

**UNIVERSIDADE DO EXTREMO SUL CATARINENSE – UNESC
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

WILLIAM BERTAN DA SILVA

**INTERFACEAMENTO ENTRE DISPOSITIVOS MÓVEIS E JOGOS
ELETRÔNICOS DE COMPUTADORES UTILIZANDO O ACELERÔMETRO
BASEADO NO SISTEMA OPERACIONAL ANDROID**

CRICIÚMA

2012

WILLIAM BERTAN DA SILVA

**INTERFACEAMENTO ENTRE DISPOSITIVOS MÓVEIS E JOGOS
ELETRÔNICOS DE COMPUTADORES UTILIZANDO O ACELERÔMETRO
BASEADO NO SISTEMA OPERACIONAL ANDROID**

Trabalho de Conclusão de Curso, apresentado para
obtenção do grau de Bacharel no curso de Ciência da
Computação da Universidade do Extremo Sul
Catarinense, UNESC.

Orientador: Prof^o. MSc. Paulo João Martins.

CRICIÚMA

2012

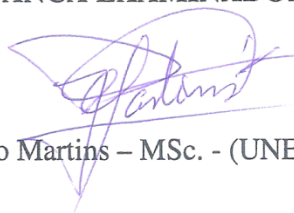
WILLIAM BERTAN DA SILVA

**INTERFACEAMENTO ENTRE DISPOSITIVOS MÓVEIS E JOGOS
ELETRÔNICOS DE COMPUTADORES UTILIZANDO O ACELERÔMETRO
BASEADO NO SISTEMA OPERACIONAL ANDROID**

Trabalho de Conclusão de Curso aprovado pela Banca Examinadora para obtenção do Grau de Bacharel, no Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC, com Linha de Pesquisa em Dispositivos Móveis.

Criciúma, 27 de junho de 2012.

BANCA EXAMINADORA



Prof. Paulo João Martins – MSc. - (UNESC) - Orientador



Prof. Sérgio Coral – Esp. - (UNESC)



Prof. Fabrício Giordani – Esp. - (UNESC)

**Dedico este trabalho aos meus amigos, pela
inspiração e descontração...**

AGRADECIMENTOS

Agradeço primeiramente a Deus, por me dar discernimento suficiente para a conclusão deste trabalho.

Aos meus pais Luiz e Cleci, e aos meus irmãos e irmãs, por me criarem com uma consciência moral bem definida e um incentivo à busca constante do conhecimento.

A minha namorada Francielle, por ser (muito) paciente comigo em todos os momentos, e me dar apoio e amor quando mais precisei.

Aos meus amigos Alisson e Ramon, por nas horas em que mais precisava de concentração e dedicação ao projeto, me levavam para as “Noites de CS” e passeios no shopping...

Ao Grupo Folclórico Italiano Valsugana e seus participantes, que me animaram nas horas difíceis desta caminhada e me proporcionaram muitos momentos felizes entre ensaios, apresentações e reuniões.

E a todos que de certo modo influenciaram para que eu pudesse conseguir vencer esta etapa, deixo aqui meu muito obrigado!

“I am not who I was,' he whispered, gripping the edges of the column, 'but I know who I am.'... 'And I won't give up.”

Christopher Paolini – Inheritance

RESUMO

Atualmente estamos em uma era da computação ubíqua, tendendo ficar conectados com tudo e com todos. Um dos meios para se alcançar esta computação pervasiva é o uso de dispositivos móveis aliados aos meios de comunicação sem fio, que proporcionam as interações e conectividades necessárias. Entre diversas tecnologias disponíveis ao usuário para escolha, está a Plataforma Android, em grande expansão visando o usuário com interfaces simplificadas e funcionais, além de auxiliar os desenvolvedores para a criação de aplicativos cada vez melhores e robustos. Os dispositivos móveis vêm com diversas tecnologias aliadas, um exemplo é o acelerômetro ao qual faz a captura da aceleração do dispositivo em três eixos distintos, com isto podendo ser utilizado para captura de movimentos que o usuário realiza com o aparelho. Esta integração entre as tecnologias é muito bem vista pela plataforma Android, ao qual já facilita sua utilização e integração. Outro nicho que esta em grande expansão é a área de jogos eletrônicos, onde os usuários estão mais exigentes com relação a formas de interação e imersão aos jogos. Visando uma integração com as tecnologias inovadoras desta era com o almejado pelos usuários, este projeto faz o uso do acelerômetro presente em um dispositivo móvel Android como forma de interação, ao modo de um joystick, realizando a conexão entre o aparelho do usuário e um servidor onde o jogo é executado. Esta nova funcionalidade ao aparelho foi desenvolvida utilizando-se do kit de desenvolvimento da plataforma Android (aplicação móvel) e Java (aplicação servidora) que proveram de todos os recursos necessários, sendo os testes realizados com o emulador disponível pelos desenvolvedores da plataforma e em dispositivos reais, sendo priorizado o uso de jogos eletrônicos *open source*. Através deste estudo espera-se que novas inovações com as tecnologias presentes e futuras possam ser criadas e difundidas pelas comunidades social e científica.

Palavras-chave: Plataforma Android. Dispositivos móveis. Acelerômetro. Jogos eletrônicos.

ABSTRACT

Currently we are in an era of ubiquitous computing, and tend to stay connected with everything and everyone. One way to achieve this pervasive computing is the use of mobile devices combined with wireless communication, which offer the necessary interactions and connectivity. Among several technologies available for the user to choose from, Android platform is one of them, in great expansion aimed at the users with simplified and functional interfaces, and helping developers to create better and robust applications. Mobile devices come with several combined technologies, an example is the accelerometer which makes the capture of the acceleration of the device in three different axes, thus can be used for motion capture that the user performs with the device. This integration between the technologies is quite welcome by the Android platform, which already facilitates their use and integration. Another niche that is in large expansion is the area of electronic games, where users are more demanding with regard to forms of interaction and immersion to the games. Aiming to integrate with innovative technologies that were desired by the users, this project is the use of accelerometer found in a mobile device Android as a way of interaction, like a joystick, making the connection between the user device and a server where the game is running. This new functionality to the device was developed using the development kit of the Android platform (mobile application) and Java (server application) which provided all the necessary resources, the tests were performed with the emulator available by the platform developers and in real devices, was prioritized the use of open source electronic games. Through this study it is expected that new innovations and future technologies can be created and disseminated by the social and scientific communities.

Keywords: Android Platform. Mobile devices. Accelerometer. Electronic games.

LISTA DE ILUSTRAÇÕES

Figura 1 - Tennis for Two	20
Figura 2 - Linha do tempo dos portáteis.....	25
Figura 3 - Linha do tempo da história do Android.....	42
Figura 4 - Arquitetura do Sistema Android.....	44
Figura 5 - Ciclo de vida de uma <i>Activity</i>	48
Figura 6 - Representação dos três eixos direcionais.....	53
Figura 7 - Android SDK e AVD Manager	62
Figura 8 - Dispositivo móvel com a plataforma 2.2 sendo emulado.....	62
Figura 9 - LogCat	63
Figura 10 - Modelo conceitual da aplicação móvel.....	63
Figura 11 - Diagrama de caso de uso da aplicação móvel.....	64
Figura 12 - Classes da Aplicação Móvel.....	65
Figura 13 - Tela inicial da aplicação móvel	66
Figura 14 - Atributos e Métodos da classe <i>Connectivity.java</i>	67
Figura 15 - Atributos e Métodos da classe <i>UserScreen.java</i>	68
Figura 16 - Classes do pacote <i>repository</i>	69
Figura 17 - Classes da aplicação Desktop	69
Figura 18 - Tela de Log da aplicação Desktop.....	70
Figura 19 - Tela Principal da aplicação Desktop.....	71
Figura 20 - Métodos da classe <i>MyFrame</i>	71

LISTA DE TABELAS

Tabela 1 - Performance e características do infravermelho.	29
Tabela 2 - Performance e características do Bluetooth.	30
Tabela 3 - Comparação entre Tecnologias WLAN	32
Tabela 4 - Características do padrão HomeRF	34
Tabela 5 - Performance e características da rede local sem fio sob padrão 802.11 e utilizando salto de frequências	36
Tabela 6 - Os quatro componentes possíveis de uma Aplicação Android	48
Tabela 7 - Principais métodos do ciclo de vida de uma <i>Activity</i> Android.....	49
Tabela 8 - Sensores comuns no Android	54
Tabela 9 - Valores de retorno dos Sensores	55

LISTA DE ABREVIATURAS E SIGLAS

A2DP	<i>Advanced Audio Distribution Profile</i>
AAC	<i>Advanced Audio Coding</i>
ADT	<i>Android Development Toolkit</i>
API	<i>Application Programming Interface</i>
ASL	<i>Apache Software License</i>
AVC	<i>Advanced Video Coding</i>
BRAN	<i>Broadband Radio Access Networks</i>
CD	<i>Compact Disc</i>
DECT	<i>Digital Enhanced Cordless Telecommunications</i>
DSSS	<i>Direct Sequence Spread Spectrum</i>
DVD	<i>Digital Versatile Disc</i>
EA	Electronic Arts
ETSI	<i>European Telecommunications Standards Institute</i>
FHSS	<i>Frequency Hopping Spread Spectrum</i>
FPS	<i>First-Person Shooter</i>
Gb	Gigabit
GHz	GigaHertz
GPL	<i>General Public License</i>
GPS	<i>Global Positioning System</i>
GSM	<i>Global System for Mobile Communications</i>
HIPERLAN	<i>High Performance Radio Lan</i>

HTML	<i>HyperText Markup Language</i>
IAPP	<i>Inter-Access Point Protocol</i>
IDE	<i>Integrated Development Environment</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
ISM	<i>Industrial, Scientific and Medical</i>
Kbps	Kilobits por segundo
LAN	<i>Local Area Network</i>
m/s ²	Metro por segundo ao quadrado
MAC	<i>Media Access Control</i>
Mb	Megabit
Mbps	Megabit por segundo
ME	<i>Micro Edition</i>
MP3	<i>Moving Picture Experts Group 1 Audio Layer 3</i>
MPEG-4	<i>Moving Picture Experts Group Phase 4</i>
mW	<i>Miliwatt</i>
NDK	<i>Native Development Kit</i>
NFC	<i>Near Field Communication</i>
Nm	Nanômetro
OFDM	<i>Orthogonal Frequency-Division Multiplexing</i>
OHA	<i>Open Handset Alliance</i>
PC	<i>Personal Computer</i>
PDA	<i>Personal Digital Assistant</i>

PSN	<i>Playstation Network</i>
QoS	<i>Quality of Service</i>
RAM	<i>Random Access Memory</i>
RF	<i>Rádio Frequência</i>
RFCOMM	<i>Radio Frequency Communications</i>
SDK	<i>Software Development Kit</i>
SE	<i>Standard Edition</i>
SGL	<i>Scalable Games Language</i>
SOHO	<i>Small Office/Home Office</i>
SQL	<i>Structured Query Language</i>
SSL	<i>Secure Sockets Layer</i>
USB	<i>Universal Serial Bus</i>
WLAN	<i>Wireless Local Area Network</i>

SUMÁRIO

1 INTRODUÇÃO.....	16
1.1 OBJETIVO GERAL.....	17
1.2 OBJETIVOS ESPECÍFICOS	17
1.3 JUSTIFICATIVA	17
1.4 ESTRUTURA DO TRABALHO	19
2 JOGOS ELETRÔNICOS	20
2.1 HISTÓRICO GERAL	20
2.2 CONSOLES	21
2.2.1 Atari 2600.....	21
2.2.2 Mega Drive (Genesis).....	22
2.2.3 Super Nintendo (SNES).....	22
2.2.4 Playstation	22
2.2.5 Nintendo 64	23
2.2.6 Playstation 2	23
2.2.7 Xbox.....	23
2.2.8 Xbox 360	24
2.2.9 Playstation 3	24
2.2.10 Nintendo Wii	25
2.3 PORTÁTEIS.....	25
2.4 COMPUTADORES	26
2.5 MERCADO ATUAL.....	26
3 CONECTIVIDADE WIRELESS	27
3.1 FUNDAMENTOS SOBRE FREQUÊNCIA.....	27
3.2 INFRAVERMELHO	29
3.3 BLUETOOTH.....	30
3.3.1 Bluetooth e Android	31
3.4 WIRELESS LAN	31
3.4.1 HomeRF	33
3.4.2 HIPERLAN/1 e HIPERLAN/2.....	35
3.4.3 Bluetooth	35
3.4.4 Família IEEE 802.11	36

3.4.5 Wi-Fi e Android	39
4 SISTEMA OPERACIONAL ANDROID	41
4.1 HISTÓRICO	41
4.2 PRINCIPAIS CARACTERÍSTICAS	42
4.2.1 Multiprocesso e App Widget	42
4.2.2 Toque, Gestos e Multitoque	43
4.3 ARQUITETURA	44
4.3.1 <i>Kernel Linux</i>	45
4.3.2 Bibliotecas	45
4.3.3 <i>Android Runtime</i>	46
4.3.4 <i>Framework</i> de Aplicação	47
4.3.5 Camada de Aplicação	47
4.4 ANATOMIA DE APLICAÇÕES	47
4.5 VERSÕES OFICIAIS	51
4.6 LICENÇA DE USO	52
5 ACELERÔMETRO	53
5.1 ACELERÔMETRO E ANDROID	54
6 TRABALHOS RELACIONADOS	57
6.1 NIIME	57
6.2 WEEWHEEL	57
6.3 ZOOZCONTROL	58
6.4 M-COMMERCE E A PLATAFORMA GOOGLE ANDROID	58
7 DESENVOLVIMENTO DA Aplicação	60
7.1 METODOLOGIA	60
7.1.1 Estudo das Ferramentas	61
7.1.2 Modelo Conceitual	63
7.2 IMPLEMENTAÇÃO	65
7.2.1 Aplicação Móvel	65
7.2.2 Aplicação Desktop	69
7.2.3 Funcionamento	73
7.3 RESULTADOS OBTIDOS	74
8 CONCLUSÃO	76
REFERÊNCIAS	78

BIBLIOGRAFIA COMPLEMENTAR.....	80
--------------------------------	----

1 INTRODUÇÃO

Com o intuito de diminuir o tamanho de objetos e agregar diversas funcionalidades a eles, é comum encontrar atualmente aparelhos que outrora eram de tamanhos exorbitantes e não executavam grandes tarefas, agora com tamanhos minúsculos executando centenas de vezes mais tarefas do que aparentam ser possível.

Um dos objetos que agregaram inúmeras funcionalidades pode-se citar os dispositivos móveis, celulares e os mais atuais, *tablets*. No início da difusão dos dispositivos móveis eram comuns celulares serem usados apenas para a comunicação, ou seja, ligações. Na atualidade, ao procurar por um celular, tem-se como base o que se pode desfrutar do mesmo, quais as capacidades que ele pode ter e o que se consegue fazer nele, qual o seu diferencial.

Com o mesmo princípio temos a evolução dos jogos eletrônicos, trazendo maior realismo nos gráficos, controles, forma de jogar. Uma das novidades são os controles sem fios, proporcionando melhor mobilidade e conforto. Assim também temos os periféricos (*mouse*, teclado, entre outros) os quais fazem uso de algum tipo de conexão *wireless* (RF, Wi-Fi, Bluetooth).

Conforme a competitividade aumenta no mercado onde estão inseridos os fabricantes de dispositivos móveis, ao lançar uma nova funcionalidade no seu produto também normalmente acabam por agregar um novo padrão, sendo que outros fabricantes tentarão se adequar para inserir em seus produtos funcionalidades similares. Com isto pode-se então dizer que um padrão para os produtos está sendo criado. Como um exemplo pode-se ter o celular, onde os atuais vêm com acelerômetro, tela *touchscreen*, processadores potentes, memórias dedicadas, processamento de vídeo independente, câmeras de alta resolução, Wi-Fi entre outras características. Assim tem-se uma noção de que novos produtos irão vir com estas funcionalidades e talvez até com outras e assim aumentando a gama de diversificação e diferencial.

A integração destes dispositivos com os já existentes tornam a vida mais fácil. Imagine controlar as luzes da casa, verificar e controlar a temperatura do ar condicionado, visualizar a câmera do jardim tudo isto com o uso de um celular por exemplo.

Um dos responsáveis por fazer os diferentes *hardwares* e *softwares* se comunicarem são as interfaces, que atuam como uma ferramenta para possibilitar a troca de

informações entre os componentes de forma a possibilitar que ambos compreendam informações um do outro.

Os novos consoles de jogos estão trazendo novidades ao mundo dos *Gamers*, além de gráficos e novos estilos de jogabilidade incríveis estão trazendo um novo conceito de interatividade. O X-Box 360 da Microsoft traz o atual e revolucionário Kinect onde conta com uma câmera infravermelha para capturar os movimentos do jogador e refletir em ações nos seus jogos trazendo uma imersão maior com o jogo e inúmeras aplicações e diversão. Com o console Playstation 3 da Sony veio o Move, controle com acelerômetro para pegar com precisão os movimentos do jogador e conta também com uma câmera chamada Playstation Eye. O console Wii da Nintendo também traz como o Move um controle (Wii Remote) para captura dos movimentos.

1.1 OBJETIVO GERAL

Utilização de dispositivos móveis na forma de um joystick para jogos eletrônicos, fazendo uso do sensor de acelerômetro.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos do projeto consistem em:

- a) descrever as características do desenvolvimento para dispositivos móveis com Android;
- b) apresentar a arquitetura e funcionamento de dispositivos móveis com Android;
- c) estudar e aplicar as capacidades do acelerômetro em ambiente móvel;
- d) documentar a evolução dos consoles em jogos eletrônicos;
- e) desenvolver uma aplicação utilizando-se do acelerômetro para captura de movimentos do dispositivo móvel para controle de jogo eletrônico em um servidor remoto.

1.3 JUSTIFICATIVA

Segundo Lee (2005) quando houve uma miniaturização dos computadores para os PDAs, *Tablets* entre outros, forneceu-se uma nova forma de interação com os computadores.

Surgiu com isto uma necessidade de portar as aplicações *desktops* para este novo ambiente móvel. Com as diferenças entre as plataformas houve alguns problemas, tais como: custos elevados de desenvolvimento, questões técnicas com os dispositivos móveis, conectividade imperfeita, complexidade de desenvolvimento das aplicações.

Uma aplicação móvel traz algumas oportunidades que acabam por aumentar os benefícios em qualquer negócio, sendo alguns exemplos: aumento de produtividade e vantagem competitiva. Também há o fato de que estas aplicações envolvem novas tecnologias e conceitos ao quais as maiorias das empresas ainda estão por aprender seu uso (MALLICK, 2003, tradução nossa).

Jogos de computadores possuem uma grande popularidade e representam uma grande parte das vendas na indústria dos jogos eletrônicos, tanto que muitos deles antes somente lançados para consoles dedicados agora já estão disponíveis também para computadores permitindo assim o jogador escolher qual plataforma utilizar. A interface utilizada pelo jogador para sua interação com o jogo é uma das formas mais importantes no design de um jogo, sendo importante ampliar a gama de possíveis periféricos utilizados como controle, tais como *mouse*, teclado, *gamepads*, *joysticks* entre outros (MORRISON, 2002, tradução nossa).

Segundo Lecheta (2010) os usuários de celulares estão procurando por diversas funcionalidades em seus aparelhos, como GPS, jogos, aplicativos corporativos, entretenimento, TV Digital entre outros. Empresas estão visando aumentar seus lucros investindo em desenvolvimento de suas aplicações para os dispositivos móveis. Estes desenvolvedores buscam uma plataforma ágil e moderna como também os usuários buscam uma interface amigável e de fácil acesso, para suprir estas necessidades e visando uma padronização de desenvolvimento para plataformas móveis, a Google junto com outras empresas de telefonia móvel fundou a Open Handset Alliance (OHA) e criaram a plataforma Android.

Baseado em um sistema operacional Linux, foi criado o Android que é uma nova plataforma para ambientes móveis, contendo uma interface visual rica, recursos de GPS, aplicações já instaladas (aplicações nativas) e conta com um ambiente de desenvolvimento inovador e flexível. Outra vantagem dele é que podem ser desenvolvidas aplicações utilizando a linguagem Java e seus recursos. Por ser de código livre possibilita desenvolvedores de todo o mundo contribuir tanto para novas funcionalidades quanto para a correção de falhas (LECHETA, 2010).

O Android vem com suporte para o Acelerômetro. Ele serve para medir a aceleração, podendo fazer leituras em três eixos (X, Y e Z) sendo sua leitura feita com base na taxa de alteração de velocidade, ou seja, ele mede quão rápido a velocidade do dispositivo está sendo alterada em uma determinada direção (FELKER, 2010, tradução nossa; MEIER, 2010, tradução nossa).

Com base nestas justificativas fica aberta uma nova forma de interação entre periféricos e jogos eletrônicos que seria a utilização do acelerômetro e dispositivos móveis como uma interface de controle em jogos eletrônicos.

1.4 ESTRUTURA DO TRABALHO

No capítulo 1 é explicada uma breve descrição sobre o tema da proposta, especificando o objetivo geral e os objetivos específicos e argumentado uma justificativa para o problema proposto.

No capítulo 2 é descrito sobre os jogos eletrônicos, sua história, os consoles que marcaram sua evolução, os consoles portáteis e computadores, chegando ao mercado atual da indústria dos jogos eletrônicos.

A conectividade sem fio (*wireless*) é tratada no capítulo 3, onde é feita uma explicação dos fundamentos sobre frequência para se ter conhecimentos básicos para um melhor entendimento dos assuntos abordados em seguida, ao qual é descrito sobre as tecnologias de infravermelho, Bluetooth e as redes sem fio (*wireless LAN*).

A história e as principais características do sistema operacional Android são descritas no capítulo 4, sendo também detalhada sua arquitetura, as versões oficiais lançadas e sobre as licenças aos quais foram lançadas.

As características, utilização do acelerômetro são abordadas no capítulo 5, como também sua integração ao Sistema Android.

No capítulo 6 são detalhados alguns trabalhos relacionados com o proposto, suas plataformas e funcionalidades.

O sistema proposto é apresentado no capítulo 7, ao qual foi detalhado a metodologia empregada, os recursos aplicados e o desenvolvimento da aplicação.

No capítulo 8 estão as conclusões obtidas com os resultados obtidos, como também sugestões de trabalhos futuros e dificuldades encontradas.

2 JOGOS ELETRÔNICOS

Conforme descrito por Morrison (2002, tradução nossa), a indústria de vídeo game cresceu se tornando um importante setor para a economia mundial. Jogos eletrônicos agora são financiados em escalas comparadas a filmes de Hollywood. Com esta passagem pode-se perceber como a indústria de jogos eletrônicos está crescendo e se destacando mundialmente.

A seguir, conforme Arkade [2011?], será descrito um pouco a trajetória desta indústria que movimenta bilhões anualmente e com uma tendência de aumentar a cada ano¹.

2.1 HISTÓRICO GERAL

Conforme descrito em Arkade [2011?] foi nos computadores onde surgiram os primeiros jogos eletrônicos, sendo estes computadores considerados para a época supermodernos. Eles eram de imenso porte e não disponível ao público em geral. Sua utilização estava restrita aos governos, laboratórios e faculdades.

Tennis for Two ostenta o título de primeiro game criado no mundo. Sua criação deu-se em um laboratório de pesquisas militares nos Estados Unidos, ele foi programado em uma máquina equivalente a um computador analógico e a forma de se interagir era por meio de um osciloscópio. A Figura 1 demonstra uma imagem do jogo.

Figura 1 - Tennis for Two



Fonte: ARKADE [2011?]

¹ Mais informações disponíveis em: <http://www.papodeempreendedor.com.br/oportunidades/o-mercado-dos-jogos-eletronicos/>

Por serem, na década de 60, artigos de luxo, os computadores e equipamentos necessários para se jogar jogos eletrônicos eram praticamente exclusivos de cientistas e pesquisadores. Mas a evolução dos games foi iniciada, inúmeras plataformas foram lançadas, sendo que sua popularização e evolução, tais quais conhecemos nos dias atuais, deve-se com o advento dos consoles domésticos.

2.2 CONSOLES

A seguir serão listados alguns dos principais consoles que revolucionaram alguns aspectos, desde interatividade a gráficos.

2.2.1 Atari 2600

Criado em 1977 pela empresa Atari, foi ele quem realmente foi decisivo pela popularização dos videogames. Dentre suas especificações estava uma memória de 128 bytes. Sua gama de jogos era limitada, quando somente a própria Atari desenvolvia games para o console, porém ao liberar para terceiros o desenvolvimento, surgiu a gigante Activision que criou grandes clássicos para o console, porém além de alavancar as vendas e aumentar a diversidade, esta abertura as *third parties*² pode ser considerado também sua maldição. O mercado ficou tão saturado de centenas de jogos de péssima qualidade.

Suas vendas totais foram mais de 40 milhões de unidades, e sua biblioteca de jogos é formada por mais de 900 títulos, sendo que menos de um terço destes foram considerados clássicos dos games. Foi descontinuada sua comercialização em 1990, e devido ao surgimento dos computadores pessoais e a preços mais acessíveis, este último começou a derrubar o mercado dos consoles.

² Empresas de terceiros que não estão ligadas diretamente no produto primário. Neste caso, são outras desenvolvedoras de jogos que não a própria Atari.

2.2.2 Mega Drive (Genesis)

Criado pela Sega no final de 1988 e após um fracasso com outro console, o Master System, foi lançado o Mega Drive (nome oficial Genesis), com isto iniciou-se o que se chamou da quarta-geração de consoles.

Na época, ainda estavam em alta os jogos em fliperamas, assim a Sega ganhou muitos usuários por lançar inúmeros *ports*³ dos clássicos de fliperamas. Seus gráficos já eram superiores aos seus antecessores e contava com uma boa interatividade com o usuário, e gráficos que na época impressionavam.

O final de sua vida útil foi em 1997 e contava com quase mil jogos em sua plataforma e mais de 40 milhões de unidades vendidas mundialmente. Dentre seus jogos formaram-se clássicos como *Sonic the Hedgehog* e *Streets of Rage*.

2.2.3 Super Nintendo (SNES)

Outro console que obteve grande destaque durante a quarta geração foi o Super Nintendo da Nintendo. Lançado em 1990, dois anos após o Mega Drive, fez com que a criadora equipasse seu console com um chip gráfico e de som superior aos concorrentes, mesmo com seu processador mais lento, obteve êxito, ganhando em números de vendas do Mega Drive com mais de 49 milhões de unidades vendidas.

Descontinuado em 1996, contava com acervo de quase três mil jogos, clássicos como *Super Mario World* e franquias tais quais *Donkey Kong*, *Megaman* e *Final Fantasy*. É considerado um dos consoles mais bem-sucedidos do mundo.

2.2.4 Playstation

A chegada da quinta geração dos consoles veio com aparelhos menos conhecidos como o Jaguar e o 3DO, e um pouco mais difundido o Sega Saturn, porém foi o console de uma empresa que até então não estava na área dos jogos eletrônicos, a Sony, que se consolidou com o Playstation e ficou por certo tempo como o console mais vendido, sendo

³ Derivado da palavra portabilidade (*portability*), em relação a programas e jogos, é um processo de adaptação de código criado em uma plataforma de modo que seu resultado final também seja compatível com outras plataformas.

apenas superado pelo sucessor Playstation 2, também da Sony (MUNDO ESTRANHO, 2011).

Nesta geração os jogos em 3D se tornaram acessíveis. Gráficos que impressionavam os *gamers*⁴. O console foi descontinuado em 2006, e deixando em seu histórico mais de 100 milhões de consoles vendidos e um acervo de quase 2.500 títulos.

2.2.5 Nintendo 64

A empresa Nintendo também teve sua participação nos consoles da quinta geração, com o Nintendo 64 no ano de 1996. Surpreendeu por ainda utilizar a tecnologia de cartuchos para armazenamento dos jogos, os concorrentes já utilizavam o CD como mídia.

2.2.6 Playstation 2

Na próxima geração, a Sony que havia surpreendido a todos, lançou em março de 2000 o Playstation 2. Sucesso de vendas a ponto de que o primeiro lote não atendeu a tanta demanda.

Visando a uma nova ideia de “central de entretenimento” o Playstation 2 aderiu o conceito. Além de jogos também têm a possibilidade de reproduzir músicas e filmes, e em sua porta USB poderiam ser ligados alguns *gadgets*⁵, além da retro-compatibilidade com os jogos do seu antecessor, o Playstation.

2.2.7 Xbox

Na sexta geração, mesma do Playstation 2 da Sony e GameCube da Nintendo, a Microsoft se lançou no mercado de consoles. Muito conhecida pelos softwares para computadores, com seu console não foi muito diferente, lançado em 2001 logo foi o segundo mais bem vendido desta geração com aproximadamente 25 milhões de unidades vendidas e cerca de mil títulos para o console.

⁴ Termo usado comumente para designar aqueles que passam grande parte do tempo livre jogando ou aprendendo sobre jogos eletrônicos.

⁵ Palavra em inglês para dispositivo é um equipamento com uma função e propósito específico, prático e útil. Normalmente são considerados os eletrônicos portáteis como celulares leitores de mp3, PDAs, entre outros.

A principal característica é a facilidade em realizar um *port* de um jogo de computador para a plataforma, assim teve vários títulos exclusivos e famosos como *Doom 3*. A criação da *Xbox Live* auxiliou nas suas vendas, onde jogadores poderiam se encontrar para bate-papo, realizar partidas online, dentre outras funcionalidades.

2.2.8 Xbox 360

Na sexta geração dos consoles a Microsoft foi a última a iniciar, na geração seguinte ela foi a primeira, iniciando assim a sétima geração dos consoles e a que perdura atualmente. Lançando o Xbox 360 no final de 2005, a Microsoft investiu em um *hardware* mais robusto, alta definição e controles sem fio, além de aprimorar a rede *Xbox Live*.

Mesmo com um problema em suas primeiras versões ocasionados por uma falha de *hardware* e aquecimento, ainda conquistou vários jogadores com suas franquias exclusivas como *Halo*, *Gears of War*, *Fable* e tem mais de 55 milhões de consoles vendidos e conta com um acervo de mais de 900 títulos. A *Xbox Live* contribui com games, expansões, vídeos e conteúdos exclusivos para os mais de 25 milhões de assinantes (MUNDO ESTRANHO, 2011).

O que revolucionou a plataforma foi no ano de 2010 o lançamento do Kinect. Um equipamento no qual faz a leitura dos movimentos dos usuários e aumenta a interação entre o jogador e o game.

2.2.9 Playstation 3

Lançado pela Sony em 2006 tem o diferencial e aposta na alta definição sendo o primeiro console com compatibilidade em mídias *blu-ray*, mídias estas que possuem maior capacidade de armazenamento quando comparados a um DVD comum (MUNDO ESTRANHO, 2011).

O início foi difícil para a plataforma, devido ao *hardware* complicado e o tipo de mídia, muitos desenvolvedores hesitaram em relação ao alto custo para o desenvolvimento de jogos. Porém esta situação foi revertida com o lançamento de kits de desenvolvimento que auxiliaram no desenvolvimento.

Com o crescimento da comunidade online, a Sony lança a PSN, ao estilo da *Xbox Live*, que atende milhares de jogadores para conteúdos exclusivos e partidas em rede.

Tendo cerca de 52 milhões de unidades comercializadas, não é a primeira no ranking de vendas. Para a diversificação lançou o seu próprio controle a base de movimentos, o PS Move.

2.2.10 Nintendo Wii

A Nintendo, fora do mercado de consoles por mais de uma década, voltou e produziu o Nintendo Wii, console este que com 86 milhões de unidades vendidas alcançou o título do mais vendido no mundo e conta com um acervo de mais de 900 títulos.

A aposta da Nintendo foi para o público *gamer* e o público casual, este último garantiu uma grande fatia das vendas. O motivo está relacionado ao formato de interatividade do jogador, ao qual faz uso do Wii-mote, o precursor dos controles baseados no movimento do usuário. Tanto foi o sucesso que este tipo de controle fez que inspirou a concorrência a seguir a mesma linha e tentar conquistar uma fatia deste mercado, assim a Sony criou o PS Move e a Microsoft o Kinect (MUNDO ESTRANHO, 2011).

2.3 PORTÁTEIS

Os consoles portáteis tiveram início em 1989 com o Game Boy. Provendo aos usuários mobilidade e diversão, diversos portáteis foram lançados posteriormente, como se pode visualizar na Figura 2 a linha do tempo dos portáteis.

Figura 2 - Linha do tempo dos portáteis



Fonte: ARKADE [2011?]

2.4 COMPUTADORES

Em conjunto ao mercado dos consoles também se teve a evolução dos computadores. Tendo seu início na década de 90, os computadores se tornaram cada vez mais pessoais e com isto se popularizaram, e junto surgiu o mercado de jogos eletrônicos.

Com os computadores surgiram alguns gêneros como o *First Person Shooter* (FPS) e alguns títulos clássicos. O poder destas máquinas também cresceram, como exemplo, na época do lançamento do jogo *Doom* para computadores, uma máquina com o processador 486 era considerada topo de linha, e isto que possuía, em média, apenas 16 Mb de memória RAM, hoje em dia os computadores podem possuir mais de 4 Gb, isto é 250 vezes mais, e ainda os computadores modernos contam com processadores velozes, placas de vídeo de última geração com processadores gráficos exclusivos e periféricos.

2.5 MERCADO ATUAL

Na atualidade os jogos eletrônicos estão mais presentes no cotidiano da população. Podendo ser jogados em lugares indistintos com dispositivos móveis ou no conforto de sua casa com os consoles, computadores e demais dispositivos.

Morrison (2002, tradução nossa) descreveu o motivo pelos quais jogos eletrônicos são tão bem vindos à sociedade moderna. Ele faz uma comparação com os filmes atuais, aos quais poucas pessoas são capazes de assistir um filme e não sentir que em algum momento não estão fazendo parte de determinada cena. Sendo que os jogos eletrônicos possibilitam esta imersão e ainda dão ao usuário o controle deste mundo.

Reforçando a ideia de Morrison temos Dawson (2010, tradução nossa) quando afirma que a indústria de jogos para computadores é um rival para o melhor que Hollywood pode oferecer. Em uma imersão do jogador a um jogo por horas, ele não fica apenas limitado a assistir seu herói participar das batalhas, ele acaba se tornando o herói.

Os fios tiveram um grande papel na evolução dos consoles e seus periféricos, conectando-os e proporcionando ao usuário imersão aos jogos e aplicações. Com a evolução das tecnologias empregadas em conectividade, a tecnologia sem fio (*wireless*) vem substituindo os fios das conexões possibilitando imersões do usuário com novas formas de interação.

3 CONECTIVIDADE WIRELESS

Segundo Mallick (2003, tradução nossa), tecnologias de conexões sem fios são um dos tópicos mais abordados em dispositivos móveis e todos possuem uma opinião sobre o estado da tecnologia das redes 3G, o efeito do Bluetooth nas redes pessoais, e como as redes locais sem fios dominarão o mercado. Redes sem fios (*wireless networks*) possuem vários propósitos, em alguns deles são para substituir o uso de cabos, enquanto em outros para disponibilizar acesso a dados de locais remotos.

Segundo Muller (2000, tradução nossa) as pessoas gastam um grande tempo tentando esconder os vários cabos e fios que conectamos os equipamentos. É um fio conectando um mouse e outro do teclado ao computador, fios das caixas acústicas, microfone de uma estação multimídia, cabos conectando a câmera, dispositivos de armazenamentos, celulares para transferência de dados. Chegando a um ponto em que as nossas pastas fiquem cheias com os cabos para conectar os diversos dispositivos que carregamos conosco. Em uma sociedade cada vez mais conectada, conectar as coisas se tornou um grande problema.

3.1 FUNDAMENTOS SOBRE FREQUÊNCIA

Segundo Mallick (2003, tradução nossa), algumas tecnologias sem fio, como a *Wireless Local Area Network* (WLAN)⁶, transmitem suas informações via ondas de rádio. Em um processo chamado de modulação, os dados são convertidos e transmitidos pelas ondas de rádio, em algo denominado ondas de carregamento, pois carregam os dados. Existem alguns tipos de técnicas para a modulação, sendo que cada um possui suas vantagens e desvantagens em termos de eficiência e requerimentos de energia. A seguir Mallick (2003, tradução nossa) descreve um pequeno resumo sobre cada uma das técnicas mais utilizadas.

***Narrowband Technology* (Tecnologia de banda estreita)** são sistemas de rádio que utilizam a banda estreita transmitem e recebem dados em uma frequência específica. A banda da frequência é deixada a mais estreita possível para permitir a informação de passar. Interferências são evitadas coordenando diferentes usuários para diferentes frequências. O receptor filtra todos os outros sinais, exceto da faixa designada. Para uma companhia que se utiliza desta tecnologia, é necessário uma licença do governo.

⁶ Faz a conexão entre dois ou mais dispositivos utilizando-se de algum método de distribuição sem fios.

Spread Spectrum (Espectro espalhado), por definição, troca a eficiência da banda por confiança, integridade e segurança. Isto consome mais largura de banda do que a tecnologia de banda estreita, porém produz sinais mais altos e fáceis dos receptores detectarem quando estes sabem os parâmetros em que o sinal está sendo distribuído. Para o resto o sinal de espectro espalhado aparentará como ruído de fundo. Existem duas variações desta tecnologia:

- a) *Frequency-Hopping Spread Spectrum (FHSS)*: esta é o salto de frequência, ela utiliza uma banda estreita que rapidamente fica circulando por uma gama de frequências. Tanto o emissor quanto o receptor sabem o padrão que está sendo utilizado. A ideia é que se uma determinada frequência estiver bloqueada, outra poderá estar livre, se este não for o caso, o dado é reenviado. Quando sincronizado apropriadamente, o resultado é um canal lógico simples onde as informações serão transmitidas. Normalmente a taxa de transferência é por volta de 1 Mbps;
- b) *Direct-Sequence Spread Spectrum (DSSS)*: esta é a sequência direta, DSSS espalha o sinal simultaneamente sobre algumas faixas de rádio frequências. Cada bit transmitido tem um padrão de bit redundante chamado chip. Quanto maior o chip, mais perto do dado original consegue-se recuperar. Bits mais longos requerem uma largura de banda maior. Para os receptores que não esperam o sinal, o DSSS aparece como ruído de pouca energia e é descartado. Requer mais energia que o FHSS, porém podem ter taxas de transferências de até 2 Mbps.

Orthogonal Frequency Division Multiplexing (OFDM) transmite os dados de forma paralela, em oposição às técnicas de FHSS e DSSS. Esta técnica protege contra interferências, tendo em vista que o sinal é transmitido sobre várias frequências paralelamente. OFDM tem alta eficiência em seu espectro, isto quer dizer que mais dados podem ser transmitidos por uma pequena largura de banda. Porém isto também é seu problema, é de difícil implementação quando comparados ao FHSS e DSSS, e consome grandes quantidades de energia.

3.2 INFRAVERMELHO

Alguns equipamentos são conectados via conexões ópticas, como é o caso do infravermelho. Este método utiliza um feixe de luz com tamanho de ondas que variam de 780 a 950 nanômetros (nm) para transferências tanto de voz como de dados. Uma das desvantagens do seu uso é que necessita um caminho livre de obstáculos e direto entre os dispositivos e com limitação em suas distâncias de até um metro. Porém, com equipamentos e diferentes métodos a distância e velocidades atingidas podem variar, sendo um exemplo que a tecnologia infravermelho também pode ser utilizada como redes de computadores (*infrared networking*) atingindo, dependendo de equipamentos e topologias utilizadas, velocidades de até 16 Mbps. (MULLER, 2000, tradução nossa; MULLER, 2002, tradução nossa).

Na Tabela 1 é demonstrada as especificações de performance e características do infravermelho.

Tabela 1 - Performance e características do infravermelho.

Característica/Funcionalidade	Performance
Tipo de conexão	Infravermelho, ângulo pequeno (30° ou menos)
Espectro	Óptico, 850 nm
Poder de transmissão	100 mW
Taxa de transferência	Até 16 Mbps
Distância	Até 1 metro
Dispositivos suportados	2
Canais de voz	1
Segurança dos dados	O alcance curto e pequeno ângulo do raio infravermelho oferecem por si só uma forma simples de segurança; porém não há outras formas de segurança em nível de conexão
Endereçamento	Cada dispositivo possui um identificador físico de 32 bits que é usado para estabelecer a conexão com outro dispositivo

Fonte: MULLER (2000, tradução nossa)

3.3 BLUETOOTH

Segundo Muller (2002, tradução nossa), o Bluetooth é uma tecnologia sem fio omnidirecional, que provém um alcance limitado em transmissão de voz e dados sobre a frequência de banda não licenciada de 2.4 GHz, possibilitando assim conexões com os mais variados dispositivos sejam móveis ou fixos, ao qual esta comunicação, não exclusivamente, seria feita através de cabos. Até oito dispositivos – um mestre e sete escravos – podem realizar uma comunicação entre si no que se denomina de *piconet*⁷. Na Tabela 2 é listado algumas características da tecnologia Bluetooth trabalhando com velocidade de 1 Mbps e na frequência de 2.4 GHz.

Tabela 2 - Performance e características do Bluetooth.

Característica/Funcionalidade	Performance
Tipo de conexão	Espectro espalhado (salto de frequência)
Espectro	Banda 2.4 GHz ISM (industrial, científico e médico)
Poder de transmissão	1 mW
Taxa de transferência	1 Mbps utilizando o salto de frequência
Distância	Até aproximadamente 10 metros
Dispositivos suportados	Até 8 dispositivos por <i>piconet</i>
Canais de voz	Ate 3 canais sincronizados
Segurança dos dados	Para autenticação uma chave de 128 bits; para encriptação o tamanho da chave é configurado entre 8 e 128 bits
Endereçamento	Cada dispositivo possui um <i>Media Access Control</i> (MAC) de 48 bits que é utilizado para estabelecer conexão com outro dispositivo

Fonte: MULLER (2002, tradução nossa)

⁷ Rede *ad-hoc* conectando um grupo de no máximo oito dispositivos (MALLICK, 2003, tradução nossa)

Uma das coisas que os usuários podem realizar utilizando do Bluetooth é transferir e/ou sincronizar dados apenas tendo os aparelhos dentro do alcance um do outro. Por exemplo, através de uma câmera transferir as imagens para um computador ou diretamente para uma impressora sem a necessidade de utilizar cabos, carregar arquivos, abrir aplicações, ou cliques em botões. A tecnologia é combinada com a troca de circuito (*circuit switching*) e troca de pacotes (*packet switching*), fazendo que seja possível transferir tanto voz como também dados, sendo que assim um exemplo seria que o usuário de um telefone celular com a tecnologia poderia utilizar um adaptador para atender as chamadas do telefone, mesmo que este esteja dentro de uma maleta ou bolsa (MULLER, 2002, tradução nossa).

3.3.1 Bluetooth e Android

Conforme Conder e Darcey (2010, tradução nossa) a API do Bluetooth apenas foi disponibilizada no Android 2.0 SDK. Claramente nem todos os dispositivos Android possuem o *hardware* necessário, porém esta tecnologia já se tornou uma característica popular e que os desenvolvedores da plataforma podem utilizar para a sua vantagem. Quando presente no dispositivo possibilita algumas funcionalidades às aplicações, a seguir são listadas algumas:

- a) procurar, descobrir e interagir com dispositivos Bluetooth;
- b) estabelecer conexões RFCOMM com a possibilidade de transferir dados entre os dispositivos utilizando canais de dados (*data streams*);
- c) controlar conexões ponto-a-ponto ou multiponto com aparelhos Bluetooth e gerenciar múltiplas conexões.

Os desenvolvedores Android poderão encontrar a API Bluetooth dentro do pacote `android.bluetooth`, sendo que a aplicação em que for utilizada deve explicitar ao usuário as permissões de: `android.permission.BLUETOOTH` para conectar a aparelhos Bluetooth; `android.permissions.BLUETOOTH_ADMIN` para possibilitar controle sobre o *hardware* da tecnologia e seus serviços no dispositivo, incluindo as tarefas de habilitar/desabilitar o Bluetooth e realizar varreduras por dispositivos ativos (CONDER; DARCEY; 2010, tradução nossa; MEIER, 2010, tradução nossa).

3.4 WIRELESS LAN

Para Muller (2000, tradução nossa), outra opção de conexão é a rede de área local (*Local Area Network*), que quando associada a um padrão sem fio como o 802.11, descrito pelo *Institute of Electrical and Electronic Engineers* (IEEE), se tornam uma *Wireless LAN* (Rede de área local sem fio, WLAN).

Segundo Morinaga, Kohno e Sampei (2002, tradução nossa), com o início das WLAN, novas tecnologias e aplicações foram surgindo expandindo assim o mercado. Há uma contínua expansão nas áreas da saúde, finanças, pequenos negócios e educação. Utilizando-se de alguns padrões como o IEEE 802.11 e HIPERLAN consegue-se um alto nível de transferências de dados, dando suporte a serviços como voz orientada por pacotes ou conexões, qualidade de serviços do inglês *Quality of Service* (QoS), transferências de arquivos, compartilhamentos de periféricos, acesso a Internet, entre outros. Um dos pontos positivos em relação a redes LAN via cabos é a mobilidade e o baixo custo.

Na Tabela 3 pode-se ver uma comparação entre algumas diferentes tecnologias de WLAN.

Tabela 3 - Comparação entre Tecnologias WLAN

Padrão	Frequência	Taxa de Transferência	Alcance	Pontos de interesse
802.11	2.4 GHz	1 – 2 Mbps	100 metros	A primeira especificação aprovada da família 802.11
802.11a	5 GHz	54 Mbps	50 metros	Utiliza modulação OFDM para alcançar altas taxas de transferência; Primeiros produtos foram disponibilizados em 2002
802.11b	2.4 GHz	11 Mbps	100 metros	Grande penetração no mercado entre os padrões WLAN, com produtos comerciais disponíveis desde 1999
802.11g	2.4 GHz	54 Mbps	100 metros	Aprovado pela IEEE-AS no outono de 2002; Possui retro-compatibilidade com 802.11b

HomeRF	2.4 GHz	10 Mbps	50 metros	HomeRF não atingiu sucesso comercial
HIPERLAN/1	5 GHz	Teoricamente 20 Mbps	-	HIPERLAN/1 não atingiu sucesso comercial
HIPERLAN/2	5 GHz	54 Mbps	150 metros	Projetado para integração com outras redes, incluindo redes com fio LANs, IEEE 1394 (<i>Firewire</i>), e redes móveis 3G.

Fonte: MALLICK (2003, tradução nossa)

Segundo Muller (2002, tradução nossa) existe vários padrões para redes de tecnologia sem fio, cada uma tendo melhor desempenho em certo ambiente comparada as outras, com suas características principais. Com o foco em um ambiente doméstico será descrito a seguir alguns padrões comumente utilizados para estas redes (HomeRF, Bluetooth, HIPERLAN/1, HIPERLAN/2, 802.11).

3.4.1 HomeRF

Segundo Mallick (2003, tradução nossa), o nome já sugestivo do padrão nos remete que ele é uma tecnologia sem fio visando as redes domésticas. Baseada no padrão 802.11 FHSS possui melhorias para atender as necessidades do consumidor médio.

Conforme Muller (2002, tradução nossa), o padrão HomeRF 2.0 foi baseado nos padrões IEEE 802.11b e *Digital Enhanced Cordless Telecommunication* (DECT), sendo este último mundialmente popular para telefones domésticos sem fio. Operando na banda não registrada de 2.4 GHz, foi projetado para suportar tanto transferência de voz como de dados. Oferece velocidades de transferências comparadas ao padrão 802.11b e suporta os mesmos tipos de aparelhos terminais tanto em configurações ponto-a-ponto como em configurações multiponto. Transmissão de até 10 Mbps com alcance de aproximadamente 15 metros da base, sua transmissão dá-se pelo método de espectro espalhado com salto de frequência (*spread-spectrum frequency hopping*). Ao encontrar alguma interferência, como a de fornos microondas ou outra rede WLAN, por exemplo, se adapta movendo-se para uma frequência diferente.

A maior vantagem da tecnologia HomeRF sobre a 802.11b é a sua capacidade de se adaptar a interferências de fontes externas (fornos microondas, telefones sem fio, entre outros). Sendo que a 802.11b simplesmente retransmite o pacote ou aguarda o término da interferência, funcionando bem com dados, porém com voz ocasiona chiados e cortes. Como “saltador de frequências” consegue coexistir com outros dispositivos que utilizam do mesmo método de transmissão. Outra vantagem é a reserva de uma parte da banda através de canais isócronos para serviços de voz, assim continua-se podendo utilizar das facilidades do identificador de chamadas, transferências de ligações, chamadas em espera, e com uma ótima qualidade de voz, não havendo problemas na comunicação enquanto o protocolo lida com as interferências (MULLER, 2002, tradução nossa).

Com a adoção pela indústria do padrão 802.11b, a tecnologia HomeRF acabou não atingindo grande número de produtos e usuários. As vantagens sobre a tecnologia 802.11b são a facilidade de uso, custo, e o suporte a telefonia, todas as áreas que começaram a ser utilizadas por produtos com 802.11b e posteriormente pelas especificações 802.11x. Por estas e outras razões o Consórcio HomeRF foi descontinuado no início de 2003, fazendo assim o padrão HomeRF um padrão morto (MALLICK, 2003, tradução nossa).

Na Tabela 4 pode-se ver as características do padrão HomeRF.

Tabela 4 - Características do padrão HomeRF

Característica/Funcionalidade	Performance
Tipo de conexão	Espectro espalhado de 50 saltos/segundo
Espectro	Banda 2.4 GHz ISM
Poder de transmissão	100 mW
Taxa de transferência	1.6 Mbps com HomeRF 1.0; 10 Mbps com HomeRF 2.0; 25 Mbps com HomeRF 3.0(previsão)
Distância	Até aproximadamente 50 metros
Dispositivos suportados	Até 127
Canais de voz	Até 4 dispositivos ativos
Segurança dos dados	Compressão de dados com a utilização do algoritmo LZRW3-A
Endereçamento	Identificador de rede com 48 bits, possibilitando operações concorrentes de múltiplas redes

Fonte: MULLER (2002, tradução nossa), adaptação nossa.

3.4.2 HIPERLAN/1 e HIPERLAN/2

Criado pela *European Telecommunications Standards Institute* (ETSI) em 1992 sob o nome de *High-Performance Radio Local Area Network* (HIPERLAN), foi proposta para atender a necessidade de altas velocidades em curto alcance na comunicação sem fio. A primeira versão é chamada de HIPERLAN/1, baseada no padrão *Ethernet* e sua transmissão de rádio feita por GSM e na faixa de banda de 5 GHz. Devido ao não sucesso comercial sua distância de operação e taxas de transferências não são definidas, porém de acordo com as especificações (em teoria) teria taxas de transferência de dados próximo aos 23,5Mbps (MALLICK, 2003, tradução nossa).

A geração seguinte, o HIPERLAN/2, surgiu pela ETSI *Broadband Radio Access Networks* (BRAN), e continuou com a utilização da faixa de banda de 5 GHz, porém com a utilização da tecnologia OFDM. Com isto possibilitou taxas de transferências de até 54 Mbps e alcance aproximado de até 150 metros. HIPERLAN/2 foi definido para vários ramos do mercado onde WLAN é utilizado, tais quais: redes empresariais, SOHO, e ponto de acesso 3G sem fio. Para estas finalidades incorporou alguns serviços, como o QoS em tempo real, um eficiente consumo de energia para dispositivos portáteis, forte segurança e interoperabilidade com *Ethernet*, IEEE 1394 (*Firewire*), e sistemas móveis 3G. Porém até 2002 ainda não houve adoção da tecnologia tanto pelo setor doméstico como pelo corporativo (MALLICK, 2003, tradução nossa).

3.4.3 Bluetooth

Segundo Muller (2002, tradução nossa) e Tanenbaum (2002, tradução nossa) esta tecnologia não foi desenvolvida para suportar as redes sem fios LANs, mas sim criadas para substituir os cabos periféricos dos computadores e dispositivos de mão. Operando a velocidades taxadas como lentas, na faixa de 30 a 400 kbps por uma distância de até 10 metros, suportam *piconets* entre *laptops*, PDAs, dispositivos móveis, entre outros. Seu principal diferencial com a tecnologia infravermelho é a não necessidade de que os dispositivos estejam direcionados uns aos outros como também o seu alcance superior. Estas redes formadas por Bluetooth também dão suporte a canais de voz.

Por contar com uma arquitetura mestre-escravo (*master-slave*) pode estender seu alcance (10 metros) para além, conectando outras *piconets* entre si. Como o HomeRF, utiliza-se do salto de frequência, e por possuir o salto mais rápido consegue identificar e solucionar rapidamente possíveis interferências ocasionadas por outros dispositivos ou outras redes. Em resumo tem-se que o Bluetooth não prove largura de banda e alcance requeridos para uma aplicação WLAN, porem é recomendado sua utilização para substituição de cabos entre periféricos e redes *ad-hoc*, tanto para transferências de dados quanto de voz (MULLER, 2002, tradução nossa; TANENBAUM, 2002, tradução nossa).

3.4.4 Família IEEE 802.11

A seguir será descrito sobre os tipos de rede contidos dentro da família 802.11.

3.4.4.1 IEEE 802.11

Esta especificação, aprovada em julho de 1997, foi o primeiro padrão de WLAN a ser definido. Utiliza os mesmos protocolos de troca que as redes *Ethernet* via cabos, utilizando a banda não licenciada de rádio na frequência de 2.4 GHz. Dois tipos de técnicas de modulação de frequência são utilizados, o FHSS e DSSS. Porém produtos com este padrão não são mais utilizados, sendo substituídos no mercado pelos seus sucessores 802.11a e 802.11b, estes trouxeram maior taxa de transferência a menor custo (MALLICK, 2003, tradução nossa). Na

Tabela 5 é possível observar algumas características do padrão 802.11.

Tabela 5 - Performance e características da rede local sem fio sob padrão 802.11 e utilizando salto de frequências

Característica/Funcionalidade	Performance
Tipo de conexão	Espectro espalhado (Sequência direta ou salto de frequência)
Espectro	Banda 2.4 GHz ISM
Poder de transmissão	100 mW
Taxa de transferência	1 Mbps ou 2 Mbps utilizando o salto de frequência; 11 Mbps utilizando a sequência direta
Distância	Até aproximadamente 100 metros entre o ponto de acesso e o

	cliente
Dispositivos suportados	Múltiplos dispositivos por ponto de acesso; Múltiplos pontos de acesso por rede
Canais de voz	Voz sobre IP
Segurança dos dados	Autenticação: senha entre o ponto de acesso e o cliente via <i>Wired Equivalent Privacy</i> (WEP) Encriptação: padrão de 40 bits; opcional de 128 bits
Endereçamento	Cada dispositivo possui um <i>Media Access Control</i> (MAC) de 48 bits que e utilizado para estabelecer conexão com outro dispositivo

Fonte: MULLER (2002, tradução nossa)

3.4.4.2 IEEE 802.11a

Segundo Mallick (2003, tradução nossa) e Muller (2002, tradução nossa), esta especificação foi a sucessora do 802.11 e também uma alternativa ao 802.11b, transmitindo a 5 GHz e a taxas de transmissão de até 54 Mbps. Diferentemente das 802.11 e 802.11b, utiliza modulação OFDM. Devido à diferença de frequência, é incompatível com dispositivos e redes que utilizam 802.11b, que trabalham a 2.4 GHz.

A seguir Mallick (2003, tradução nossa) demonstra alguns pontos positivos e negativos sobre a especificação 802.11a, sendo alguns pontos também são suportados por Muller (2002, tradução nossa).

Por utilizar a frequência de 5 GHz e modulação OFDM, isto traz alguns benefícios sobre a 802.11b. Primeiro referente a máxima taxa de transmissão por canal, de 11 Mbps para até 54 Mbps. Isto é um aumento significativo, devido que estes canais são divididos entre os usuários do mesmo ponto de acesso. Convém muito este aumento para utilizações como dispositivos de multimídia sem fios, transferências grandes de arquivos, e acesso rápido a Internet. Após pode-se citar que por utilizar esta faixa de frequência acaba por diminuir potenciais conflitos e interferências, isto devido o fato desta faixa de 5 GHz ser menos utilizada.

Porém também existem desvantagens. A utilização de alta frequência acaba por limitar a distância de uso, enquanto a 802.11b pode chegar a ate 100 metros, a 802.11a acaba

sendo limitada entre 25 a 50 metros apenas. Também há que a modulação OFDM requer mais energia que a modulação DSSS, levando assim a um maior consumo de energia nos produtos que utilizam 802.11a.

3.4.4.3 IEEE 802.11b

Ele é o padrão mais popular da família 802.11. Suas especificações foram aprovadas junto com a 802.11a em 1999, sendo que desde então alcançou uma grande aceitação no mercado de redes sem fios. Foi baseada na versão DSSS da 802.11 e utiliza a frequência de 2.4 GHz. Pela implementação de DSSS ser mais fácil que OFDM, seus produtos alcançaram o mercado cedo, sendo que também a frequência de 5 GHz ser restrita em alguns países, acabaram por ajudar a alavancar a sua popularidade pelo mundo (MALLICK, 2003, tradução nossa; MULLER, 2002, tradução nossa).

Algumas vantagens e desvantagens são descritas por Mallick (2003, tradução nossa). Uma vantagem é a utilização da faixa de 2.4 GHz, praticamente disponível universalmente, sendo que o sinal em 2.4 GHz também possui melhor penetração em barreiras físicas como paredes comparadas as altas frequências. Porém também é o seu ponto negativo, pois esta faixa não é registrada, isto significa que qualquer um pode utilizá-la sem possuir uma licença, assim pode acabar gerando muitas interferências e a frequência acabar ficando saturada, sendo alguns exemplos de dispositivos que utilizam esta mesma faixa de frequência: a maioria dos telefones sem fio e fornos microondas.

Segundo Muller (2002, tradução nossa) e Mallick (2003, tradução nossa), o padrão 802.11b pode se comunicar com dispositivos em distâncias de até 100 metros. Porém as taxas de transferências vão diminuindo conforme o dispositivo se afasta do ponto de acesso, sendo bem limitada na ponta mais distante e melhor aproveitada quando bem próximo.

3.4.4.4 Outros padrões da família IEEE 802.11

Segundo Mallick (2003, tradução nossa), o padrão **IEEE 802.11g** traz para a frequência 2.4 GHz altas taxas de transferências. Isto porque utiliza a modulação OFDM, assim consegue taxas de transferências de até 54 Mbps, sendo que as demais especificações

são similares as do padrão 802.11b. Mantendo também a compatibilidade com dispositivos 802.11b, limitado as suas especificações, acabou por ser bem aceito no mercado.

A seguir serão demonstrados alguns outros padrões da família 802.11 descritos por Mallick (2003, tradução nossa), sendo que as áreas de concentrações foram segurança, qualidade do serviço e interoperabilidade.

- a) **IEEE 802.11e**: visando a prover melhor capacidade de qualidade de serviço para possibilitar uma comunicação de voz confiável e complementar os sistemas 802.11b.
- b) **IEEE 802.11f**: seu objetivo é desenvolver as práticas recomendadas para um *Inter-Access Point Protocol* (IAPP) para alcançar a interoperabilidade de vários fornecedores de pontos de acesso.
- c) **IEEE 802.11h**: o objetivo é aumentar a segurança e mecanismos de autenticação na camada MAC do 802.11.

3.4.5 Wi-Fi e Android

Segundo Conder e Darcey (2010, tradução nossa), o sensor de Wi-Fi pode ler o status de uma rede e determinar o ponto de acesso mais próximo. A SDK do Android prove um conjunto de APIs para recuperar informações sobre as redes Wi-Fi disponíveis ao aparelho e detalhes das conexões destas redes. Estas são separadas das APIs do `SensorManager`. Com estas informações é possível recuperar uma série de outras informações e utilizá-las, por exemplo, para determinar a força do sinal, achar pontos de acesso de interesse, realizar ações quando conectado a um determinado ponto de acesso.

Conforme Conder e Darcey (2010, tradução nossa) e Meier (2010, tradução nossa), o `WifiManager` representa o Serviço de Conectividade Wi-Fi do Android. Ele pode ser utilizado para configurar conexões de redes Wi-Fi, gerenciar a conexão atual, procurar por pontos de acessos e monitorar mudanças na conexão. Sobre as permissões da aplicação descritas no arquivo `AndroidManifest.xml`, a permissão `CHANGE_WIFI_STATE` é necessário quando a aplicação está acessando informações sobre as redes sem fio, como também para modificar seu estado, já a permissão `ACCESS_WIFI_STATE` é necessário para requisitar qualquer informação do dispositivo Wi-Fi.

No contexto atual a conectividade *wireless* está inserida em vários dispositivos do cotidiano, neste cenário tem-se os dispositivos móveis, aos quais podem acompanhar o

usuário por diversas localidades e proporcionar uma conectividade entre equipamentos e tecnologias. Nestes dispositivos podem rodar diversos sistemas operacionais para o controle, gerenciamento e processamento, sendo um destes o sistema operacional Android, ao qual esta entrando e ampliando no mercado.

4 SISTEMA OPERACIONAL ANDROID

Alguns anos atrás os celulares não contavam com toda a gama de recursos que possuem atualmente, tais quais GPS, câmera, tecnologia *touchscreen*, tocador de músicas e vídeos, entre outros. Além das tecnologias empregadas no *hardware*, existe o *software*, onde os fabricantes mantinham fechado o sistema que rodava sobre seus celulares, dificultando a evolução e melhoria de recursos (DIMARZIO; POLO, 2008, tradução nossa, MEIER, 2010, tradução nossa).

Para mudar este cenário, e trazer melhorias tanto para os desenvolvedores quanto aos usuários finais, foi criada a *Open Handset Alliance* (OHA), liderada pela gigante e revolucionária Google e diversos integrantes que marcam presença em fatores tecnológicos tais quais a HTC, Motorola, Sony Ericsson, Intel, entre outras. A OHA tem seu objetivo focado na definição de uma plataforma aberta e única para dispositivos móveis, visando os usuários e os desenvolvedores, sendo que para satisfazer seus objetivos foi criado o Android. (LECHETA, 2010).

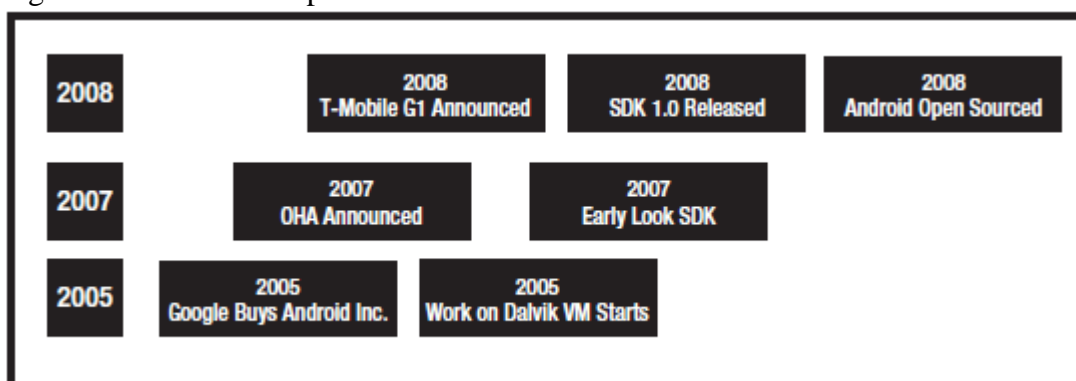
4.1 HISTÓRICO

Segundo Hashimi et al (2010, tradução nossa) quando o Android chegou ao mundo de sistemas móveis, os dispositivos utilizavam uma variedade de sistemas operacionais proprietários (Symbian OS, Microsoft Windows Mobile, iPhone OS, entre outros), sendo que nenhum se tornou um padrão para os dispositivos. Neste cenário onde o desenvolvimento com as APIs disponíveis eram muito restritivas e fracas quando comparados ao ambiente *Desktop* de programação, entra a Google com um sistema operacional para plataformas móveis que promete conforto, código-fonte aberto e um *framework*⁸ de desenvolvimento de ponta.

Android Inc. contou com nomes importantes e chaves como Andy Rubin, Rich Miner, Nick Sears e Chris Whit, todos estes fundadores desta companhia, adquirida pela Google em 2005 que iniciou o desenvolvimento desta plataforma.

⁸ Coleção de bibliotecas de softwares que fornece uma *Application Programming Interface* (API) definida.

Figura 3 - Linha do tempo da história do Android



Fonte: HASHIMI et al (2010, tradução nossa)

Na Figura 3 pode-se ver uma linha do tempo da história da plataforma, onde mostra os principais acontecimentos, desde a compra da Android Inc. pela Google até o pronunciamento da abertura do código fonte.

Em meados de 2007, algumas empresas tecnológicas mundialmente conhecidas (Sprint Nextel, Motorola, Samsung, Sony Ericsson, Intel, Google entre *outras*⁹) se uniram ao redor da plataforma e fundaram a *Open Handset Alliance* (OHA), tendo como principais objetivos uma inovação rápida e uma melhor abrangência das necessidades dos consumidores, sendo como chave para alcançar os objetivos o lançamento da plataforma Android, esta lançada sobre o código aberto na Licença Apache versão 2.0, com isto os fabricantes de dispositivos não precisam pagar taxas de licença para colocar em seus aparelhos o sistema operacional Android.

4.2 PRINCIPAIS CARACTERÍSTICAS

Segundo Steele e To (2010, tradução nossa) a plataforma possui algumas características maiores ao qual o torna um diferencial contra os demais. A seguir será descrito algumas destas importantes características do Android.

4.2.1 Multiprocesso e App Widget

Conforme explicado por Steele e To (2010, tradução nossa) o sistema operacional Android não limita o processador a um único processo por vez, sendo que faz o

⁹ Lista completa disponível em http://www.openhandsetalliance.com/oha_members.html

gerenciamento de prioridades e threads de uma aplicação. Com isto há o benefício da utilização de multitarefa, ou multiprocesso como também é conhecido, abrindo ao usuário e desenvolvedores novas possibilidades, por exemplo, durante a execução de um jogo pelo usuário, em *background* uma aplicação que verifica, em tempo real, o nível de estoque e dispara um alerta caso necessário. Esta característica traz o benefício de uma experiência rica aos usuários.

Os *AppWidget* podem ser descritos como mini aplicações, aos quais podem ser “anexadas” em outras para executar determinadas tarefas, serviços. Um exemplo é o tocador de música que fica na *Home Screen* do usuário. Sendo que a *Home Screen* é uma aplicação rodando, em sua interface há este *Widget* do tocador de música, ao qual concede ao usuário algumas interações, como executar, pausar, trocar de música sem a necessidade de abrir a aplicação completa do tocador musical. Um outro exemplo é um *Widget* que atualiza e mostra ao usuário a temperatura ambiente, sendo atualizado seus valores por um processo em *background*.

4.2.2 Toque, Gestos e Multitoque

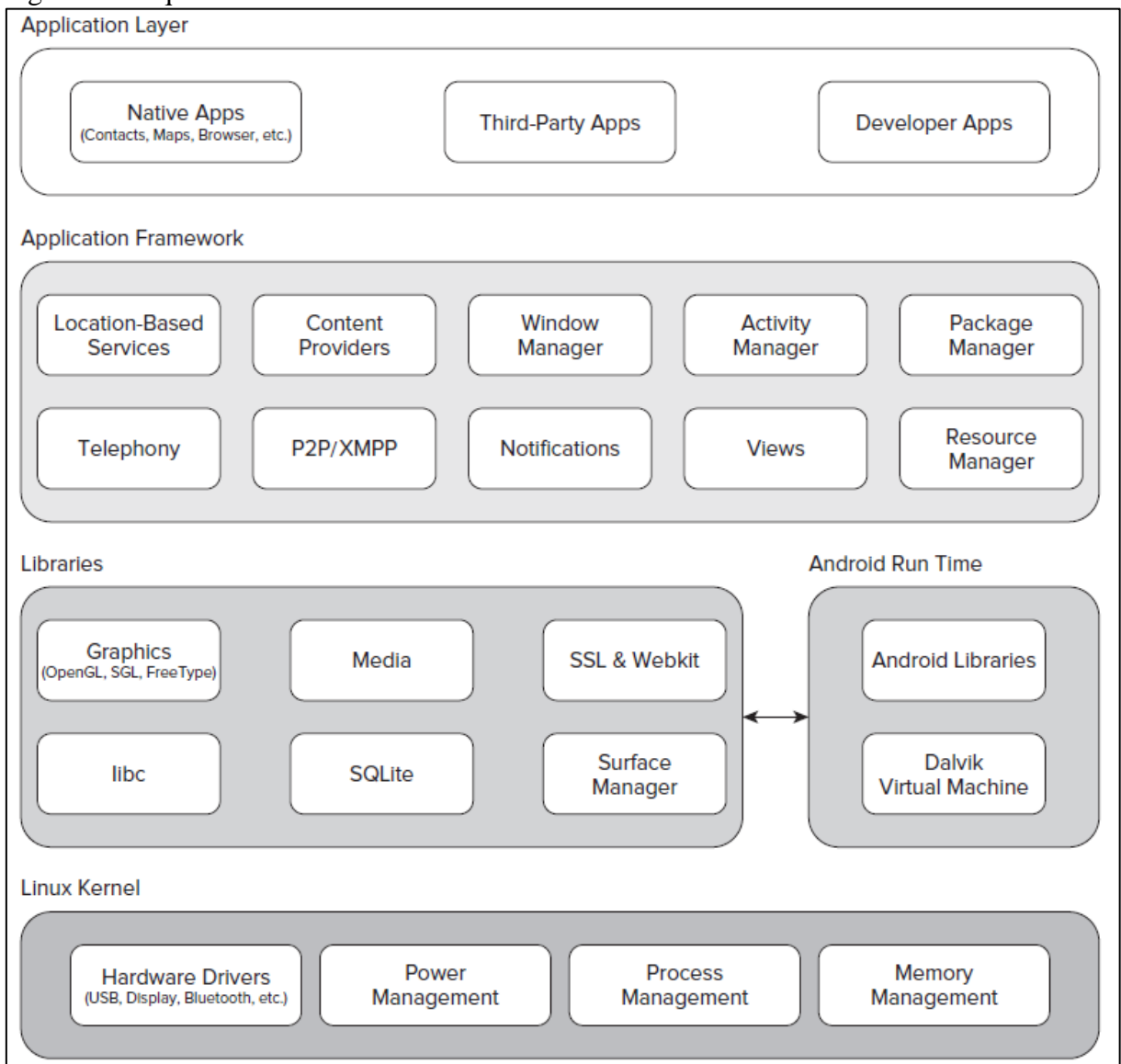
As telas sensíveis ao toque (*touchscreen*) estão presentes nos celulares mais modernos, tanto como se fossem um padrão. O *touchscreen* é uma forma intuitiva de interação com um aparelho. Sua utilização é feita de forma quase que natural, sendo que dar-se ao pressionar um dedo sobre a tela. Um exemplo é a forma prática de pressionar e arrastar para interagir com um gráfico. A função de multitoque (*multitouch*) dá ao usuário a possibilidade de utilizar mais de um dedo simultaneamente, sendo utilizado normalmente para *zoom* ou realizar a rotação de algo na tela. Ainda existem os gestos, aos quais se podem definir certos padrões de toques para executar determinadas funções (STEELE; TO, 2010, tradução nossa).

O Android vem por padrão com *multitouch* em sua interface, Karch (2010, tradução nossa) sugere que cabe ao desenvolvedor utilizar ou não as vantagens destes recursos em suas aplicações. Steele e To (2010, tradução nossa) salientam que muitos dos eventos gerados pelo *touch* são disponíveis ao desenvolvedor de forma transparente e sem a necessidade de implementações complicadas e confusas.

4.3 ARQUITETURA

Segundo Meier (2010, tradução nossa) a arquitetura do Android é composta por alguns componentes, melhores demonstrados pela Figura 4 e detalhadas posteriormente segundo Ableson, Sen e King (2011, tradução nossa), Burnette (2009, tradução nossa) e Meier (2010, tradução nossa). Mas basicamente é composto por um *Kernel* em Linux e coleções de bibliotecas em C/C++ que são expostas ao *framework* para prover os serviços e gerenciamento das aplicações.

Figura 4 - Arquitetura do Sistema Android



Fonte: MEIER (2010, tradução nossa)

4.3.1 *Kernel Linux*

O Android foi criado sobre uma base sólida, o *Kernel* do Linux versão 2.6. Este provê ao sistema a camada de abstração de *hardware*, possibilitando assim ser portado para diferentes plataformas, usa internamente, o Linux como um gerenciador de memória, gerenciador de processos, e outros serviços do sistema operacional.

Isto de forma abstrata ao usuário, sendo assim o usuário não se deparará com chamadas a programas e comandos em Linux, da mesma forma como é abstrata aos desenvolvedores, porém a estes é pertinente a informação de que existe o *Kernel* em Linux e como é seu funcionamento, pois durante o processo de desenvolvimento pode-se utilizar de alguns utilitários para interagir diretamente com o Linux, como o caso do *adb shell command*¹⁰ na qual abre uma *shell* do Linux e pode executar comandos diretamente no aparelho, sendo sua utilização para examinar arquivos do sistema, visualizar processos ativos, como também aplicar restrições de segurança.

4.3.2 Bibliotecas

A camada acima do *Kernel* contém as bibliotecas nativas do Android. São bibliotecas compartilhadas escritas em C ou C++, compiladas para uma arquitetura de *hardware* em específico usada em um aparelho e pré-instalada pelo fabricante.

Algumas das mais importantes bibliotecas são:

- a) gerenciador de superfície: utiliza um compositor e gerenciador de janelas. Ao invés de desenhar diretamente no *buffer* da tela, os comandos de desenho vão para outro *buffer* de imagem, que combinados com outras imagens formam o que o usuário visualiza. Este sistema possibilita a criação de efeitos, como as transições suaves entre as telas;
- b) gráficos 2D e 3D: elementos bidimensionais e tridimensionais são combinados em uma interface única e simples. A biblioteca utilizará de aceleração 3D via *hardware* se este tiver suporte, ou um renderizador por *software* quando não presente. Faz uso da *Scalable Games Language* (SGL) e OpenGL ES;
- c) codecs de mídias: é capaz de reproduzir e gravar áudio e vídeo em diferentes formatos, incluindo AAC, AVC (H.264), H.263, MP3 e MPEG-4;

¹⁰ Maiores informações disponíveis em: <http://d.android.com/guide/developing/tools/adb.html>

- d) base de dados SQL: vêm nativo com uma versão do SQLite, mesma base de dados utilizada no navegador Firefox e no iPhone da Apple. Com ele é possível utilizar de armazenamento persistente nas aplicações;
- e) *browser engine*: para um bom desempenho ao mostrar conteúdos em HTML, ele vêm com o WebKit, com licença open source, ele também está presente no navegador da Google, o Chrome, navegador da Apple, o Safari, no iPhone da Apple e na plataforma S60 da Nokia, sendo considerado de fato um padrão para a maioria das plataformas móveis. Também conta com o *Secure Sockets Layer* (SSL) do projeto Apache;
- f) sendo estas as bibliotecas nativas mais importantes, a plataforma não é limitada somente a estas bibliotecas, sendo possível a partir da versão 1.5 o desenvolvedor criar e implantar suas próprias através do *Native Development Toolkit* (NDK).

4.3.3 Android Runtime

Também sobre o *Kernel* do Linux, está o *Android Runtime* (Tempo de execução do Android), que é o que o diferencia de um dispositivo móvel com implementação Linux. Incluindo as bibliotecas do núcleo Java e a máquina virtual Dalvik, ele é onde estão rodando as aplicações, e com as bibliotecas, formam a base do *framework* de aplicação.

As bibliotecas do núcleo Java estão próximas de um ambiente completo de desenvolvimento Java, porém são diferentes das versões *Standart Edition* (SE) e *Micro Edition* (ME) da plataforma.

A máquina virtual Dalvik é uma máquina virtual baseada em registradores que foi otimizada para possibilitar que um aparelho possa rodar múltiplas instâncias de forma eficiente. Outra característica é que também faz o gerenciamento de memória de baixo nível.

4.3.4 Framework de Aplicação

O *framework* de aplicação prove as classes necessárias para a criação de aplicações para o Android. Também faz a abstração do acesso ao *hardware* e gerencia a interface do usuário e recursos das aplicações. As partes mais importantes são:

- a) *Activity Manager*: controlador do ciclo de vida das aplicações, é responsável por manter uma *backstack* (“pilha”) da navegação do usuário;
- b) *Content Providers*: Faz um encapsulamento de dados que precisam ser compartilhados entre aplicações, como por exemplo, os contatos da agenda;
- c) *Resource Manager*: os recursos são qualquer coisa que vão com sua aplicação que não são códigos;
- d) *Location Manager*: o dispositivo Android sempre sabe onde ele está;
- e) *Notificating Manager*: eventos como uma nova mensagem, alertas e muito mais podem ser mostrados ao usuário de forma discreta;

4.3.5 Camada de Aplicação

A última camada, e a mais alta no diagrama de arquitetura, é a Camada de Aplicação, esta se pode comparar a uma ponta de um iceberg, ao qual está fora da água é o que o usuário final irá visualizar e interagir, sendo que a superfície inferior do iceberg e embaixo d’água não é visualmente mostrada ao usuário, porém os desenvolvedores devem conhecer o que o sistema está executando e de que forma o faz.

Todas as aplicações, nativas ou de terceiros, são feitas na camada de aplicação, ou seja, utilizam as mesmas bibliotecas de API. Isto significa que permite o usuário selecionar uma aplicação que pode substituir uma tela nativa, como por exemplo, a listagem dos contatos poderia ser substituída por alguma tela de terceiros.

A camada de aplicação roda em conjunto com o Android *Runtime*, usando as classes e serviços disponibilizados a partir do *framework* da aplicação.

4.4 ANATOMIA DE APLICAÇÕES

Segundo Steele e To (2010, tradução nossa) uma aplicação Android consiste em várias funcionalidades tais como editar uma anotação, tocar uma música, disparar um alarme,

ou abrir um contato do telefone. Estas podendo ser classificadas em quatro diferentes componentes Android. Na Tabela 6 é demonstrado cada uma destes componentes e sua classe base Java.

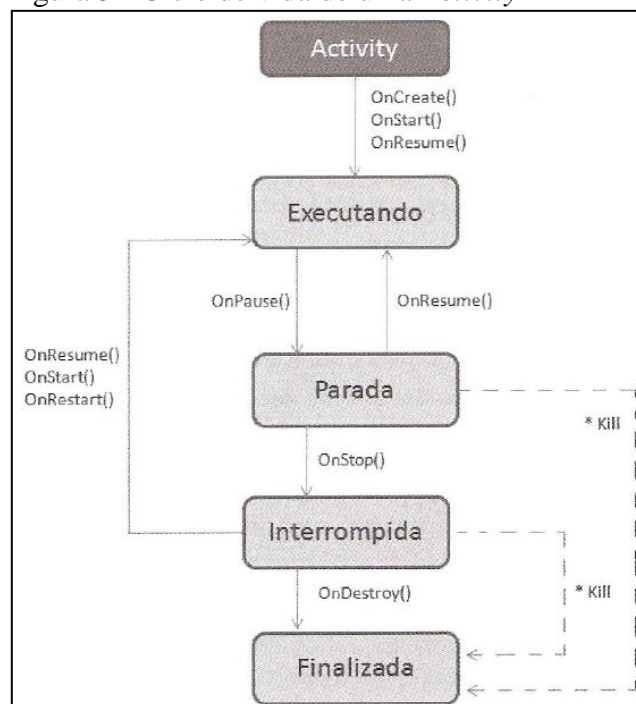
Tabela 6 - Os quatro componentes possíveis de uma Aplicação Android

Funcionalidade	Classe Base Java	Exemplo
Algo focado que o usuário pode fazer	Activity	Editar uma anotação, jogar um jogo
Processo <i>background</i> ¹¹	Service	Tocar uma música, atualizar ícone de temperatura
Recebe mensagens	BroadcastReceiver	Aciona alarme sobre um evento
Armazena e recupera dados	ContentProvider	Abrir um contato da agenda

Fonte: STEELE; TO, 2010, tradução nossa

Segundo Hashimi et al (2010, tradução nossa), uma atividade (*Activity*) é um conceito de interface de usuário, que normalmente representa uma única tela na aplicação. Ela possui um ciclo de vida bem definido, como pode ser observado na Figura 5 a seguir.

Figura 5 - Ciclo de vida de uma *Activity*



Fonte: PEREIRA, L.; SILVA, M. (2009)

¹¹ Que fica em segundo plano, sem necessariamente possuir uma interface.

Este ciclo de vida não é exclusivamente controlado pelo desenvolvedor, e em certos casos nem pelo usuário, pois o sistema operacional pode acabar eliminando uma atividade que não esteja no topo da pilha de processamento quando for necessário liberar recursos para o dispositivo (ABLESON; SEN; KING, 2011, tradução nossa).

A Tabela 7 demonstra maiores informações sobre as fases do ciclo de vida da atividade e destaca os principais métodos de uma atividade.

Tabela 7 - Principais métodos do ciclo de vida de uma *Activity* Android

Método	Significado
<code>onCreate()</code>	Chamado quando a <i>Activity</i> é criada.
<code>onRestart()</code>	Chamada se a <i>Activity</i> está sendo reiniciada.
<code>onStart()</code>	Chamada quando a <i>Activity</i> está sendo visível na tela para o usuário.
<code>onResume()</code>	Chamada quando a <i>Activity</i> começa a interagir com o usuário. Este método é sempre chamado independente se está iniciando ou reiniciando.
<code>onPause()</code>	Chamada quando a <i>Activity</i> está sendo pausada ou solicitando processamento e outros recursos do dispositivo. O desenvolvedor pode utilizar este método para salvar informações para posteriormente quando a <i>Activity</i> voltar a ser ativada recuperar e continuar do mesmo ponto interrompido.
<code>onStop()</code>	Chamada para parar a <i>Activity</i> e transita para uma fase não visível e de eventos subsequentes no ciclo.
<code>onDestroy()</code>	Chamada quando uma <i>Activity</i> é totalmente removida da memória do sistema. Podendo ser chamada pelo método <code>onFinish()</code> ou diretamente invocada pelo sistema para liberar recursos.

Fonte: Adaptado de ABLESON; SEN; KING (2011, tradução nossa)

Os serviços (*Services*) do Android relembram os serviços vistos no Windows e em demais plataformas, são processos no *background* que ficam rodando por um período de

tempo e não requerem interação do usuário. Podem ser iniciados e parados por qualquer componente, sendo que quando está rodando, qualquer componente pode se conectar a ele, e um serviço pode parar a si próprio. Porém se ele está ativo, ele está consumindo recursos e isto afeta diretamente no estado do dispositivo como no desempenho e duração da bateria. Exemplo de serviços é um tocador de música continuar tocando a música mesmo quando sua *Activity* for destruída, realizar atualizações com a Internet de notícias (HASHIMI et al, 2010, tradução nossa; MURPHY, 2009, tradução nossa; STEELE; TO, 2010, tradução nossa).

Segundo Steele e To (2010, tradução nossa) e Ableson, Sen e King (2011, tradução nossa), um `BroadcastReceiver` escuta por mensagens em *broadcast*¹² relevantes para disparar algum evento. O próprio sistema Android pode disparar mensagens por *broadcast* para informar as demais aplicações de algum evento que ocorreu, como exemplo podemos citar o botão da câmera foi pressionado, o nível da bateria está baixo, uma nova aplicação foi instalada, um SMS foi recebido, entre outras. Aplicações também podem enviar mensagens por *broadcast* para informar algum evento que possa ser relevante a demais aplicações. Seu ciclo de vida é simples, um único método `onReceive()` é chamado quando a mensagem chega ao `BroadcastReceiver`, após isto ele volta ao estado inativo e fica aguardando até a próxima mensagem relevante.

Segundo Burnette (2009, tradução nossa), Murphy (2010, tradução nossa) e Lecheta (2010), no Android é possível armazenar informações de diferentes formas, como utilizando de banco de dados, sistema de preferências ou até mesmo arquivos. Porém estas informações são restritas ao acesso somente das aplicações que as criaram, ficando proibido o acesso das demais aplicações a estes dados. Para resolver este problema de acesso, o Android possui a classe `android.content.ContentProvider`, que em seu conceito defini-se como um provedor de conteúdo. Seu funcionamento pode-se dizer que é contrária dos `BroadcastReceiver`, que captam avisos de aplicações e sistema, ele acaba por compartilhar dados com outras aplicações, transformando assim públicas estas informações que antes eram privadas apenas da aplicação criadora. O Sistema Android também possui `ContentProvider` nativos, como um exemplo pode-se citar o provedor de conteúdos que permite ler os contatos cadastrados na agenda do celular.

¹² Neste contexto significa o envio de uma mensagem a todas as aplicações interessadas (CONDER; DARCEY; 2010, tradução nossa).

4.5 VERSÕES OFICIAIS

Conforme demonstrado em *The Complete Guide To Google Android* (2011, tradução nossa) pode-se verificar as principais mudanças conforme a evolução do Sistema Android.

a) Android 1.5 (Cupcake – Abril de 2009):

- captura e reprodução de vídeos,
- upload direto para o Youtube e Picasa,
- suporte ao Bluetooth A2DP,
- *Widgets* e pastas na Área de Trabalho,
- opção de copiar/colar estendido a páginas web,
- *software* do teclado atualizado;

b) Android 1.6 (Donut – Setembro de 2009):

- galeria aceita excluir múltiplas fotos,
- aperfeiçoamento do *Android Market*,
- pesquisa por voz mais rápida e integrada,
- galeria combinada de fotos e vídeos,
- através de *Widgets* realizar pesquisas pela *Home Screen*,
- *Text-to-Speech*: serviço de leitura de textos;

c) Android 2.0/2.1 (Eclair– Outubro de 2009):

- interface melhorada,
- *wallpapers* animados,
- nova interface dos navegadores com HTML5,
- Microsoft Exchange,
- nova lista de contatos,
- Bluetooth 2.1,
- aperfeiçoamento do *software* do teclado,
- flash e zoom digital para a câmera;

d) Android 2.2 (Froyo – Maio de 2010):

- novos *Widgets* para a *Home Screen*,
- mensagens unificadas,

- assistência de GPS,
 - suporte para cartões microSD,
 - USB *Tethering*¹³,
 - suporte ao Flash 10.1;
- e) Android 2.3 (Gingerbread – Dezembro de 2010):
- NFC (*Near Field Communications*),
 - fechamento automático de aplicativos caso sobrecarregamento do aparelho,
 - melhor gerenciamento da bateria.

4.6 LICENÇA DE USO

Segundo Ableson, Sen e King (2011, tradução nossa) o Android é distribuído em duas diferentes licenças de código aberto. O *Kernel* do Linux é lançado sobre a licença GNU *General Public License* (GPL) como é de necessidade para qualquer um que está licenciando um *Kernel* de um sistema operacional de código aberto. A plataforma Android, excluindo o *Kernel*, é licenciado sobre a *Apache Software License* (ASL). Ambos os tipos de licenças são orientadas a códigos abertos, porém a maior diferença é que a licença Apache é considerada mais amigável as causas comerciais.

Eddy, Eddy (2009, tradução nossa) exemplificam de forma mais direta o que seria a licença Apache, sendo que a licença Apache v2 permite que fabricantes e os distribuidores de serviço de telefonia móvel a fazerem mudanças no *software* sem a necessidade de liberar estas mudanças aos demais.

O Android vem trazendo modernidade e padronizando o desenvolvimento aos dispositivos móveis. Com o auxílio de sensores consegue-se obter alguns tipos de dados do mundo real e realizar processamento sobre eles, um destes sensores que merece destaque é o acelerômetro, importante para sensores de orientação e medição da aceleração em um determinado eixo.

¹³ Ato de usar o aparelho móvel como um modem portátil e compartilhar esta conexão com outro dispositivo, possibilitando também o compartilhamento de Internet (KARCH, 2010, tradução nossa).

5 ACELERÔMETRO

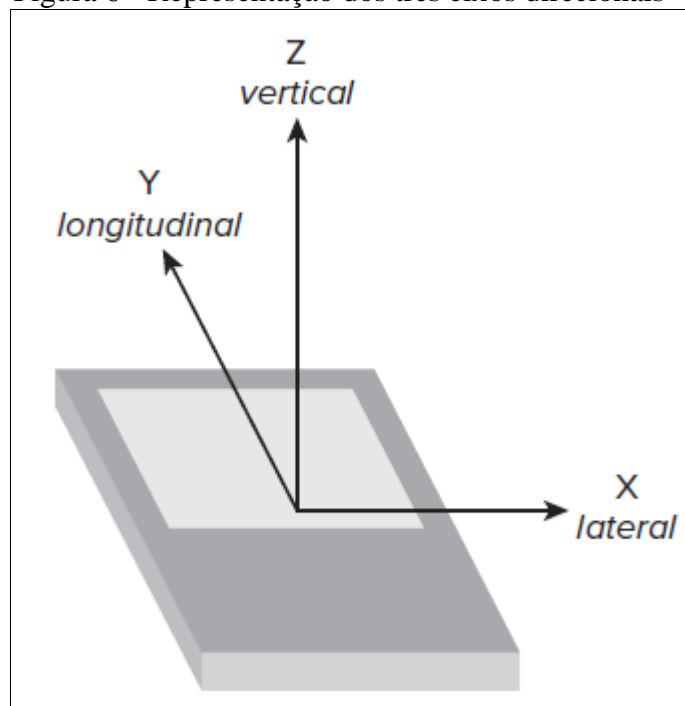
Segundo Felker (2010, tradução nossa), os acelerômetros, como seu nome sugere, é algo para se medir a aceleração. Conforme Meier (2010, tradução nossa), eles também podem ser referenciados de sensores de gravidade, devido ao fato de não possuir a habilidade de diferenciar entre a aceleração causada por um movimento e a gravidade. Assim sendo, quando o sensor esta em repouso, a leitura para o eixo-z (cima/baixo) será de -9.8m/s^2 .

A aceleração é definida como um valor de mudança na velocidade, portanto acelerômetros medem quão rápidos a velocidade do dispositivo mudou em uma determinada direção. Utilizando um acelerômetro podem-se detectar movimentos, e mais utilmente, o valor de mudança na velocidade deste movimento (MEIER, 2010, tradução nossa).

Um fato importante a ser destacado é que o acelerômetro não mede velocidade, portanto não é possível diretamente e baseado em apenas uma leitura determinar a velocidade. Para tal é necessário medir as mudanças na aceleração durante o tempo.

A aceleração pode ser mensurada sobre três eixos direcionais, sendo eles: esquerda-direita (horizontal), frente-atrás (longitudinal) e cima-baixo (vertical). Pode-se ver em uma melhor visualização com a Figura 6 que demonstra uma representação dos três eixos direcionais (CONDER; DARCEY; 2010, tradução nossa; MORIMOTO, 2009).

Figura 6 - Representação dos três eixos direcionais



Fonte: MEIER (2010, tradução nossa)

5.1 ACELERÔMETRO E ANDROID

Segundo Meier (2010, tradução nossa), após iniciarem com câmeras e tocadores de áudio, dispositivos móveis agora contam com sistema GPS, acelerômetros e telas sensíveis ao toque. Enquanto estas novas tecnologias abrem portas para novas possibilidades no desenvolvimento de *softwares*, as aplicações móveis disponíveis acabam por ficar para trás das capacidades dos *hardwares*.

O Android SDK contém APIs para diversos tipos de *hardwares*, entre eles: baseados em localização (como um GPS), a câmera, áudio, conexões de rede, Bluetooth, e dentre vários outros se encontra o Acelerômetro. Alguns destes *hardwares* fazem parte do conjunto de sensores disponíveis, sendo estes listados na Tabela 8, com a sua classe de acesso e descrição (MEIER, 2010, tradução nossa).

Estes sensores são disponíveis através da classe `android.hardware.SensorManager`, ao qual funciona como o gerenciador de sensores. Há algumas classes relacionadas aos sensores, como `Sensor` representa um sensor em particular, `SensorEvent` representa as leituras captadas pelo sensor e o `SensorEventListener` que é a interface utilizada para receber os eventos gerados quase que em tempo real (ABLESON; SEN; KING, 2011, tradução nossa).

Tabela 8 - Sensores comuns no Android

Classe	Descrição
<code>Sensor.TYPE_ACCELEROMETER</code>	Medidas de aceleração sobre três eixos em m/s^2
<code>Sensor.TYPE_GYROSCOPE</code>	Giroscópio que retorna a orientação do dispositivo sobre três eixos em graus
<code>Sensor.TYPE_LIGHT</code>	Sensor de luz ambiente na unidade lux
<code>Sensor.TYPE_MAGNETIC_FIELD</code>	Medidas do campo magnético em microtesla sobre os três eixos
<code>Sensor.TYPE_ORIENTATION</code>	Mede orientação do dispositivo sobre três eixos em graus
<code>Sensor.TYPE_PRESSURE</code>	Medidas de pressão que retorna um único valor em kilopascal

`Sensor.TYPE_PROXIMITY` Mede a distância em que o dispositivo está longe de outro objeto em metros

`Sensor.TYPE_TEMPERATURE` Mede a temperatura ambiente em graus Celsius

Fonte: Adaptado de ABLESON; SEN; KING (2011, tradução nossa) e MEIER (2010, tradução nossa)

Segundo Ableson, Sen e King (2011, tradução nossa) e Meier (2010, tradução nossa), é possível coletar os valores dos sensores quando implementado o `SensorEventListener` e registrando-o para o `SensorManager`. Será chamado o método `onSensorChanged()` quando os valores lidos forem alterados, e chamado o método `onAccuracyChanged()` quando a precisão for alterada.

O `SensorEvent` disponibiliza alguns campos, como o `accuracy`, ao qual retorna quão preciso estão sendo as leituras. O `Sensor` que é a referencia ao sensor criado pelo `SensorEvent`, o `timestamp` que retorna quando o evento ocorreu na forma de nanosegundos, e `values[3]` sendo os valores provenientes do sensor para cada eixo, sendo que podem haver variações pois há sensores que somente possuem um valor (`values[1]`). Os retornos dos valores dos sensores podem ser vistos na Tabela 9, ao qual mostra o tipo do sensor, quantos valores retorna e a composição destes valores.

Tabela 9 - Valores de retorno dos Sensores

Tipo do Sensor	Quantidade de Valores	Composição dos Valores
<code>TYPE_ACCELEROMETER</code>	3	<code>value[0]</code> : Lateral <code>value[1]</code> : Longitudinal <code>value[2]</code> : Vertical
<code>TYPE_GYROSCOPE</code>	3	<code>value[0]</code> : Azimute <code>value[1]</code> : Pitch <code>value[2]</code> : Roll
<code>TYPE_LIGHT</code>	1	<code>value[0]</code> : Iluminação
<code>TYPE_MAGNETIC_FIELD</code>	3	<code>value[0]</code> : Lateral

		value[1] : Longitudinal
		value[2] : Vertical
TYPE_ORIENTATION	3	value[0] : Azimute
		value[1] : Pitch
		value[2] : Roll
TYPE_PRESSURE	1	value[0] : Pressão
TYPE_PROXIMITY	1	value[0] : Distância
TYPE_TEMPERATURE	1	value[0] : Temperatura

Fonte: Adaptado de MEIER (2010, tradução nossa)

6 TRABALHOS RELACIONADOS

Foram encontrados alguns projetos e aplicações e trabalhos acadêmicos relacionados ao trabalho proposto, sendo que alguns não foram encontrados referências atualizadas. Para o sistema operacional Android é comumente encontrado aplicações que emulam o teclado e *mouse*, e que enviam alguns comandos para o computador, como comandos de apresentação de slides, tocadores de musica e vídeos, porém tendo carência com a integração a jogos eletrônicos.

6.1 NIIME

O projeto NiiMe tem como princípio utilizar o acelerômetro do celular como um *joystick* no computador. Desenvolvido em Python e *open-source*, está disponível para aparelhos Nokia Symbian S60. Além da funcionalidade de joystick ainda permite utilizar o aparelho como *mouse*/teclado e até mesmo para tocar bateria no computador (ITH, 2009; ARACIOLO, 2009).

É possível encontrar alguns vídeos¹⁴ mostrando algumas das funcionalidades da aplicação. Porém ao tentar acessar o site do desenvolvedor¹⁵ aparece a mensagem informando que o site foi suspenso. Tentado encontrar alguma outra noticia sobre a aplicação, porém após 2009 não foi mais encontrado referências.

6.2 WEEWHEEL

Segundo Araciolo (2009), foi desenvolvido em Python e sua utilização é para os telefones Nokia Symbian e Python para S60. Sua funcionalidade é capturar os movimentos do acelerômetro do dispositivo e convertê-los em sinais de *joystick*, *mouse* ou sinais de teclado direto para o PC. A conexão com o computador é feita via Bluetooth, assim necessitando no computador um dispositivo Bluetooth.

Na Internet há disponível um vídeo¹⁶ onde é demonstrado sua utilização real entre um dispositivo N95 da Nokia e um computador pessoal rodando o jogo *Need for Speed Most*

¹⁴ Vídeo disponível em: <http://www.youtube.com/watch?v=00PKV5xCL24>

¹⁵ Site do desenvolvedor: <http://www.niime.com/>

¹⁶ Vídeo disponível em: <http://www.youtube.com/watch?v=iAWuYgD-njY>

Wanted da *Electronic Arts* (EA). Pelo vídeo consegue-se notar que há em certos momentos falhas, ou seja, o movimento não é repassado ao computador.

Ao acessar a página do desenvolvedor¹⁷, utilizando o antivírus gratuito Avast 6.0.1289 atualizado até a data de 17/11/2011, alertou ameaça de *Trojan*¹⁸ no site. Aberto via sistema de cache de páginas disponibilizada pela Google, informando que o cache da página é datado de 14/11/2011, ao qual se podem retirar mais algumas informações sobre a aplicação. No site informa que além da utilização do acelerômetro tem-se a possibilidade de mapear as teclas do aparelho para ações customizadas e gravação de macros, utilização do dispositivo como *mouse*/teclado. Porém a última notícia do site é datada de 10 de dezembro de 2010, tentado localizar alguma outra notícia mais recente sobre a aplicação, porém sem sucesso.

6.3 ZOOZCONTROL

Desenvolvido pela empresa ZooZMobile para os dispositivos da Apple como iPhone e iPod Touch, o ZooZControl faz conexão via Wi-Fi com o computador e permite customização dos botões, possui *force feedback*, tecnologia que alguns jogos utilizam para uma melhor imersão do jogador ao jogo fazendo com que o controle vibre dependendo da intensidade do impacto. Disponível em duas versões, a *Lite* que apenas disponibiliza pre-configurações dos controles para dois jogos, *Microsoft Flight Simulator X* e *Need for Speed ProStreet*, e a versão *Pro* ao qual o usuário pode customizar suas configurações. Como requerimentos estão ter o Windows XP/Vista e um roteador Wi-Fi. Um vídeo demonstrativo foi disponibilizado pela empresa desenvolvedora¹⁹ (ZOOZ ..., 2010).

6.4 M-COMMERCE E A PLATAFORMA GOOGLE ANDROID

Este trabalho de conclusão de curso, desenvolvido pelo acadêmico Diego Gomes Antoneli, para o curso de Ciência da Computação pela Universidade do Extremo Sul Catarinense – UNESC, no ano de 2011, para obtenção do título de Bacharel, com orientação da prof. MSc. Leila Laís Gonçalves e co-orientação do prof. Gilberto Vieira da Silva, teve como objetivo principal o desenvolvimento de um protótipo de *m-commerce* utilizando a

¹⁷ Site do desenvolvedor: <http://www.weewheel.com/>

¹⁸ Software que parece desempenhar uma função desejável para o usuário antes de executar ou instalar, mas (talvez para além da função esperada) rouba informações ou prejudica o sistema. O termo é derivado da história Cavalo de Tróia da mitologia grega

¹⁹ Vídeo demonstrativo disponível em: http://www.youtube.com/watch?v=QljS_Ian_tM

plataforma Google Android a partir de tecnologias de intercambio de dados. Para alcançar os objetivos foram necessários estudos no desenvolvimento utilizando-se de Java e XML, Web Services com implementação da técnica de programação JAX-WS e banco de dados MySQL. Utilizando a plataforma móvel Android como aplicação para acessar o conteúdo disponibilizado no Webservice.

7 DESENVOLVIMENTO DA APLICAÇÃO

Este projeto consiste no desenvolvimento de um aplicativo para o dispositivo móvel com o Sistema Android onde foi possível utilizar as capacidades do acelerômetro para o controle em jogos e aplicações de computador, funcionando como um controle sem fio ao estilo do Move da Microsoft e do Wii Remote da Nintendo, com isto agregando uma nova funcionalidade ao aparelho móvel.

7.1 METODOLOGIA

Foi efetuado primeiramente um levantamento bibliográfico para ampliar o nível de referências, sendo que este levantamento foi realizado em maioria com obras de língua estrangeira devido a facilidade de acesso.

Posteriormente fora estudado e descrito a arquitetura do sistema operacional Android, a fim de obter o conhecimento de suas funcionalidades para o desenvolvimento na plataforma.

Foi descrito a indústria de jogos eletrônicos, início e evolução pelo tempo até os tempos atuais, descrevendo os principais consoles que marcaram na história.

Exposto os requisitos de aplicações móveis para se obter uma implementação robusta e discutido de forma sucinta o desenvolvimento em plataformas móveis.

Para ampliar o conhecimento e a aplicabilidade das tecnologias envolvidas, foi necessário um estudo sobre o tipo de sensor acelerômetro, e das conectividades sem fio Bluetooth e Wi-Fi.

Por fim, após ter realizado um estudo das tecnologias envolvidas, foi efetuado o desenvolvimento da interface responsável pelo controle de jogos eletrônicos de computadores via dispositivo móvel com sistema operacional Android, utilizando-se do sensor de acelerômetro e conectividade wireless.

Para o desenvolvimento da aplicação foi necessário alguns recursos, todos disponibilizados pelo próprio acadêmico. Recursos de *hardware* foram utilizados para contemplar os objetivos, sendo eles um computador com sistema operacional Windows/Linux a título de integração entre o aplicativo no dispositivo móvel e o jogo eletrônico, sendo os dois sistemas operacionais para testes em um ambiente multiplataforma; foi utilizado um adaptador Bluetooth para a comunicação via esta tecnologia; roteador *Wireless* colocando

assim o dispositivo móvel e o computador em uma mesma rede; dispositivo móvel com sistema operacional Android e disponibilidade de sensor de acelerômetro e conexão Wi-Fi para os testes com a aplicação.

Também foram necessários alguns recursos de *softwares*, sendo necessários um ambiente de desenvolvimento Eclipse com a SDK Android para o desenvolvimento da aplicação para o dispositivo móvel e um ambiente de desenvolvimento NetBeans para o desenvolvimento da aplicação do computador (servidor), sendo que ambos os *softwares* são encontrado na Internet, nos sites dos desenvolvedores de forma gratuita. Para os jogos eletrônicos que serão realizados testes, deu-se uma priorização por jogos eletrônicos gratuitos e *open source*.

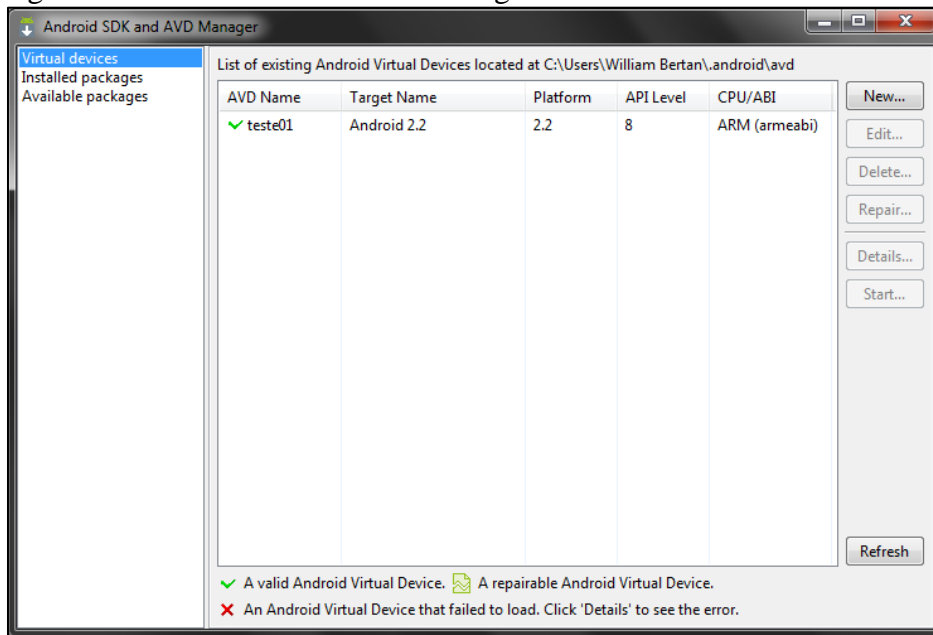
7.1.1 Estudo das Ferramentas

A aplicação para o dispositivo móvel foi desenvolvido utilizando a IDE (Integrated Development Environment) Eclipse Helios Service Release 2 Build id: 20110218-0911, esta fora selecionada por possuir uma melhor integração com a Android SDK e seu conjunto de ferramentas.

Ainda sobre o desenvolvimento móvel, foi utilizado as ferramentas Android SDK 2.2, revisão 2 que é a API da plataforma utilizada no projeto; *Android Development Toolkit* (ADT) versão 12 é um *plugin* para a IDE Eclipse ao qual proporciona um ambiente integrado, sendo assim, criar novos projetos de desenvolvimento para a plataforma, criar aplicações de interface com usuários, entre outros recursos; *Android SDK and AVD Manager*, responsável pelo gerenciamento das das API's e AVD's instaladas e disponíveis para uso, sua interface é intuitiva e simples como pode ser observado na Figura 7; *Android Virtual Device* (AVD), simula a plataforma Android em um dispositivo móvel, sendo que a Figura 8 reflete no emulador rodando a plataforma Android 2.2; e, *LogCat* ficando como responsável por auxiliar, ao desenvolver, depurar erros, sendo que ele mostra um log completo do sistema operacional e mensagens internas de aplicativos, como é demonstrado na Figura 9.

Para o desenvolvimento do aplicativo responsável por receber no computador servidor os comandos, foi utilizado a IDE Netbeans 7.0.1 pela sua facilidade no desenvolvimento de interfaces gráficas, em conjunto com a JDK 1.7.

Figura 7 - Android SDK e AVD Manager



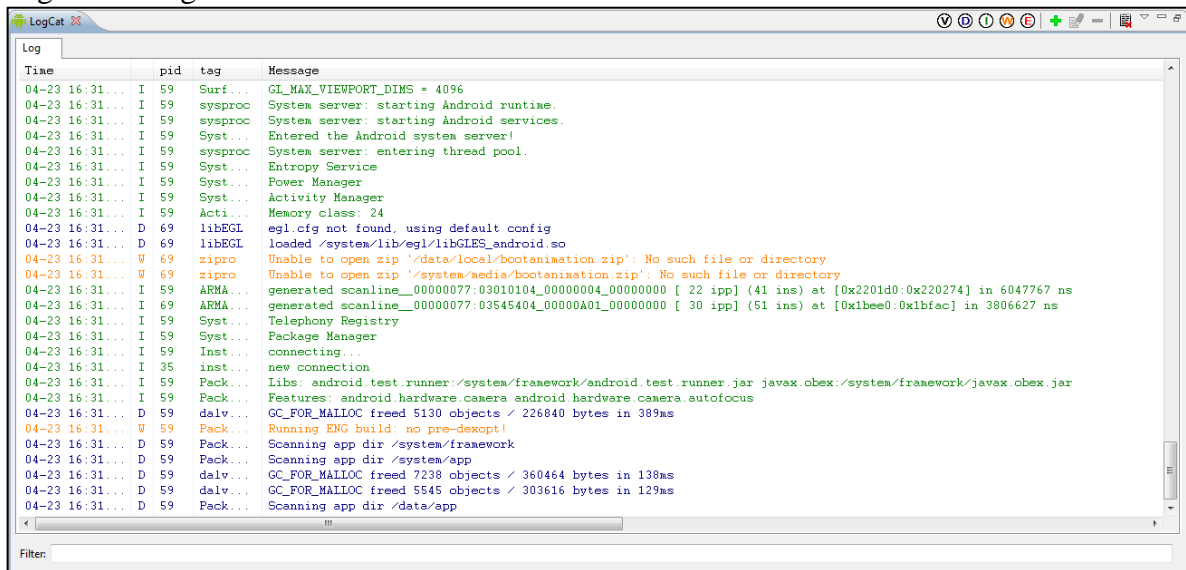
Fonte: GOOGLE INC. ([2012?]a, tradução nossa).

Figura 8 - Dispositivo móvel com a plataforma 2.2 sendo emulado



Fonte: GOOGLE INC. ([2012?]b, tradução nossa).

Figura 9 - LogCat

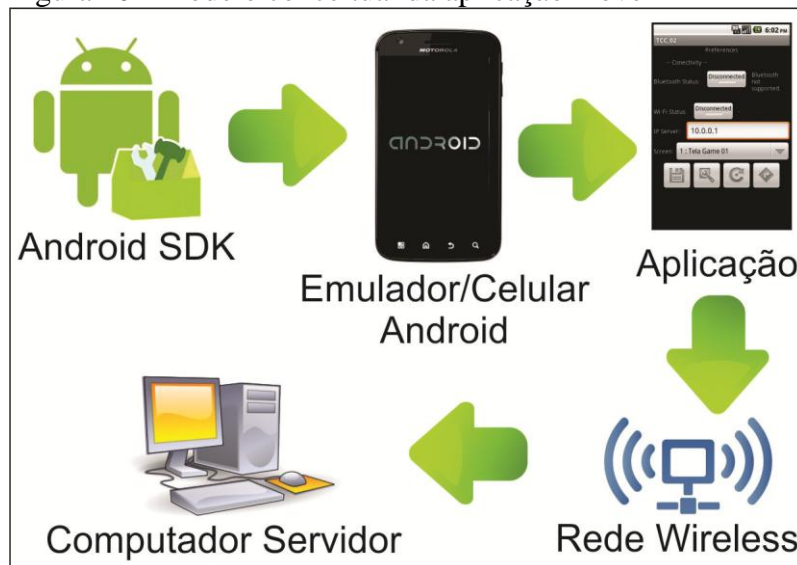


Fonte: GOOGLE INC. ([2012?]c, tradução nossa).

7.1.2 Modelo Conceitual

O projeto envolve em sua estrutura tecnologias e softwares diversos. Na Figura 10 é demonstrado o modelo conceitual da aplicação móvel.

Figura 10 - Modelo conceitual da aplicação móvel



Fonte: Do autor.

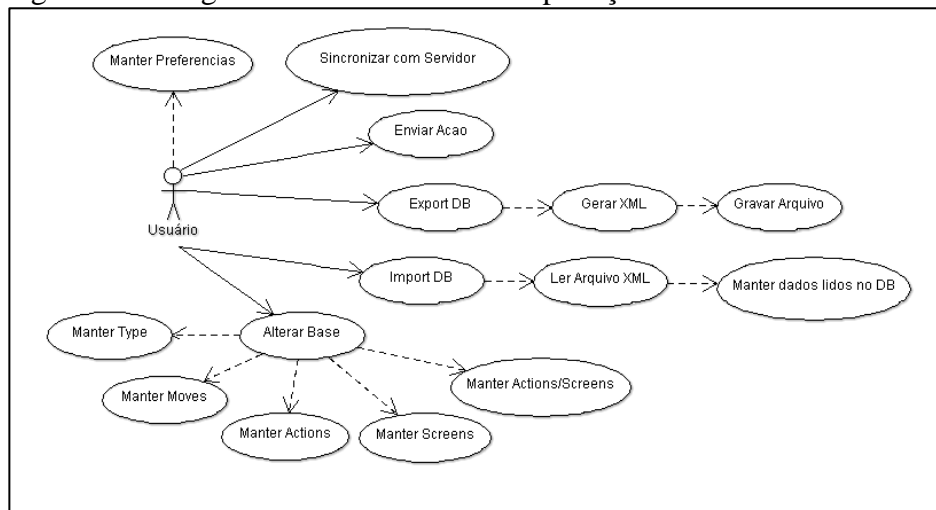
Como pode ser observado na Figura 10, a aplicação móvel é composta pelo Android SDK, um emulador ou celular com o Sistema Operacional Android, a aplicação desenvolvida, conexão com Rede *Wireless*, que no caso é através de Wi-Fi, e o computador

servidor ao qual estará rodando a aplicação *Server* e o jogo. A comunicação entre a aplicação e o Servidor é dado por meio de um padrão de códigos e valores criado pelo desenvolvedor, sendo esta troca de mensagens através da tecnologia de *Sockets* e *ObjectStreams*.

A aplicação do Servidor utiliza a tecnologia Java e IDE NetBeans para seu desenvolvimento, como também utiliza algumas bibliotecas para a interpretação de XML, recurso este utilizado em parte no projeto. A aplicação *server* trabalha sobre a porta 50000, não sendo plausível de alteração pelo usuário, através de *Sockets* para a comunicação com a aplicação móvel, sendo que a cada mensagem recebida é processada e executada uma ação específica.

Na Figura 11 demonstra-se o caso de uso da aplicação móvel, nela pode-se notar as principais funcionalidades da aplicação, ao qual seriam: Sincronizar com Servidor, responsável por atualizar a lista das ações no *server*; Alterar Base, onde pode-se realizar alterações das informações contidas na base de dados; Importar e Exportar o Banco de Dados (*Import/Export DB*), que possibilita de criar backups da base já cadastrada e recuperar estas informações posteriormente; Enviar Ação, sendo esta talvez a mais importante da aplicação, pois é ela em que faz a comunicação com o servidor e envia o comando a ser executado remotamente.

Figura 11 - Diagrama de caso de uso da aplicação móvel



Fonte: Do autor.

7.2 IMPLEMENTAÇÃO

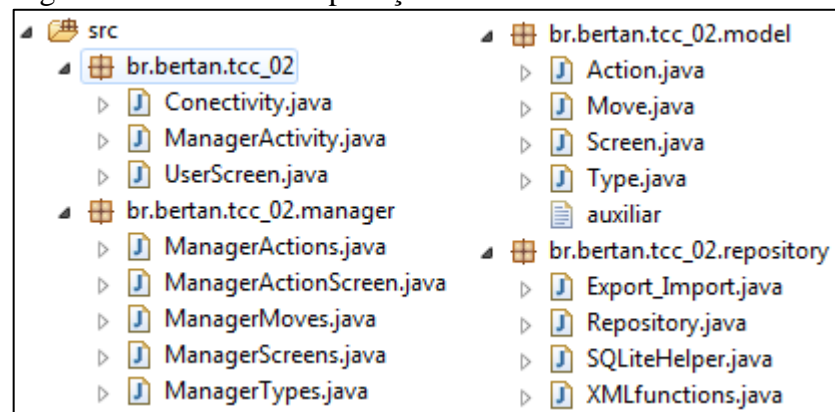
Com a escolha das tecnologias envolvidas no projeto, e após o estudo destas, foi dado continuidade das etapas metodológicas, seguindo com a implementação da aplicação móvel e posteriormente para a aplicação do servidor.

7.2.1 Aplicação Móvel

Para esta parte do projeto, foram utilizados todos os recursos para a plataforma Android descritos na metodologia.

O tipo de organização das classes foi baseado no modelo MVC²⁰, sendo que foi utilizada a separação das classes referentes ao Modelo, às classes de Visão foram separadas já pelo estilo de desenvolvimento da plataforma Android por meio de XMLs responsáveis pelos layouts do usuário, e a camada de Controle foi subdividida em outras três (*repository*, *manager*, e as classes do pacote principal). A Figura 12 ilustra as classes desta etapa do desenvolvimento que a seguir serão melhores detalhadas.

Figura 12 - Classes da Aplicação Móvel



Fonte: Do autor.

A aplicação conta com o pacote padrão, que contem as classes *Activities* principais que serão mostradas ao usuário, onde a classe *Conectivity.java* mostra a tela inicial da aplicação, solicitando as configurações das preferências de conexões, IP do servidor, tela configurada pelo usuário, e possibilitando salvar estas opções, acessar a tela de

²⁰ Modelo MVC é um modelo de desenvolvimento de aplicações utilizado na Engenharia de Software. Este modelo visa separar a lógica da aplicação da interface do usuário. Maiores informações em: <http://pt.wikipedia.org/wiki/MVC>

Gerenciamento (*Manager*) da aplicação, sincronizar a base de dados com o servidor e, após a configuração e seleção da tela, iniciar e possibilitar a comunicação da tela personalizada com o servidor através das ações do usuário. A Figura 13 mostra esta tela principal.

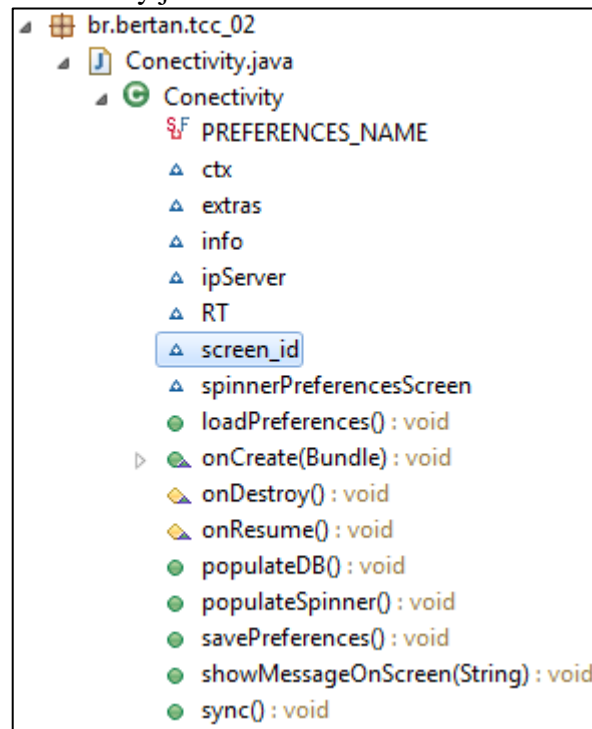
Figura 13 - Tela inicial da aplicação móvel



Fonte: Do autor.

Na Figura 14 listam-se os atributos e métodos da classe `Conectivity.java`. Os métodos `loadPreferences()` e `savePreferences()` são os responsáveis por ao iniciar a tela no método `onCreate(Bundle)`, seja carregado com as informações do usuário, e também por salvar as alterações efetuadas nas configurações, respectivamente. Outro método pertinente desta classe, é o `sync()`, ao qual é responsável por compilar todas as *Actions* cadastradas pelo usuário no formato XML, enviar ao servidor para que este possa identificar, extrair as informações e atualizar sua própria lista de *Actions*.

Figura 14 - Atributos e Métodos da classe Conectivity.java



Fonte: Do autor.

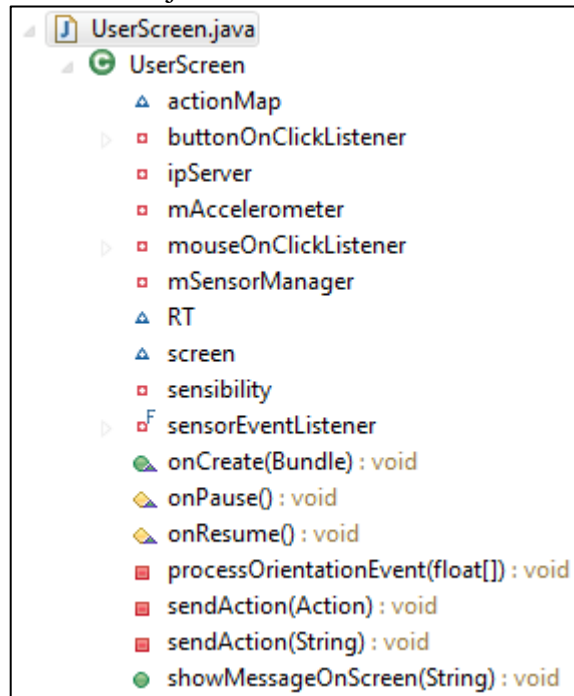
Na Figura 15 temos a classe `UserScreen.java`, sendo esta muito importante pois é ela em que mostra ao usuário a tela selecionada previamente, disponibiliza os botões personalizados, trata os sensores e realiza o envio da ação desejada.

O método `processOrientationEvent(float[])` é o responsável por tratar os dados provenientes dos sensores de orientação²¹. Nele vai ser verificado a direção que o aparelho foi movimentado (direita, esquerda, cima, baixo) nos eixos *pitch* (eixo y) e *roll* (eixo z), chamar a função de envio de ações e enviá-la ao servidor para que esta seja executada. Os eixos ao qual o acelerômetro faz referência foram explicados no capítulo 5.

Para o envio da ação ao servidor, têm-se dois métodos polimórficos, `sendAction(Action)` e `sendAction(String)`. Sendo que o segundo apenas realiza a busca da *Action* através de seu nome e posteriormente faz a chamada ao primeiro método descrito, ao qual, já tendo a *Action*, realiza a conexão propriamente dita com o servidor, sendo esta conexão efetuada através de *Sockets*, e a mensagem enviada por *ObjectStream*.

²¹ É a combinação do sensor de campo magnético, que funciona como uma bússola eletrônica, e o acelerômetro, que determina o *pitch* (eixo y) e o *roll* (eixo z) (MEIER, 2010, tradução nossa).

Figura 15 - Atributos e Métodos da classe UserScreen.java



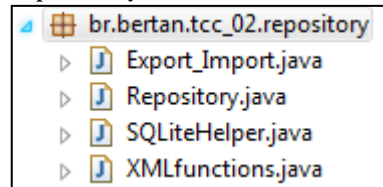
Fonte: Do autor.

Foi adotado um padrão para as mensagens enviado ao servidor, sendo eles no seguinte formato: "Type;Move", exemplo: "1;57". Sendo que o servidor, após sincronizado com a aplicação, também possui a lista das Ações, ele pode interpretar o comando requerido e reproduzir no computador. Este sistema de troca de mensagens foi escolhido, por serem simples e funcional, porem está mais suscetível a erros, como por exemplo, para o comando de sincronização foi adotado a mensagem "99;99", se o usuário cadastrar um *Type* e *Move* com estes códigos respectivos, podem ocorrer um mau funcionamento da aplicação.

Na Figura 16 expõem-se as classes pertencentes ao pacote `repository`, sendo que são classes responsáveis pela persistência dos dados da aplicação. A classe `Export_Import.java` contém as rotinas para exportar e importar a base de dados através de um arquivo XML, sendo necessário a utilização da classe `XMLfuncions.java` onde há métodos de manipulação de arquivos com formato XML. A classe `SQLiteHelper.java` fica responsável como uma classe utilitária para abrir, criar e atualizar o banco de dados no

SQLite, sendo que a classe `Repository.java` é onde possui os métodos CRUD²² dos dados e onde fará a interação com o banco de dados para a persistência.

Figura 16 - Classes do pacote *repository*



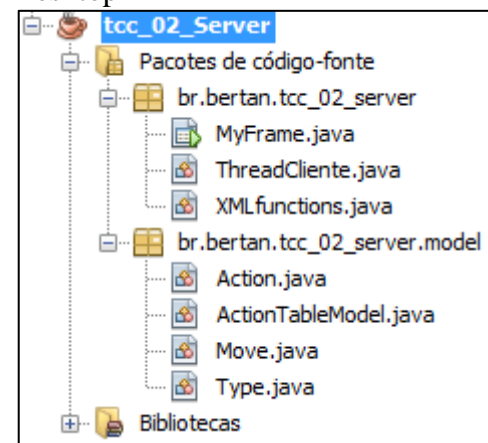
Fonte: Do autor.

7.2.2 Aplicação Desktop

Após implementado a aplicação móvel, foi dado início a aplicação desktop, ao qual é responsável por coletar os dados enviados pelo dispositivo portátil, identificar a ação solicitada e executá-la.

O desenvolvimento desta etapa também seguiu um modelo baseado no modelo MVC, porém possuindo apenas a camada Modelo em destaque, sendo que as camadas de Controle e Visão foram mescladas devido a simplicidade das ações desenvolvidas pela aplicação. A Figura 17 ilustra melhor a divisão realizada durante o desenvolvimento da aplicação.

Figura 17 - Classes da aplicação Desktop



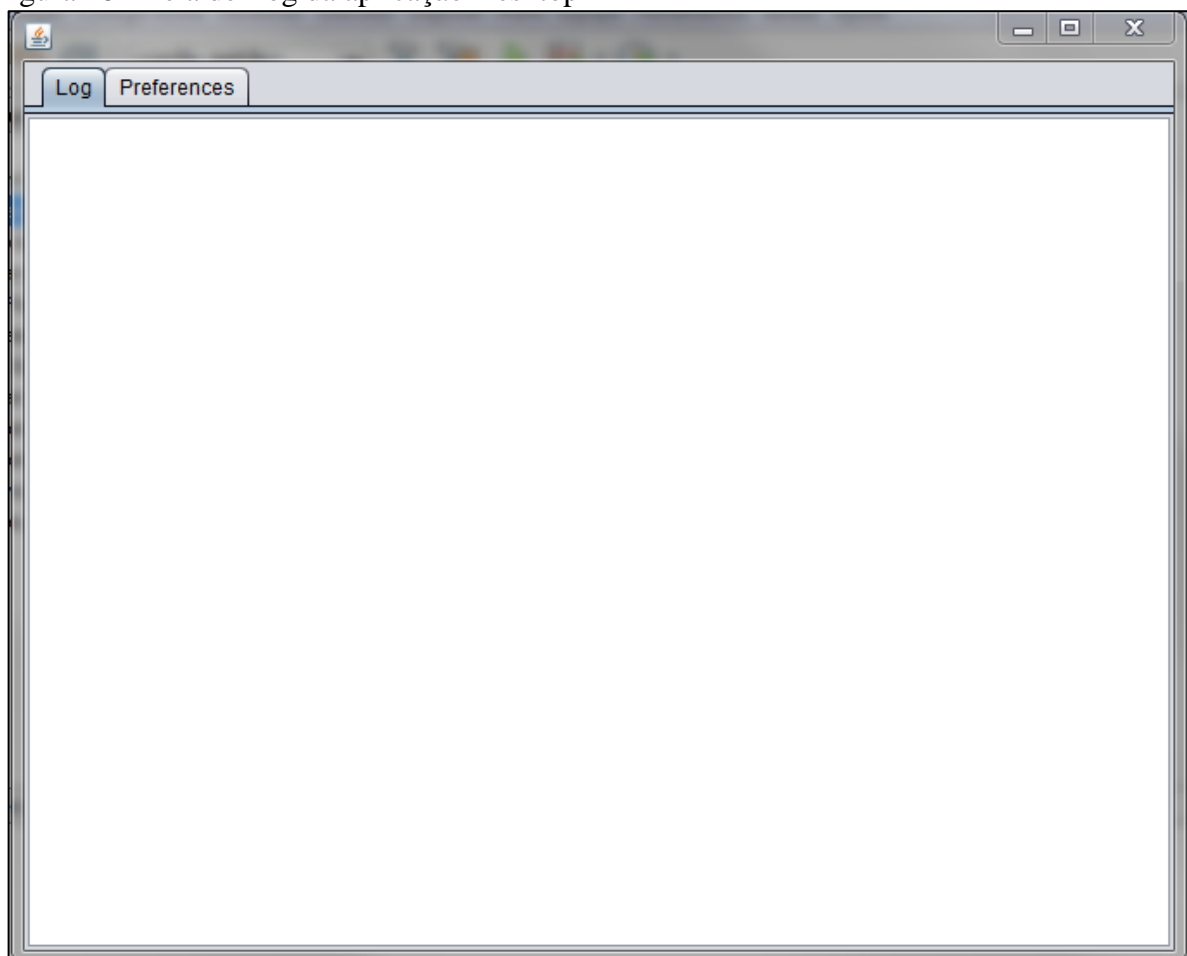
Fonte: Do autor.

²² Acrônimo para *Create, Read, Update e Delete*, consideradas estas as quatro operações básicas que se utilizam em um banco de dados relacional (FELKER, 2010, tradução nossa).

Dentro do pacote `model`, onde estão os modelos utilizados, apenas fora utilizado a classe `Action.java`, idêntica da aplicação móvel, sendo esta a única classe necessária nesta etapa, pois o propósito desta aplicação é a execução da ação solicitada pelo dispositivo móvel.

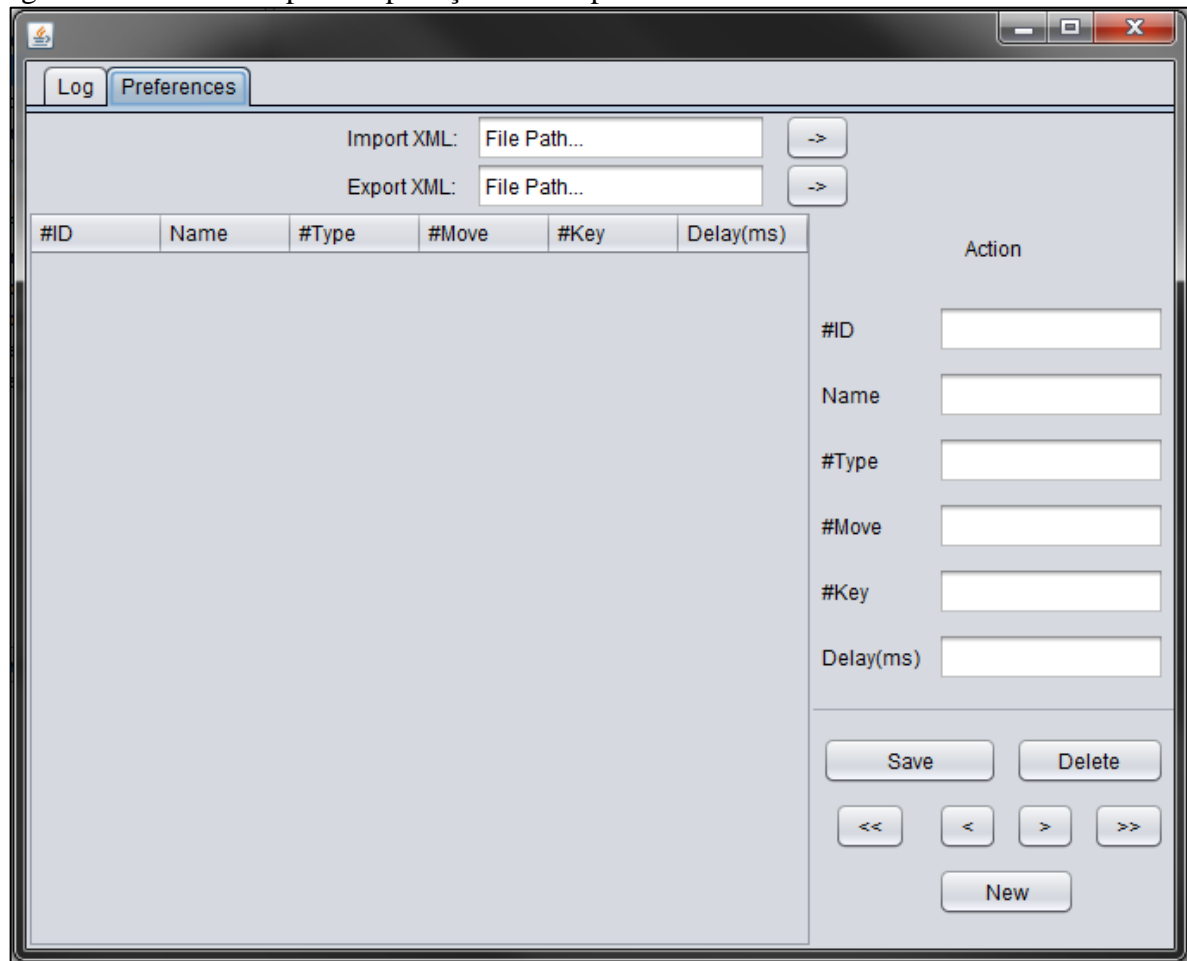
No pacote padrão, encontram-se algumas classes que mesclam as camadas de Controle e Visão, aos quais são responsáveis pelo que o usuário vê e pelo processamento das interações realizadas entre aplicação desktop x usuário e aplicação desktop x aplicação móvel. No arquivo `MyFrame.java` é onde está a tela que o usuário irá visualizar, com todo o conjunto de componentes e as ações das interações. As Figura 18 e 19 demonstram respectivamente a tela de logs da aplicação e a tela principal, estas que são criadas pela classe `MyFrame`, e a Figura 20 que mostra os métodos implementados dentro da classe, sendo que os principais serão descritos posteriormente.

Figura 18 - Tela de Log da aplicação Desktop



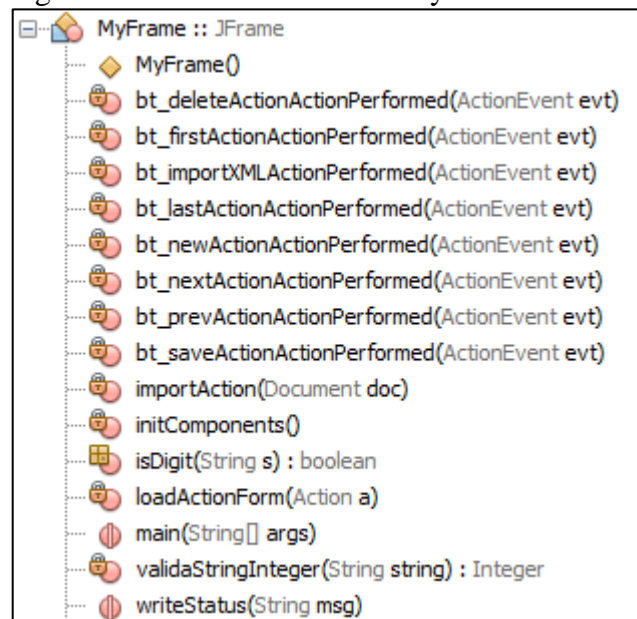
Fonte: Do autor.

Figura 19 - Tela Principal da aplicação Desktop



Fonte: Do autor.

Figura 20 - Métodos da classe MyFrame



Fonte: Do autor.

Referente aos métodos da classe `MyFrame.java`, os que iniciam com a sigla “bt” tratam uma ação executada sobre um botão da tela. Os métodos `validaStringInteger()` e `isDigit()` trabalham em conjunto e são responsáveis por verificar se uma `String` é um número, caso positivo, ele é convertido de `String` para `Integer`, caso contrário é retornado o valor zero. O método `writeStatus()` serve como um auxiliar sendo que sua funcionalidade é escrever diretamente no log que é mostrado ao usuário a `String` passado por parâmetro. No método `loadActionForm()` ele é utilizado quando alguma linha da tabela onde é mostrado as Ações é selecionada, sendo o sua função é preencher o formulário lateral a tabela com os dados da Ação escolhida. E o método mais importante desta classe é o `importAction()`, sendo que este é o método responsável por interpretar o XML recebido da aplicação móvel e popular a tabela de Ações.

Da mesma forma que na aplicação móvel, foi necessário a utilização da classe `XMLfuncions.java` onde há métodos de manipulação de arquivos com formato XML. A classe mais importante da aplicação é a `ThreadCliente.java`, sendo que esta é executada a cada vez que a aplicação móvel envia algum comando para a aplicação desktop interpretar.

A classe `ThreadCliente` ao ser executada, recebe da aplicação móvel comandos no seguinte formato, já explicados no capítulo 7.2.1 Aplicação Móvel, “`Type;Move`”, exemplo: “`1;57`”, sendo a exceção o comando de sincronização, ao qual é fixo o valor de “`99;99`”. Ao receber o comando, este é dividido e interpretado, sendo que para o funcionamento correto da aplicação, esta já deve ter sido sincronizada com a aplicação móvel a título de possuir todas as `Actions` possíveis. Quando separados, primeiramente são verificadas as particularidades, que nesta aplicação a única é o comando de sincronização, caso não for, trata-se de uma ação a ser executada.

Ao identificar que se trata de uma ação, a aplicação realiza a busca pela `Action` correspondente ao `Type` e `Move` recebidos na mensagem, uma vez encontrado, passa-se por mais uma verificação, a do `Type` da ação requerida. Se for uma ação do tipo *keyboard*, o valor da tecla também deve ser diferente de 0 (padronizado na aplicação que zero é para quando não há correspondente no teclado), se for válido é chamado a classe `Robot` do Java, utilizada para que a aplicação possa controlar mouse e teclado, chamando o método `keyPress()` e `keyRelease()` passando por parâmetros o código da tecla que está na `Action` localizada. Se a ação for do tipo *mouse*, é verificado o movimento, pois a aplicação

trata que: 2, 3, 4 e 5 são responsáveis pelos movimentos, respectivamente, para cima, baixo, esquerda e direita; 6, 7 e 8 são responsáveis pelos cliques, respectivamente, do botão esquerdo, botão direito e botão central. Sendo que estas ações com o *mouse* são executadas pela classe `Robot` utilizando o método `mouseMove()` para os movimentos na tela, passando por parâmetro a posição na tela onde o *mouse* irá ser posicionado, e os métodos `mousePress()` e `mouseRelease()` para os botões de cliques, sendo passado por parâmetro os valores da máscara mantidos pela classe `InputEvent` (exemplo: `InputEvent.BUTTON1_MASK`).

7.2.3 Funcionamento

A aplicação utiliza-se do princípio de troca de mensagens entre as a aplicação móvel e a servidora que foram desenvolvidas. Estas mensagens são o que possibilitam do servidor saber qual ação que fora selecionada no dispositivo móvel e assim executá-la no jogo. A implementação deu-se de forma unidirecional, assim sendo, somente a aplicação móvel envia informações e a aplicação desktop apenas recebe e interpreta, pois para o servidor, neste momento, não é necessário o envio de mensagens, sendo que seria mais útil quando há informações que poderiam ter sido processadas e retornadas a origem, ou como forma de confirmação do comando executado, sendo estes dois últimos não contemplados pelas aplicações.

O funcionamento da aplicação é de forma simples, na aplicação móvel têm-se as ações (`Actions`), estas que foram formadas com o auxílio dos tipos (`Types`) e movimentos (`Moves`) dentre outras configurações, e na aplicação desktop têm-se as ações (`Actions`). Toda a base é mantida no dispositivo móvel, decidido assim para manter a portabilidade da aplicação viável e facilitada. Com isto, a aplicação servidora (*desktop*), pode ser denominada uma aplicação “oca”, ao qual possui todos os métodos necessários para a execução e interpretação das mensagens, porém não possui os dados a serem consultados (no caso as `Actions`), assim, se faz necessário sempre ter que realizar a sincronização, uma vez conectados, entre as duas aplicações, onde na prática o que ocorre é que as `Actions` armazenadas na aplicação móvel são enviadas e armazenadas em memória de execução pela aplicação servidora. Decidiu-se utilizar a sincronização para manter a escalabilidade da aplicação *desktop*, sendo que se pode utilizá-la com diferentes usuários móveis e em diferentes locais uma mesma aplicação.

Após ter realizado a sincronização das `Actions`, o usuário ao clicar em um botão de sua tela (`Screen`) na aplicação móvel, envia ao servidor o tipo e movimento (no formato “`Type;Move`”), sendo que cabe ao servidor localizar a ação correspondente e executá-la. É possível de alterar as `Actions` do servidor sem alterá-las na base do dispositivo móvel, para isto têm-se uma tabela visível ao usuário e plausível de seleção para posterior alteração, sendo que as alterações apenas são válidas para a aplicação servidora até que esta seja finalizada ou sobrescrevida (nova sincronização). Para alterar diretamente na base é necessário fazer-se através do dispositivo móvel.

A aplicação é instalada já com uma base padrão, e necessária, para o correto funcionamento da aplicação, porém o usuário possui a liberdade de adicionar novas ações. A base não contempla todas as teclas do teclado, assim basta selecionar um *sprite* para a tecla e configurá-la com o seu código referente ao seu `KeyEvent` do Java. A aplicação não contempla novo tipos, por exemplo, um *gamepad* ou “tapete de dança”, porém a aplicação suporta o cadastramento destes e a utilização das ações, sendo a limitação na aplicação do servidor, que teria que ser melhorada para reconhecer novos tipos e suas ações.

7.3 RESULTADOS OBTIDOS

Com a pesquisa desenvolvida neste trabalho, pôde-se obter duas aplicações para plataformas distintas, ao qual quando combinadas se unem para alcançar os objetivos propostos no projeto inicial.

A primeira constitui-se da aplicação para a plataforma móvel Android. Criada com o intuito de construir uma aplicação fácil de utilizar, com telas intuitivas e práticas, utilizando-se de recursos tecnológicos avançados e aprimorados. Sua utilização ficou simplificada e possibilita variedade nas ações oferecidas, assim sendo uma forma de adaptação ao usuário que poderá utilizar e personalizar, até certo ponto, a aplicação às suas necessidades. Com o auxílio dos dados obtidos através de leituras sobre os seus sensores, principalmente o acelerômetro, é possível identificar movimentos realizados pelo usuário, mesmo que até o momento implementado o básico (cima, baixo, direita e esquerda), já é o suficiente para apreciar o quanto esta tecnologia fornece de possibilidades.

A segunda aplicação desenvolveu-se para a plataforma Java para Desktop, sendo esta dotada de uma interface simples, porém que possibilita ajustes imediatos, se necessários. Também traz ao usuário um retorno das ações executadas através da tela de log, assim o

usuário saberá o que está acontecendo em sua máquina. Internamente esta aplicação trabalha em conjunto com a aplicação móvel, recebendo a tabela de ações personalizadas do usuário, e as ações a serem executadas.

Esta harmonia entre as duas aplicações propicia atingir o objetivo macro, a utilização de dispositivos móveis na forma de um joystick para jogos eletrônicos, fazendo uso do sensor de acelerômetro. A realização desta pesquisa proporcionou a descrição tanto da área de jogos eletrônicos, em grande expansão e que já foi muito explorado, porém que com a outra parte da pesquisa, acelerômetro, sensores e dispositivos móveis, abrem mais um nicho que possibilita novas explorações na área, revolução na forma de interação.

Como conclusão desta etapa, todos os objetivos propostos foram atingidos, sendo como resultado duas aplicações de código livre e robustas, disponíveis aos acadêmicos interessados tanto em sua utilização, como em melhorias e também como base para novos projetos.

8 CONCLUSÃO

Ao longo da pesquisa foram efetuados estudos aos quais proporcionaram o entendimento sobre a Plataforma Google Android, seus aspectos funcionais e características de desenvolvimento. Além do conhecimento adquirido para a plataforma móvel, obtiveram-se também conhecimentos aprofundados no que tange a tecnologias sem fio.

Através de toda esta gama de conhecimentos adquiridos, tornou-se viável alcançar o objetivo principal desta pesquisa que consiste na utilização de dispositivos móveis na forma de um joystick para jogos eletrônicos, fazendo uso do sensor de acelerômetro.

Para se alcançar este ponto principal, os estudos realizados e aplicações testes forneceram um amplo conhecimento sobre os assuntos abordados para o acadêmico, abrindo novas possibilidades de projetos que poderão trazer benefícios às comunidades científicas e sociais.

O resultado da pesquisa se mostrou favorável e com sucesso foram atingidos todos os objetivos almejados, tanto o geral quanto os específicos. Sendo que estes resultados consistem-se, além do conhecimento teórico, de duas aplicações distintas, a aplicação móvel, desenvolvida unicamente para a plataforma Android, que utiliza-se de conectividade sem fio para conexão com o computador servidor e o uso de acelerômetro para captura de movimentos do usuário; e a outra aplicação, desenvolvida sobre a linguagem Java, tornando a aplicação multiplataforma, com o propósito de servidor para os dados recebidos da aplicação móvel, têm a função de interpretar estas informações e executá-las no computador remoto o que o usuário solicitou através do dispositivo móvel, focando o seu uso em jogos eletrônicos.

Poucas foram as dificuldades encontradas. Os materiais de estudo proporcionaram um entendimento prático e funcional das ferramentas utilizadas, porém não foi possível a implementação da comunicação entre a aplicação móvel com o servidor através da tecnologia Bluetooth. Sendo que para o ambiente Android a manipulação de conexões e canais de transmissão (recepção e envio) de dados são de fácil manipulação, porém para a implementação no servidor esta tecnologia ainda está imatura e com alguns problemas de compatibilidade com adaptadores Bluetooth. A implementação utilizando-se de Java e Bluetooth em ambiente desktop não é impossível, porém para a pesquisa não é relevante este canal de comunicação visando que o foco é a utilização do acelerômetro, sendo este alcançado junto com a comunicação via Wi-Fi.

Para a conclusão da pesquisa, obteve-se uma gama de conhecimentos suficientes para o início e desenvolvimento de novas ideias de aplicações em diversas áreas, tais como educação e entretenimento. Os recursos disponibilizados na plataforma móvel, e as tecnologias envolvidas facilitam no desenvolvimento e com isto aumentam o incentivo para este nicho de possibilidades.

O acelerômetro proporciona grandes possibilidades de uso, para movimentos básicos, como os utilizados nesta pesquisa, é de fácil implementação e manipulação dos dados, porém para uma aplicação em que se deseja coletar movimentos elaborados do usuário, requer uma complexidade alta tanto do processamento das informações providas das leituras como do algoritmo que o fará. O seu uso pode ser ampliado não somente para a área de jogos eletrônicos, como também para as diversas áreas, podem-se utilizá-lo, por exemplo, para o controle de robô em ambientes de riscos, e um uso atual é nos discos rígidos de computadores que podem detectar quedas e assim travar as cabeças de gravação para não danificarem os dados do usuário.

Para sugestões em trabalhos futuros que seguem esta linha de pesquisa:

- a) implementar a comunicação entre a aplicação móvel com o servidor utilizando a tecnologia Bluetooth;
- b) ampliar as capacidades das aplicações para a inserção e funcionamento de novos dispositivos, como por exemplo um tapete de dança, e suas funcionalidades estarem disponíveis para uso na aplicação móvel;
- c) implementar feedback ao usuário da aplicação servidora para a móvel, como vibração do dispositivo em impactos no jogo eletrônico;
- d) implementar a utilização de movimentos complexos do usuário, como por exemplo, o movimento de sacar a espada, ou de se proteger com um escudo;
- e) ampliar o uso para além de jogos eletrônicos, podendo trazer a funcionalidade para a área de acessibilidade para pessoas deficientes com dificuldades de utilizarem computadores, facilitando a inclusão destas no meio digital.

REFERÊNCIAS

- ABLESON, W. Frank; SEN, Robi; KING, Chris. **Android in Action**. Second Edition [S. l.]: Manning Publications, 2011.
- ANTONELI, Diego Gomes. **M-Commerce e a Plataforma Google Android**. 2011. 90 f. Trabalho de Conclusão de Curso (Bacharelado) – Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense – UNESC, Criciúma, BR – SC, 2011.
- ARACIOLO, Marcel Pinheiro. **Transforme seu celular com acelerômetro em um controle para PCS**. 2009. Disponível em: <<http://www.mobideia.com/2009/07/transforme-seu-celular-com-acelerometro.html>>. Acesso em: 17 nov. 2011.
- ARKADE: A História dos Videogames. Curitiba: Arkade, [2011?].
- BURNETTE, Ed. **Hello, Android: Introducing Google's Mobile Development Platform**. 2nd edition [S. l.]: Pragmatic Bookshelf, 2009.
- CONDER, Shane; DARCEY, Lauren. **Android Wireless Application Development: Developer's Library**. Second Edition. [S. l.]: Addison Wesley, 2010.
- DAWSON, Michael. **Beginning C++ Through Game Programming**. Third Edition. Boston: Course Technology PTR, 2010.
- DIMARZIO, J. F.; POLO, G. L. **Android, A Programmer's Guide**. [S. l.]: McGraw-Hill Osborne Media, 2008.
- EDDY, John; EDDY, Patricia Di Giacomo. **Google on the Go: Using an Android-Powered Mobile Phone**. Indiana: QUE, 2009.
- FELKER, Donn. **Android Application Development For Dummies**. [S. l.]: For Dummies, 2010.
- GOOGLE INC.. **Managing AVDs with AVD Manager**. [2012?]a. Disponível em: <<http://developer.android.com/guide/developing/devices/managing-avds.html>>. Acesso em: 12 jun. 2012.
- _____. **Using the Android Emulator**. [2012?]b. Disponível em: <<http://developer.android.com/guide/developing/devices/emulator.html>>. Acesso em: 12 jun. 2012.
- _____. **Logcat**. [2012?]c. Disponível em: <<http://developer.android.com/guide/developing/tools/logcat.html>>. Acesso em: 12 jun. 2012.
- HASHIMI, Sayed; KOMATINENI, Satya; MACLEAN, Dave. **Pro Android 2**. [S. l.]: Apress, 2010.

ITH. **Niime 1.3b**: N95 como Mouse/Console. 2009. Disponível em: <<http://www.n95br.com/niime/>>. Acesso em: 17 nov. 2011.

KARCH, Marziah. **Android for Work**: Productivity for Professionals. [S. I.]: Apress, 2010.

LECHETA, Ricardo R. **Google Android**. 2 ed. São Paulo: Novatec, 2010.

MALLICK, Martyn. **Mobile and Wireless Design Essentials**. New Jersey: John Wiley & Sons, 2003.

MEIER, Reto. **Professional Android™ 2 Application Development**. New York: Wiley Publishing, 2010.

MORIMOTO, Carlos E.. **Smartphones**: Guia Prático. [S. I.]: Gdh Press e Sul Editores, 2009.

MORINAGA, Norihiko; KOHNO, Ryuji; SAMPEI, Seiichi. **Wireless Communication Technologies**: New Multimedia Systems. 2002.

MORRISON, Michael. **Sams Teach Yourself Game Programming in 24 Hours**. Indianapolis: Sams Publishing, 2002.

MULLER, Nathan J. **Bluetooth Demystified**. New York: McGraw-Hill Professional Publishing, 2000.

MULLER, Nathan J. **Wireless A to Z**. New York: McGraw-Hill Professional, 2002.

MUNDO ESTRANHO: A era de ouro dos Games. São Paulo: Abril, n. 118, dez. 2011. Mensal.

MURPHY, Mark L. **Beginning Android**. New York: Apress, 2009.

MURPHY, Mark L. **Beginning Android 2**. New York: Apress, 2010.

PEREIRA, Lúcio C. O.; SILVA, Michel L. **Android para desenvolvedores**. Rio de Janeiro. Brasport. 2009.

STEELE, James; TO, Nelson. **The Android Developer's Cookbook**: Building Applications with the Android SDK. New York: Addison-Wesley Professional, 2010.

TANENBAUM, Andrew S.. **Computer Networks**. 4th Edition [S. I.]: Prentice Hall, 2002.

THE COMPLETE GUIDE TO GOOGLE ANDROID. [S. I.]: Idg Communications, [2011?].

ZOOZ Control. 2010. Disponível em: <<http://www.zoozmobile.com/zoozcontrol.htm>>. Acesso em: 17 nov. 2011.

BIBLIOGRAFIA COMPLEMENTAR

ANDERSON, Richard. **Professional XML**. Rio de Janeiro: Ciência Moderna, 2001.

CADENHEAD, Roger; LEMAY, Laura. **Aprenda em 21 dias Java 2**. Rio de Janeiro: Campus, 2005.

DEITEL, H.M; DEITEL, P. J. **Java: como programar**. 6. ed. São Paulo: Pearson Education do Brasil, 2005.

FAN, Joel; RIES, Eric; TENITCHI, Calin. **Black art of Java game programming**. Corte Madera: Waite Group, 1996.

FRIESEN, Jeff. **Learn Java for Android Development**. [S. l.]: Apress, 2010.

LARAMEE, Francois Dominic. **Secrets of the Game Business**. Massachusetts: Charles River Media, 2003.

LEE, Valentino; SCHNEIDER, Heather; SCHELL, Robbie. **Aplicações móveis: arquitetura, projeto e desenvolvimento**. São Paulo: Pearson/Makron Books, 2005.

MCLAUGHLIN, Brett. **Java & XML**. 2 ed. Sebastopol: O'Reilly, 2001.

MENDES, Antonio. **Programando com XML**. Rio de Janeiro: Elsevier, 2004.

MENDONÇA, Alexandre; FRANÇA, Paulo Roberto; ZELENOVSKY, Ricardo. **Hardware: programação virtual de I/O e interrupções**. Rio de Janeiro: MZ, 2001.

MINEI, Ina; LUCEK, Julian. **MPLS-Enabled Applications: Emerging Developments and New Technologies**. New York: Wiley, 2005.

MITCHELL, John C. **Concepts in Programming Languages**. Cambridge: Cambridge University Press, 2002.

MORONTA, Lewis. **Game Development with ActionScript**. Boston: Premier Press, 2004.

MYERS, Glenford J. **The Art of Software Testing**. 2 ed. New York: Wiley, 2004.

PATTERSON, David A.; HENNESSY, John L. **Computer Organization and Design: The Hardware/Software Interface**. 3 ed. San Francisco: Morgan Kaufmann, 2004.

SIERRA, Kathy; BATES, Bert. . **Use a cabeça!:** Java. 2. ed. Rio de Janeiro: Alta Books, 2007.

SULLIVAN, Dan O'; IGOE, Tom. **Physical Computing: Sensing and Controlling the Physical World with Computers**. [S. l.]: Thomson, 2004.

TANENBAUM, A. S. **Organização Estruturada de Computadores**. 4 ed. Rio de Janeiro: LTC, 2001.

TORRES, Gabriel. **Hardware**: curso completo. 4 ed. Rio de Janeiro: Axcel Books, 2001.

ZELENOVSKY, R.; MENDONÇA, A. **PC**: um Guia Prático de Hardware e Interfaceamento. Rio de Janeiro: MZ, 1999.

APÊNDICE A – ARTIGO CIENTÍFICO

Interfaceamento entre Dispositivos Móveis e Jogos Eletrônicos de Computadores Utilizando o Acelerômetro Baseado no Sistema Operacional Android

William Bertan da Silva¹, Paulo João Martins¹

¹Curso de Ciência da Computação – Universidade do Extremo Sul Catarinense (UNESC)
Criciúma – SC – Brasil

w_toco@hotmail.com, pjm@unesc.net

***Abstract.** This paper has by objective show the use of accelerometer sensor in mobile devices with Android. Giving a brief understand about the emergent platform Android, the important characteristics and the integration with another technologies like the Wi-Fi wireless transmission and the accelerometer. Using as well the multiplatform Java for the development of the server application, bringing a new role to the mobile device. Through this research, it hope that new innovations with the present technologies and the emergent can be developed and broadcasted by the social and scientific communities.*

***Resumo.** Este artigo tem por objetivo demonstrar a utilização do sensor de acelerômetro presente em dispositivos móveis Android. Proporcionando um breve entendimento sobre a utilização da plataforma emergente Android, suas principais características como também sua integração com outras tecnologias como as transmissões sem fios Wi-Fi e o próprio Acelerômetro. Também utilizando-se da multiplataforma Java para criação da aplicação servidora, assim agregando uma nova funcionalidade ao dispositivo móvel. Através deste estudo espera-se que novas inovações com as tecnologias presentes e futuras possam ser criadas e difundidas pelas comunidades social e científica.*

1. Introdução

Com o intuito de diminuir o tamanho de objetos e agregar diversas funcionalidades a eles, é comum encontrar atualmente aparelhos que outrora eram de tamanhos exorbitantes e não executavam grandes tarefas, agora com tamanhos minúsculos executando centenas de vezes mais tarefas do que aparentam ser possível.

Um dos objetos que agregaram inúmeras funcionalidades pode-se citar os dispositivos móveis, celulares e os mais atuais, tablets. No início da difusão dos dispositivos móveis eram comuns celulares serem usados apenas para a comunicação, ou seja, ligações. Na atualidade, ao procurar por um celular, tem-se como base o que se pode desfrutar do mesmo, quais as capacidades que ele pode ter e o que se consegue fazer nele, qual o seu diferencial.

Conforme a competitividade aumenta no mercado onde estão inseridos os fabricantes de dispositivos móveis, ao lançar uma nova funcionalidade no seu produto também normalmente acabam por agregar um novo padrão, sendo que outros fabricantes tentarão se adequar para inserir em seus produtos funcionalidades similares. Com isto pode-se então dizer que um padrão para os produtos está sendo criado. Como um exemplo pode-se ter o celular, onde os atuais vêm com acelerômetro, tela touchscreen, processadores potentes, memórias dedicadas, processamento de vídeo independente, câmeras de alta resolução, Wi-Fi

entre outras características. Assim tem-se uma noção de que novos produtos irão vir com estas funcionalidades e talvez até com outras e assim aumentando a gama de diversificação e diferencial.

A integração destes dispositivos com os já existentes tornam a vida mais fácil. Imagine controlar as luzes da casa, verificar e controlar a temperatura do ar condicionado, visualizar a câmera do jardim tudo isto com o uso de um celular por exemplo.

Os novos consoles de jogos estão trazendo novidades ao mundo dos Gamers, além de gráficos e novos estilos de jogabilidade incríveis estão trazendo um novo conceito de interatividade. O X-Box 360 da Microsoft traz o atual e revolucionário Kinect onde conta com uma câmera infravermelha para capturar os movimentos do jogador e refletir em ações nos seus jogos trazendo uma imersão maior com o jogo e inúmeras aplicações e diversão. Com o console Playstation 3 da Sony veio o Move, controle com acelerômetro para pegar com precisão os movimentos do jogador e conta também com uma câmera chamada Playstation Eye. O console Wii da Nintendo também traz como o Move um controle (Wii Remote) para captura dos movimentos.

2. Jogos Eletrônicos

Conforme descrito por Morrison (2002, tradução nossa), a indústria de vídeo game cresceu se tornando um importante setor para a economia mundial. Jogos eletrônicos agora são financiados em escalas comparadas a filmes de Hollywood. Com esta passagem pode-se perceber como a indústria de jogos eletrônicos está crescendo e se destacando mundialmente.

Em conjunto ao mercado dos consoles também se teve a evolução dos computadores. Tendo seu início na década de 90, os computadores se tornaram cada vez mais pessoais e com isto se popularizaram, e junto surgiu o mercado de jogos eletrônicos.

Com os computadores surgiram alguns gêneros como o First Person Shooter (FPS) e alguns títulos clássicos. O poder destas máquinas também cresceram, como exemplo, na época do lançamento do jogo Doom para computadores, uma máquina com o processador 486 era considerada topo de linha, e isto que possuía, em média, apenas 16 Mb de memória RAM, hoje em dia os computadores podem possuir mais de 4 Gb, isto é 250 vezes mais, e ainda os computadores modernos contam com processadores velozes, placas de vídeo de última geração com processadores gráficos exclusivos e periféricos.

Na atualidade os jogos eletrônicos estão mais presentes no cotidiano da população. Podendo ser jogados em lugares indistintos com dispositivos móveis ou no conforto de sua casa com os consoles, computadores e demais dispositivos.

Morrison (2002, tradução nossa) descreveu o motivo pelos quais jogos eletrônicos são tão bem vindos à sociedade moderna. Ele faz uma comparação com os filmes atuais, aos quais poucas pessoas são capazes de assistir um filme e não sentir que em algum momento não estão fazendo parte de determinada cena. Sendo que os jogos eletrônicos possibilitam esta imersão e ainda dão ao usuário o controle deste mundo.

Reforçando a ideia de Morrison temos Dawson (2010, tradução nossa) quando afirma que a indústria de jogos para computadores é um rival para o melhor que Hollywood pode oferecer. Em uma imersão do jogador a um jogo por horas, ele não fica apenas limitado a assistir seu herói participar das batalhas, ele acaba se tornando o herói.

Os fios tiveram um grande papel na evolução dos consoles e seus periféricos, conectando-os e proporcionando ao usuário imersão aos jogos e aplicações. Com a evolução

das tecnologias empregadas em conectividade, a tecnologia sem fio (wireless) vem substituindo os fios das conexões possibilitando imersões do usuário com novas formas de interação.

3. Conectividade Wireless

Segundo Mallick (2003, tradução nossa), tecnologias de conexões sem fios são um dos tópicos mais abordados em dispositivos móveis e todos possuem uma opinião sobre o estado da tecnologia das redes 3G, o efeito do Bluetooth nas redes pessoais, e como as redes locais sem fios dominarão o mercado. Redes sem fios (wireless networks) possuem vários propósitos, em alguns deles são para substituir o uso de cabos, enquanto em outros para disponibilizar acesso a dados de locais remotos.

Segundo Muller (2000, tradução nossa) as pessoas gastam um grande tempo tentando esconder os vários cabos e fios que conectamos os equipamentos. É um fio conectando um mouse e outro do teclado ao computador, fios das caixas acústicas, microfone de uma estação multimídia, cabos conectando a câmera, dispositivos de armazenamentos, celulares para transferência de dados. Chegando a um ponto em que as nossas pastas ficarem cheias com os cabos para conectar os diversos dispositivos que carregamos conosco. Em uma sociedade cada vez mais conectada, conectar as coisas se tornou um grande problema.

3.1. Wi-Fi e Android

Segundo Conder e Darcey (2010, tradução nossa), o sensor de Wi-Fi pode ler o status de uma rede e determinar o ponto de acesso mais próximo. A SDK do Android prove um conjunto de APIs para recuperar informações sobre as redes Wi-Fi disponíveis ao aparelho e detalhes das conexões destas redes. Estas são separadas das APIs do SensorManager. Com estas informações é possível recuperar uma série de outras informações e utilizá-las, por exemplo, para determinar a força do sinal, achar pontos de acesso de interesse, realizar ações quando conectado a um determinado ponto de acesso.

Conforme Conder e Darcey (2010, tradução nossa) e Meier (2010, tradução nossa), o WifiManager representa o Serviço de Conectividade Wi-Fi do Android. Ele pode ser utilizado para configurar conexões de redes Wi-Fi, gerenciar a conexão atual, procurar por pontos de acessos e monitorar mudanças na conexão. Sobre as permissões da aplicação descritas no arquivo AndroidManifest.xml, a permissão `CHANGE_WIFI_STATE` é necessário quando a aplicação está acessando informações sobre as redes sem fio, como também para modificar seu estado, já a permissão `ACCESS_WIFI_STATE` é necessário para requisitar qualquer informação do dispositivo Wi-Fi.

4. Sistema Operacional Android

Alguns anos atrás os celulares não contavam com toda a gama de recursos que possuem atualmente, tais quais GPS, câmera, tecnologia touchscreen, tocador de músicas e vídeos, entre outros. Além das tecnologias empregadas no hardware, existe o software, onde os fabricantes mantinham fechado o sistema que rodava sobre seus celulares, dificultando a evolução e melhoria de recursos [DIMARZIO; POLO, 2008, tradução nossa, MEIER, 2010, tradução nossa].

Para mudar este cenário, e trazer melhorias tanto para os desenvolvedores quanto aos usuários finais, foi criada a Open Handset Alliance (OHA), liderada pela gigante e revolucionária Google e diversos integrantes que marcam presença em fatores tecnológicos tais quais a HTC, Motorola, Sony Ericsson, Intel, entre outras. A OHA tem seu objetivo

focado na definição de uma plataforma aberta e única para dispositivos móveis, visando os usuários e os desenvolvedores, sendo que para satisfazer seus objetivos foi criado o Android. [LECHETA, 2010].

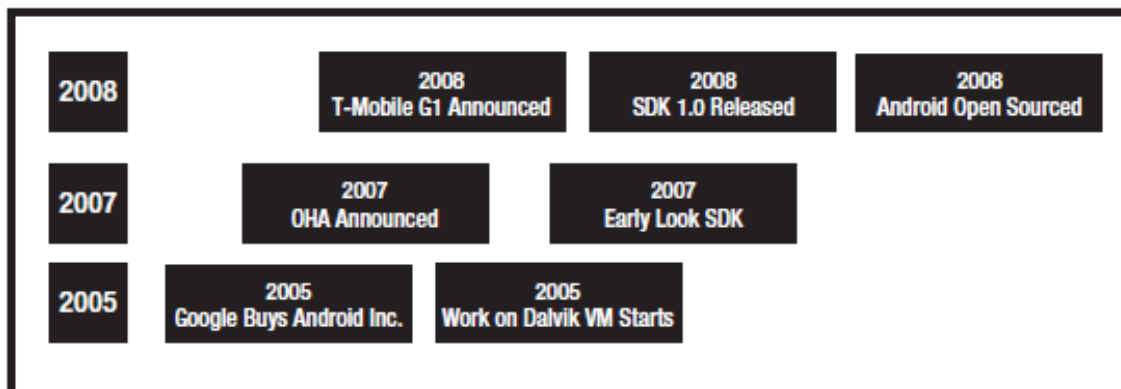


Figura 1. Linha do tempo da história do Android

Na Figura 1 pode-se ver uma linha do tempo da história da plataforma, onde mostra os principais acontecimentos, desde a compra da Android Inc. pela Google até o pronunciamento da abertura do código fonte.

Em meados de 2007, algumas empresas tecnológicas mundialmente conhecidas (Sprint Nextel, Motorola, Samsung, Sony Ericsson, Intel, Google entre outras) se uniram ao redor da plataforma e fundaram a Open Handset Alliance (OHA), tendo como principais objetivos uma inovação rápida e uma melhor abrangência das necessidades dos consumidores, sendo como chave para alcançar os objetivos o lançamento da plataforma Android, esta lançada sobre o código aberto na Licença Apache versão 2.0, com isto os fabricantes de dispositivos não precisam pagar taxas de licença para colocar em seus aparelhos o sistema operacional Android.

4.1. Principais Características

Segundo Steele e To (2010, tradução nossa) a plataforma possui algumas características maiores ao qual o torna um diferencial contra os demais. A seguir será descrito algumas destas importantes características do Android.

4.1.1 Multiprocesso e App Widget

Conforme explicado por Steele e To (2010, tradução nossa) o sistema operacional Android não limita o processador a um único processo por vez, sendo que faz o gerenciamento de prioridades e threads de uma aplicação. Com isto há o benefício da utilização de multitarefa, ou multiprocesso como também é conhecido, abrindo ao usuário e desenvolvedores novas possibilidades, por exemplo, durante a execução de um jogo pelo usuário, em background uma aplicação que verifica, em tempo real, o nível de estoque e dispara um alerta caso necessário. Esta característica traz o benefício de uma experiência rica aos usuários.

Os AppWidget podem ser descritos como mini aplicações, aos quais podem ser “anexadas” em outras para executar determinadas tarefas, serviços. Um exemplo é o tocador de música que fica na Home Screen do usuário. Sendo que a Home Screen é uma aplicação rodando, em sua interface há este Widget do tocador de música, ao qual concede ao usuário algumas interações, como executar, pausar, trocar de música sem a necessidade de abrir a

aplicação completa do tocador musical. Um outro exemplo é um Widget que atualiza e mostra ao usuário a temperatura ambiente, sendo atualizado seus valores por um processo em background.

4.1.2 Toque, Gestos e Multitoque

As telas sensíveis ao toque (touchscreen) estão presentes nos celulares mais modernos, tanto como se fossem um padrão. O touchscreen é uma forma intuitiva de interação com um aparelho. Sua utilização é feita de forma quase que natural, sendo que dar-se ao pressionar um dedo sobre a tela. Um exemplo é a forma prática de pressionar e arrastar para interagir com um gráfico. A função de multitoque (multitouch) dá ao usuário a possibilidade de utilizar mais de um dedo simultaneamente, sendo utilizado normalmente para zoom ou realizar a rotação de algo na tela. Ainda existem os gestos, aos quais se podem definir certos padrões de toques para executar determinadas funções [STEELE; TO, 2010, tradução nossa].

O Android vem por padrão com multitouch em sua interface, Karch (2010, tradução nossa) sugere que cabe ao desenvolvedor utilizar ou não as vantagens destes recursos em suas aplicações. Steele e To (2010, tradução nossa) salientam que muitos dos eventos gerados pelo touch são disponíveis ao desenvolvedor de forma transparente e sem a necessidade de implementações complicadas e confusas.

4.2 Arquitetura

Segundo Meier (2010, tradução nossa) a arquitetura do Android é composta por alguns componentes, melhores demonstrados pela Figura 2. Mas basicamente é composto por um Kernel em Linux e coleções de bibliotecas em C/C++ que são expostas ao framework para prover os serviços e gerenciamento das aplicações.

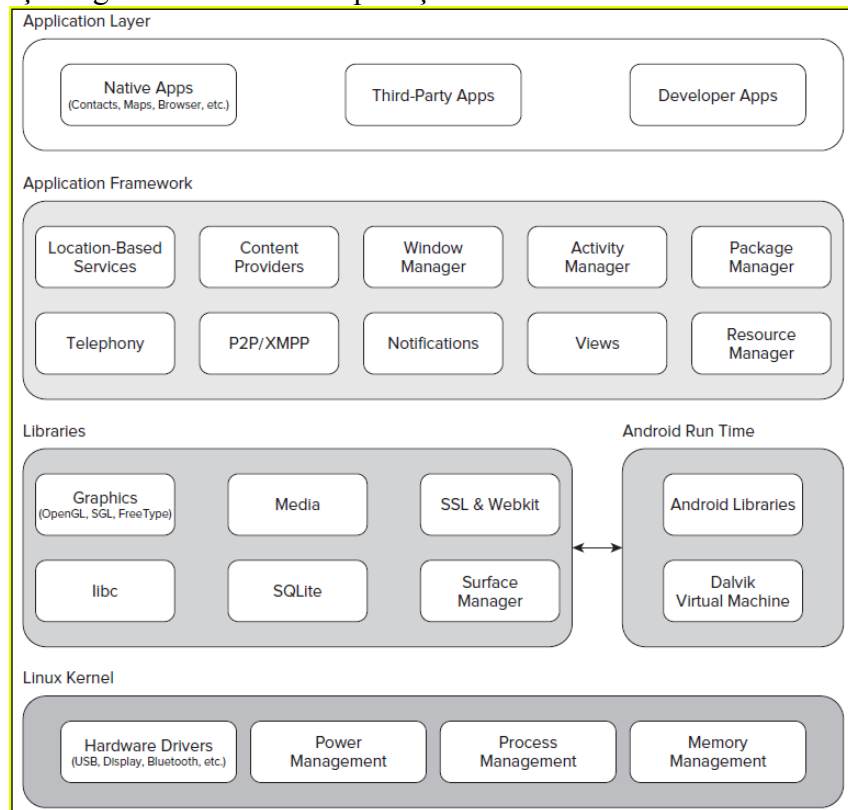


Figura 2. Arquitetura do Sistema Android

5. Acelerômetro

Segundo Felker (2010, tradução nossa), os acelerômetros, como seu nome sugere, é algo para se medir a aceleração. Conforme Meier (2010, tradução nossa), eles também podem ser referenciados de sensores de gravidade, devido ao fato de não possuir a habilidade de diferenciar entre a aceleração causada por um movimento e a gravidade. Assim sendo, quando o sensor esta em repouso, a leitura para o eixo-z (cima/baixo) será de -9.8m/s^2 .

A aceleração é definida como um valor de mudança na velocidade, portanto acelerômetros medem quão rápidos a velocidade do dispositivo mudou em uma determinada direção. Utilizando um acelerômetro podem-se detectar movimentos, e mais utilmente, o valor de mudança na velocidade deste movimento (MEIER, 2010, tradução nossa).

Um fato importante a ser destacado é que o acelerômetro não mede velocidade, portanto não é possível diretamente e baseado em apenas uma leitura determinar a velocidade. Para tal é necessário medir as mudanças na aceleração durante o tempo.

A aceleração pode ser mensurada sobre três eixos direcionais, sendo eles: esquerda-direita (horizontal), frente-atrás (longitudinal) e cima-baixo (vertical). Pode-se ver em uma melhor visualização com a Figura 6 que demonstra uma representação dos três eixos direcionais [CONDER; DARCEY; 2010, tradução nossa; MORIMOTO, 2009].

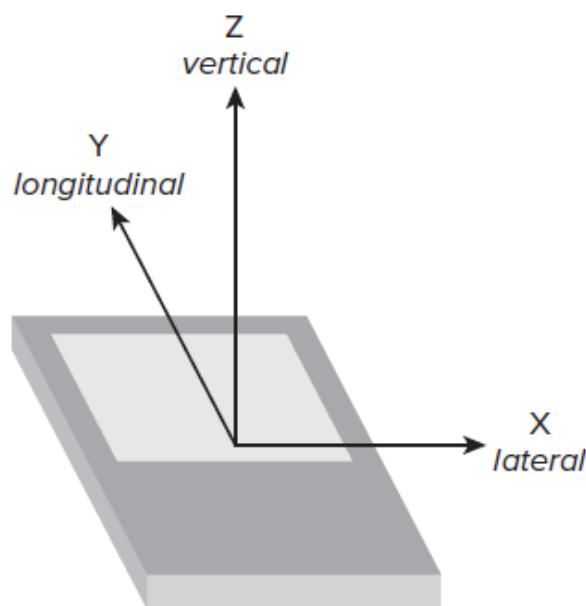


Figura 3. Representação dos três eixos direcionais

5.1 Acelerômetro e Android

Segundo Meier (2010, tradução nossa), após iniciarem com câmeras e tocadores de áudio, dispositivos móveis agora contam com sistema GPS, acelerômetros e telas sensíveis ao toque. Enquanto estas novas tecnologias abrem portas para novas possibilidades no desenvolvimento de softwares, as aplicações móveis disponíveis acabam por ficar para trás das capacidades dos hardwares.

O Android SDK contém APIs para diversos tipos de hardwares, entre eles: baseados em localização (como um GPS), a câmera, áudio, conexões de rede, Bluetooth, e dentre vários outros se encontra o Acelerômetro. Alguns destes hardwares fazem parte do conjunto de

sensores disponíveis, sendo estes listados na Tabela 1, com a sua classe de acesso e descrição [MEIER, 2010, tradução nossa].

Estes sensores são disponíveis através da classe `android.hardware.SensorManager`, ao qual funciona como o gerenciador de sensores. Há algumas classes relacionadas aos sensores, como `Sensor` representa um sensor em particular, `SensorEvent` representa as leituras captadas pelo sensor e o `SensorEventListener` que é a interface utilizada para receber os eventos gerados quase que em tempo real [ABLESON; SEN; KING, 2011, tradução nossa].

Tabela 1. Sensores comuns no Android

Classe	Descrição
<code>Sensor.TYPE_ACCELEROMETER</code>	Medidas de aceleração sobre três eixos em m/s ²
<code>Sensor.TYPE_GYROSCOPE</code>	Giroscópio que retorna a orientação do dispositivo sobre três eixos em graus
<code>Sensor.TYPE_LIGHT</code>	Sensor de luz ambiente na unidade lux
<code>Sensor.TYPE_MAGNETIC_FIELD</code>	Medidas do campo magnético em microtesla sobre os três eixos
<code>Sensor.TYPE_ORIENTATION</code>	Mede orientação do dispositivo sobre três eixos em graus
<code>Sensor.TYPE_PRESSURE</code>	Medidas de pressão que retorna um único valor em kilopascal
<code>Sensor.TYPE_PROXIMITY</code>	Mede a distância em que o dispositivo está longe de outro objeto em metros
<code>Sensor.TYPE_TEMPERATURE</code>	Mede a temperatura ambiente em graus Celsius

Segundo Ableson, Sen e King (2011, tradução nossa) e Meier (2010, tradução nossa), é possível coletar os valores dos sensores quando implementado o `SensorEventListener` e registrando-o para o `SensorManager`. Será chamado o método `onSensorChanged()` quando os valores lidos forem alterados, e chamado o método `onAccuracyChanged()` quando a precisão for alterada.

O `SensorEvent` disponibiliza alguns campos, como o `accuracy`, ao qual retorna quão preciso estão sendo as leituras. O `Sensor` que é a referência ao sensor criado pelo `SensorEvent`, o `timestamp` que retorna quando o evento ocorreu na forma de nanosegundos, e `values[3]` sendo os valores provenientes do sensor para cada eixo, sendo que podem haver variações pois há sensores que somente possuem um valor (`values[1]`). Os retornos dos valores dos sensores podem ser vistos na Tabela 2, ao qual mostra o tipo do sensor, quantos valores retorna e a composição destes valores.

Tabela 2. Valores de retorno dos Sensores

Tipo do Sensor	Quantidade de Valores	Composição dos Valores
<code>TYPE_ACCELEROMETER</code>	3	<code>value[0]</code> : Lateral <code>value[1]</code> : Longitudinal <code>value[2]</code> : Vertical
<code>TYPE_GYROSCOPE</code>	3	<code>value[0]</code> : Azimute <code>value[1]</code> : Pitch <code>value[2]</code> : Roll

TYPE_LIGHT	1	value[0] : Iluminação
TYPE_MAGNETIC_FIELD	3	value[0] : Lateral value[1] : Longitudinal value[2] : Vertical
TYPE_ORIENTATION	3	value[0] : Azimute value[1] : Pitch value[2] : Roll
TYPE_PRESSURE	1	value[0] : Pressão
TYPE_PROXIMITY	1	value[0] : Distância

6. Metodologia

Após ter realizado um estudo das tecnologias envolvidas, foi efetuado o desenvolvimento da interface responsável pelo controle de jogos eletrônicos de computadores via dispositivo móvel com sistema operacional Android, utilizando-se do sensor de acelerômetro e conectividade wireless.

Para o desenvolvimento da aplicação foi necessário alguns recursos, todos disponibilizados pelo próprio acadêmico. Recursos de hardware foram utilizados para contemplar os objetivos, sendo eles um computador com sistema operacional Windows/Linux a título de integração entre o aplicativo no dispositivo móvel e o jogo eletrônico, sendo os dois sistemas operacionais para testes em um ambiente multiplataforma; foi utilizado um adaptador Bluetooth para a comunicação via esta tecnologia; roteador Wireless colocando assim o dispositivo móvel e o computador em uma mesma rede; dispositivo móvel com sistema operacional Android e disponibilidade de sensor de acelerômetro e conexão Wi-Fi para os testes com a aplicação.

Também foram necessários alguns recursos de softwares, sendo necessários um ambiente de desenvolvimento Eclipse com a SDK Android para o desenvolvimento da aplicação para o dispositivo móvel e um ambiente de desenvolvimento NetBeans para o desenvolvimento da aplicação do computador (servidor), sendo que ambos os softwares são encontrado na Internet, nos sites dos desenvolvedores de forma gratuita. Para os jogos eletrônicos que serão realizados testes, deu-se uma priorização por jogos eletrônicos gratuitos e open source.

6.1 Modelo Conceitual

O projeto envolve em sua estrutura tecnologias e softwares diversos. Na Figura 4 é demonstrado o modelo conceitual da aplicação móvel.



Figura 4. Modelo conceitual da aplicação móvel

Como pode ser observado na Figura 4, a aplicação móvel é composta pelo Android SDK, um emulador ou celular com o Sistema Operacional Android, a aplicação desenvolvida, conexão com Rede Wireless, que no caso é através de Wi-Fi, e o computador servidor ao qual estará rodando a aplicação Server e o jogo.

A aplicação do Servidor utiliza a tecnologia Java e IDE NetBeans para seu desenvolvimento, como também utiliza algumas bibliotecas para a interpretação de XML, recurso este utilizado em parte no projeto. A aplicação server trabalha sobre a porta 50000, não sendo plausível de alteração pelo usuário, através de Sockets para a comunicação com a aplicação móvel, sendo que a cada mensagem recebida é processada e executada uma ação específica.

Na Figura 5 demonstra-se o caso de uso da aplicação móvel, nela pode-se notar as principais funcionalidades da aplicação, ao qual seriam: Sincronizar com Servidor, responsável por atualizar a lista das ações no server; Alterar Base, onde pode-se realizar alterações das informações contidas na base de dados; Importar e Exportar o Banco de Dados (Import/Export DB), que possibilita de criar backups da base já cadastrada e recuperar estas informações posteriormente; Enviar Ação, sendo esta talvez a mais importante da aplicação, pois é ela em que faz a comunicação com o servidor e envia o comando a ser executado remotamente.

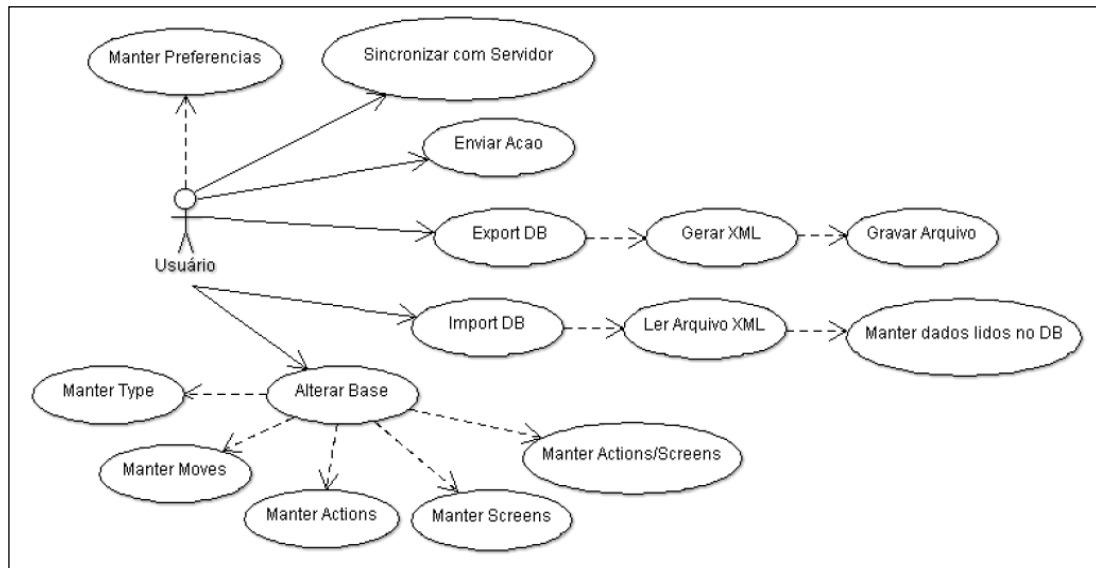


Figura 5. Diagrama de caso de uso da aplicação móvel

7. Implementação

Com a escolha das tecnologias envolvidas no projeto, e após o estudo destas, foi dado continuidade das etapas metodológicas, seguindo com a implementação da aplicação móvel e posteriormente para a aplicação do servidor.

7.1 Aplicação Móvel

O tipo de organização das classes foi baseado no modelo MVC, sendo que foi utilizada a separação das classes referentes ao Modelo, às classes de Visão foram separadas já pelo estilo de desenvolvimento da plataforma Android por meio de XMLs responsáveis pelos layouts do usuário, e a camada de Controle foi subdividida em outras três (repository, manager, e as

classes do pacote principal). A Figura 6 ilustra as classes desta etapa do desenvolvimento. A Figura 7 ilustra a tela principal da aplicação quando aberta pelo usuário.

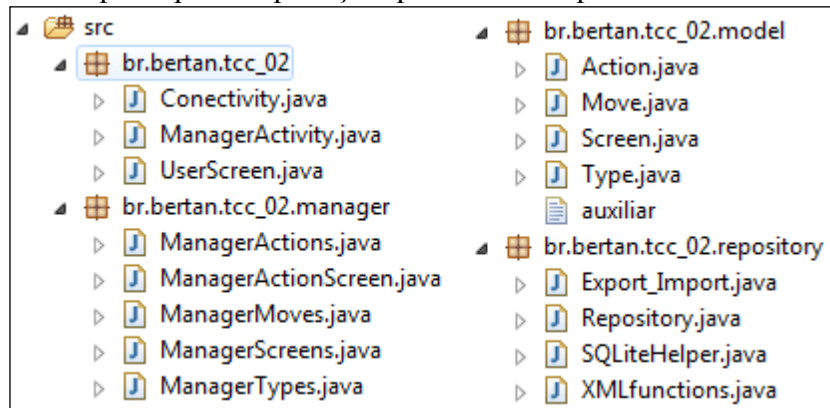


Figura 6. Classes da Aplicação Móvel

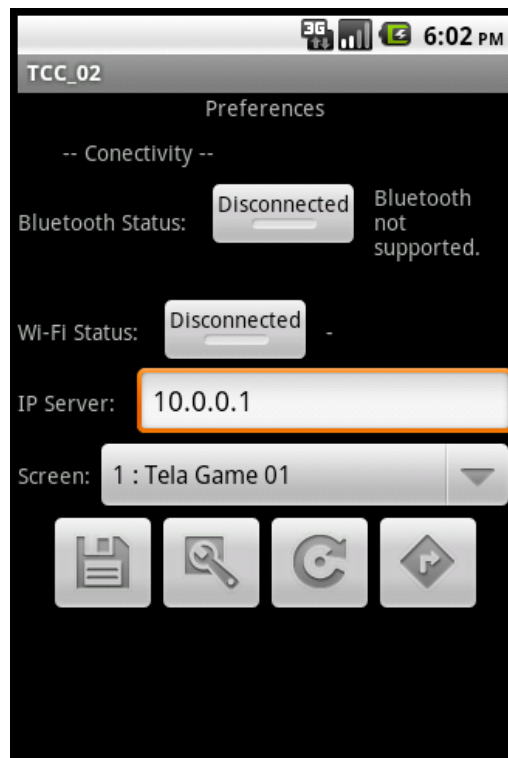


Figura 7. Tela inicial da aplicação móvel

7.2 Aplicação Desktop

Após implementado a aplicação móvel, foi dado início a aplicação desktop, ao qual é responsável por coletar os dados enviados pelo dispositivo portátil, identificar a ação solicitada e executá-la.

O desenvolvimento desta etapa também seguiu um modelo baseado no modelo MVC, porém possuindo apenas a camada Modelo em destaque, sendo que as camadas de Controle e Visão foram mescladas devido a simplicidade das ações desenvolvidas pela aplicação. A Figura 8 ilustra melhor a divisão realizada durante o desenvolvimento da aplicação. A Figura 9 ilustra a tela principal da aplicação quando aberta pelo usuário.

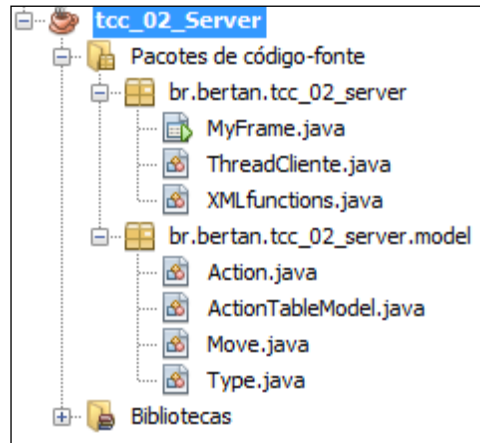


Figura 8. Classes da aplicação Desktop

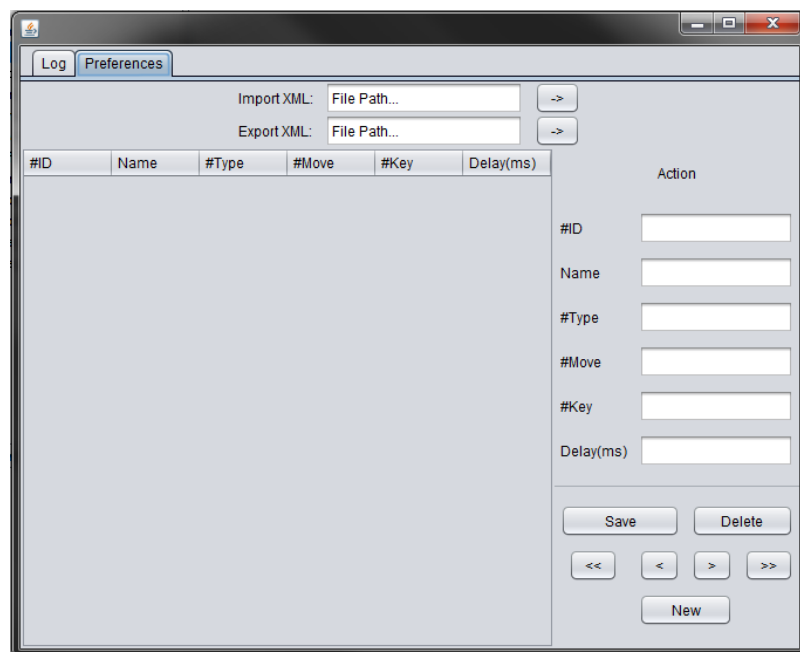


Figura 9. Tela inicial da aplicação móvel

7.2 Funcionamento

A aplicação utiliza-se do princípio de troca de mensagens entre a aplicação móvel e a servidora que foram desenvolvidas. Estas mensagens são o que possibilitam do servidor saber qual ação que fora selecionada no dispositivo móvel e assim executá-la no jogo. A implementação deu-se de forma unidirecional, assim sendo, somente a aplicação móvel envia informações e a aplicação desktop apenas recebe e interpreta, pois para o servidor, neste momento, não é necessário o envio de mensagens, sendo que seria mais útil quando há informações que poderiam ter sido processadas e retornadas a origem, ou como forma de confirmação do comando executado, sendo estes dois últimos não contemplados pelas aplicações.

O funcionamento da aplicação é de forma simples, na aplicação móvel têm-se as ações (Actions), estas que foram formadas com o auxílio dos tipos (Types) e movimentos (Moves) dentre outras configurações, e na aplicação desktop têm-se as ações (Actions). Toda a base é mantida no dispositivo móvel, decidido assim para manter a portabilidade da aplicação viável

e facilitada. Com isto, a aplicação servidora (desktop), pode ser denominada uma aplicação “oca”, ao qual possui todos os métodos necessários para a execução e interpretação das mensagens, porém não possui os dados a serem consultados (no caso as Actions), assim, se faz necessário sempre ter que realizar a sincronização, uma vez conectados, entre as duas aplicações, onde na prática o que ocorre é que as Actions armazenadas na aplicação móvel são enviadas e armazenadas em memória de execução pela aplicação servidora. Decidiu-se utilizar a sincronização para manter a escalabilidade da aplicação desktop, sendo que se pode utilizá-la com diferentes usuários móveis e em diferentes locais uma mesma aplicação.

Após ter realizado a sincronização das Actions, o usuário ao clicar em um botão de sua tela (Screen) na aplicação móvel, envia ao servidor o tipo e movimento (no formato “Type;Move”), sendo que cabe ao servidor localizar a ação correspondente e executá-la. É possível de alterar as Actions do servidor sem alterá-las na base do dispositivo móvel, para isto têm-se uma tabela visível ao usuário e plausível de seleção para posterior alteração, sendo que as alterações apenas são válidas para a aplicação servidora até que esta seja finalizada ou sobrescrevida (nova sincronização). Para alterar diretamente na base é necessário fazer-se através do dispositivo móvel.

A aplicação é instalada já com uma base padrão, e necessária, para o correto funcionamento da aplicação, porém o usuário possui a liberdade de adicionar novas ações. A base não contempla todas as teclas do teclado, assim basta selecionar um sprite para a tecla e configurá-la com o seu código referente ao seu KeyEvent do Java. A aplicação não contempla novo tipos, por exemplo, um gamepad ou “tapete de dança”, porém a aplicação suporta o cadastramento destes e a utilização das ações, sendo a limitação na aplicação do servidor, que teria que ser melhorada para reconhecer novos tipos e suas ações.

8. Resultados Obtidos

Com a pesquisa desenvolvida neste trabalho, puderam-se obter duas aplicações para plataformas distintas, ao qual quando combinadas se unem para alcançar os objetivos propostos no projeto inicial.

A primeira constitui-se da aplicação para a plataforma móvel Android. Criada com o intuito de construir uma aplicação fácil de utilizar, com telas intuitivas e práticas, utilizando-se de recursos tecnológicos avançados e aprimorados. Sua utilização ficou simplificada e possibilita variedade nas ações oferecidas, assim sendo uma forma de adaptação ao usuário que poderá utilizar e personalizar, até certo ponto, a aplicação às suas necessidades. Com o auxílio dos dados obtidos através de leituras sobre os seus sensores, principalmente o acelerômetro, é possível identificar movimentos realizados pelo usuário, mesmo que até o momento implementado o básico (cima, baixo, direita e esquerda), já é o suficiente para apreciar o quanto esta tecnologia fornece de possibilidades.

A segunda aplicação desenvolveu-se para a plataforma Java para Desktop, sendo esta dotada de uma interface simples, porém que possibilita ajustes imediatos, se necessários. Também traz ao usuário um retorno das ações executadas através da tela de log, assim o usuário saberá o que está acontecendo em sua máquina. Internamente esta aplicação trabalha em conjunto com a aplicação móvel, recebendo a tabela de ações personalizadas do usuário, e as ações a serem executadas.

Esta harmonia entre as duas aplicações propicia atingir o objetivo macro, a utilização de dispositivos móveis na forma de um joystick para jogos eletrônicos, fazendo uso do sensor de acelerômetro. A realização desta pesquisa proporcionou a descrição tanto da área de jogos eletrônicos, em grande expansão e que já foi muito explorado, porém que com a outra parte da

pesquisa, acelerômetro, sensores e dispositivos móveis, abrem mais um nicho que possibilita novas explorações na área, revolução na forma de interação.

9. Conclusão

Através de toda esta gama de conhecimentos adquiridos, tornou-se viável alcançar o objetivo principal desta pesquisa que consiste na utilização de dispositivos móveis na forma de um joystick para jogos eletrônicos, fazendo uso do sensor de acelerômetro.

O resultado da pesquisa se mostrou favorável e com sucesso foram atingidos todos os objetivos almejados, tanto o geral quanto os específicos. Sendo que estes resultados consistem-se, além do conhecimento teórico, de duas aplicações distintas, a aplicação móvel, desenvolvida unicamente para a plataforma Android, que utiliza-se de conectividade sem fio para conexão com o computador servidor e o uso de acelerômetro para captura de movimentos do usuário; e a outra aplicação, desenvolvida sobre a linguagem Java, tornando a aplicação multiplataforma, com o propósito de servidor para os dados recebidos da aplicação móvel, têm a função de interpretar estas informações e executá-las no computador remoto o que o usuário solicitou através do dispositivo móvel, focando o seu uso em jogos eletrônicos.

Poucas foram as dificuldades encontradas. Os materiais de estudo proporcionaram um entendimento prático e funcional das ferramentas utilizadas, porém não foi possível a implementação da comunicação entre a aplicação móvel com o servidor através da tecnologia Bluetooth. Sendo que para o ambiente Android a manipulação de conexões e canais de transmissão (recepção e envio) de dados são de fácil manipulação, porém para a implementação no servidor esta tecnologia ainda está imatura e com alguns problemas de compatibilidade com adaptadores Bluetooth. A implementação utilizando-se de Java e Bluetooth em ambiente desktop não é impossível, porém para a pesquisa não é relevante este canal de comunicação visando que o foco é a utilização do acelerômetro, sendo este alcançado junto com a comunicação via Wi-Fi.

O acelerômetro proporciona grandes possibilidades de uso, para movimentos básicos, como os utilizados nesta pesquisa, é de fácil implementação e manipulação dos dados, porém para uma aplicação em que se deseja coletar movimentos elaborados do usuário, requer uma complexidade alta tanto do processamento das informações provindas das leituras como do algoritmo que o fará. O seu uso pode ser ampliado não somente para a área de jogos eletrônicos, como também para as diversas áreas, podem-se utilizá-lo, por exemplo, para o controle de robô em ambientes de riscos, e um uso atual é nos discos rígidos de computadores que podem detectar quedas e assim travar as cabeças de gravação para não danificarem os dados do usuário.

Referências

- Ableson, W. Frank; Sen, Robi; King, Chris. Android in Action. Second Edition [S. l.]: Manning Publications, 2011.
- Conder, Shane; Darcey, Lauren. Android Wireless Application Development: Developer's Library. Second Edition. [S. l.]: Addison Wesley. 2010.
- Dawson, Michael. Beginning C++ Through Game Programming. Third Edition. Boston: Course Technology PTR, 2010.
- Dimarzio, J. F.; Polo, G. L. Android, A Programmer's Guide. [S. l.]: McGraw-Hill Osborne Media, 2008.

- Felker, Donn. Android Application Development For Dummies. [S. l.]: For Dummies, 2010.
- Karch, Marziah. Android for Work: Productivity for Professionals. [S. l.]: Apress, 2010.
- Lecheta, Ricardo R. Google Android. 2 ed. São Paulo: Novatec, 2010.
- Mallick, Martyn. Mobile and Wireless Design Essentials. New Jersey: John Wiley & Sons, 2003.
- Meier, Reto. Professional Android™ 2 Application Development. New York: Wiley Publishing, 2010.
- Morimoto, Carlos E.. Smartphones: Guia Prático. [S. l.]: Gdh Press e Sul Editores, 2009.
- Morrison, Michael. Sams Teach Yourself Game Programming in 24 Hours. Indianapolis: Sams Publishing, 2002.
- Muller, Nathan J. Bluetooth Demystified. New York: McGraw-Hill Professional Publishing, 2000.
- Steele, James; To, Nelson. The Android Developer's Cookbook: Building Applications with the Android SDK. New York: Addison-Wesley Professional, 2010.