Values (CSV)*

Importante para a construção do projeto aqui proposto e componente da Linguagem *Java* é o *Comma Separated Value*, conhecido pela sigla CSV, é um formato de arquivo, uma

estratégia de estruturação comum para arquivos de formatação de texto. Este formato de arquivo é usado com frequência para trocar dados entre aplicações diferentes.

O CSV pode ser traduzido como Valores Separados por Vírgula. Segundo Rouse (2006, p. 1):

Nos computadores, um arquivo CSV (valores separados por vírgula) contém os valores em uma tabela como uma série de linhas de texto ASCII organizados de modo que cada valor da coluna é separada por uma vírgula do valor da próxima coluna e cada linha inicia uma nova linha. Um arquivo CSV é uma forma de coletar os dados de qualquer tabela de modo que possa ser transmitido como entrada para outra aplicação orientada para a mesa, como um banco de dados relacional aplicação. *Microsoft Excel*, uma planilha principal ou aplicação de banco de dados relacional, pode ler arquivos CSV. Um arquivo CSV é por vezes referido como um arquivo simples.

De acordo com Suski (2013) o CSV é diferente de outros formatos de arquivos separados por delimitadores, pois usa um caractere, aspas duplas, em torno de caracteres reservados, que podem ser vírgulas ou novas linhas. Assim Suski (2013, p. 1) elucida que “o formato do arquivo CSV não requer uma codificação específica de caráter, ordem de *bytes*, ou formato terminador de linha”. E para um melhor entendimento da função do CSV, Suski (2013, p. 1) explica detalhadamente:

- Cada registro é uma linha terminada por uma alimentação de linha (ASCII / LF = 0x0A) ou um retorno de carro e um par de alimentação de linha (ASCII / CRLF = 0x0D 0x0A), no entanto, quebras de linha podem ser incorporados.
- Os campos são separados por vírgulas.
- Em algumas implementações CSV, esquerda e à direita espaços ou tabulações, ao lado vírgulas, são cortados. Esta prática é controversa e, de fato, é especificamente proibido pela RFC 4180, que afirma: "Os espaços são considerados como parte de um campo e não deve ser ignorado".
- Os campos com vírgulas incorporadas devem ser delimitados por caracteres de aspas duplas.
- Os campos com caracteres de aspas duplas incorporados devem ser delimitados por caracteres de aspas duplas, e os personagens de aspas duplas incorporadas devem ser representados por um par de caracteres de aspas duplas.
- Os campos com quebras de linha embutidos devem ser delimitados por caracteres de aspas duplas.
- Os campos com esquerda ou à direita espaços devem ser delimitados por caracteres de aspas duplas.

- Os campos podem sempre ser delimitados por caracteres de aspas duplas, se necessário ou não.
- O primeiro registro em um arquivo CSV pode conter nomes de colunas em cada um dos campos.

Quase todas as planilhas e sistemas de gerenciamento de banco de dados apoiam o formato de arquivo CSV o tornando um formato bastante simples, sendo que diversas linguagens de programação tem bibliotecas disponíveis que suportam os arquivos em CSV, até as aplicações de *softwares* modernas estão habilitadas a suportar tanto as importações quanto as exportações do CSV, pelo fato de o formato do mesmo ser reconhecido de forma ampla.

### 3.5 *Contact Provider*

*Contact provider*, ou provedor de contatos é um poderoso componente do *Android* que gerencia o repositório central de informações sobre os contatos do dispositivo móvel, de modo flexível. O provedor de contatos é visto no aplicativo de contatos do aparelho móvel, sendo possível o acesso dos dados em sua própria aplicação e realização de transferência de dados entre o dispositivo e os serviços *on-line*.

Suportando um número grande de dados, o provedor procura gerir o máximo deles para cada usuário, a API do provedor de contatos inclui o conjunto de classes e *interfaces* tornando mais fácil a recuperação de dados dos contatos do celular, bem como a modificação dos mesmos.

### 3.6 *vCard*

O *vCard* tem o propósito de ligar-se a mensagens de *e-mail*, também mensagens instantâneas, pois é um formato de arquivo padrão para eletrônicos que funciona como um cartão de visitas anexo as mensagens, contendo nomes e endereços, fotografias e, ainda, números de telefone, o que justifica sua menção neste trabalho. Através do *vCard* é possível estruturar e recuperar dados de um determinado contato, pois o mesmo segue um padrão de estruturação de informações de multimídia sendo utilizado por diversas aplicações.

De acordo com Rouse (2005, p. 1):

*vCard* foi desenvolvido com a cooperação da *Apple*, *AT&T*, *IBM* e *Siemens*, que transformou em uma especificação da indústria, e da *Internet Mail Consortium* (IMC), em 1996. Um *vCard* contém um nome, informações de endereço, data e hora, e, opcionalmente, fotografias, logotipos de empresas, clipes de som e informações de posicionamento geográficos. É basicamente apenas um cartão.

Rouse (2005) salienta, ainda, que o *vCard* constitui-se de uma especificação da Indústria Multimídia, em que os desenvolvedores de aplicativos de softwares possam criar programas

que processam *vCards* e que os mesmos contenham imagens, sons e texto, possibilitando, além de sua visualização, importação, arrastá-lo e soltá-lo para um outro endereço ou para outro aplicativo.

### 3.7 *Android Development Tools (ADT)*

Traduzido como Ferramentas de Desenvolvimento para *Android*, o *Android Development Tools* é um *plugin*, ou seja, uma extensão para o *Eclipse* IDE.

Segundo Ferreira (2012), este *plugin* prepara para o desenvolvimento de aplicações para o *Android* de forma simples e integrada, permitindo criação, construção, empacotamento, instalação, depuração e teste das aplicações *Android* de modo rápido e fácil.

### 3.8 *Java Development Kit (JDK)*

Para que seja possível desenvolver aplicativos em *Android* é preciso que esteja configurado um ambiente contendo a última versão do JDK, como é conhecido o *Java Development Kit*. É também necessária à configuração *Android* SDK, sendo que é recomendada a utilização do IDE para a codificação.

O *Android* SDK permite que os desenvolvedores elaborem as aplicações a partir de um dispositivo virtual para os aparelhos de celular e tablet, desde jogos a utilitários que façam uso das funções oferecidas pelos aparelhos, como *touchscreen*, telefonia GSM, Câmera, GPS, bússola, acelerômetro, *Bluetooth*, EDGE, 3G e *WiFi*. A plataforma apresenta suporte para mídias de áudio, vídeo e imagem, nos formatos MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF, bem como acelerador gráfico 3D, baseados no *OpenGL ES*. Os dados podem ser armazenados em *SQLite* e a plataforma traz um navegador integrado com base no código livre do motor *WebKit*. (XAVIER, 2013)



## 4 *BackShare: Uma Aplicação de Backup e Compartilhamento de Contatos para Android*

Este capítulo descreve a aplicação desenvolvida neste trabalho. Será detalhada as tecnologias utilizadas, assim como o desenvolvimento e funcionamento do sistema de *backup* para salvar em um ambiente *online* os contatos do celular que possui o SO *Android*, mostrando assim os resultados obtidos.

As etapas para execução do Trabalho de Conclusão de Curso *BackShare* uma Aplicação de *Backup* e Compartilhamento de Contatos para *Android* foram: o estudo das técnicas de estruturação de arquivos, o qual proporcionou a escolha e utilização da técnica que melhor se aplicou ao projeto desenvolvido. Essas técnicas visam a organização dos atributos dos contatos do celular (nome, sobrenome, *e-mail*, número, etc), como mostrado na Figura 1.

### vCard 3.0

```
BEGIN:VCARD
VERSION:3.0
N:Gump;Forrest;Mr.
FN:Forrest Gump
ORG:Bubba Gump Shrimp Co.
TITLE:Shrimp Man
PHOTO;VALUE=URL;TYPE=GIF:http://www.example.com/dir_photos/my_photo.gif
TEL;TYPE=WORK,VOICE:(111) 555-1212
TEL;TYPE=HOME,VOICE:(404) 555-1212
ADR;TYPE=WORK;;;100 Waters Edge;Baytown;LA;30314;United States of America
LABEL;TYPE=WORK:100 Waters Edge\nBaytown, LA 30314\nUnited States of America
ADR;TYPE=HOME;;;42 Plantation St.;Baytown;LA;30314;United States of America
LABEL;TYPE=HOME:42 Plantation St.\nBaytown, LA 30314\nUnited States of America
EMAIL;TYPE=PREF,INTERNET:forrestgump@example.com
REV:2008-04-24T19:52:43Z
END:VCARD
```

*Figura 1 – Exemplo da estrutura de um arquivo vCard 3.0*

Posteriormente foi feito o estudo sobre programação para *Android*, nessa etapa houve a instalação dos *softwares* necessários, tais como: a IDE (*Integrated Development Environment*) de Desenvolvimento *Eclipse*, o SDK que contém o emulador e todas as ferramentas necessárias para o desenvolvimento e o *plugin* ADT (*Android Development Tools*), fornecido pela *Google*. Após as instalações, foi criada uma configuração virtual AVD (*Android Virtual Device*) com

um SDC (*Secure Digital Card*) de 200 Mb, para que o emulador pudesse simular a execução do sistema em um celular real com um SDC.

Em seguida foi feito *wireframes* do sistema, ou seja, foi desenhado um protótipo simulando o funcionamento da aplicação. Foi desenvolvido um aplicativo simples para manipulação de arquivos, o mesmo foi utilizado posteriormente com a técnica de estruturação de arquivos.

Para mostrar o funcionamento do sistema em um dispositivo móvel real, foi utilizado o *Smartphone Galaxy SIII* e foram tirados *print screens* da tela do mesmo. Foram feitos testes em 4 dispositivos móveis e um emulador e os resultados obtidos serão mostrados na Seção 4.3. Os mesmos estão organizados em uma tabela para melhor entendimento dos dados.

## 4.1 Instalação e Configuração dos *Softwares*

Para desenvolvimento da aplicação deste trabalho, foi utilizado o Sistema Operacional *Windows 8 64-bits* pela facilidade que o mesmo proporcionou na instalação dos programas utilizados. Foi necessário também a instalação, configuração de bibliotecas e *plugins* para a implementação do sistema, os quais estão detalhados a seguir:

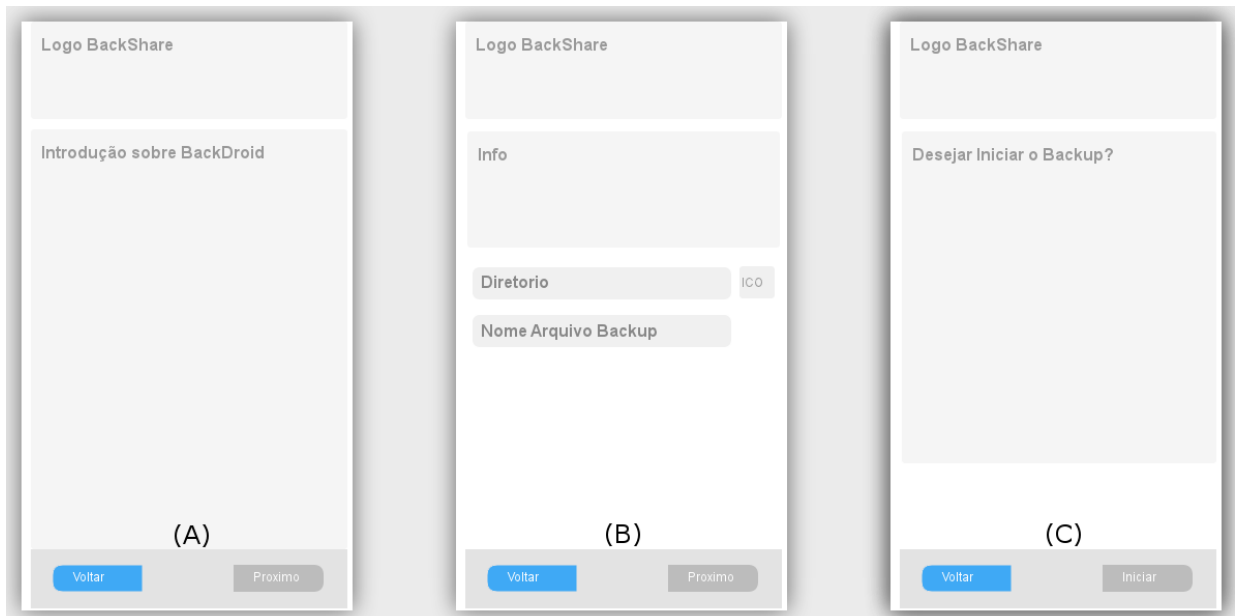
- **Eclipse:** é um IDE para desenvolvimento *Java* que suporta várias outras linguagens a partir de *plugins* como *C/C++*, plataforma *Android*, etc. Neste trabalho foi utilizado a versão *Juno* para implementação do aplicativo, usando como linguagem de programação *Java*.
- **SDK (*Software Development Kit*):** é um Kit de Desenvolvimento de Aplicativos da *Microsoft* que inclui documentação, código e utilitários para que programadores consigam desenvolver as suas aplicações de acordo com um padrão de desenvolvimento para o sistema operativo em questão. O mesmo foi instalado e configurado para implementação do aplicativo.
- **ADT (*Android Development Tools*):** é um *plugin* que contém ferramentas para facilitar a criação de projetos *Android*. O mesmo foi instalado no *Eclipse*.

Após a instalação da IDE *Eclipse* e do kit de desenvolvimento SDK, fez-se necessário a configuração do AVD, responsável por carregar a imagem do sistema e simular o *hardware* e *software* de um sistema *Android*, o mesmo foi configurado da seguinte forma: versão 4.2.2 (API level 17) do Sistema Operacional *Android*, *Screen 4.0"* (correspondente ao tamanho da tela do dispositivo), resolução(px) 350 X 600 e memória RAM de 512 MB.

## 4.2 Funcionamento do Sistema *BackShare*

O sistema de *Backup* e Compartilhamento de contatos foi desenvolvido para plataforma *Android* e atende usuários em geral, que necessitem salvar informações importantes de contatos telefônicos em um ambiente online, tais como, nome, número, *e-mail*, endereço, etc.

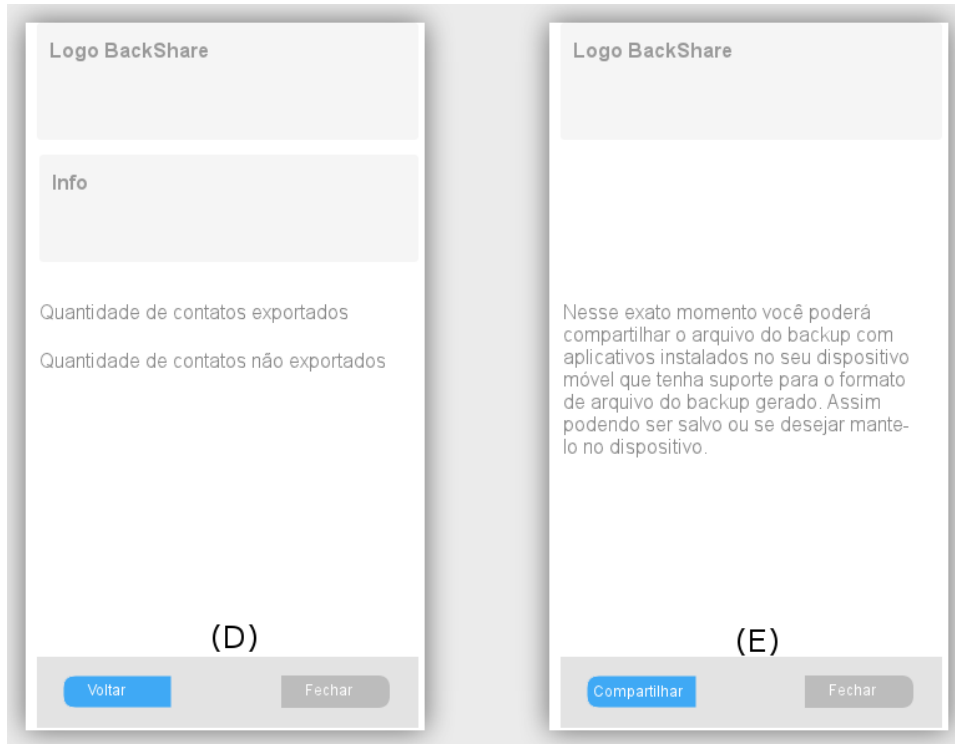
As Figuras 2 e 3 ilustram os *wireframes* da aplicação em questão, criados no início do projeto, como um protótipo visual do *layout* e funcionalidades da aplicação.



**Figura 2** – Wireframes das Três Telas Iniciais

O sistema em geral consiste em exportar informações dos contatos telefônicos para que possam ser armazenados e acessados remotamente independente de onde o usuário esteja.

Ao executar a aplicação é exibido inicialmente informações sobre o sistema (*Wireframe A*), posteriormente o usuário pode selecionar o local para salvar o arquivo de *backup* (*Wireframe B*) e em seguida o mesmo executa a ação de salvar na tela correspondente ao *Wireframe C*.



**Figura 3 – Wireframes das Telas Finais**

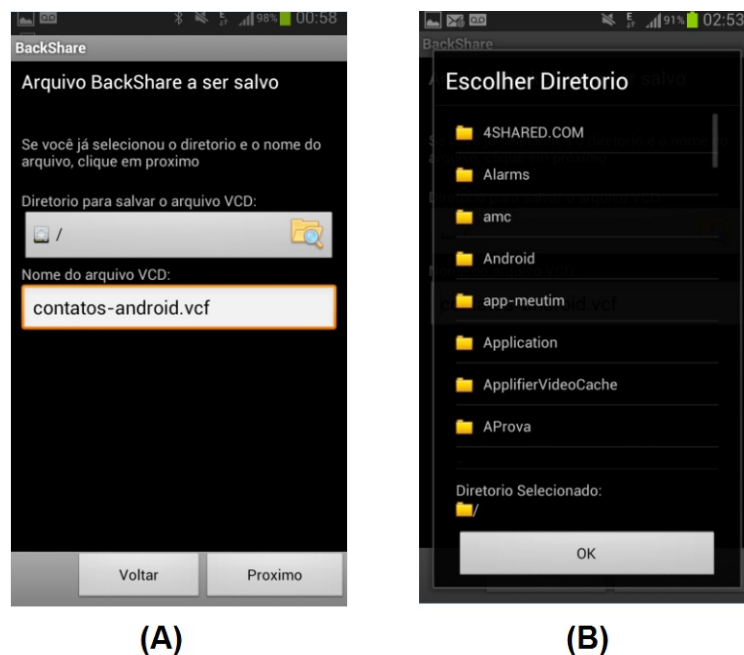
Em seguida é exibido a tela informando o processo de armazenamento dos contatos (*wireframe D*), posteriormente é mostrado para o usuário a tela de compartilhamento onde o mesmo pode compartilhar o arquivo de *backup* com outros aplicativos de armazenamento do celular (*wireframe E*).

Ao instalar o aplicativo no dispositivo móvel, é apresentado a tela inicial (Figura 4) ao usuário, na qual contém informações importantes do funcionamento da aplicação juntamente com a logomarca da mesma e dois botões (Voltar e Próximo). O usuário inicialmente não pode executar a ação de voltar, porque o botão correspondente a essa ação está desabilitado, estando habilitado apenas o botão próximo. Essa tela inicial é importante, pois é nela que está definido o objetivo da aplicação, para manter o usuário informado sobre o que a mesma vai desempenhar. O código correspondente a essa tela pode ser visto no Apêndice A.



**Figura 4 – Tela de Apresentação**

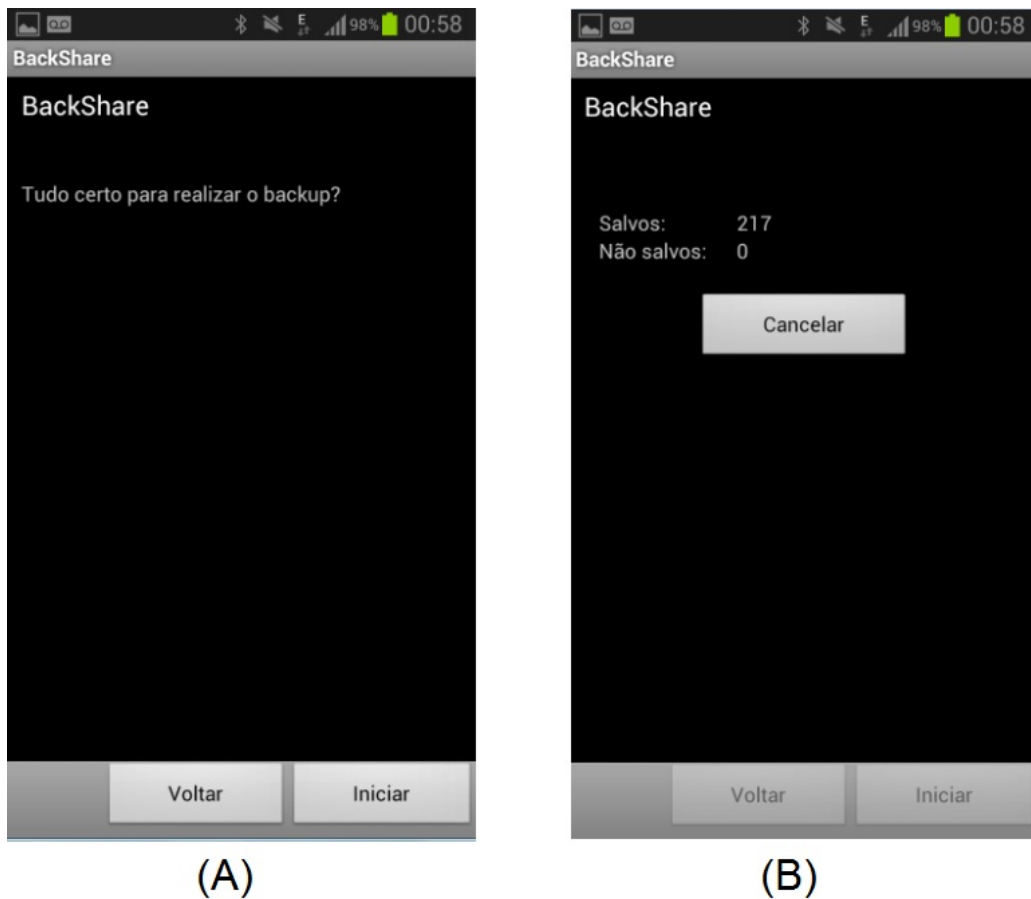
Na tela seguinte (Figura 5-A) o usuário pode escolher o local no dispositivo móvel que vai salvar o arquivo clicando no botão que contém um ícone simbolizando uma pasta, ao clicar no mesmo é exibido uma subtela (Figura 5-B) para escolher a pasta em que o arquivo é armazenado. Em seguida o usuário deve inserir o nome do arquivo com a extensão *.vcf*. Nesta tela os botões Voltar e Próximo estão habilitados, pois o usuário pode voltar para a tela anterior ou seguir para a próxima tela. O código correspondente a Figura 5 está inserido no Apêndice B.



**Figura 5 – Telas para Selecionar Diretorio e Nome do Arquivo**

Após o usuário selecionar o diretório e inserir o nome do arquivo de *backup*, aparece uma tela perguntando se o mesmo está preparado para efetuar o *backup* dos contatos, como mostrado na Figura 6-A. Esta tela contém dois botões, o botão Voltar e o Iniciar, sendo que o botão Voltar executa a ação de voltar para a tela anterior e o botão Iniciar dar início ao processo de *backup* dos contatos.

Quando o usuário clicar no botão Iniciar aparece a tela de processos (Figura 6-B), que contém informações sobre todo o andamento do *backup*, ou seja, quantos contatos foram salvos ou não. A tela contém também um botão Cancelar para que o usuário possa efetuar a ação de abortar o processo a qualquer momento, estando assim, desabilitados os botões de Voltar e Iniciar, pois o usuário não pode efetuar mais nenhuma ação que não seja cancelar. O código desta tela pode ser visto no Apêndice C.



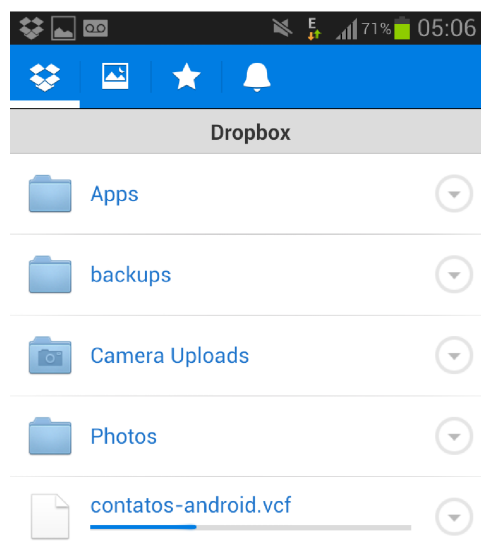
**Figura 6 – Telas de Processo**

Na tela seguinte, como mostrado na Figura 7-A, o usuário ao clicar no botão Compartilhar, pode realizar o compartilhamento do arquivo gerado pelo sistema *BackShare* com demais aplicativos, podendo assim, armazená-lo em qualquer ambiente *online* que tenha suporte para armazenamento de arquivos, tais como, por *E-mail*, *DropBox*, *SkyDrive* e etc, desde que os mesmos estejam instalados no dispositivo (Figura 7-B). O código dessas telas pode ser visto no Apêndice D.



**Figura 7 – Tela de Compartilhamento**

A Figura 8 mostra o compartilhamento do arquivo de *backup* com o aplicativo *DropBox* que já está instalado no dispositivo, facilitando assim, o armazenamento.



**Figura 8 – Compartilhamento com DropBox**

### 4.3 Resultados obtidos dos testes realizados em Dispositivos Móveis com SO *Android*

Com o objetivo de verificar se o sistema desenvolvido neste trabalho funciona corretamente, foram feitos testes em 4 (quatro) dispositivos móveis e um emulador do *Android*, onde cada um possuía características diferentes, tais como, versão do Sistema Operacional *Android* e modelo do *Smartphone*.

Os testes são apresentados em uma tabela, para melhor organização e entendimento das informações. Os mesmos permitiram avaliar o funcionamento do *backup* de contatos nos dispositivos mostrando assim os modelos de celulares, versões do SO em que foram realizados os testes e a quantidade de contatos que foram salvos e não salvos, os quais podem ser visualizados na tabela abaixo:

*Tabela 1 – Resultados Obtidos*

<b>Modelo Celular</b>	<b>Versão SO</b>	<b>Contatos Salvos</b>	<b>Contatos Não Salvos</b>
Emulador AVD	4.2.2	3	0
<i>Samsung Galaxy SIII</i>	4.1.2	645	0
<i>Galaxy Win I8550</i>	4.1.2	422	0
<i>Optimus L3 II Dual E435</i>	4.1.1	159	0
<i>Galaxy Pocket Plus S5301</i>	4.0	96	0

As imagens dos testes podem ser vistas no Apêndice E.



## 5 Conclusão e Trabalhos Futuros

### 5.1 Conclusão

O trabalho foi desenvolvido com o objetivo de apresentar a evolução dos dispositivos móveis e as dificuldades das pessoas nos dias de hoje em evitar a perda total dos dados dos contatos do celular. Devido a essa necessidade foi desenvolvido o sistema de *backup* de contatos, nomeado *BackShare*.

O sistema foi implementado para dispositivos móveis que possuem o Sistema Operacional *Android*. O mesmo foi criado para pessoas em geral, que desejam ter os seus contatos armazenados na rede mundial de computadores (*internet*), evitando assim, a perda total dos mesmos.

Com o aplicativo *BackShare* é possível oferecer aos usuários um meio de evitar a perda total dos dados dos seus contatos em caso de danos ou furto do seu dispositivo móvel, pois todas as informações dos contatos celulares estarão armazenados na *internet*, podendo ser por *E-mail*, *DropBox*, *SkyDrive* e *Mega*, desde que esses aplicativos estejam instalados no celular.

Com o desenvolvimento do *BackShare* foi possível estudar o *vCard* que é um método de estruturação de contatos, mais conhecido como arquivo de cartão de visitas, sendo o mesmo muito utilizado por várias aplicações no mundo inteiro. Ainda foi estudado o *ContactProvider*, traduzido como Provedor de Contatos, componente poderoso que gerencia o repositório central de informações sobre os contatos do dispositivo móvel, o mesmo foi utilizado neste trabalho. O Provedor de Contatos inclui um amplo conjunto de classes de contato e *interfaces* que facilitam tanto recuperação como a modificação dos dados.

Finalizando, através do percurso realizado nesse desenvolvimento e a conclusão dele decorrente, foi possível conhecer uma nova maneira de armazenamento de informações relevantes do dispositivo móvel, visto que, atualmente o furto de celulares tem aumentado devido a evolução dos mesmos, e conseqüentemente dados importantes são perdidos.

## 5.2 Trabalhos Futuros

Atualmente o *BackShare* realiza o *backup* dos contatos em um arquivo e os compartilha em um ambiente *online* podendo ser acessado de qualquer ambiente computacional que esteja conectado à *internet*.

Para trabalhos futuros, os objetivos a serem implementados serão: (a) implementar importação de contatos diretamente do sistema, ou seja, abrir o arquivo e inserir os contatos novamente no dispositivo móvel; (b) armazenamento de imagens juntamente com os dados dos contatos.

# Referências

- ABREU, Leonardo Marques. **Usabilidade de Telefones Celulares Com Base em Critérios Ergonômicos**. 2005. Disponível em: <[http://www2.dbd.puc-rio.br/pergamum/tesesabertas/0310214\\_05\\_cap\\_02.pdf](http://www2.dbd.puc-rio.br/pergamum/tesesabertas/0310214_05_cap_02.pdf)>. Acesso em: 19 jan. 2014.
- AMOROSO, Danilo. **O que é Computação em Nuvens?** 2012. Disponível em: <<http://www.tecmundo.com.br/computacao-em-nuvem/738-o-que-e-computacao-em-nuvens-.htm>>. Acesso em: 24 jan. 2014.
- ANDRADE, Bruno Henrique et al. **Desenvolvimento de uma Aplicação Embarcada em Celular Visando Controle de Robô Via Wi-Fi**. Revista Brasileira de Computação Aplicada, 2011. Acesso em: 27 dez. 2013.
- FERREIRA, Rodrigo Lopes. **Mini Curso Android**. 2012. Disponível em: <<http://www.dniel.com.br/dniel/cursoAndroid.pdf>>. Acesso em: 01 fev. 2014.
- GUIMARAES, Adionel Dayani. **Introdução às Telecomunicações Móveis**. 1998. Disponível em: <[http://www.inatel.br/revista/busca/doc\\_download/224-introducao-as-telecomunicacoes-moveis+&cd=1&hl=pt-BR&ct=clnk&gl=br](http://www.inatel.br/revista/busca/doc_download/224-introducao-as-telecomunicacoes-moveis+&cd=1&hl=pt-BR&ct=clnk&gl=br)>. Acesso em: 20 jan. 2014.
- INDRUSIAK, Leandro Soares. **Linguagem Java**. 1996. Disponível em: <<http://www.cin.ufpe.br/~arfs/introjava.pdf>>. Acesso em: 20 jan. 2014.
- LEMONS, Antônio Carlos. **Plataforma Android**. Minas Gerais, 2011.
- MACHADO, Renato. **Sistemas de telefonia celular no Brasil**. 2013. Disponível em: <<http://coral.ufsm.br/gpscom/professores/Renato%20Machado/Telefonia/TelCelular17Renato.pdf>>. Acesso em: 27 dez. 2013.
- OGLIARI, Ricardo. **As principais facilidades no desenvolvimento Android**. ITWEB, 2012. Disponível em: <<http://itweb.com.br/blogs/as-principais-facilidades-no-desenvolvimento-android/>>. Acesso em: 18 mar. 2013.
- PACHECO, Marco Antônio Júnior; CASTRO, Reinaldo de Oliveira. **Um estudo de caso da plataforma Android com Interfaces Adaptativas**. 2011. Disponível em: <[http://fgh.escoladenegocios.info/revistaalumni/artigos/Artigo\\_Marco%20Antonio.pdf](http://fgh.escoladenegocios.info/revistaalumni/artigos/Artigo_Marco%20Antonio.pdf)>. Acesso em: 19 jan. 2014.
- PANKIEWICZ, Igor. **Roubo de Celulares**. 2009. Disponível em: <<http://www.tecmundo.com.br/seguranca/1626-roubo-de-celulares.htm>>. Acesso em: 23 jan. 2014.
- PASSOS, Thiago de Souza. **Android, Arquitetura e Desenvolvimento**. Minas Gerais, 2009.

PEIXOTO, Berenice. **O Desenvolvimento da Telefonia Celular Pré-paga no Brasil e o Consumo da População de Baixo Poder Aquisitivo: análise dos fatores determinantes do período recente.** Universidade Federal da Bahia, Salvador, 2007.

PELLANDA, Eduardo Campos. **Comunicação móvel: das potencialidades aos usos e aplicações.** 2009. Disponível em: <<http://www.intercom.org.br/papers/nacionais/2008/resumos/R3-1727-1.pdf>>. Acesso em: 23 jan. 2014.

PERBONI, Marcos. **O que é Android – Sua História.** 2013. Disponível em: <<http://marcosvperboni.wordpress.com/em-breve-noticias-de-funcoes-do-android-para-leigos/>>. Acesso em: 17 dez. 2013.

PEREIRA, Mirella M; GUEDES, Leonardo G De. **Perspectivas das Comunicações Móveis no Brasil.** 2004. Disponível em: <[http://www.researchgate.net/publication/229052239\\_PERPECTIVAS\\_DAS\\_COMUNICAES\\_MVEIS\\_NO\\_BRASIL](http://www.researchgate.net/publication/229052239_PERPECTIVAS_DAS_COMUNICAES_MVEIS_NO_BRASIL)>. Acesso em: 23 jan. 2014.

RABELLO, Ramon Ribeiro. **Android: um novo paradigma de desenvolvimento móvel.** 2013. Disponível em: <[http://www.cesar.org.br/site/files/file/WM18\\_Android.pdf](http://www.cesar.org.br/site/files/file/WM18_Android.pdf)>. Acesso em: 18 dez. 2013.

ROCHA, Arnóbio. **4G e as Gerações do Telefone Celular.** 2013. Disponível em: <<http://arnobiorocha.com.br/2013/04/24/4g-e-as-geracoes-da-telefoniam-celular/>>. Acesso em: 21 jan. 2014.

ROUSE, Margaret. **O que é vCard?** 2005. Disponível em: <<http://whatis.techtarget.com/definition/vCard>>. Acesso em: 13 jan. 2014.

ROUSE, Margaret. **Arquivo de valores separados por vírgulas (CSV).** 2006. Disponível em: <<http://searchsqlserver.techtarget.com/definition/comma-separated-values-file&prev=/search%3Fq%3Dcomma%2Bseparated%2Bvalue%26start%3D10%26sa%3DN%26biw%3D1366%26bih%3D629>>. Acesso em: 02 jan. 2014.

SOBRAL, João Bosco Mangureira; CLARO, Daniela Barreiro. **Programação em Java.** 2008. Disponível em: <<http://homes.dcc.ufba.br/~dclaro/download/Programando%20em%20Java.pdf>>. Acesso em: 27 dez. 2013.

SUSKI, Sigrid. **Valores Separados por Vírgula.** 2013. Disponível em: <[http://www.sigsiu.net/csv\\_files.html&prev=/search%3Fq%3Dcomma%2Bseparated%2Bvalue%26start%3D10%26sa%3DN%26biw%3D1366%26bih%3D629](http://www.sigsiu.net/csv_files.html&prev=/search%3Fq%3Dcomma%2Bseparated%2Bvalue%26start%3D10%26sa%3DN%26biw%3D1366%26bih%3D629)>. Acesso em: 02 jan. 2014.

TAVARES, Mônica. **Número de celulares no Brasil é maior do que o de habitantes.** O Globo, 2010. Disponível em: <<http://oglobo.globo.com/economia/numero-de-celulares-no-brasil-maior-que-de-habitantes-2924116>>. Acesso em: 15 mar. 2013.

TELES, Ariel Soares. **Plataforma Android.** 2010. Disponível em: <[http://www.deinf.ufma.br/~fssilva/mestrado/icm/seminarios/introducao\\_android.pdf](http://www.deinf.ufma.br/~fssilva/mestrado/icm/seminarios/introducao_android.pdf)>. Acesso em: 19 dez. 2013.

XAVIER, Andressa. **Android SDK**. 2013. Disponível em: <<http://www.baixaki.com.br/download/android-sdk.htm>>. Acesso em: 04 fev. 2014.

XAVIER, Jonas et al. **Estudo da Evolução da Telefonia Móvel no Brasil**. 2006. Disponível em: <[http://www.inicepg.univap.br/cd/INIC\\_2006/inic/inic/07/INIC0000860.ok.pdf](http://www.inicepg.univap.br/cd/INIC_2006/inic/inic/07/INIC0000860.ok.pdf)>. Acesso em: 21 jan. 2014.

## APÊNDICE A – Classe Apresentação

```
package tcc.android.backshare;

import java.util.regex.Matcher;
import java.util.regex.Pattern;

import tcc.android.backshare.R;
import android.os.Bundle;
import android.text.util.Linkify;
import android.widget.TextView;

public class Introducao extends TelasActivity {

    @Override
    protected void onCreate( Bundle savedInstanceState )
    {
        setContentView( R.layout.introducao );
        super.onCreate( savedInstanceState );

        setProximoActivity( ConfigurarVCF.class );
    }
}
```

*Figura 9 – Código Tela Apresentação*

## APÊNDICE B – Algoritmo para Selecionar Diretório e Nome do Arquivo

```

package tcc.android.backshare;

import java.io.IOException;

public class ConfigurarVCF extends TelasActivity
{
    public final static int DIALOG_ESCOLHERARQ = 1;
    public final static int DIALOG_SEMSDCARD = 2;

    private EscolherArq _escolher_arq = null;

    // o diretório para salvar
    private String _diretorio;

    @Override
    protected void onCreate( Bundle savedInstanceState
    {
        setContentView( R.layout.configurar_vcf );
        super.onCreate( savedInstanceState );

        setProximoActivity( Fazer.class );

        // pega prefixo do diretório do cartão sd
        String prefixo_sdcard = obterPrefixoDir();
        if( prefixo_sdcard == null )
            showDialog( DIALOG_SEMSDCARD );

        // inicializa classe escolherarq
        _escolher_arq = new EscolherArq( this );
        _escolher_arq.setMode( EscolherArq.MODE_DIR );
        _escolher_arq.setDismissListener(

```

*Figura 10 – Código - Selecionar Diretório e Nome do Arquivo (Parte 1)*

```

        new DialogInterface.OnDismissListener() {
            public void onDismiss( DialogInterface dialog )
            {
                if( _escolher_arq.getOk() ) {
                    _diretorio = _escolher_arq.getPath();
                    AtualizaDirButton();
                }
            }
        } );
        _escolher_arq.setPathPrefix( "/sdcard" );

        Button path_button = (Button)findViewById( R.id.path );
        path_button.setOnClickListener( new View.OnClickListener() {
            public void onClick( View view ) {
                onBrowse();
            }
        } );
    }

    @Override
    protected void onPause() {
        super.onPause();

        SharedPreferences.Editor editor = getSharedPreferences().edit();

        // diretorio e nome do arquivo
        editor.putString( "path", _diretorio );
        EditText nomearq = (EditText)findViewById( R.id.filename );
        editor.putString( "filename", nomearq.getText().toString() );

        editor.commit();
    }

```

**Figura 11** – Código - Selecionar Diretório e Nome do Arquivo (Parte 2)



```

@Override
protected void onResume() {
    super.onResume();

    SharedPreferences prefs = getSharedPreferences();

    _diretorio = prefs.getString( "path", "/" );
    AtualizaDirButton();
    EditText filename = (EditText)findViewById( R.id.filename );
    filename.setText( prefs.getString( "filename",
        "contatos-android.vcf" ) );
}

static protected String obterPrefixoDir()
{
    // verifica o status do cartão sd
    String estado = Environment.getExternalStorageState();
    if( !Environment.MEDIA_MOUNTED.equals( estado ) &&
        !Environment.MEDIA_MOUNTED_READ_ONLY.equals( estado ) )
    {
        // sem cartão sd habilitado
        return null;
    }

    // pega o diretório do sd
    String dir_sdcard;
    try {
        dir_sdcard = Environment.getExternalStorageDirectory()
            .getCanonicalPath();
        if( dir_sdcard.charAt( dir_sdcard.length() - 1 ) == '/' )
            dir_sdcard =

```

**Figura 12** – Código - Selecionar Diretório e Nome do Arquivo (Parte 3)

```
    }  
    catch( IOException e ) {  
        dir_sdcard = null;  
    }  
  
    return dir_sdcard;  
}  
  
protected void AtualizaDirButton()  
{  
    Button path_button = (Button)findViewById( R.id.path );  
    path_button.setText(  
        _escolher_arq.prettyPrint( "/sdcard" + _diretorio, true ) );  
}  
  
protected void onBrowse()  
{  
    // pega diretorio  
    Button path_button = (Button)findViewById( R.id.path );  
  
    // definir um caminho  
    _escolher_arq.setPath( path_button.getText().toString() );  
  
    showDialog( DIALOG_ESCOLHERARQ );  
}
```

**Figura 13** – Código - Selecionar Diretório e Nome do Arquivo (Parte 4)

## APÊNDICE C – Algoritmo de Processos

```

public boolean ObterProximosContatos( Backup.ContactData contact )
{
    // configurar cursor
    if( _cursor == null )
    {
        // obtém todos os contatos agregados
        _cursor = _activity.getContentResolver().query(
            ContactsContract.Contacts.CONTENT_URI,
            new String[] {
                ContactsContract.Contacts._ID,
                ContactsContract.Contacts.DISPLAY_NAME,
            }, null, null, null );
    }

    // caso não existam contactos mais agregados, abortar
    if( _cursor == null ) return false;
    if( !_cursor.moveToNext() ) {
        _cursor.close();
        _cursor = null;
        return false;
    }

    // obter ID do contato agregado
    Long id = _cursor.getLong( _cursor.getColumnIndex(
        ContactsContract.Contacts._ID ) );

    // cria contato
    contact.setName( _cursor.getString( _cursor.getColumnIndex(
        ContactsContract.Contacts.DISPLAY_NAME ) ) );

    // obtém todos os dados de contato referentes ao contato agregado

```

**Figura 14** – Código - Armazenamento de Contatos em Arquivo (Parte 1)

```

Cursor cur = _activity.getContentResolver().query(
    ContactsContract.Data.CONTENT_URI,
    new String[]{
        ContactsContract.Data.MIMETYPE,
        ContactsContract.Data.IS_PRIMARY,
        ContactsContract.Data.DATA1,
        ContactsContract.Data.DATA2,
        ContactsContract.Data.DATA4,
    },
    ContactsContract.Data.CONTACT_ID + " = ? AND " +
        ContactsContract.Data.MIMETYPE + " IN ( ?, ?, ?, ?, ?, ? ) ",
    new String[] {
        String.valueOf( id ),
        CommonDataKinds.Phone.CONTENT_ITEM_TYPE,
        CommonDataKinds.Email.CONTENT_ITEM_TYPE,
        CommonDataKinds.Organization.CONTENT_ITEM_TYPE,
        CommonDataKinds.StructuredPostal.CONTENT_ITEM_TYPE,
        CommonDataKinds.Note.CONTENT_ITEM_TYPE,
        CommonDataKinds.Event.CONTENT_ITEM_TYPE,
    },
    ContactsContract.Data.IS_SUPER_PRIMARY + " DESC, " +
        ContactsContract.Data.RAW_CONTACT_ID + ", " +
        ContactsContract.Data.IS_PRIMARY + " DESC" );
while( cur.moveToNext() )
{
    String type = cur.getString( cur.getColumnIndex(
        ContactsContract.Data.MIMETYPE ) );

    // adiciona numero de telefones
    if( type.equals( CommonDataKinds.Phone.CONTENT_ITEM_TYPE ) )

```

**Figura 15** – Código - Armazenamento de Contatos em Arquivo (Parte 2)

```

// adicionar endereços postais
else if( type.equals( CommonDataKinds.StructuredPostal.CONTENT_ITEM_TYPE ) )
    contact.addAddress( contact.new AddressDetail(
        convertBackendTypeToType( CommonDataKinds.StructuredPostal.class,
            cur.getInt( cur.getColumnIndex(
                CommonDataKinds.StructuredPostal.TYPE ) ) ),
        cur.getString( cur.getColumnIndex(
            CommonDataKinds.StructuredPostal.FORMATTED_ADDRESS ) ) ) );

// adicionar companhias e títulos
else if( type.equals( CommonDataKinds.Organization.CONTENT_ITEM_TYPE ) )
    contact.addOrganisation( contact.new OrganisationDetail(
        cur.getString( cur.getColumnIndex(
            CommonDataKinds.Organization.COMPANY ) ),
        cur.getString( cur.getColumnIndex(
            CommonDataKinds.Organization.TITLE ) ) ) );

// adicionar notas
else if( type.equals( CommonDataKinds.Note.CONTENT_ITEM_TYPE ) )
    contact.addNote( cur.getString( cur.getColumnIndex(
        CommonDataKinds.Note.NOTE ) ) );

// adicionar aniversário
else if( type.equals( CommonDataKinds.Event.CONTENT_ITEM_TYPE ) ) {
    int event = cur.getInt( cur.getColumnIndex(
        CommonDataKinds.Event.TYPE ) );
    if( event == CommonDataKinds.Event.TYPE_BIRTHDAY )
        contact.setBirthday( cur.getString( cur.getColumnIndex(
            CommonDataKinds.Event.START_DATE ) ) );
}

```

**Figura 16** – Código - Armazenamento de Contatos em Arquivo (Parte 3)

## APÊNDICE D – Algoritmo para Compartilhamento do Backup Gerado

```
package tcc.android.backshare;

import java.util.regex.Matcher;
import java.util.regex.Pattern;

import tcc.android.backshare.R;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.text.util.Linkify;
import android.widget.TextView;

public class Compartilhar extends TelasActivity {

    @Override
    protected void onCreate( Bundle savedInstanceState )
    {
        setContentView( R.layout.compartilhar );
        super.onCreate( savedInstanceState );

        Intent shareIntent = new Intent(Intent.ACTION_SEND);
        shareIntent.setType("*/*");
        shareIntent.putExtra(Intent.EXTRA_STREAM, Uri.parse(Environment.getExternalStorageDirectory()+"/.
startActivity(Intent.createChooser(shareIntent, ""));|

    }

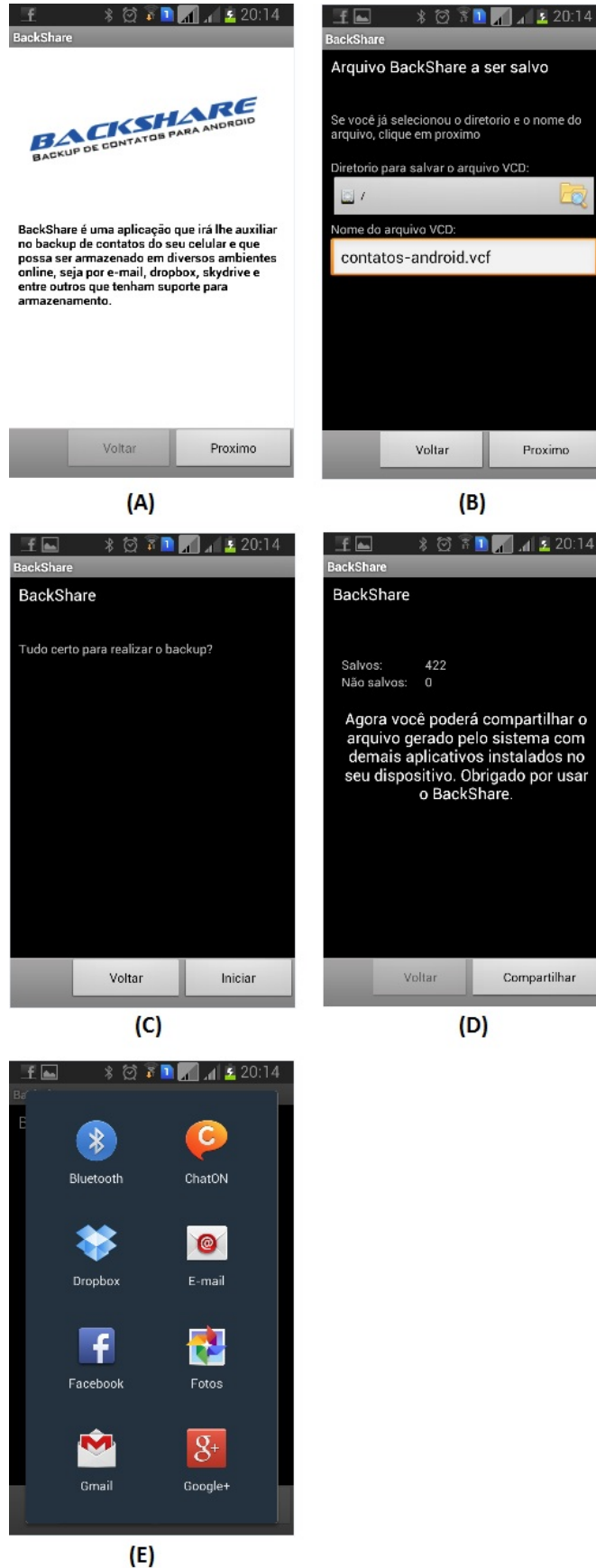
}
```

*Figura 17 – Código - Compartilhamento do Backup Gerado*

## APÊNDICE E – Teste Realizados



*Figura 18 – Galaxy Pocket Plus S5301*

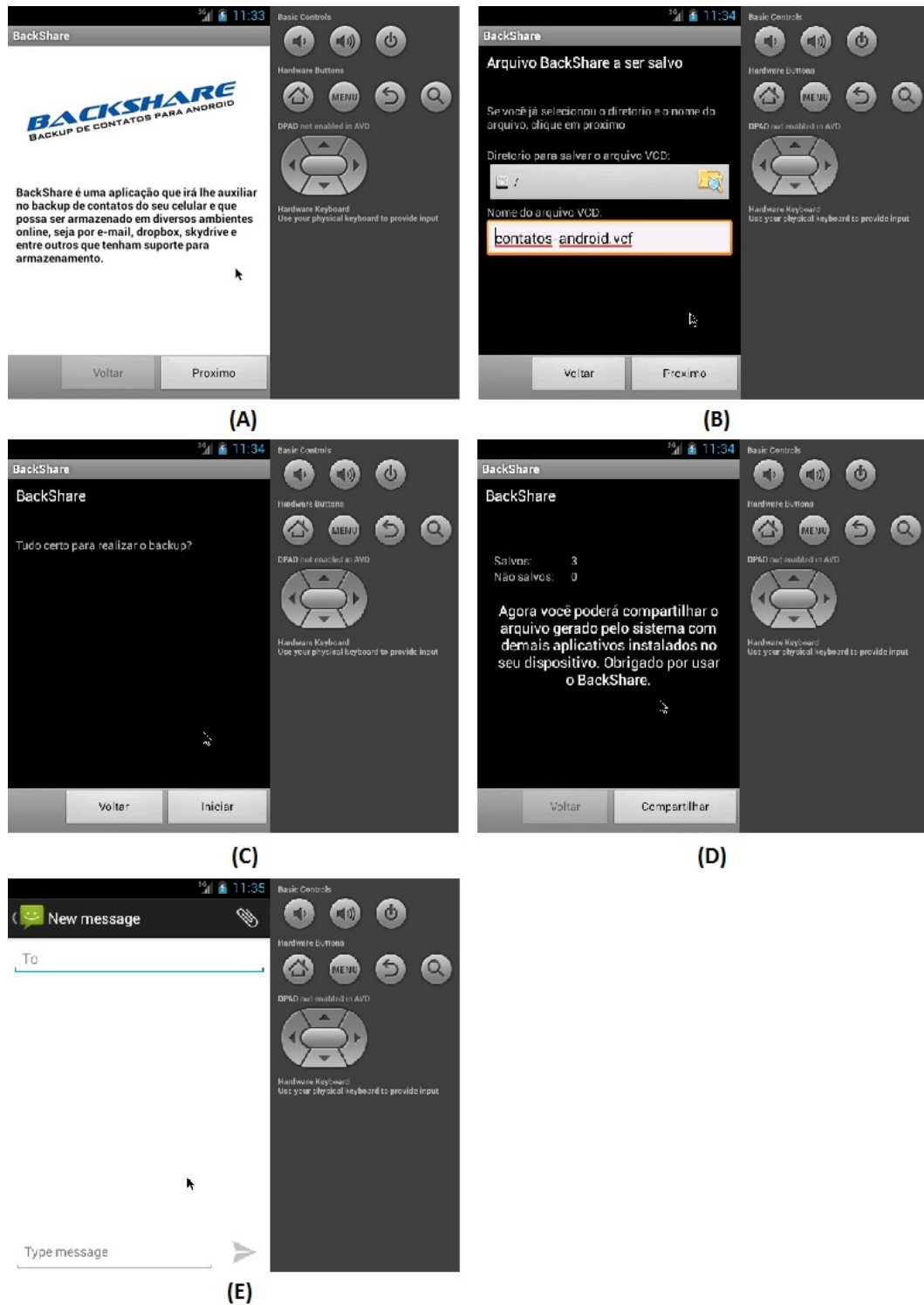


*Figura 19 – Galaxy Win I8550*





*Figura 20 – Optimus L3 II Dual E435*



*Figura 21 – Emulador AVD*