

Maurício César Pinto Pessoa

Home e-Care: Uma ferramenta para o monitoramento de quedas em pessoas da terceira idade através de *Smartwatches*

São Luís - MA

2016

Maurício César Pinto Pessoa

Home e-Care: Uma ferramenta para o monitoramento de quedas em pessoas da terceira idade através de *Smartwatches*

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Universidade Federal do Maranhão
Centro de Ciências Exatas e Tecnológicas
Curso de Ciência da Computação

Orientador: Prof. Dr. Geraldo Braz Junior

São Luís - MA

2016

Pessoa, Maurício César Pinto.

Home e-care: uma ferramenta para o monitoramento de quedas em pessoas da terceira idade através de smartwatches / Maurício César Pinto Pessoa. – São Luís, 2016.

53 f.

Orientador: Geraldo Braz Junior.

Monografia (Graduação) – Universidade Federal do Maranhão, Curso de Ciência da Computação, 2016.

1. Computação vestível. 2. Detecção de queda. 3. Android. 4. Smartwatches. I. Título.

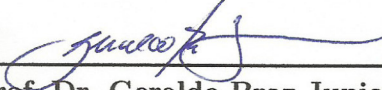
CDU 004.5

Maurício César Pinto Pessoa

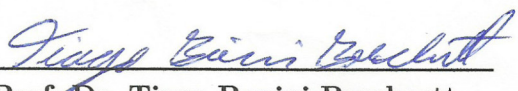
Home e-Care: Uma ferramenta para o monitoramento de quedas em pessoas da terceira idade através de *Smartwatches*

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

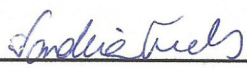
Trabalho aprovado. Nota 10.0 (Dez). São Luís - MA, 11 de Março de 2016:



Prof. Dr. Geraldo Braz Junior
Orientador
Universidade Federal do Maranhão



Prof. Dr. Tiago Bonini Borchardt
Examinador 1
Universidade Federal do Maranhão



Prof. Msc. Vandecia Rejane Monteiro Fernandes
Examinador 2
Universidade Federal do Maranhão

São Luís - MA
2016

À minha avó Rosa, que inspirou este trabalho.

Agradecimentos

Agradeço a Deus, por me dar saúde e a força necessária para superar os desafios da construção desse trabalho, além de colocar na minha vida todos as pessoas mencionadas abaixo.

Agradeço aos meus pais, Gerisval Alves Pessoa e Conceição de Maria Pinto Silva Pessoa, que me criaram e me educaram do menino ao homem, e por todo o amor, carinho e conhecimento recebido no passar dos anos.

Ao meu irmão, Alexandre, pelas brincadeiras e momentos de cobrança. E pelas conversas que ajudaram a melhorar o trabalho.

Agradeço à Amanda, minha querida namorada e companheira, pela paciência, amor e carinho, e por sempre estar presente e me apoiar antes e durante o desenvolvimento desse trabalho.

Ao meu orientador, Geraldo Braz Junior, que com sua experiência e conhecimento me ensinou o caminho das pedras, ajudando a tornar esse trabalho muito mais rico e bem escrito.

Agradeço meus amigos Hedvan Fernandes e Ruy Oliveira, que muito me ajudaram com pequenas revisões a escrita da monografia. E ao meu amigo de longa data, Victor Simonetti, que me ajudou a melhor formular a ideia desse trabalho.

Agradeço aos demais amigos da UFMA, que em várias conversas de corredor foram me dando novas ideias para incrementar esse trabalho.

Agradeço aos meus amigos e companheiros programadores Android do chat Android do StackOverflow, ou simplesmente *room-15*. Em especial o Adam McNeilly, Eric Cugota e David Breneisen por me ajudarem com as ideias iniciais do projeto e com os problemas no código do *Home e-Care*.

Agradeço ao Google e ao StackOverflow, pela quase infinita fonte de conhecimento que essas ferramentas provêm.

Por fim, agradeço a todos os demais que de alguma forma contribuíram para que esse trabalho se tornasse realidade.

*“Ciência da Computação está tão relacionada aos computadores,
quanto a Astronomia aos telescópios.
A Ciência não estuda ferramentas.
Ela estuda como nós as utilizamos, e o que descobrimos com elas.”
(Edsger Dijkstra)*

Resumo

Com o envelhecimento da população mundial, questões sobre a qualidade de vida dessas pessoas assumem grande importância. Como essas pessoas normalmente possuem a saúde mais frágil, elas ficam cada vez mais dependentes da ajuda de terceiros na medida em que vão envelhecendo, principalmente na ocorrência de acidentes como quedas. Por esse motivo, torna-se necessário a existência de métodos automatizados de monitoramento e notificação da ocorrência de tais acidentes. É possível utilizar os sensores presentes nos *smartwatches*, como o acelerômetro, para monitorar as atividades realizadas por uma pessoa, e com base nas informações coletadas, aplica-se um algoritmo classificador para determinar a ocorrência de eventos de queda. Os algoritmos baseados em limiares costumam ser utilizados como classificadores, por serem robustos, possuírem baixa complexidade e um impacto energético minimizado. Nesse contexto, foi desenvolvido o *Home e-Care*, uma aplicação para *Android Wear* que monitora as atividades de seus usuários e notifica os contatos de emergência na ocorrência de quedas. Posteriormente, foram realizados testes de eficiência, e concluiu-se que o sistema desenvolvido e os algoritmos baseados em limiares caracterizam uma alternativa robusta e viável para a detecção automatizada de quedas.

Palavras-chaves: Detecção de quedas. Android. Smartwatches. Computação vestível.

Abstract

The elderly population is growing worldwide, and with that the concerns about their quality of life become more important. As the elderly normally have frail health, which tends to get worse as they age, they grow even more dependent on someone else, specifically in the event of an accident such as a fall. For that reason, an automated monitoring and notification mechanism for accidents is necessary. The activities of daily living (ADL) can be monitored using sensors embedded in a smartwatch, such as an accelerometer. A classification algorithm can be implemented to detect fall events using the sensors data. Threshold based algorithms (TBA) are normally used as classifiers, since they are simple, robust, and have minimal energy impact. On this context, an Android Wear application called *Home e-Care* was developed to monitor the ADL of users, detect the occurrence of falls, and notify emergency contacts when they happen. Next, some tests were made to verify the application efficiency, and it was concluded that *Home e-Care* and the TBA are a robust and viable solution for automated fall detection.

Keywords: Fall detection. Android. Smartwatches. Wearable computing.

Lista de ilustrações

Figura 1 – Projeção da pirâmide populacional do Brasil para 2050.	14
Figura 2 – Visão geral do Android.	19
Figura 3 – Camada do Kernel/HAL	20
Figura 4 – Camada de bibliotecas	20
Figura 5 – Camada de ambiente de execução	21
Figura 6 – Camada de estrutura de aplicativos	22
Figura 7 – Camada de aplicativos	22
Figura 8 – Cartões no <i>Android Wear</i>	23
Figura 9 – Comunicação no <i>Android Wear</i>	24
Figura 10 – Exemplo de conexão do <i>smartphone</i> com múltiplos nós.	25
Figura 11 – Abstração de um acelerômetro.	26
Figura 12 – Acelerômetro MEMS visto em um microscópio.	26
Figura 13 – Funcionamento de um acelerômetro MEMS.	27
Figura 14 – Botão de emergência do sistema <i>ERICA</i>	28
Figura 15 – Captura de tela do <i>iFall</i>	29
Figura 16 – <i>F2D</i> rodando em diversos dispositivos.	30
Figura 17 – Arquitetura básica de sistemas de detecção de quedas.	31
Figura 18 – Visão geral do Home e-Care.	33
Figura 19 – Módulos do <i>Home e-Care</i>	34
Figura 20 – Gráfico de magnitude da aceleração durante uma queda.	36
Figura 21 – Algoritmo de detecção de quedas utilizado no <i>Home e-Care</i>	38
Figura 22 – Capturas de tela do <i>smartwatch</i>	39
Figura 23 – Algoritmo de detecção de quedas estendido do <i>Home e-Care</i>	40
Figura 24 – Fluxo de dados do módulo do <i>smartphone</i>	41
Figura 25 – Fluxo de dados do módulo <i>web</i>	42
Figura 26 – Alerta recebido através do Twitter.	42
Figura 27 – Resultado dos testes de ADL.	45
Figura 28 – Resultado dos testes de queda.	46

Lista de tabelas

Tabela 1 – Versões do Android	19
Tabela 2 – Parâmetros de configuração do algoritmo de detecção de quedas.	43
Tabela 3 – Matriz de confusão dos experimentos realizados.	45

Lista de abreviaturas e siglas

API	Application Programming Interface - Interface de Programação de Aplicações
HTTP	Hypertext Transfer Protocol - Protocolo de Transferência de Hipertexto
SMS	Short Message Service - Serviço de Mensagens Curtas
WEKA	Waikato Environment for Knowledge Analysis - Ambiente Waikato para Análise de Conhecimento
OMS	Organização Mundial da Saúde
GPS	Global Positioning Systems - Sistema de Posicionamento Global
TA	Tecnologia Assistiva
PERS	Personal Emergency Response System - Sistema Pessoal de Atendimento Emergencial
OHA	Open Handset Alliance
HAL	Hardware Abstraction Layer - Camada de Abstração de Hardware
VM	Virtual Machine - Máquina virtual
SO	Sistema Operacional
SDK	Source Development Kit - Kit de Desenvolvimento de Software/Aplicativos
IDE	Integrated Development Environment - Ambiente de Desenvolvimento Integrado
NDK	Native Development Kit - Kit de Desenvolvimento Nativo
MEMS	Microelectromechanical systems - Sistemas Microeletromecânicos
TBA	Threshold Based Algorithm - Algoritmo Baseado em Limiares
ADL	Activities of Daily Living - Atividades da Vida Cotidiana
SVM	Support Vector Machine - Máquina de vetores de suporte

Lista de símbolos

m/s^2 metro por segundo ao quadrado

ms milissegundo

s segundo

Sumário

1	INTRODUÇÃO	14
1.1	Objetivos	16
1.2	Estrutura	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Tecnologia Assistiva	17
2.1.1	Sistema Pessoal de Atendimento Emergencial (PERS)	17
2.2	Paradigma de Computação Móvel usando Android	18
2.2.1	Arquitetura Android	18
2.2.2	Android Wear	23
2.3	Captura de Movimentos com Acelerômetros	25
3	TRABALHOS RELACIONADOS	28
3.1	ERICA	28
3.2	iFall	29
3.3	F2D	30
3.4	Metodologias de Detecção de Quedas com Classificador	30
3.5	Arquitetura Básica de Metodologias de Detecção de Quedas	31
4	HOME E-CARE: METODOLOGIA E IMPLEMENTAÇÃO	33
4.1	Visão Geral	33
4.2	Módulo de Detecção de Quedas	35
4.3	Módulo do Smartphone	40
4.4	Módulo Web	41
5	RESULTADOS	43
5.1	Experimentos Realizados	43
5.2	Análise dos Resultados Obtidos	44
6	CONCLUSÃO	48
	REFERÊNCIAS	50

1 Introdução

A população mundial está envelhecendo, e o número total de idosos vai dobrar até 2050, sendo que no Brasil esse número deverá triplicar. Segundo relatos da Organização Mundial da Saúde (OMS), pela primeira vez na história da humanidade a população de idosos no mundo irá superar a população de até 15 anos, inclusive no Brasil, conforme a pirâmide populacional da Figura 1 (OMS, 2015a).

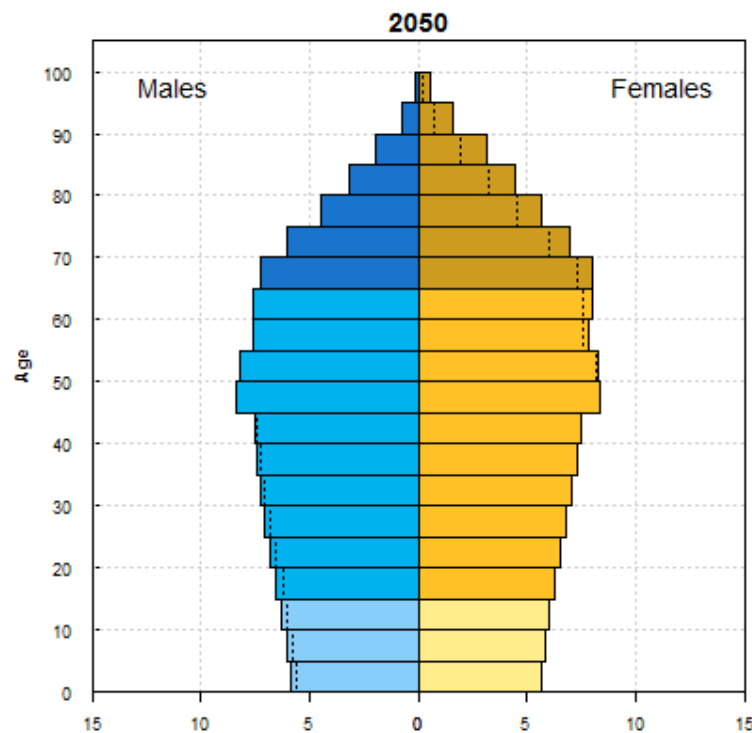


Figura 1 – Projeção da pirâmide populacional do Brasil para 2050.

Fonte: (ONU, 2015)

Pessoas acima de 60 anos normalmente possuem as condições físicas mais debilitadas além de outras possíveis complicações de saúde oriundas da idade, tornando-os cada vez mais dependentes da ajuda de terceiros na medida em que vão envelhecendo (OMS, 2015b).

O problema é que nem sempre o idoso poderá contar com a ajuda de outra pessoa o tempo inteiro. Essa é a preocupação de muitas famílias que possuem os parentes idosos, pois eles precisam trabalhar e resolver os seus próprios problemas, enquanto os seus relativos ficam sozinhos em suas residências, suscetíveis aos mais diversos tipos de acidentes domésticos, como quedas não intencionais.

Se torna perigoso para um idoso permanecer muito tempo sozinho, ou até mesmo morar longe de outras pessoas, devido ao risco de acidentes domésticos, principalmente

pela demora no atendimento, pois muitas das vezes ele pode se encontrar debilitado demais para pedir ajuda.

A demora para ser atendido costuma ser mais perigosa que o acidente por si só, pois as quedas podem resultar em diversas fraturas e lesões temporárias ou permanentes, além do fator psicológico onde o idoso se torna cada vez mais dependente de terceiros para realizar suas atividades rotineiras, devido ao medo de recorrência dos acidentes.

Os estudos conduzidos por [Wild, Nayak e Isaacs \(1981\)](#) indicam que 50% dos idosos que permanecem no chão após uma queda por um período de tempo superior a uma hora acabam falecendo dentro de um período de 6 meses após o acidente.

Segundo a [OMS \(2007\)](#), a cada ano, aproximadamente 28% a 35% das pessoas acima de 65 anos sofrem ao menos uma queda e esse número sobe para 32% a 42% nas pessoas que tem mais de 70 anos. E como em muitos casos o idoso pode se encontrar sozinho, aumentam as chances do acidente evoluir para algo mais grave, ou potencialmente fatal.

Para sanar esse problema, se torna necessário o uso de um Sistema Pessoal de Atendimento Emergencial, do inglês *Personal Emergency Response System* (PERS), para notificar de forma imediata parentes ou responsáveis pelo idoso na ocorrência de acidentes, com o objetivo de agilizar o resgate e minimizar os riscos decorrentes da demora, ou da completa falta de atendimento.

Tal notificação pode ser gerada através de softwares desenvolvidos para dispositivos como *smartphones* e *smartwatches*, que contam com aproximadamente um bilhão de usuários no mundo em 2015, sendo que somente no Brasil são 48,6 milhões de usuários, e esse número só tende a aumentar nos próximos anos ([EMARKETER, 2014](#)).

O mercado de *smartwatches* já conta com vendas nas casas dos milhões de unidades, e segundo a *International Data Corporation* (IDC), esse número deve alcançar 173 milhões de unidades vendidas em 2019 ([IDC, 2015](#)), o que torna essa combinação uma boa escolha para realizar um estudo de caso da automação de detecção e notificação de situações de risco.

É possível detectar quedas através do uso de um acelerômetro de 3 eixos, conforme foi provado em diversos estudos ([PANAGIOTIS et al., 2014](#); [KAZI et al., 2014](#); [LUQUE et al., 2014](#)). Se tal aplicação for criada, além de uma forma de comunicar os contatos de emergência no caso de algum acidente, é possível agilizar o atendimento de emergência para o idoso.

1.1 Objetivos

O objetivo desse trabalho é testar e validar algoritmos de detecção de queda, além de desenvolver o *Home e-Care*, um sistema que fica responsável por detectar e notificar a ocorrência de quedas através do uso do acelerômetro encontrados em *smartwatches* e da conexão *web* presente nos *smartphones*. As notificações não irão requerer nenhuma ação por conta da pessoa acidentada. O *Home e-Care*, portanto, deverá agilizar o atendimento médico que o idoso necessitar após a ocorrência de uma queda.

Os objetivos específicos deste trabalho estão enumerados a seguir:

- Analisar a API de comunicação do *Android Wear*;
- Fazer um levantamento de técnicas e algoritmos de detecção de quedas utilizando acelerômetros;
- Coletar os dados de aceleração fornecidos pelo *smartwatch*;
- Realizar experimentos com os dados de aceleração e exibir os resultados obtidos, verificando a eficiência do *Home e-Care*.

1.2 Estrutura

Esse trabalho está dividido em 6 capítulos. No Capítulo 1 é feita a apresentação do problema que motivou este trabalho, além dos seus objetivos gerais e específicos. No Capítulo 2 é apresentada a fundamentação teórica, com um levantamento de tópicos relevantes à resolução do problema apresentado. O Capítulo 3 mostra ferramentas e estudos relacionados a este trabalho. O Capítulo 4 detalha a metodologia proposta neste trabalho, além de levantar detalhes da implementação do *Home e-Care*. O Capítulo 5 descreve os testes realizados com o algoritmo do *Home e-Care* e analisa os resultados obtidos. Por fim, no Capítulo 6 são apresentadas as conclusões e contribuições realizadas, além de propostas para trabalhos futuros.

2 Fundamentação Teórica

Neste capítulo apresentam-se os termos, tecnologias e conceitos que são utilizados no decorrer deste trabalho. Inicialmente definem-se os conceitos de Tecnologia Assistiva (TA) e de Sistemas Pessoais de Atendimento Emergencial, além de demonstrar como elas são utilizadas para auxiliar a vida dos idosos. Posteriormente são apresentados o sistema operacional móvel Android e *Android Wear*, utilizados na implementação de referência deste trabalho, definindo suas arquiteturas e funcionamento.

2.1 Tecnologia Assistiva

Conforme o Comitê de Ajudas Técnicas (CAT) da Secretaria Especial dos Direitos Humanos, a Tecnologia Assistiva (TA) é definida como:

Tecnologia Assistiva é uma área do conhecimento, de característica interdisciplinar, que engloba produtos, recursos, metodologias, estratégias, práticas e serviços que objetivam promover a funcionalidade, relacionada à atividade e participação, de pessoas com deficiência, incapacidades ou mobilidade reduzida, visando sua autonomia, independência, qualidade de vida e inclusão social (CAT, 2009).

Ou seja, pode-se dizer que a TA engloba qualquer recurso, computacional ou não, que visa facilitar a vida das pessoas que possuem qualquer tipo de deficiência, como a mobilidade reduzida dos idosos.

2.1.1 Sistema Pessoal de Atendimento Emergencial (PERS)

Um Sistema Pessoal de Atendimento Emergencial, do inglês *Personal Emergency Response System* (PERS), é um sistema eletrônico de alarme pessoal responsável por detectar a presença de situações de risco para a saúde de seu usuário. Quando uma situação de emergência for detectada, esse sistema deve alertar os contatos de emergência do usuário, solicitando assistência médica imediata (EDLICH et al., 1992).

Segundo os estudos realizados por Heinbüchner et al. (2010) e Porter (2003), os idosos que utilizam um PERS sentem-se mais seguros e confiantes para realizarem as tarefas cotidianas, pois eles possuem uma garantia de assistência médica no caso de algum acidente doméstico ou emergência médica. Portanto, o uso de um PERS auxilia o idoso a viver a sua vida de forma autônoma e independente.

2.2 Paradigma de Computação Móvel usando Android

O Android é um sistema operacional (SO) móvel baseado no núcleo do Linux e foi criado pela Google em conjunto com a Open Handset Alliance (OHA), um grupo composto por grandes empresas do ramo de tecnologia e telefonia móvel (OHA, 2007).

Esse sistema foi projetado para rodar em dispositivos móveis com telas de toque e normalmente esse SO é visto rodando em *smartphones* e *tablets*, mas não se limita somente a estes dispositivos, podendo também ser encontrado em relógios inteligentes (*smartwatches*), televisões, videogames, carros, entre outros. (LEAL, 2015).

A Google disponibiliza gratuitamente um kit de desenvolvimento (SDK) para aplicações Android, contendo diversas ferramentas de desenvolvimento e depuração, APIs, um emulador de dispositivos Android e um ambiente integrado de desenvolvimento (IDE) conhecido como *Android Studio*.

A linguagem de programação mais utilizada para o desenvolvimento dessas aplicações é a Java (LEAL, 2015), porém também é possível desenvolver utilizando outras linguagens como C ou C++ através do Kit de Desenvolvimento Nativo (NDK) também fornecido pela empresa (GUIDE, 2016).

A plataforma Android possui o seu código-fonte aberto, disponibilizado sob a licença Apache, o que possibilita para qualquer pessoa ou empresa criar sua própria versão do Android sem necessariamente ter que disponibilizar as mudanças com terceiros.

Efetivamente, cada fabricante tem a liberdade de criar um Android diferente dos que rodam nos dispositivos de seus concorrentes (LEAL, 2015), além de possibilitar o surgimento de versões customizadas do SO como o *CyanogenMod*, que oferecem um conjunto de funcionalidades maior do que as versões oficiais dos fabricantes (COMMUNITY, 2015).

O Android é o SO móvel mais utilizado no mundo, no final do segundo semestre de 2015 ele possuía 82,8% da fatia de mercado global de dispositivos móveis (IDC, 2016). Ele teve sua primeira versão lançada em 2008 e desde então o sistema já passou por diversas atualizações, adicionando novas funcionalidades e melhorias, além de corrigir erros das versões mais antigas. A Tabela 1 apresenta as versões mais utilizadas e seus respectivos nomes, níveis de API (ou número da versão) e distribuição de mercado.

2.2.1 Arquitetura Android

O Android pode ser dividido em cinco camadas de software trabalhando em conjunto, cada uma rodando em um nível de abstração diferente, conforme ilustrado pela Figura 2. Essas camadas estão divididas em: Kernel Linux, camada de abstração de hardware, bibliotecas, ambiente de execução, estrutura de aplicativos (framework) e a camada dos aplicativos (LEE, 2011).

Tabela 1 – Versões do Android

Versão	Nome da versão	Nível de API	Distribuição
2.2	Froyo	8	0,1%
2.3.3 - 2.3.7	Gingerbread	10	2,7%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2,5%
4.1.x		16	8,8%
4.2.x	Jelly Bean	17	11,7%
4.3		18	3,4%
4.4	KitKat	19	35,5%
5.0	Lollipop	21	17,0%
5.1		22	17,1%
6.0	Marshmallow	23	1,2%

Fonte: Traduzido de (ANDROID, 2016e)

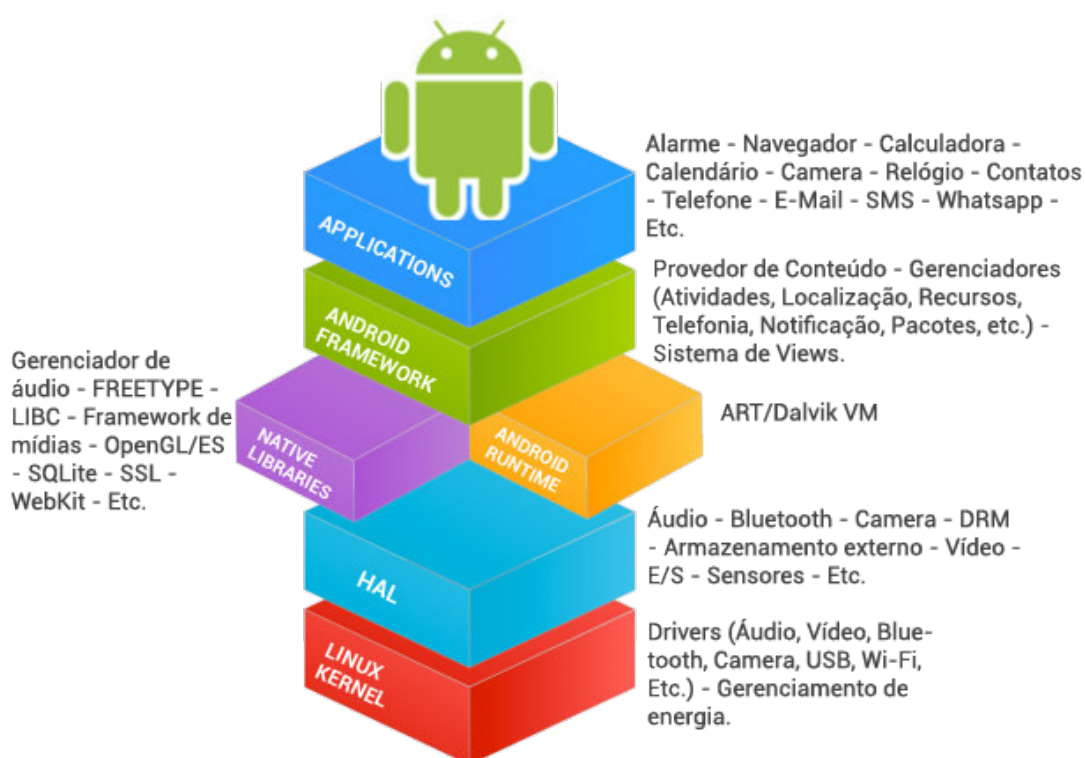


Figura 2 – Visão geral do Android.

Fonte: Traduzido de (ANDROID, 2016a).

A camada kernel Linux é o núcleo do sistema operacional Android. Essa é a camada responsável por todas os serviços de baixo nível da plataforma e contém os drivers dos diversos dispositivos de hardware presentes nos *smartphones*.

Essa camada trabalha em conjunto com a camada de abstração de hardware (HAL), conforme ilustrado pela Figura 3, provendo uma interface entre os dispositivos de hardware e as aplicações Android. O núcleo do Android também é responsável pelo gerenciamento de energia, memória, processos, redes, entre outros.

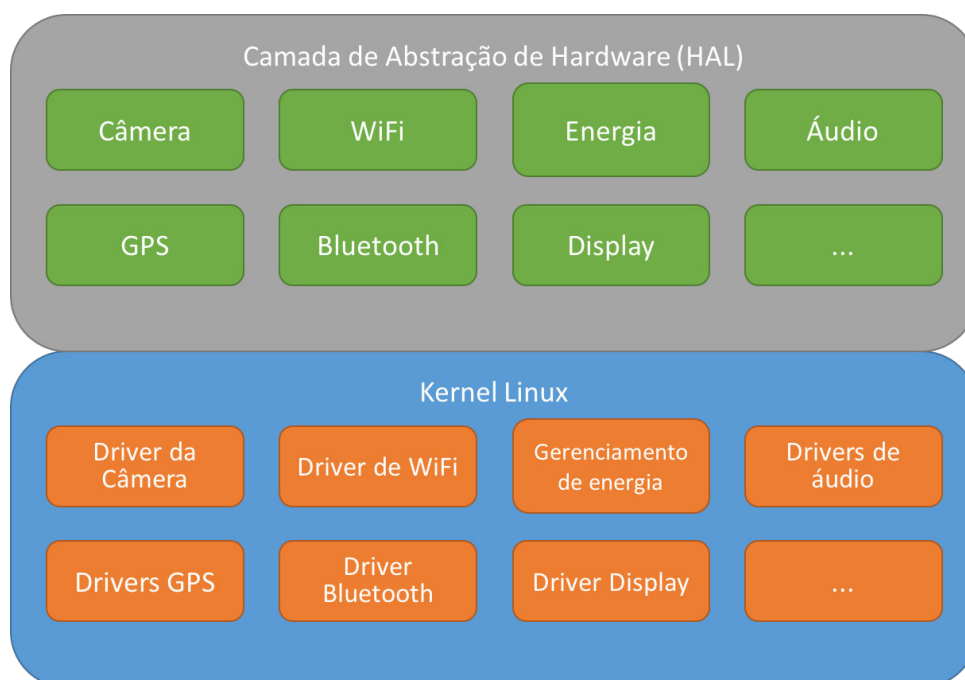


Figura 3 – Camada do Kernel/HAL

Fonte: Acervo do autor.

A camada de bibliotecas, ilustrada pela Figura 4, consiste em um conjunto de bibliotecas, normalmente escritas em C/C++, que provêm diversas funcionalidades para o sistema operacional.



Figura 4 – Camada de bibliotecas

Fonte: Acervo do autor.

Entre elas, cita-se como exemplos: SQLite, que fornece o suporte para bases de dados; WebKit, que fornece funcionalidades de navegação web; OpenGL que fornece funcionalidades para gráficos 3D; Entre outras. Todas essas bibliotecas podem ser acessadas através de frameworks disponibilizados para os desenvolvedores Android.

A camada do ambiente de execução (*runtime*), ilustrada pela Figura 5, é constituída de várias bibliotecas centrais que possibilitam o desenvolvimento de aplicações para Android

utilizando a linguagem de programação Java, além de incluir uma máquina virtual (VM) conhecida como Dalvik, para Androids até a versão 4.4, ou ART para Androids a partir da versão 5.0.

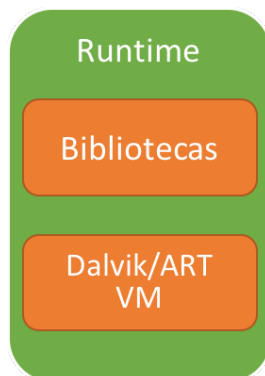


Figura 5 – Camada de ambiente de execução

Fonte: Acervo do autor.

Essas duas máquinas virtuais foram criadas especialmente para a plataforma Android, de forma a otimizar o uso de recursos e energia em ambientes com capacidade de processamento mais limitado e onde o uso de bateria é um fator importante.

Para as aplicações poderem ser executadas na VM, os programas escritos em Java são convertidos em um código de máquina conhecido como *Java Bytecode* e posteriormente em *Dalvik Bytecode* para então serem executados. Cada aplicação roda em um processo separado no sistema operacional, dentro de uma instância da máquina virtual Dalvik/ART (ANDROID, 2016b; LEE, 2011).

A camada de estrutura de aplicativos (*Application Framework*), ilustrada pela Figura 6, expõe para os desenvolvedores todas as APIs e recursos em forma de classes Java, que podem ser utilizadas pelos aplicativos Android, tais como (LEE, 2011; SMITH; FRIESEN, 2011):

- Gerenciador de atividades: Componente responsável por gerenciar o ciclo de vida das aplicações Android;
- Provedores de conteúdo: Componentes responsáveis por encapsular informações das aplicações Android e torná-las disponíveis para outras aplicações, viabilizando uma fácil comunicação entre aplicações;
- Gerenciador de localização: Componente que permite uma aplicação Android acessar a localização física do dispositivo na qual ela está rodando;
- Gerenciador de pacotes: Esse componente é responsável por manter informações sobre as aplicações instaladas no sistema e permite

- Gerenciador de telefonia: Permite às aplicações Android saber das capacidades de telefonia do dispositivo móvel, além de permitir a realização e recebimento de chamadas telefônicas;
- Gerenciador de sensores: Provê o acesso aos mais diversos sensores disponíveis no dispositivo Android, como o acelerômetro, giroscópio, magnetômetro, etc.;
- Gerenciador de Views: Esse componente disponibiliza diversos componentes gráficos reutilizáveis para as aplicações Android, como botões, listas, layouts, etc.

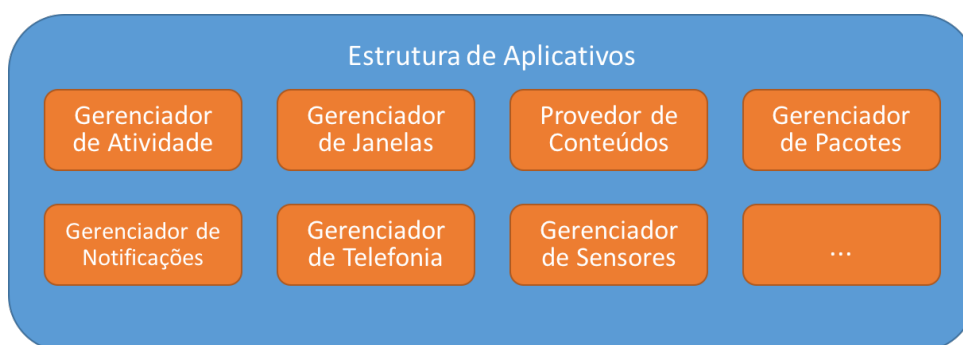


Figura 6 – Camada de estrutura de aplicativos

Fonte: Acervo do autor.

A camada de aplicativos representa todas as aplicações de um dispositivo Android, como agenda de contatos, SMS, navegador web, telefone, calendários, jogos, etc., conforme ilustrado pela Figura 7.

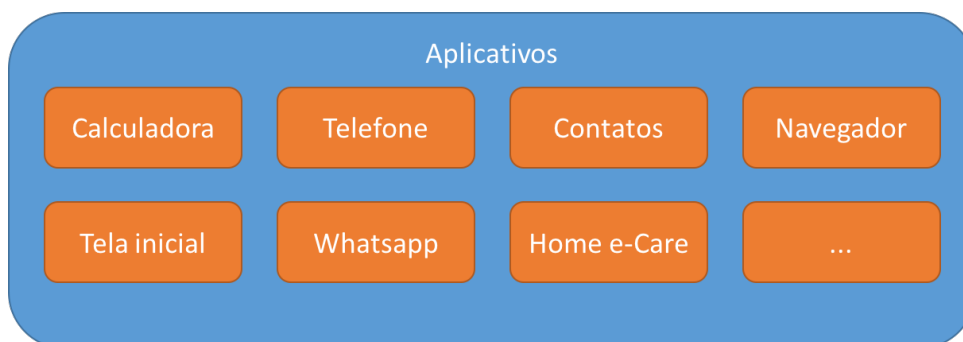


Figura 7 – Camada de aplicativos

Fonte: Acervo do autor.

Essa camada inclui tanto as aplicações que já vem embarcadas de fábrica no dispositivo, quanto as instaladas pela loja virtual *Play Store*, ou através de fontes terceiras. Essa camada não faz distinção entre aplicações nativas e de desenvolvedores terceiros, ou seja, independente da fonte, as aplicações ficam localizadas na camada de aplicativos.

2.2.2 Android Wear

Android Wear é uma versão especial do sistema operacional Android feito para rodar em dispositivos vestíveis, mais especificamente em relógios inteligentes, ou *smartwatches*. Esses dispositivos são fabricados com telas redondas ou quadradas e podem ser controlados pela tela de toque ou por comandos de voz (LEAL, 2015).

O *Android Wear* foi desenhado para ser simples e prático de usar, ele possui uma interface de usuário otimizada para telas pequenas, com ênfase em um fluxo de informações em formato de cartões, conforme ilustrado na Figura 8.

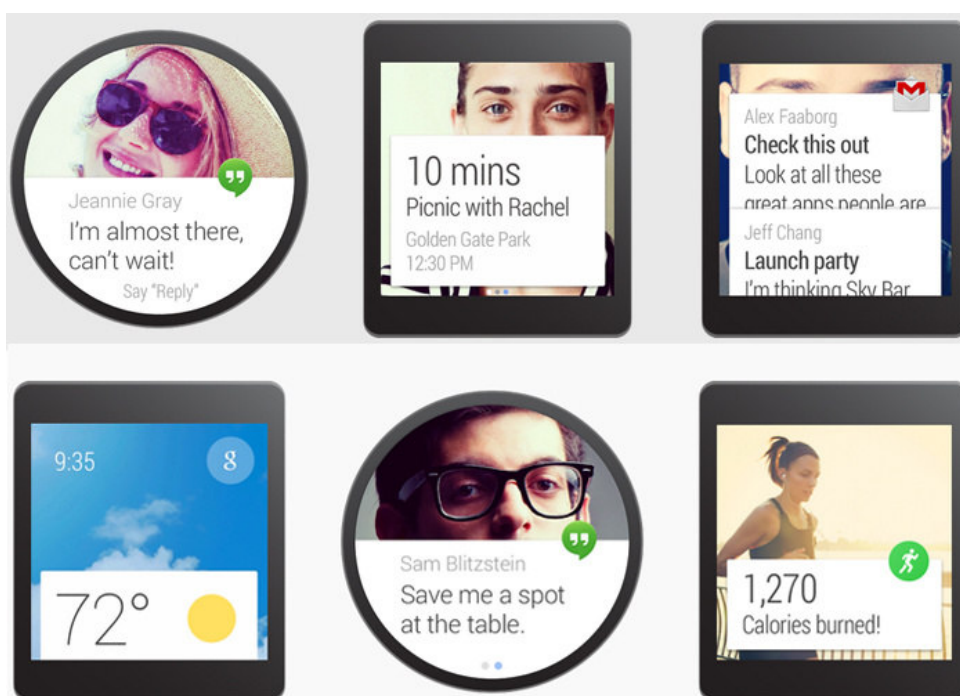


Figura 8 – Cartões no *Android Wear*

Fonte: (OJEDA-ZAPATA, 2015)

Esses cartões mostram informações relevantes ao contexto do usuário, requerendo pouca ou nenhuma interação por parte do usuário para exibi-las, além das notificações que normalmente são exibidas no *smartphone* (ANDROID, 2016c; LEAL, 2015).

Os *smartwatches* Android, assim como os *smartphones*, possuem uma gama de sensores que podem ser utilizados no desenvolvimento de aplicações para a plataforma, tais como acelerômetro, giroscópio, sensor de luz, GPS, etc. Entretanto, a presença de um sensor em particular varia de acordo com o fabricante e modelo do *smartwatch* (LEAL, 2015; ANDROID, 2016g).

Em termos de estrutura, o *Android Wear* é um SO completo, similar ao Android que roda nos *smartphones*, e possui todas as camadas de abstração presentes na Seção 2.2.1, ou seja, o *Android Wear* usa a linguagem Java na programação de suas aplicações, faz uso das mesmas APIs e bibliotecas, possui a máquina virtual Dalvik/ART, etc.

Os relógios Android não possuem conexão direta com a internet, portanto para se comunicar com o mundo exterior e ter acesso a todas as funcionalidades, esses dispositivos precisam estar pareados a um *smartphone* ou *tablet* rodando a versão do Android 4.3 ou superior, através de Bluetooth ou Wi-Fi.

Porém os desenvolvedores não possuem acesso às chamadas de baixo nível, então para criar aplicações que se comuniquem com o *smartphone* ou *tablet* é necessário utilizar as APIs de alto nível do *Android Wear SDK* disponibilizadas pelo *Google Play Services*, conforme é ilustrado na Figura 9 (LEAL, 2015; ELEKS, 2014).

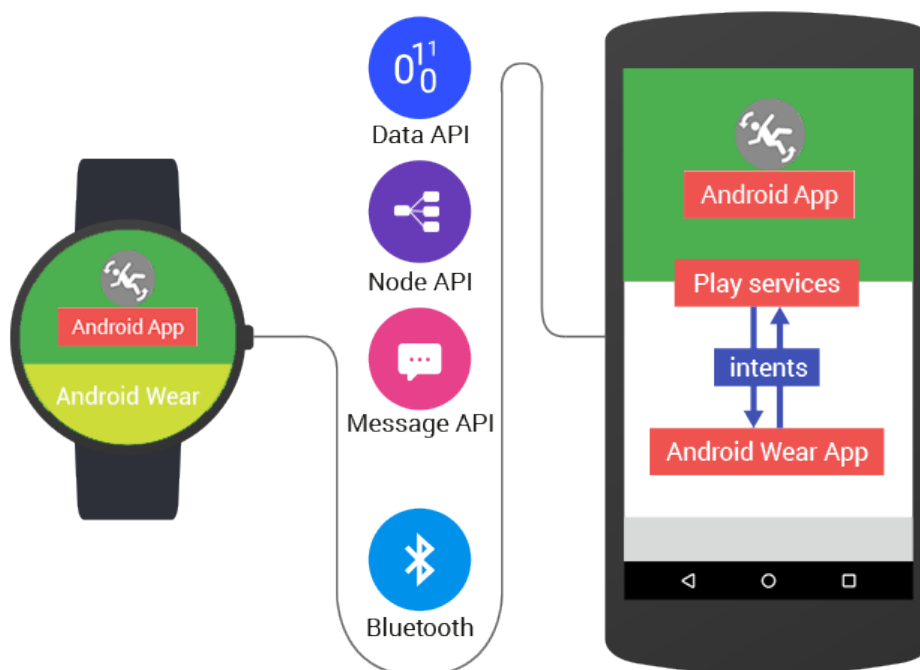


Figura 9 – Comunicação no *Android Wear*

Fonte: Adaptado e traduzido de (ELEKS, 2015).

O *Android Wear SDK* é constituído por três APIs, conhecidas como *Node API*, *Message API* e *Data API*, e todos os dados trafegados por esses canais são privados para cada aplicação. (LEAL, 2015; ELEKS, 2014).

A *Node API* é responsável pelo gerenciamento de conexão dos dispositivos tanto no *smartphone/tablet* quanto no *smartwatch*. Essa API notifica quando um dispositivo se conecta ou se desconecta, ou seja, ele informa quando ele entra e sai do raio de alcance do *Bluetooth* ou *Wi-Fi*.

A *Node API* também é responsável por listar todos os dispositivos aos quais o aparelho está conectado e é possível haver mais de um *smartwatch* pareado com um *smartphone/tablet*, conforme ilustrado na Figura 10 (LEAL, 2015).

A *Message API* fornece métodos que facilitam a troca de mensagens entre os

dispositivos conectados, podendo se originar tanto do celular quanto do relógio. Essa API normalmente é utilizada para enviar mensagens curtas, e o seu envio ocorre somente em uma direção, ou do *smartwatch* para o *smartphone*, ou vice-versa (LEAL, 2015; GOOGLE, 2015).

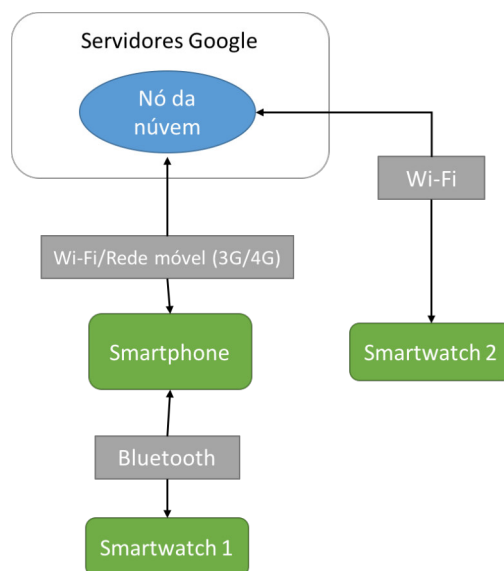


Figura 10 – Exemplo de conexão do *smartphone* com múltiplos nós.

Fonte: Traduzido de (ANDROID, 2016f)

A *Data API* funciona como um repositório compartilhado de dados entre os dispositivos conectados. Com essa API é possível criar, modificar e remover dados nesse repositório, como textos, imagens, arquivos em geral, etc. e essas operações podem ser feitas a partir de qualquer um dos dispositivos conectados. Também é possível notificar todos os dispositivos conectados quando uma modificação é feita em qualquer um desses dados.

2.3 Captura de Movimentos com Acelerômetros

O acelerômetro é um dispositivo responsável por medir forças de aceleração ou a vibração aplicadas sobre uma estrutura. Essa força é medida em metros por segundo ao quadrado (m/s^2) ou em termos de força G. Uma unidade de força G equivale à aceleração da gravidade aplicada sobre o acelerômetro, ou seja, no planeta Terra uma única unidade de força G é igual a $9,8m/s^2$ (ANDREJASIC, 2008; CORINNE, 2016). Para detectar a variação de aceleração, o acelerômetro utiliza uma massa de prova móvel ligada a molas e um sensor que detecta a movimentação dessa massa e, conforme ilustrado pela Figura 11.

Os acelerômetros podem ser fabricados utilizando diversos materiais e utilizando diversas técnicas de produção, entretanto para que seja possível embarcar um acelerômetro

em dispositivos pequenos e com fonte de energia limitada, como *smartphones* ou *smartwatches*, faz-se necessário que ele seja muito compacto e consuma pouca energia.

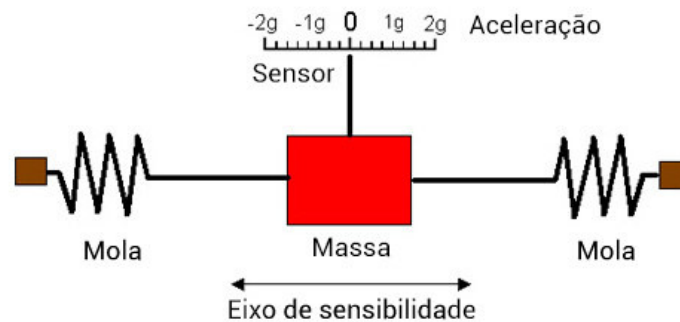


Figura 11 – Abstração de um acelerômetro.

Fonte: (INNOVENTIONS, 2015).

Por esse motivo, os acelerômetros encontrados nesses dispositivos são baseados na tecnologia de Sistemas Microeletromecânicos, ou no inglês *Microelectromechanical Systems* (MEMS), onde elementos mecânicos, como engrenagens, membranas, molas, entre outros, são miniaturizados e encapsulados de forma similar a um chip eletrônico através da técnica de micro fabricação (ANDREJASIC, 2008). A Figura 12 demonstra como um acelerômetro MEMS se parece quando visto através de um microscópio.

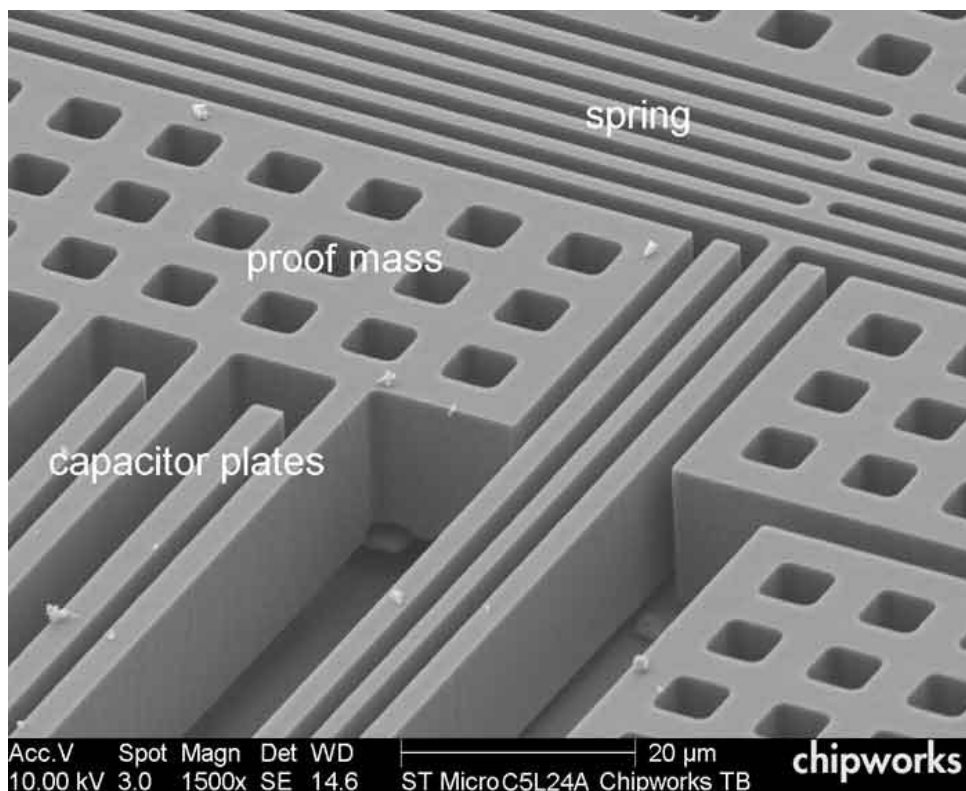


Figura 12 – Acelerômetro MEMS visto em um microscópio.

Fonte: (DIXON-WARREN, 2011).

Os acelerômetros MEMS utilizam uma massa de prova microscópica presa em pequenas molas. O acelerômetro também possui um conjunto de pequenas placas fixas conectadas a pequenos capacitores, e a massa de prova possui placas móveis localizadas entre essas placas fixas. Quando uma força é aplicada sobre o acelerômetro, a massa de prova e suas placas se movem, alterando a capacitância dos capacitores do conjunto, conforme ilustrado na Figura 13, ou seja, existe uma relação direta entre a aceleração aplicada no conjunto e a capacitância resultante (ANDREJASIC, 2008).

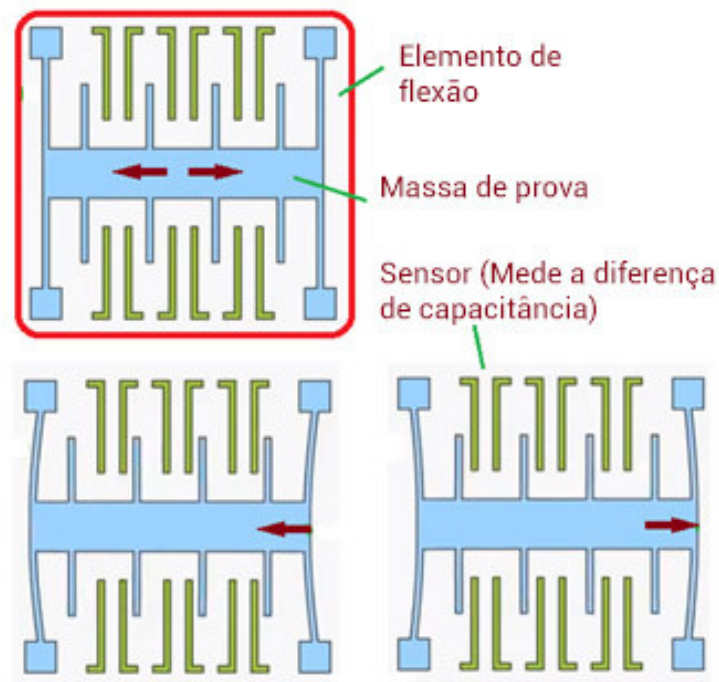


Figura 13 – Funcionamento de um acelerômetro MEMS.

Fonte: (GUJARATI, 2015).

3 Trabalhos Relacionados

Nesse capítulo são apresentados estudos e ferramentas comerciais que assim como o *Home e-Care*, trabalham com detecção de quedas e notificação de contatos de emergência. Além disso, é apresentado um resumo da arquitetura básica presente em todos esses trabalhos.

3.1 ERICA

O *ERICA* é um PERS criado pela empresa australiana *VitalCall* que utiliza botões de emergência para prover atendimento emergencial aos idosos em caso de acidentes (VITALCALL, 2015).

O botão de emergência é um equipamento dedicado de monitoramento pessoal. Ele consiste em um botão que pode estar presente em locais fixos da residência, ou que pode ser carregado por uma pessoa quando apresentados em formato de pulseira ou cordão conforme demonstrado pela Figura 14.



Figura 14 – Botão de emergência do sistema *ERICA*.

Fonte: Adaptado de (VITALCALL, 2015).

Esse botão deve ser pressionado pelo idoso na ocorrência de uma emergência que necessite de assistência imediata, como uma queda. Ao ser pressionado, esse dispositivo notifica de forma automática um ou mais contatos de emergência, indicando que o idoso necessita de ajuda.

Sistemas baseados em botões de emergência, como o *ERICA*, são produtos bem difundidos tanto no mercado brasileiro quanto internacional além de serem simples de fabricar e utilizar.

Entretanto, a principal desvantagem dessa abordagem se dá pela total dependência da notificação ser gerada por uma ação direta do idoso, tornando-o ineficiente em alguns casos em que o idoso não está apto a apertar o botão, como por exemplo, em uma queda com ocorrência de desmaio (RAJENDRAN et al., 2008).

3.2 iFall

O *iFall* é uma aplicação móvel proposta por Sposaro e Tyson (2009) que utiliza *smartphones* Android para realizar detecção de quedas. O *iFall* detecta quedas utilizando o acelerômetro presente no *smartphone* em conjunto com uma combinação entre dois algoritmos, um baseado em limiares e outro que detecta a inclinação do *smartphone*. Caso o algoritmo detecte uma aceleração que supere um determinado limiar e o *smartphone* esteja em uma inclinação horizontal em relação com o solo, o *iFall* infere que aquele evento corresponde a uma queda. A Figura 15 ilustra a tela de testes desse aplicativo.

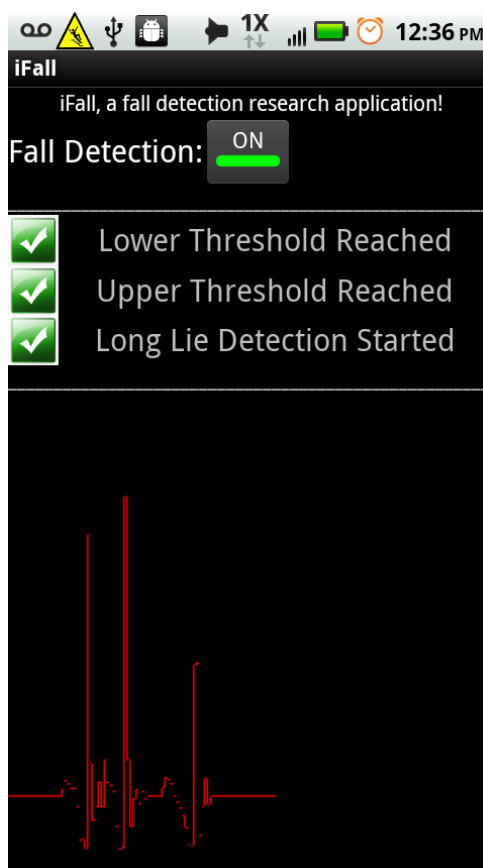


Figura 15 – Captura de tela do *iFall*.

Fonte: (SPOSARO, 2009).

O *iFall*, assim como qualquer outra solução de detecção de quedas que utilizam *smartphones*, tem como principal vantagem o seu custo reduzido de implantação, se comparado com equipamentos dedicados como os botões de emergência. Porém, como um

smartphone não é fixado em nenhuma parte do corpo, o algoritmo de detecção de quedas pode ter sua eficiência reduzida, e normalmente o *smartphone* costuma descarregar em menos de um dia (ABBATE et al., 2010).

3.3 F2D

O *F2D* é uma aplicação de detecção de quedas desenvolvido para Android e *Android Wear* e foi criado por Panagiotis et al. (2015). A Figura 16 demonstra essa aplicação rodando em um *smartphone* e em dois *smartwatches*. O *F2D* utiliza o acelerômetro de 3 eixos presente nos *smartwatches* em conjunto com um algoritmo de detecção de quedas baseado em limiares. Esse algoritmo leva em consideração algumas informações do usuário, como peso, idade e altura de forma a melhorar a eficiência da detecção de quedas.

Esse sistema funciona independentemente de uma estação central e permite que o usuário seja monitorado mesmo quando ele não estiver dentro de sua residência. Entretanto, sistemas de detecção baseados em *smartwatches* Android como o *F2D* tem como principal desvantagem a baixa autonomia de bateria desses relógios, que não costumam durar mais que um dia (RAWASSIZADEH; PRICE; PETRE, 2015).

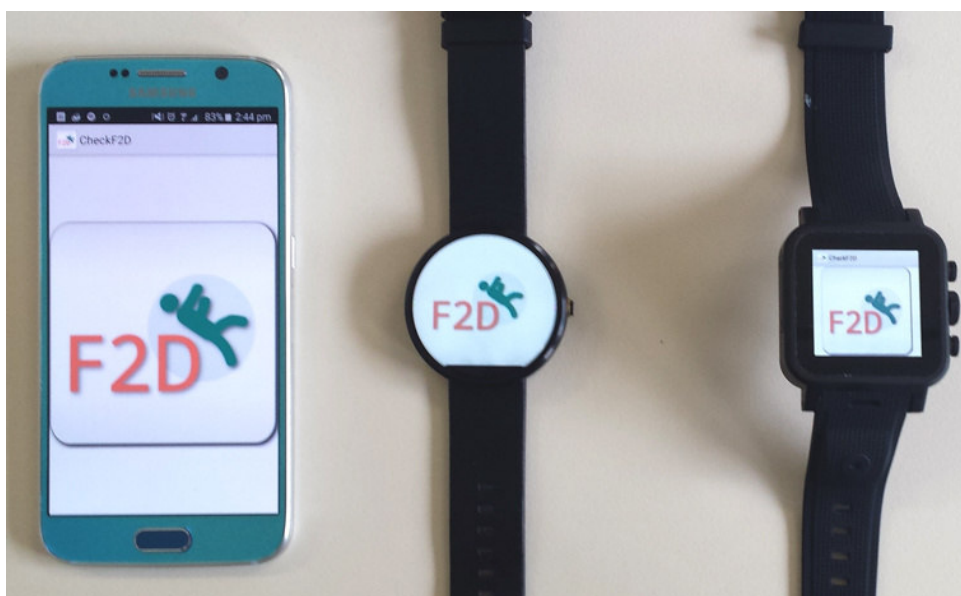


Figura 16 – *F2D* rodando em diversos dispositivos.

Fonte: (TAM, 2015).

3.4 Metodologias de Detecção de Quedas com Classificador

O trabalho proposto por Kulkarni (2011) utiliza o acelerômetro e o magnetômetro presentes em *smartphones* Android em seu algoritmo de detecção de quedas. Os dados de ambos os sensores são utilizados em conjunto com um classificador gerado pelo Ambiente

Waikato para Análise de Conhecimento, do inglês *Waikato Environment for Knowledge Analysis* (WEKA). Quando um padrão de queda é detectado em ambos os sensores, é inferido que houve a ocorrência de uma queda e um alerta é gerado e enviado para os contatos de emergência do usuário.

O WEKA é uma pacote de software feito em Java que agrega diversos algoritmos de aprendizagem de máquinas. Tais algoritmos podem ser aplicados diretamente sobre dados coletados previamente, ou em tempo real através de um classificador Java gerado pelo WEKA com base em um ou mais desses algoritmos (HALL et al., 2009).

A principal vantagem de se utilizar um classificador como o WEKA vem de sua melhor precisão e adaptabilidade quando comparado com algoritmos mais simples de detecção de quedas. Entretanto, o uso contínuo desse algoritmo de detecção de quedas resultou na diminuição de 43,49% da duração da bateria do *smartphone*, quando comparado com situações onde a aplicação de detecção de quedas não estava sendo executada.

3.5 Arquitetura Básica de Metodologias de Detecção de Quedas

Segundo Habib et al. (2014), todos os sistemas de detecção e prevenção de quedas baseados em *smartphones* possuem a mesma arquitetura básica conforme apresentado na Figura 17. Esses sistemas dividem-se em três partes: Percepção, Análise e Comunicação.

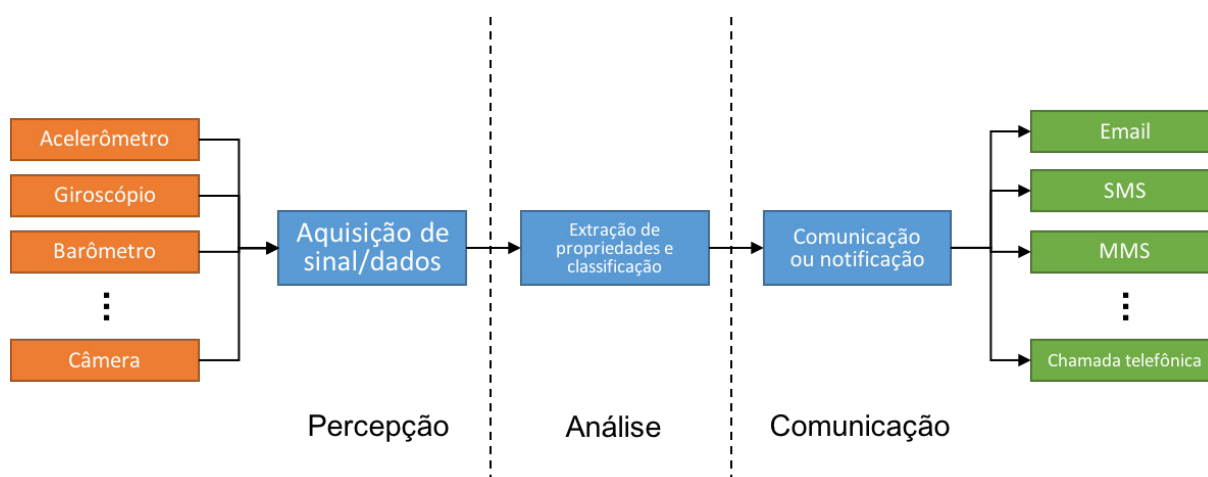


Figura 17 – Arquitetura básica de sistemas de detecção de quedas.

Fonte: Traduzido de (HABIB et al., 2014)

A primeira parte desses sistemas é caracterizada pela percepção das grandezas físicas através de um ou mais sensores normalmente presentes nos *smartphones*, como o acelerômetro, o giroscópio, o magnetômetro, entre outros. Dentre esses sensores, o acelerômetro é o mais utilizado por esses sistemas, por prover dados de maior significância em um cenário de queda.

Após a realização da primeira etapa, o sistema deve analisar os dados coletados através dos sensores. A partir desses dados, o sistema extrai as características relevantes à detecção de quedas e então aplicam algum algoritmo para classificar se as características extraídas representam ou não um evento de queda.

Dentre esses algoritmos, o mais utilizado é o Algoritmo Baseado em Limiares, ou no inglês *Threshold Based Algorithm* (TBA), devido à sua baixa complexidade e pelo consequente baixo consumo de energia, maximizando a duração de uma carga de bateria.

Esse algoritmo compara as características extraídas dos sensores com um ou mais limiares de valores pré-determinados, e caso esses limiares sejam ultrapassados, o algoritmo classifica o evento como uma possível queda (MEHNER; KLAUCK; KOENIG, 2013).

Por fim, após a inferência por parte do algoritmo de detecção de quedas, alguns sistemas de detecção podem exibem uma mensagem, indicando que uma possível queda foi detectada e aguardam por uma confirmação manual do usuário. Caso ele não cancele o alerta, uma mensagem é enviada para os seus contatos de emergência para obter auxílio médico através de algum meio de comunicação, como e-mail, SMS, chamadas de voz, entre outros.

Alguns sistemas de detecção podem não exibir essa mensagem e enviar o alerta para os contatos de emergência imediatamente após a suspeita de uma queda (HABIB et al., 2014).

4 Home e-Care: Metodologia e Implementação

Neste capítulo, apresenta-se de forma detalhada todas as etapas que são realizadas para a detecção dos eventos de queda e sua posterior notificação para os contatos de emergência, de forma a proporcionar um entendimento aprofundado do funcionamento do *Home e-Care*.

4.1 Visão Geral

O *Home e-Care* consiste de vários algoritmos e dispositivos trabalhando em conjunto com a finalidade de agilizar o atendimento ao idoso na ocorrência de uma queda.

Tal objetivo pode ser alcançado através da detecção da queda em tempo real e sua posterior notificação automatizada para contatos de emergência através de diversos meios de comunicação. Após receberem o alerta, os contatos responsáveis pela pessoa acidentada podem acionar um resgate, como por exemplo, uma ambulância. A Figura 18 exemplifica o funcionamento geral desse esquema.

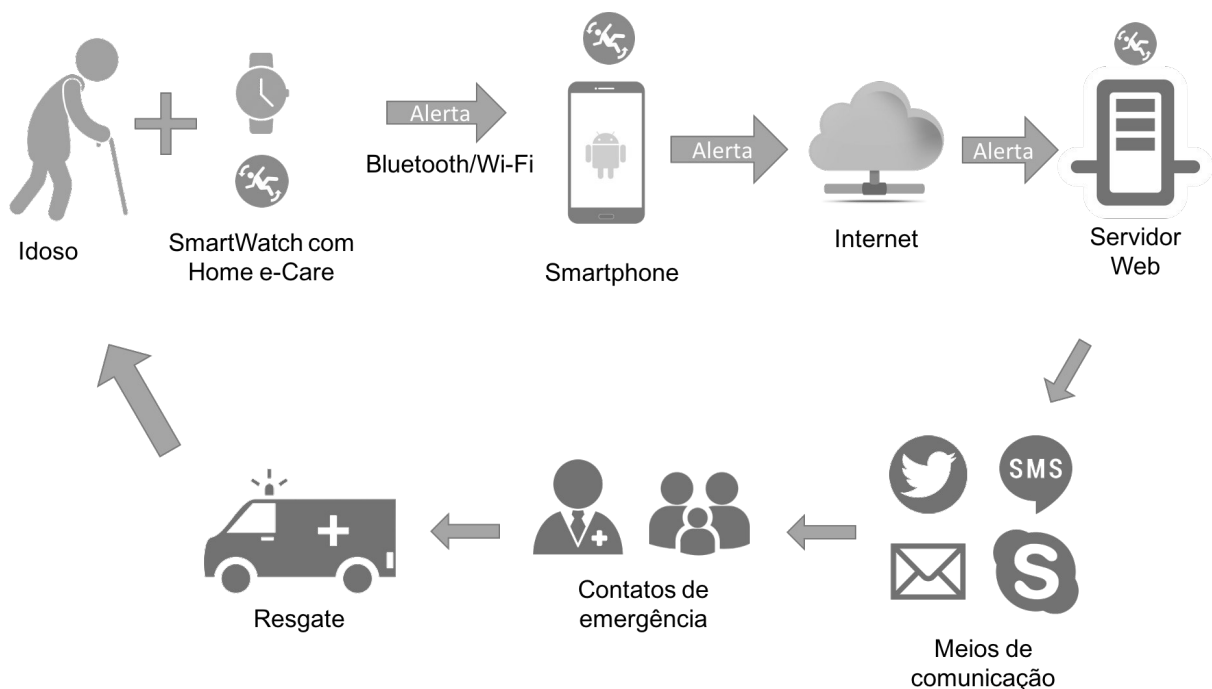


Figura 18 – Visão geral do Home e-Care.

Fonte: Acervo do autor.

O idoso veste um *smartwatch* que possui o *Home e-Care* instalado, e a partir da

ocorrência de uma queda, um alerta é gerado e enviado para o *smartphone* pareado. Ao receber a mensagem de alerta provinda do *smartwatch*, ela é encaminhada através da internet para o módulo *web*. Esse módulo então encaminha a mensagem aos contatos de emergência por todos os meios de comunicação cadastrados. Por fim, ao receber a mensagem, os contatos podem acionar um resgate para o idoso acidentado, ou tomar outras providências.

O *Home e-Care* está dividido em três módulos trabalhando em conjunto, conforme ilustrado pela Figura 19. Cada um deles é responsável por realizar uma etapa do processo de alerta aos contatos de emergência e posterior resgate da pessoa acidentada.

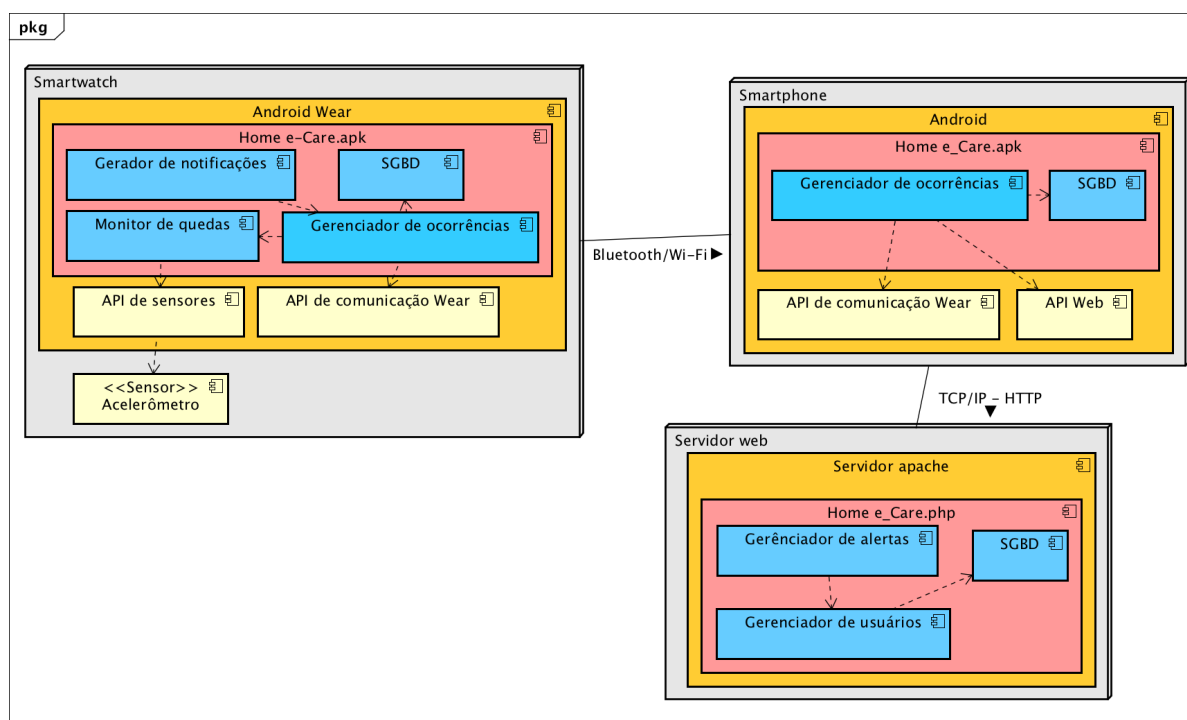


Figura 19 – Módulos do *Home e-Care*

Fonte: Acervo do autor.

O módulo de detecção de quedas que roda no *smartwatch* do idoso fica responsável por analisar os dados fornecidos pelo acelerômetro do dispositivo em tempo real, até detectar o padrão de uma queda. A partir desse momento o processo de alerta é iniciado. O módulo então envia uma mensagem de alerta para o *smartphone* pareado através da rede *Wi-Fi* ou da conexão *Bluetooth*.

Após o envio do alerta, o módulo continua a monitorar os dados fornecidos pelo acelerômetro, para então verificar se a pessoa acidentada continua caída no chão por um período prolongado de tempo, enviando um novo alerta caso isso ocorra.

O módulo de comunicação que fica instalado no *smartphone* recebe as mensagens de alerta geradas pelo módulo no *smartwatch*, armazena a ocorrência em um banco de

dados local e então às encaminha através da internet para o módulo *web*.

O módulo *web* é responsável por ficar permanentemente escutando mensagens de alerta geradas pelos *smartphones* dos idosos. Na ocorrência de um evento de queda, é realizado o registro dessa ocorrência em um banco de dados e é feita a comunicação de forma automática aos contatos de emergência cadastrados para o usuário acidentado. Através de meios como SMS, e-mail, chamadas telefônicas, redes sociais (Twitter, Facebook), entre outros.

A partir desse alerta automatizado, os contatos de emergência poderão providenciar o resgate do idoso acidentado logo após a queda ter acontecido, agilizando o tempo de atendimento e tratamento da pessoa.

4.2 Módulo de Detecção de Quedas

Esse módulo é responsável por analisar o sinal gerado pelo acelerômetro do smartwatch e detectar a ocorrência de quedas com base nessas informações. O módulo de detecção de quedas também é responsável por gerar e encaminhar alertas para o módulo do *smartphone* quando uma ocorrência de queda do idoso for detectada.

O algoritmo de detecção de quedas utilizado pelo módulo correspondente do *Home e-Care* utiliza como base o conceito do algoritmo de limiares de sinais calculados a partir do acelerômetro, excluindo a gravidade, e um ou mais limiares de aceleração fixos ou variáveis.

Mais especificamente, o algoritmo é baseado na solução adotada pelo sistema *F2D* (PANAGIOTIS et al., 2015), que utiliza dois limiares de magnitude da aceleração e um módulo de decisão.

Os dados de aceleração são obtidos através de um acelerômetro de três eixos, ou seja, é possível detectar a variação na aceleração do usuário, em m/s^2 nos eixos ortogonais X, Y e Z.

Como normalmente um dos eixos sofre com a ação da gravidade, é aplicado um *Filtro Passa Baixa* e posteriormente um *Filtro Passa Alta* para isolar o sinal gerado pela gravidade e removê-lo dos eixos (KIRCHER, 2014; ANDROID, 2016d). Em seguida, com base nessas informações, é calculada a magnitude da aceleração (M_A) nesses três eixos, conforme a Equação 4.1:

$$M_A = \sqrt{A_x^2 + A_y^2 + A_z^2} \quad (4.1)$$

onde M_A é a magnitude de aceleração e A_x , A_y , A_z são as acelerações nos eixos X, Y e Z, respectivamente.

Para analisar o comportamento dessa magnitude de aceleração durante uma queda, foi utilizado um algoritmo coletor de dados rodando no *smartwatch*, que continuamente lê os dados fornecidos pelo acelerômetro e calcula sua magnitude usando a Equação 4.1.

O coletor de dados também salva essas informações em um banco de dados interno do *smartwatch* para posterior consulta. A Figura 20 demonstra um exemplo de coleta em um experimento de queda.

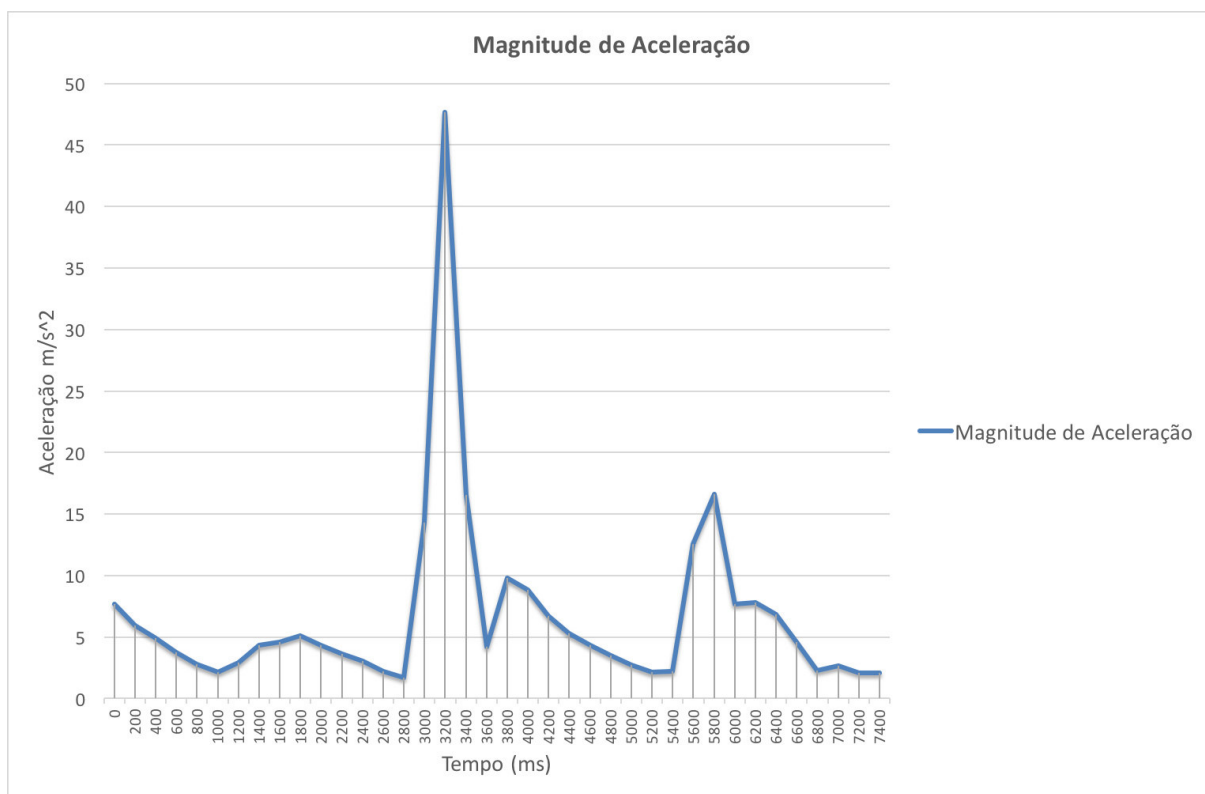


Figura 20 – Gráfico de magnitude da aceleração durante uma queda.

Fonte: Acervo do autor.

Nesse exemplo, é possível verificar que o padrão de uma queda pode ser resumido em uma breve queda na magnitude de aceleração, o que representa o período em que a pessoa se encontra em queda livre, seguida por um rápido e súbito aumento na magnitude de aceleração, representando o impacto da pessoa com o solo ou outra superfície. Posteriormente essa magnitude diminui e se estabiliza, representando o período em que a pessoa acidentada permanece no chão.

Caso o valor de magnitude ultrapasse um limiar pré-determinado, o algoritmo identifica aquele evento como uma possível queda. Porém normalmente somente essa condição é insuficiente para categorizar o evento como uma queda, pois o usuário pode ter feito apenas um movimento súbito. Assim, é necessário o uso de mais algum mecanismo para classificar aquele evento como uma queda e diminuir a ocorrência de falsos positivos (HABIB et al., 2014).

Na implementação de referência, é usado um módulo de decisão com base em um contador de ocorrências e em uma janela de tempo pré-determinada. Toda vez que a magnitude ultrapassar o limiar, o contador de ocorrências é incrementado. O processo se repete até a janela de tempo fechar e a partir desse momento, o valor do contador é averiguado e é possível inferir se aquele padrão de magnitude de aceleração caracterizou uma queda real ou se o padrão caracteriza outras atividades físicas quaisquer.

A Figura 21 ilustra o funcionamento do processo de detecção de quedas. A escolha desse tipo de algoritmo se deu por sua baixa complexidade de implementação e execução, diminuindo o consumo de energia no *smartwatch* e maximizando o tempo de duração da bateria (HABIB et al., 2014).

O *Home e-Care* lê os valores do acelerômetro presente no *smartwatch*, calcula a magnitude da aceleração e aplica o algoritmo de detecção de quedas baseado em limiares (PANAGIOTIS et al., 2015).

Uma vez que a magnitude exceda o limite superior, o algoritmo detecta que aquele evento pode representar uma possível queda, então ele abre uma janela de tempo de 6 segundos para verificar essa ocorrência. Após o primeiro pico de aceleração, o algoritmo espera um pequeno intervalo de tempo inferior a 1 segundo, e se após esse intervalo a aceleração ainda estiver acima do limite inferior, um contador interno é incrementado. Esse processo se repete enquanto a janela de tempo não ultrapassar os 6 segundos.

Após essa etapa, o algoritmo verifica o valor armazenado no contador de ocorrências, que representa quantas vezes a condição anterior foi verdadeira, se esse valor for menor que um número pequeno X , a ocorrência pode ter sido ativada por um movimento súbito da pessoa monitorada, e caso o contador seja maior que um valor Y , o evento não categoriza uma queda, e sim alguma outra atividade como uma corrida ou outra atividade física mais intensa. Porém, caso o valor do contador seja $X \leq \text{contador} < Y$, aquele evento caracteriza uma queda real e um alerta deve ser gerado. A Figura 21 demonstra o fluxo de eventos desse processo.

A escolha dos limiares superior L_s e inferior L_i , assim como dos valores X e Y utilizados nessa implementação foram feitas com base nos resultados dos experimentos realizados por Panagiotis et al. (2015).

No referido trabalho, foi descoberto que para esse algoritmo os limiares podem assumir valores entre 10 a 18m/s² e 2 a 7m/s², respectivamente, e os valores de comparação do contador podem assumir $X = 1$ e $5 \leq Y \leq 10$. Tais parâmetros variam de acordo com a idade, peso e altura da pessoa que está utilizando o *smartwatch*. Esses valores foram aplicados no *Home e-Care*, como mecanismo automatizado de detecção de quedas.

Esse módulo funciona com base em um serviço que fica rodando em *background* no *smartwatch*, continuamente lendo os dados fornecidos pelo acelerômetro, filtrando e

processando o sinal para então aplicar o algoritmo de detecção de queda propriamente dito.

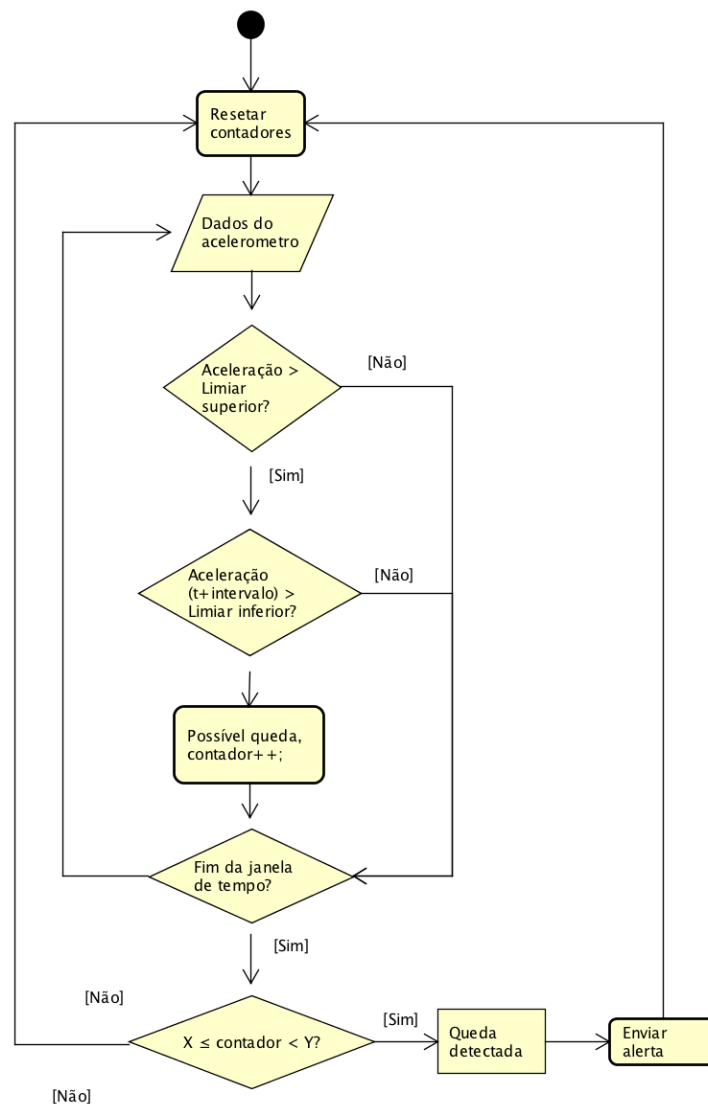


Figura 21 – Algoritmo de detecção de quedas utilizado no *Home e-Care*

Fonte: Adaptado e traduzido de (PANAGIOTIS et al., 2015)

Uma vez que um padrão de queda for detectado, o serviço utilizará do gerador de notificações para exibir uma mensagem na tela do *smartwatch* com um *timer* em contagem regressiva, dando a possibilidade do usuário cancelar o envio do alerta caso tenha sido um falso positivo, ou caso a pessoa julgar desnecessário um resgate. Caso a mensagem não seja cancelada depois de um determinado tempo, o relógio enviará uma mensagem de alerta para o *smartphone* utilizando a API de comunicação do sistema, conforme a Figura 22.

O algoritmo do *Home e-Care* também leva em consideração o movimento residual pós-queda. Uma vez que o alerta de queda for gerado, o algoritmo continuará recebendo valores do acelerômetro, com o intuito de verificar o comportamento da pessoa acidentada. Nessa etapa do processo, é aplicado um limiar de aceleração independente da etapa anterior,

esse limiar será utilizado para determinar se a pessoa conseguiu ou não se recuperar da queda dentro de um curto período pré-determinado de tempo.



(a) Notificação de queda com *timer*.

(b) Opção de cancelar o alerta no *smartwatch*.

Figura 22 – Capturas de tela do *smartwatch*.

Fonte: Acervo do autor.

Caso a pessoa não se levante até o fim desse tempo, o *smartwatch* irá gerar uma nova notificação, que assim como na primeira etapa, pode ser cancelada pelo usuário caso ele consiga e julgue necessário. O processo completo é demonstrado graficamente pelo fluxograma da Figura 23.

Segundo Bisson, Peterson e Finlayson (2015) o comportamento pós-queda pode ser separado em três categorias. A primeira é citada como *rápida recuperação inicial*, caracterizada pela permanência no chão inferior a 10 minutos após a ocorrência de uma queda, indicando que a pessoa acidentada conseguiu levantar sozinha.

A segunda categoria é chamada de *recuperação inicial tardia*, que se caracteriza por uma permanência no chão superior a 10 minutos, porém inferior a uma hora, indicando que o idoso esteve impossibilitado de se levantar por conta própria.

A terceira categoria, é chamada de *permanência prolongada*, caracterizada pela permanência da pessoa acidentada no chão por um período de tempo superior a uma hora, podendo indicar um desmaio, ou completa incapacidade de se recuperar da queda.

O módulo de detecção de quedas utiliza o tempo de 10 minutos como parâmetro de envio do segundo alerta por ele se encaixar na categoria de *recuperação inicial tardia* e ser um bom indicativo de que o idoso não conseguiu se recuperar após a queda.

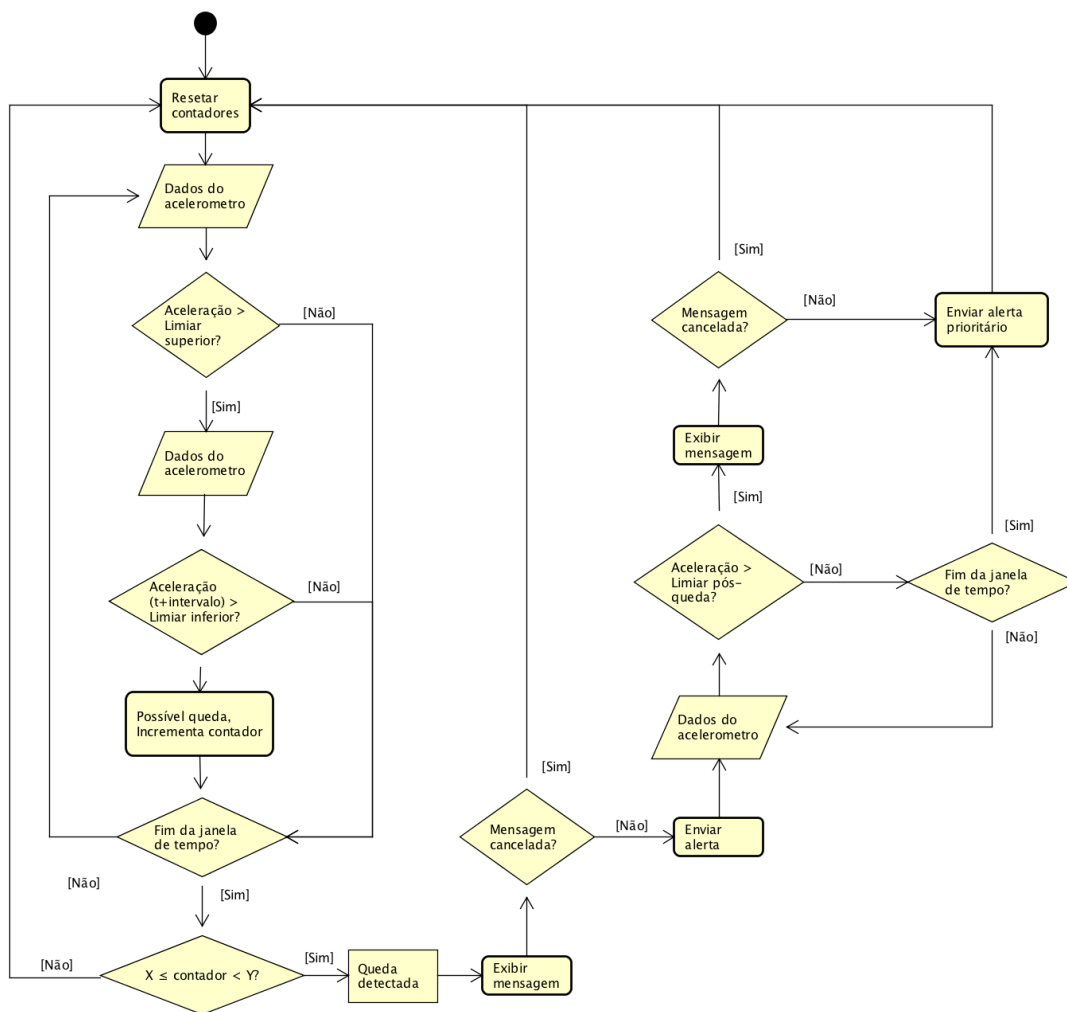


Figura 23 – Algoritmo de detecção de quedas estendido do *Home e-Care*

Fonte: Acervo do autor.

4.3 Módulo do Smartphone

O módulo do *smartphone* possui a função de manter um registro dos alertas recebidos pelo *smartwatch* e de encaminhá-los para o módulo *web*, além de garantir que essas mensagens sejam entregues corretamente, efetivamente funcionando como uma ponte entre o *smartwatch* e o mundo exterior.

Conforme é ilustrado pelo fluxo de dados da Figura 24, o *smartphone* fica ativo escutando o *smartwatch* pareado, até receber algum alerta de que uma queda foi detectada. Quando tal evento ocorrer, o *smartphone* cria um registro em seu banco de dados interno e envia o alerta para o módulo *web*, através do protocolo HTTP via POST ou GET.

Caso o envio falhe por problemas na conexão, o algoritmo realiza novas tentativas até que o envio seja concluído. Quando o alerta for enviado com sucesso, o *smartphone* irá modificar o registro em seu banco de dados para indicar que a mensagem foi enviada. O processo para envio dos alertas de atendimento urgente segue exatamente os mesmos

passos dos alertas comuns.

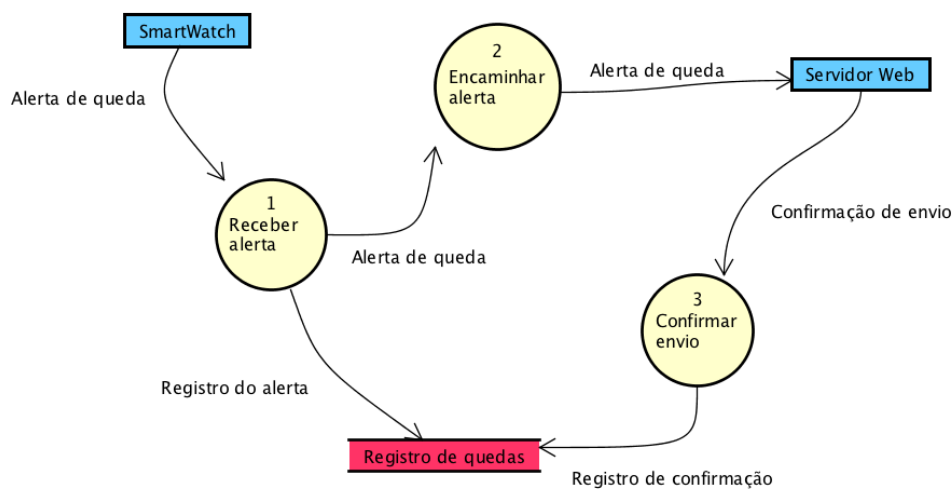


Figura 24 – Fluxo de dados do módulo do *smartphone*.

Fonte: Acervo do autor.

4.4 Módulo Web

O módulo *web* tem como principais funções receber todas as mensagens geradas pelos *smartphones* dos idosos, criar um registro dessas ocorrências e encaminhar os alertas para os respectivos contatos de emergência de cada idoso, usando os diversos meios de comunicação cadastrados no banco de dados desse módulo.

O módulo *web* escuta continuamente por alertas enviados pelos *smartphones* cadastrados no sistema. As mensagens são recebidas utilizando o protocolo HTTP, com métodos POST (ou GET). Ao receber um alerta, esse módulo salva o evento em seu banco de dados para referências futuras, e automaticamente notifica os contatos de emergência, enviando mensagens pelos meios de comunicação cadastrados, como e-mail, Twitter, SMS, Skype em texto ou voz, entre outros. Na implementação de referência está disponível somente a notificação através da rede social Twitter.

O fluxo de dados desse módulo pode ser observado e resumido conforme ilustrado na Figura 25. A Figura 26 demonstra um alerta recebido através do Twitter.

Os contatos de emergência, ao receberem a mensagem do módulo *web* ficarão cientes que a pessoa monitorada sofreu um acidente e poderão escolher a melhor forma de abordar o evento, seja ligando para a pessoa, ou chamando uma ambulância para resgatá-la.

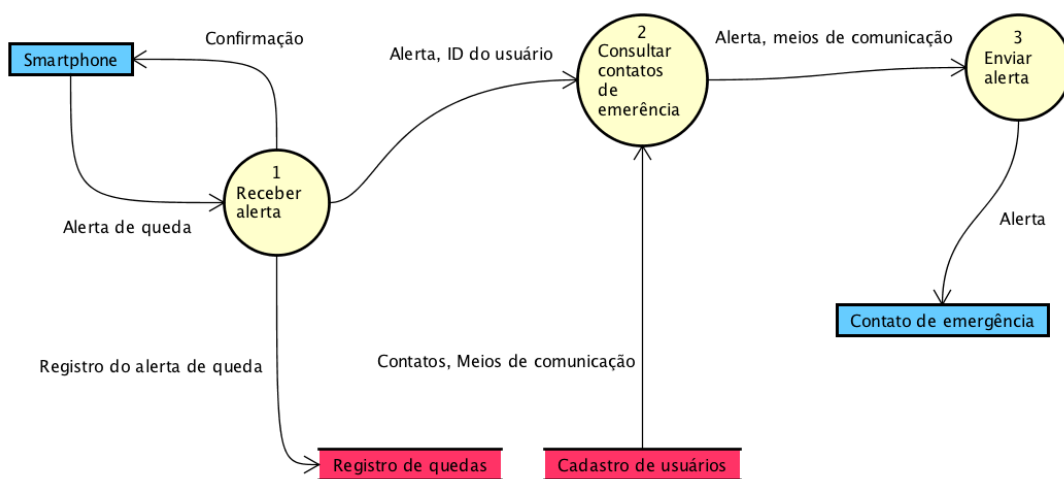


Figura 25 – Fluxo de dados do módulo *web*.

Fonte: Acervo do autor.



Figura 26 – Alerta recebido através do Twitter.

Fonte: Acervo do autor.

5 Resultados

Esse capítulo detalha os experimentos realizados com os módulos do *Home e-Care*, quais dispositivos foram usados nos testes, como os testes foram realizados e quais valores foram utilizados como parâmetros no algoritmo de detecção de quedas. Também são apresentados os resultados obtidos através desses testes, sendo então feita uma discussão sobre a eficiência do algoritmo de detecção de quedas.

5.1 Experimentos Realizados

Para verificar a eficiência e confiabilidade do algoritmo de detecção de quedas utilizado pelo módulo do *smartwatch*, foi realizada uma série de experimentos, envolvendo o uso do *smartwatch* em quedas simuladas em um colchão e em atividades da vida cotidiana, ou no inglês *Activities of Daily Living* (ADL).

Segundo [Foti e Kanazawa \(2008\)](#), as ADLs são definidas como qualquer atividade comum do cotidiano de uma pessoa, como andar, sentar, subir ou descer uma escada, tomar banho, comer, dormir, entre outros.

Todos os experimentos foram realizados com o *smartwatch Motorola Moto 360*, rodando a versão 5.1 do *Android Wear* pareado com um *smartphone Motorola Moto Maxx*, rodando a versão 5.0.2 do Android. O relógio estava fixado no braço esquerdo durante todos os testes. Os parâmetros do algoritmo de detecção de quedas, descritos na seção 4.2, estavam configurados conforme a Tabela 2.

Tabela 2 – Parâmetros de configuração do algoritmo de detecção de quedas.

Campo de configuração	Valor utilizado	Unidade de medida
Limiar superior	18	m/s^2
Limiar inferior	7	m/s^2
Limite inferior do contador	1	Número de ocorrências
Limite superior do contador	10	Número de ocorrências
Janela de tempo de detecção	6	s
Intervalo de tempo entre contagens	150	ms

Fonte: Acervo do autor.

Foram realizados um total de 240 testes, sendo que 140 correspondem a ADLs e 100 correspondem a quedas simuladas. As ADLs foram divididas nas seguintes atividades:

- Andar;
- Correr;

- Subir escadas;
- Descer escadas;
- Sentar;
- Levantar;
- Balançar o braço, ou acenar para outra pessoa.

onde cada atividade foi realizada 20 vezes. Enquanto os testes de quedas simuladas foram divididos nas seguintes categorias:

- Queda frontal;
- Queda lateral com impacto no lado direito do corpo;
- Queda lateral com impacto no lado esquerdo do corpo;
- Queda de costas;
- Queda de joelhos.

onde cada categoria foi repetida em um total de 20 vezes.

Durante a execução de cada caso de teste, foi observado se o algoritmo conseguiria categorizar corretamente a atividade que estava sendo executada, ou seja, se um alerta era gerado ou não durante as ADLs e as quedas simuladas. Os resultados foram anotados em duas tabelas, uma para os testes com ADLs e outra para os testes de quedas.

5.2 Análise dos Resultados Obtidos

Após realizar a coleta de dados, foi montada uma matriz de confusão conforme a Tabela 3. Além disso, cada caso de teste teve sua taxa de acertos calculadas conforme ilustrado nos gráficos presentes nas Figuras 27 e 28. Com esses dados, tornou-se possível analisar a eficiência do *Home e-Care*.

Foi possível detectar corretamente três a cada quatro quedas testadas, caracterizando uma sensibilidade de 75%. Dentre as 90 quedas detectadas pelo *Home e-Care*, 75 foram quedas verdadeiras, enquanto 15 foram falsos positivos, caracterizando uma precisão de 83,33%.

Entre as 140 ADLs testadas, 125 foram corretamente classificadas como tal, resultando em uma especificidade de 89,29%. De um total de 240 casos de teste realizados, 200 foram corretamente categorizados, ou seja, a acurácia do *Home e-Care* foi de 83,33%.

Tabela 3 – Matriz de confusão dos experimentos realizados.

Matriz de confusão	Queda (Real)	Não-Queda (Real)	Total Classificado
Queda (Classificado)	75	15	90
Não-Queda (Classificado)	25	125	150
Total Real	100	140	240

Sensibilidade	75%
Especificidade	89,29%
Precisão	83,33%
Acurácia	83,33%

Fonte: Acervo do autor.

Diante das informações apresentadas na matriz de confusão, é possível concluir que o *Home e-Care* obteve resultados promissores na detecção de quedas de seus usuários.

Conforme é observado na Figura 27, dentre os 7 tipos de ADL, o *Home e-Care* classificou com 100% de acurácia 3 tipos, ou seja, nos testes em que o usuário precisou andar, subir escadas e descer escadas, não houve nenhum falso positivo. Essa eficiência se deve ao fato da magnitude de aceleração não alcançar o limiar superior utilizado pelo algoritmo de detecção de quedas.

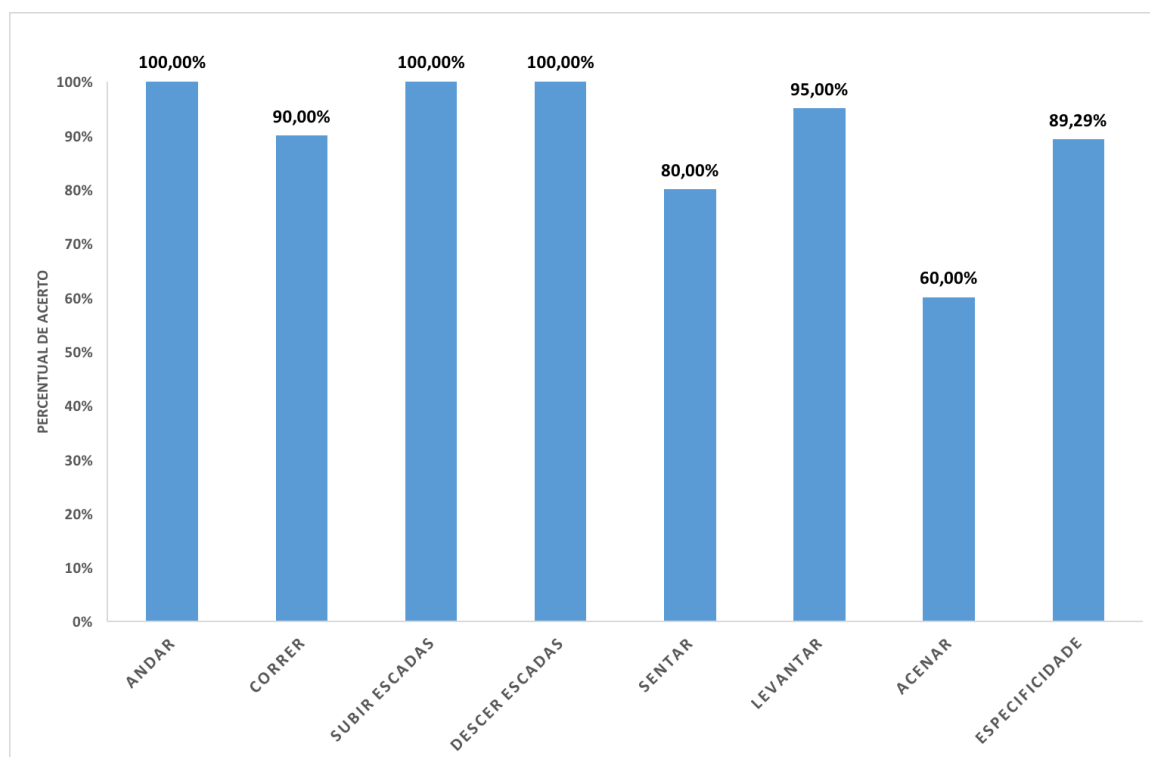


Figura 27 – Resultado dos testes de ADL.

Fonte: Acervo do autor.

Ao testar o *Home e-Care* em situações onde o usuário estava correndo, obteve-se

uma acurácia de 90%. Os casos que geraram falsos positivos ocorreram quando a corrida teve uma duração inferior à janela de tempo de 6 segundos utilizada como parâmetro do algoritmo de detecção de quedas.

Os testes caracterizados como sentar em uma cadeira resultaram em uma acurácia de 80%, sendo que os falsos positivos somente foram gerados em alguns casos em que o usuário sentou subitamente na cadeira, o que gerou um padrão de aceleração similar ao de uma queda.

Os testes em que o usuário precisou levantar-se da cadeira resultaram em uma acurácia de 95%, sendo que o único caso que gerou um falso positivo ocorreu quando o ato de se levantar foi súbito e similar a um pulo para fora da cadeira.

O *Home e-Care* teve seu pior resultado nos testes em que o usuário acenou com o braço onde o *smartwatch* estava fixado, classificando corretamente apenas 60% dessa ADL, pois nesses testes a magnitude de aceleração alcançou valores similares aos observados em uma queda real devido à rápida movimentação do pulso.

Com base nas informações exibidas na Figura 28, é possível verificar que o *Home e-Care* conseguiu melhores resultados de classificação nas quedas frontais e de costas, obtendo uma taxa de acerto de 90% em ambos os tipos de queda.

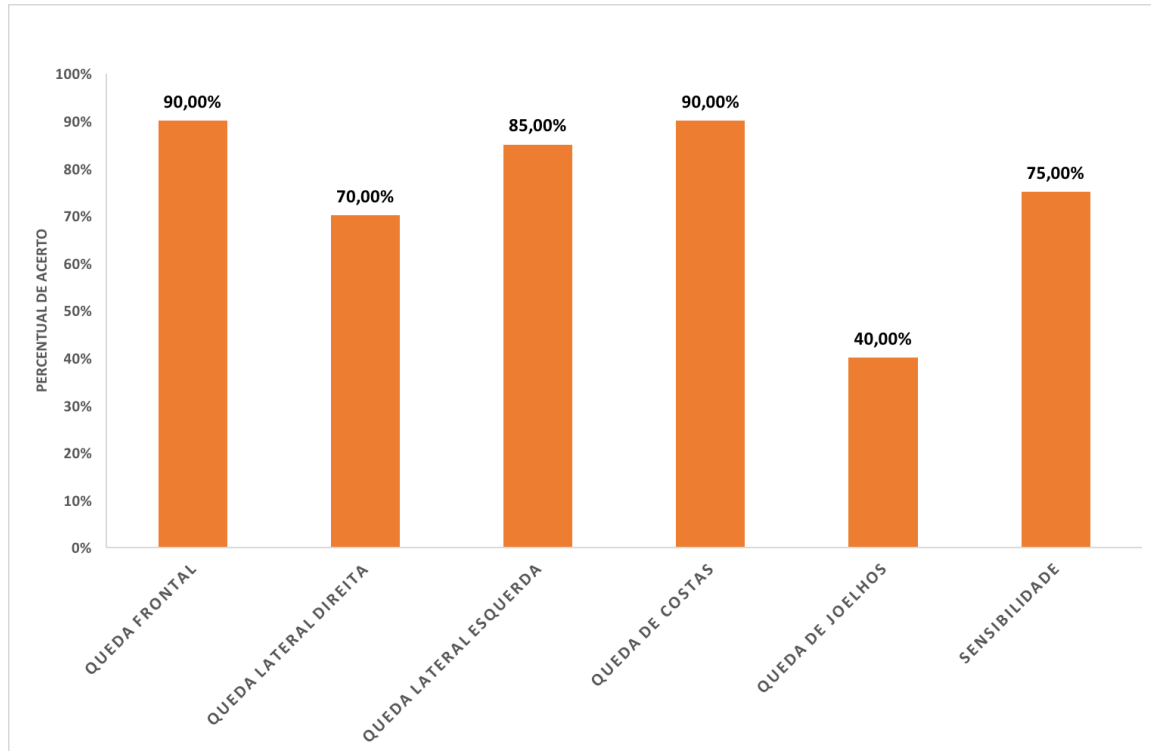


Figura 28 – Resultado dos testes de queda.

Fonte: Acervo do autor.

Os testes com quedas laterais também apresentaram acurácia satisfatória, onde

70% das quedas com impacto no lado direito e 85% das quedas com impacto no esquerdo do usuário foram corretamente classificadas. Também pode-se observar que o *Home e-Care* teve um desempenho maior na detecção das quedas com o impacto no lado esquerdo, onde o *smartwatch* estava localizado.

Por fim, observou-se que o *Home e-Care* teve o seu pior resultado nos testes onde o usuário caía de joelhos, classificando corretamente apenas 40% dessas quedas. Tal resultado se deve ao fato desse tipo de queda ser mais suave, e pela distância entre o *smartwatch* e o ponto de impacto. Portanto, a vibração causada pelo impacto não gerou uma magnitude de aceleração suficientemente alta para alcançar o limiar superior de aceleração.

6 Conclusão

Este trabalho apresentou uma solução para o problema de acidentes de queda em pessoas da terceira idade, através do desenvolvimento de um PERS automatizado denominado *Home e-Care*.

Para desenvolver esse PERS, foram coletados os dados fornecidos pelo acelerômetro de um *smartwatch* com Android Wear nas mais diversas ADLs, assim como em situações de quedas simuladas, de forma a analisar e entender os padrões de sinal gerados pelo acelerômetro durante a execução dessas atividades. Com posse dessas informações, foi utilizado um TBA com dois limiares em conjunto com um módulo de decisão como método de detecção de quedas, e a internet como meio de notificação automatizada. Além disso, por ser desenvolvido em cima do SO Android e Android Wear, o *Home e-Care* pode ser facilmente utilizado por qualquer usuário dessa plataforma.

O protótipo do *Home e-Care* utilizado nesse trabalho não implementa os seguintes itens:

- Notificação dos contatos de emergência através de SMS, e-mail, Skype e Facebook;
- Registro das ocorrências de queda nos bancos de dados de cada módulo e acesso à estas informações;
- Confirmação de envio de mensagem do módulo do *smartphone* para o módulo *web*;
- Ajuste dos limiares de magnitude de aceleração utilizados pelo TBA;
- Cadastro de usuário e contatos de emergência, esses valores foram fornecidos diretamente no código do protótipo.

Após a realização de testes com o *Home e-Care*, verificou-se que esse PERS obteve resultados promissores na detecção de quedas e notificação automatizada.

Vale ressaltar que com a criação do módulo *web* do *Home e-Care*, é possível trabalhar com notificações de uma forma genérica, pois esse módulo funciona como uma central, podendo receber alertas gerados por outros dispositivos de monitoramento além do detector de quedas do módulo do *smartwatch*. Ou seja, efetivamente é possível expandir o *Home e-Care* para monitorar e notificar outras situações de risco.

Portanto, conclui-se que o objetivo desse trabalho foi alcançado, pois o TBA utilizado pelo *Home e-Care* foi testado e validado, e os alertas foram corretamente encaminhados quando eventos de queda foram detectados. Entretanto torna-se necessário

um estudo mais aprofundado de outras técnicas além do TBA, de forma a melhorar a acurácia do PERS e torná-lo mais confiável.

Como trabalhos futuros, aponta-se as seguintes melhorias para o *Home e-Care*:

- Utilizar a técnica de fusão de sensores, ou seja, agregar os outros sensores presentes no *smartwatch* em conjunto com o acelerômetro, para melhorar a sua eficiência;
- Enviar localização do GPS ao detectar uma queda;
- Testar o *Home e-Care* em ADLs com usuários idosos, de forma a obter resultados mais próximos da realidade do seu público alvo;
- Realizar um estudo da usabilidade do *Home e-Care* com idosos;
- Ajustar os limiares de acordo com o perfil do usuário que utiliza o *Home e-Care*;
- Monitorar outras situações de risco, como atividades físicas excessivas, alterações na frequência cardíaca, entre outros;
- Comparar o desempenho do *Home e-Care* com outros classificadores, como a Máquina de vetores de suporte.

Referências

ABBATE, S. et al. *Monitoring of human movements for fall detection and activities recognition in elderly care using wireless sensor network: a survey*. [S.l.]: InTech, 2010. Citado na página 30.

ANDREJASIC, M. Mems accelerometers—seminar. *University of Ljubljana Faculty for Mathematics and Physics, Department of physics, Ljubljana*, 2008. Citado 3 vezes nas páginas 25, 26 e 27.

ANDROID. *The Android Source Code*. 2016. <<http://source.android.com/source/index.html>>. Acessado em 05 de Fevereiro de 2016. Citado na página 19.

ANDROID. *ART and Dalvik*. 2016. <<https://source.android.com/devices/tech/dalvik/index.html>>. Acessado em 05 de Fevereiro de 2016. Citado na página 21.

ANDROID. *Creative Vision for Android Wear*. 2016. <<http://developer.android.com/intl/pt-br/design/wear/creative-vision.html>>. Acessado em 07 de Fevereiro de 2016. Citado na página 23.

ANDROID. *Motion Sensors*. 2016. <http://developer.android.com/intl/pt-br/guide/topics/sensors/sensors_motion.html>. Acesso em 20 de Fevereiro de 2016. Citado na página 35.

ANDROID. *Platform Versions*. 2016. <<http://developer.android.com/intl/pt-br/about/dashboards/index.html#Platform>>. Acessado em 07 de Fevereiro de 2016. Citado na página 19.

ANDROID. *Sending and Syncing Data*. 2016. <<http://developer.android.com/intl/pt-br/training/wearables/data-layer/index.html>>. Acessado em 08 de Fevereiro de 2016. Citado na página 25.

ANDROID. *Sensors Overview*. 2016. <http://developer.android.com/intl/pt-br/guide/topics/sensors/sensors_overview.html>. Acessado em 07 de Fevereiro de 2016. Citado na página 23.

BISSON, E. J.; PETERSON, E. W.; FINLAYSON, M. Delayed initial recovery and long lie following a fall among middle-aged and older adults with multiple sclerosis. *Archives of physical medicine and rehabilitation*, Elsevier, 2015. Citado na página 39.

CAT, S. N. d. P. d. D. d. P. c. D. C. d. A. T. B. *Tecnologia Assistiva*. [S.l.]: CORDE, 2009. Citado na página 17.

COMMUNITY, C. open-source. *About CyanogenMod*. 2015. <<https://wiki.cyanogenmod.org/w/About>>. Acessado em 07 de Fevereiro de 2016. Citado na página 18.

CORINNE, T. *Accelerometer Basics*. 2016. <<https://learn.sparkfun.com/tutorials/accelerometer-basics>>. Acessado em 07 de Fevereiro de 2016. Citado na página 25.

DIXON-WARREN, S. *2 Billion Consumers Worldwide to Get Smart(phones) by 2016*. 2011. <<http://www.infoneedle.com/node/93219>>. Acesso em 10 de Fevereiro de 2016. Citado na página 26.

EDLICH, R. F. et al. Personal emergency response systems. *Journal of Burn Care & Research*, LWW, v. 13, n. 4, p. 453–459, 1992. Citado na página 17.

ELEKS. *Wearable Runtime & Connectivity Architectures*. 2014. <<http://elekslabs.com/2014/11/wearable-runtime-connectivity-architectures.html>>. Acessado em 07 de Fevereiro de 2016. Citado na página 24.

ELEKS. *Apple Watch vs Android Wear. Time to Drive Tesla Further*. 2015. <<http://elekslabs.com/2015/03/apple-watch-vs-android-wear-time-to-drive-tesla-further.html>>. Acessado em 07 de Fevereiro de 2016. Citado na página 24.

EMARKETER. *2 Billion Consumers Worldwide to Get Smart(phones) by 2016*. 2014. <<http://www.emarketer.com/Article/2-Billion-Consumers-Worldwide-Smartphones-by-2016/1011694>>. Acesso em 25 de Setembro de 2015. Citado na página 15.

FOTI, D.; KANAZAWA, L. Activities of daily living. *Pedretti's Occupational Therapy: Practice Skills for Physical Dysfunction*, v. 6, p. 146–194, 2008. Citado na página 43.

GOOGLE. *MessageApi*. 2015. <<https://developers.google.com/android/reference/com/google/android/gms/wearable/MessageApi>>. Acessado em 07 de Fevereiro de 2016. Citado na página 25.

GUIDE, A. D. *Android NDK*. 2016. <<http://developer.android.com/intl/pt-br/tools/sdk/ndk/index.html>>. Acessado em 07 de Fevereiro de 2016. Citado na página 18.

GUJARATI, P. *What is Accelerometer and how does it work on smartphones*. 2015. <<http://www.techulator.com/resources/8930-How-does-smart-phone-accelerometer-work.aspx>>. Acessado em 10 de Fevereiro de 2016. Citado na página 27.

HABIB, M. A. et al. Smartphone-based solutions for fall detection and prevention: challenges and open issues. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 14, n. 4, p. 7181–7208, 2014. Citado 4 vezes nas páginas 31, 32, 36 e 37.

HALL, M. et al. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, ACM, v. 11, n. 1, p. 10–18, 2009. Citado na página 31.

HEINBÜCHNER, B. et al. Satisfaction and use of personal emergency response systems. *Zeitschrift für Gerontologie und Geriatrie*, Springer, v. 43, n. 4, p. 219–223, 2010. Citado na página 17.

IDC. *Fueled by Growing Demand for Smart Wearables, IDC Forecasts Worldwide Wearable Shipments to Reach 173.4 Million by 2019*. 2015. <<http://www.idc.com/getdoc.jsp?containerId=prUS25903815>>. Acesso em 25 de Setembro de 2015. Citado na página 15.

IDC. *Smartphone OS Market Share, 2015 Q2*. 2016. <<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>>. Acessado em 06 de Fevereiro de 2016. Citado na página 18.

- INNOVENTIONS. *Accelerometers*. 2015. <<http://www.rotoview.com/accelerometer.htm>>. Acessado em 10 de Fevereiro de 2016. Citado na página 26.
- KAZI, S. B. et al. Fall detection using single tri-axial accelerometer. In: *ASEE 2014 Zone I Conference*. [S.l.: s.n.], 2014. Citado na página 15.
- KIRCHER, K. *Low-Pass Filter: The Basics*. 2014. <<http://www.kircherelectronics.com/blog/index.php/11-android/sensors/8-low-pass-filter-the-basics>>. Acesso em 20 de Fevereiro de 2016. Citado na página 35.
- KULKARNI, C. D. *Design and Development of a Smartphone-based Fall Detection and Alerts Generation System*. Dissertação (Mestrado) — NC State University, 2011. Citado na página 30.
- LEAL, N. G. d. V. *Dominando o Android, do básico ao avançado*. [S.l.]: Novatec, 2015. Citado 4 vezes nas páginas 18, 23, 24 e 25.
- LEE, W.-M. *Introdução ao Desenvolvimento de Aplicativos para o Android*. [S.l.]: Editora Ciência Moderla Ltda., 2011. Citado 2 vezes nas páginas 18 e 21.
- LUQUE, R. et al. Comparison and characterization of android-based fall detection systems. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 14, n. 10, p. 18543–18574, 2014. Citado na página 15.
- MEHNER, S.; KLAUCK, R.; KOENIG, H. Location-independent fall detection with smartphone. In: *ACM. Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments*. [S.l.], 2013. p. 11. Citado na página 32.
- OHA. *Open Handset Alliance*. 2007. <<http://www.openhandsetalliance.com>>. Acessado em 01 de Fevereiro de 2016. Citado na página 18.
- OJEDA-ZAPATA, J. *Android Wear Watches Now Work with the iPhone*. 2015. <<http://tidbits.com/article/16030>>. Acessado em 09 de Fevereiro de 2016. Citado na página 23.
- OMS. *WHO Global Report on Falls Prevention in Older Age*. 2007. <<http://www.who.int/mediacentre/news/releases/2015/older-persons-day/en/>>. Acesso em 25 de Setembro de 2015. Citado na página 15.
- OMS. *Number of people over 60 years set to double by 2050; major societal changes required*. 2015. <<http://www.who.int/mediacentre/news/releases/2015/older-persons-day/en/>>. Acesso em 05 de Novembro de 2015. Citado na página 14.
- OMS. *World report on Ageing And Health*. 2015. <http://apps.who.int/iris/bitstream/10665/186463/1/9789240694811_eng.pdf?ua=1>. Acesso em 29 de Outubro de 2015. Citado na página 14.
- ONU. *Projeção da pirâmide populacional brasileira para 2050*. 2015. <<http://esa.un.org/unpd/wpp/Graphs/>>. Acesso em 08 de Novembro de 2015. Citado na página 14.
- PANAGIOTIS, K. et al. Increased fall detection accuracy in an accelerometer-based algorithm considering residual movement. In: *4th International Conference on Pattern Recognition Applications and Methods, At Lisbon, Portugal*. [S.l.: s.n.], 2014. Citado na página 15.

PANAGIOTIS, K. et al. F2D: A fall detection system tested with real data from daily life of elderly people. In: *IEEE HEALTHCOM 2015, 17th International Conference on E-health Networking, Application & Services*. Boston: IEEE, 2015. Citado 4 vezes nas páginas 30, 35, 37 e 38.

PORTER, E. J. Moments of apprehension in the midst of a certainty: some frail older widows' lives with a personal emergency response system. *Qualitative health research*, Sage Publications, v. 13, n. 9, p. 1311–1323, 2003. Citado na página 17.

RAJENDRAN, P. et al. Falls, fall prevention, and fall detection technologies. In: *Eldercare technology for clinical practitioners*. [S.l.]: Springer, 2008. p. 187–202. Citado na página 29.

RAWASSIZADEH, R.; PRICE, B. A.; PETRE, M. Wearables: Has the age of smartwatches finally arrived? *Communications of the ACM*, ACM, v. 58, n. 1, p. 45–47, 2015. Citado na página 30.

SMITH, D.; FRIESEN, J. *Android Recipes*. [S.l.]: Apress, 2011. Citado na página 21.

SPOSARO, F. *iFall: Fall Monitoring System*. 2009. <<http://ww2.cs.fsu.edu/~sposaro/iFall/>>. Acesso em 18 de Fevereiro de 2016. Citado na página 29.

SPOSARO, F.; TYSON, G. ifall: an android application for fall monitoring and response. In: IEEE. *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*. [S.l.], 2009. p. 6119–6122. Citado na página 29.

TAM. *F2D*. 2015. <<http://tam.unige.ch/projects/f2d.html>>. Acesso em 18 de Fevereiro de 2016. Citado na página 30.

VITALCALL. *ERICA Medical Alarm System with Pendant or Wristband*. 2015. <<http://www.vitalcall.com.au/products/monitoring-systems/erica-personal-monitoring-system>>. Acesso em 15 de Fevereiro de 2016. Citado na página 28.

WILD, D.; NAYAK, U.; ISAACS, B. How dangerous are falls in old people at home? *Br Med J (Clin Res Ed)*, British Medical Journal Publishing Group, v. 282, n. 6260, p. 266–268, 1981. Citado na página 15.