

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

MARTIN REUS

***Detector: um sistema de monitoramento de
pessoas utilizando microcontroladores e
Smartphones***

Trabalho de Graduação.

Prof. Dr. Luciana Nedel
Orientador

Porto Alegre, julho de 2012

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Martin Reus,

Detector: um sistema de monitoramento de pessoas utilizando microcontroladores e *Smartphones* /

Martin Reus. – Porto Alegre: Graduação em Ciência da Computação da UFRGS, 2012.

60 f.: il.

Trabalho de Conclusão (bacharelado) – Universidade Federal do Rio Grande do Sul. Curso de Ciência da Computação, Porto Alegre, BR-RS, 2012. Orientador: Luciana Nedel.

1. Acelerômetros. 2. Android. 3. Java. 4. Monitoramento. 5. Sistemas embarcados. 6. Smartphones. 7. Tecnologias assistivas. I. Nedel, Luciana. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitora de Graduação: Prof^a. Valquiria Linck Bassani

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do CIC: Prof. Raul Fernando Weber

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Agradeço aos meus pais, pela criação e por todos os valores e educação que me foram passados por eles. Por sua dedicação, em todos momentos da minha vida e por sempre me mostrarem o caminho do bem.

Agradeço à minha vó, pela constante presença na minha vida e por sempre mostrar o lado alegre e bom da vida. Por ser motivo de inspiração na superação de problemas, na sua energia e pela força de vontade em viver.

Agradeço à minha orientadora que, além de conselhos na criação deste trabalho, é fonte de inspiração na sua área de pesquisa, bem como nas matérias que leciona, algumas das quais tive o privilégio de frequentar.

Agradeço também à minha namorada, por compartilhar minha vida e fazê-la especial, pelo apoio e pela compreensão em momentos difíceis; aos meus parentes, que de uma forma ou de outra influenciaram no meu crescimento pessoal e profissional, e que foram fundamentais na criação do meu caráter; aos amigos, pelo apoio na jornada, pelos momentos de alegria e companheirismo.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	7
LISTA DE FIGURAS	9
LISTA DE TABELAS	11
RESUMO	13
ABSTRACT	15
1 INTRODUÇÃO	17
1.1 Objetivos do trabalho	18
1.2 Organização	18
2 CONTEXTUALIZAÇÃO E TRABALHOS RELACIONADOS	19
2.1 Smartphones e o sistema operacional Android	19
2.1.1 <i>Android Activity</i>	20
2.1.2 <i>Android Services</i>	20
2.1.3 <i>Android BroadcastReceiver</i>	20
2.2 Machine Learning	20
2.3 Acelerômetro	22
2.3.1 Identificação em tempo real de atividade realizada por pessoas utilizando acelerômetros	23
2.3.2 <i>Hardware</i> para captura de dados de acelerômetros	23
2.3.3 Análise dos dados de acelerômetro	24
2.4 Aparelhos médicos domésticos	26
2.4.1 Bluetooth Health Device Profile	27
2.5 Trabalhos anteriores	28
2.5.1 Reconhecimento de atividades usando dados anotados de acelerômetros	28
2.5.2 Utilizando acelerômetros para reconhecimento de atividades	28
2.5.3 Plataforma de sensores móvel	28
2.5.4 Reconhecimento de atividades utilizando celular	29
2.5.5 Comentários finais	29
3 DETECTOR: UM SISTEMA DE MONITORAMENTO DE PESSOAS	31
3.1 Visão geral do Detector	31
3.1.1 Módulo de detecção de atividades	31
3.1.2 Módulo de visualização	33
3.1.3 Funcionamento geral	33

3.2	A detecção de movimento no celular	34
3.2.1	Captura de dados anotados para treinamento do classificador	34
3.2.2	Extração de características das ondas	37
3.2.3	Construção da instância classificadora de atividades	37
3.2.4	Detecção em tempo real no <i>Smartphone</i>	39
3.3	Captura de dados através de sensor infra-vermelho	41
3.4	Ferramenta de visualização WEB	42
3.4.1	Funcionamento do <i>website</i>	44
4	RESULTADOS E ANÁLISE DO <i>DETECTOR</i>	45
4.1	Metodologia de avaliação de resultados	45
4.2	A detecção de atividades utilizando <i>Smartphones</i>	45
4.2.1	Duração da bateria	46
4.2.2	Acuidade da detecção	47
4.2.3	Pesquisa preliminar sobre uso do <i>Detector</i>	48
4.2.4	Privacidade e invasividade	48
4.3	O sensor infravermelho	49
4.4	Análise do <i>website</i>	49
4.4.1	Pesquisa preliminar sobre uso do <i>website</i>	53
4.4.2	Correlação entre gráficos	54
4.4.3	Continuidade dos dados	54
4.5	Considerações sobre a utilização do <i>Detector</i>	55
5	CONCLUSÃO E TRABALHOS FUTUROS	57
5.1	Detecção de atividades	57
5.2	Outras melhorias	58
	REFERÊNCIAS	59

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
HTML	Hypertext Markup Language
IDE	Integrated Development Environment
BHDP	Bluetooth Health Device Profiles
JVM	Java Virtual Machine
TA	Tecnologia Assistiva
SD	Secure Digital
SDK	Software Development Kit
PDA	Personal Digital Assistant, também conhecido por Palm Top Computer
VM	Virtual Machine
SO	Sistema Operacional
ROM	Read Only Memory
RAM	Random Access Memory
WEB	Teia, fazendo referência à Internet.

LISTA DE FIGURAS

Figura 2.1:	Árvore de decisão para verificar se devemos jogar tênis (imagem retirada do livro (Mitchell 1997)).	21
Figura 2.2:	Acelerômetro soldado em placa de circuito impresso	22
Figura 2.3:	Gráfico de aceleração do eixo X do acelerômetro capturado no <i>Detector</i>	24
Figura 2.4:	Medidor de nível glicêmico	26
Figura 2.5:	Funcionamento do protocolo <i>BHDP</i>	27
Figura 3.1:	Celular LG P990 com Android	32
Figura 3.2:	Sensor de infravermelho	32
Figura 3.3:	Coletor de dados da Autech	33
Figura 3.4:	Diagrama de funcionamento do sistema	33
Figura 3.5:	Interface do aplicativo para captura de dados	35
Figura 3.6:	Dados do acelerômetro (eixo X) capturados a uma taxa de 20Hz	36
Figura 3.7:	Dados do acelerômetro (eixo X) capturados a uma taxa de 50Hz	36
Figura 3.8:	<i>Wii mote</i>	36
Figura 3.9:	Etapa de pré-processamento	38
Figura 3.10:	Parte do arquivo "dados.ARFF", contendo instâncias do trabalho	39
Figura 3.11:	Código fonte da construção do classificador J48	39
Figura 3.12:	Diagrama de funcionamento do <i>Detector</i>	40
Figura 3.13:	Serviço de captura dos dados	41
Figura 3.14:	Captura da tela do <i>website</i>	43
Figura 4.1:	Gráfico dos testes de duração da bateria	47
Figura 4.2:	Gráfico de um dia inteiro de captura de dados do <i>Smartphone</i>	50
Figura 4.3:	Gráfico de uma hora de captura de dados do <i>Smartphone</i>	50
Figura 4.4:	Gráfico de um dia de captura de dados do sensor infravermelho	51
Figura 4.5:	Gráfico de uma hora de captura de dados do <i>Smartphone</i> , com dados apresentados na ordem invertida	52
Figura 4.6:	Gráfico de área de um dia de captura de dados do <i>Smartphone</i>	52
Figura 4.7:	Gráfico de área de um dia de captura de dados do sensor infravermelho	52
Figura 4.8:	Gráfico do tipo <i>column</i>	53
Figura 4.9:	Gráfico do tipo <i>line</i>	53
Figura 4.10:	Gráfico que mostra a correlação entre os dados capturados	54
Figura 4.11:	Gráfico com dados incompletos em uma hora	55
Figura 4.12:	Gráfico do dia aparentemente completo	55

LISTA DE TABELAS

Tabela 2.1:	Comparativo das características dos trabalhos relacionados ao <i>Detector</i>	29
Tabela 2.2:	Comparativo das características dos trabalhos relacionados ao <i>Detector</i> (continuação)	30

RESUMO

A saúde ou limitações na locomoção de uma pessoa são fatores que podem influenciar na sua independência. Prover uma forma de monitoramento eficiente, com a menor invasividade possível, pode ser um meio de aumentar o grau de independência dessas pessoas.

O presente trabalho descreve o desenvolvimento de uma ferramenta de monitoramento de pessoas, utilizando para isto detecção de atividades por meio de acelerômetro presente em *Smartphones*. Além do celular, um sensor infravermelho conectado a um sistema embarcado coleta os dados de movimentação extra para suprir a carência dos dados do telefone. Para completar, a visualização dos dados em tempo real oferece uma maneira para melhor acompanhamento da pessoa monitorada. O sistema permite que se possa acompanhar a rotina de uma pessoa para que ela se sinta mais segura, ou para que familiares ou amigos possam acompanhar suas atividades, oferecendo-lhe, deste modo, mais independência e segurança.

O segundo capítulo elucidava um pouco da teoria da detecção das atividades, bem como apresenta algumas das ferramentas utilizadas na implementação. A seguir, a concepção de *Detector* é explicada. Uma análise dos resultados obtidos com a construção do sistema é apresentada no quarto capítulo. Finalmente, é feita a conclusão sobre os resultados obtidos, e são apontadas possíveis melhorias para o *Detector*.

Palavras-chave: Acelerômetros, Android, Java, monitoramento, sistemas embarcados, Smartphones, tecnologias assistivas.

Detector: a platform for measuring a person's activity using mobile phones and embedded hardware

ABSTRACT

A person's health or reduced mobility may impact on this person's overall autonomy. Providing them with an efficient and as minimally obtrusive as possible monitoring system may improve their independence, by reducing the need of someone having to take care of them.

This undergraduation thesis presents the implementation of *Detector*: a monitoring system able to detect some simple activities performed by a person with the aid of a Smartphone's accelerometer. To complement the lack of information from the cellphone, which will not always be detecting activities, an infrared sensor connected to an embedded system records the person's movement inside his house. Furthermore, a real time visualisation tool was developed to display the information collected from both systems. The whole software can then be used to help family members or friends to have a greater sense of security and peace of mind, knowing that everything is as it should be with someone. This person of interest can also be more self confident and independent.

A general overview of activity detection can be found in the second chapter, as well as information about the hardware and software needed for this application development. The third chapter explains how *Detector* was built. In the fourth chapter, an analysis of the results from this implementation is discussed. For the final part, this work presents a general overview of the system, pointing out some of its further improvement possibilities.

Keywords: Accelerometer, Android, Java, monitoring, embedded hardware, Smartphones, assistive technologies.

1 INTRODUÇÃO

Muitos idosos vivem sozinhos pois desejam ter sua independência. São frequentes acidentes com tapetes e outros objetos que não representariam perigo para pessoas mais jovens. Apesar de não ocorrerem somente com idosos, esses acidentes podem representar um perigo muito maior à sua saúde pelo simples fato de o corpo já não suportar agressões tão intensas. Além disso, idosos são mais propensos a acidentes vasculares, cardíacos, etc. Por estes motivos, ainda que sejam capazes de viver sozinhos e de forma independente, convém que sejam de alguma forma monitorados pela família.

A *Tecnologia Assistiva* (ou *TA*) (Bersch 2012), termo relativamente novo que crescentemente ganha atenção, diz respeito a tecnologias que permitem ou facilitam a reabilitação ou inclusão de pessoas portadoras de deficiência ou com algum tipo de mobilidade reduzida na sociedade. *TA*, entre outras coisas, descreve e identifica **recursos** e **serviços** que integrem a tecnologia atual a fim de facilitar e tornar mais seguras as vidas dessas pessoas. **Recursos** são os materiais e produtos utilizados para alcançar este objetivo.

Tomando o **recurso** da *TA* como exemplo, pode-se pensar em um sistema que monitorea a atividade de uma pessoa no seu dia-a-dia. Esse monitoramento não tem utilidade somente para indivíduos portadores de deficiência, mas também para idosos - que não propriamente têm alguma deficiência - e pessoas em geral. No caso dos idosos, um sistema de monitoramento pode ser instalado para identificar acidentes ou situações que fogem à sua rotina. Acidentes físicos (quedas, fraturas) são comuns, e ter uma maneira de detectá-los faz com que o tempo de resposta a um desses eventos possa ser fator decisivo no atendimento que o idoso terá.

A existência de um registro do histórico de atividades é muito útil para identificar, a partir de dados pré determinados, se uma pessoa tem uma atividade física considerada normal - neste caso sem riscos, como por exemplo quedas ou ociosidade excessiva. Pode ser uma ferramenta para se verificar quão regularmente ela pratica exercícios, e consequentemente a sua qualidade de vida. Monitorar uma pessoa pode significar um aumento na sua própria independência, pois evita que ela tenha de estar sob constante vigilância médica ou familiar. Se o sistema é responsivo, podemos pensar também que a segurança de sua vida é maior, pois a qualquer momento alguém responsável pode ser notificado de uma situação imprevista.

Criar um sistema de monitoramento não é tarefa fácil, pois existem diversos aspectos importantes a serem considerados. A invasão de privacidade é um deles. Como podemos monitorar uma pessoa sem ferir o seu direito de se reservar em um dado momento? Uma

outra questão relativa ao monitoramento é o quão invasivo é o sistema. Quanto menos fios, caixas e aparelhos estiverem aparentes, maior será a naturalidade do sistema e menor será o incômodo gerado pelo mesmo.

1.1 Objetivos do trabalho

O objetivo do presente trabalho é criar um sistema capaz de extrair informações contextuais importantes das atividades cotidianas realizadas por uma pessoa e, de acordo com esses dados, aferir sua integridade física. Ao mesmo tempo, o sistema deve ser o menos invasivo possível. Serão exploradas algumas das formas de se praticar esse monitoramento e grande parte do trabalho será baseado em alguns artigos e outras referências que utilizam acelerômetros para captura de movimento, tais como as pesquisas encontradas em (Ravi et al. 2005) e (Choudhury et al. 2008).

No que diz respeito ao hardware utilizado para captura e classificação dos movimentos, será usado um *Smartphone* com sistema operacional *Android*[®]. Os celulares modernos têm inúmeros recursos e grande poder computacional. Esses aparelhos comumente possuem acelerômetros (*hardware* que mede a aceleração relativa sendo aplicada sobre um objeto, no caso o telefone) capazes de medir movimento, o que os tornam ideais para o objetivo do trabalho. Além disso, o celular é um objeto que normalmente está junto e é útil à pessoa, e pode ainda ter conexão direta com a Internet, o que o torna muito interessante no quesito armazenamento e distribuição dos dados coletados.

Além disso, será utilizado *hardware* embarcado, um coletor de dados da empresa *Autech*, em conjunto com sensores de infravermelho, para medir a movimentação de um indivíduo pela sua casa. Este *hardware* poderá, por sua vez, se comunicar com um banco de dados remoto. Por fim, uma ferramenta de visualização na WEB será implementada para apresentar de uma maneira clara e objetiva os dados coletados, a fim de se monitorar a atividade física diária de uma pessoa e também possivelmente indicar anormalidades na sua rotina.

1.2 Organização

A organização do trabalho é a que segue. No capítulo 2 apresentamos uma contextualização de algumas das ferramentas usadas no decorrer do trabalho, bem como trabalhos relacionados a este; no capítulo 3 é apresentado o desenvolvimento de *Detector*, uma ferramenta de monitoramento de pessoas e visualização dos dados pela internet; o capítulo 4 apresenta os resultados do trabalho desenvolvido no capítulo 3; finalmente, no capítulo 5 são apresentadas as conclusões e possíveis aperfeiçoamentos futuros para o *Detector*.

2 CONTEXTUALIZAÇÃO E TRABALHOS RELACIONADOS

O seguinte capítulo contextualiza algumas das ferramentas e técnicas que serão utilizadas no decorrer do trabalho, como os *Smartphones* com sistema operacional *Android* e seu hardware. Uma breve explanação sobre *machine learning* também se faz necessária, dado que este tipo de algoritmo será utilizado para inferir tipos de atividades executadas pelo usuário do *Detector*. Também são apresentados pesquisas e trabalhos relacionados à captura de movimento por meio de acelerômetros.

2.1 *Smartphones* e o sistema operacional *Android*

A definição de *Smartphones*, ou celulares inteligentes¹, é bastante ampla. Além da função básica de telefonia, estes aparelhos têm recursos que comumente encontramos em um computador, mas de forma mais compacta. É possível ter acesso à Internet, instalar novas aplicações como jogos e ferramentas de produtividade. O telefone celular como era conhecido há 10 anos ganhou novas funcionalidades e passou a ser uma ferramenta de auxílio nas tarefas diárias de uma pessoa, bem como de entretenimento.

O *Android* é um sistema operacional baseado no *Kernel* do Linux. Além dos drivers de dispositivos como dos acelerômetros, *touchscreen*, câmeras e *Wifi*, o *Android* tem um extenso conjunto de bibliotecas - *libraries* - que permitem um rápido desenvolvimento de aplicações na linguagem *Java*. Além disso, o desenvolvimento para *Android* é facilitado por um plugin para a *IDE - Integrated Development Environment* - do *Eclipse*, aliado a um *SDK - Software Development Kit* - para a plataforma, que contém todas essas ferramentas e bibliotecas.

Para permitir a programação diretamente em *Java* no celular, o *Google* implementou uma versão para dispositivos móveis de uma máquina virtual chamada **Dalvik VM**. Isto permite que o programador possa utilizar uma vasta quantidade de *libraries* já existentes no mercado, para *Java*, requerendo nenhuma ou pouca adaptação de código. No decorrer do trabalho verificou-se que, exceto em raras exceções, *libraries* destinadas para implementações de *JVM* da *Sun* foram diretamente compatíveis no *Android*.

A compreensão de algumas classes comumente utilizadas no *Android* se faz necessária, pois elas são importantes no decorrer do trabalho.

¹Tradução livre

2.1.1 *Android Activity*

A *Activity*, no *Android*, é uma classe base necessária para criar interfaces visuais renderizáveis pelo sistema. Geralmente, cada *Activity* representa uma visão em um aplicativo. As *Activities* não devem ser usadas para processamentos intensos, ou que lidem com comunicação de rede, uma vez que isto visualmente "tranca" a interface de usuário. Para este propósito, devem ser usados *Services*.

2.1.2 *Android Services*

Services são classes que se comportam como processos separados. São requisitados geralmente por uma *Activity*, por outro *Service* ou ainda por *BroadcastReceiver*. Um *Service* não possui interface visual, e pode ou não compartilhar a mesma *Thread* de uma *Activity*. Portanto, um processamento intenso ou comunicação de rede é possível sem que haja um comprometimento da fluidez de interação com a interface gráfica notada pelo usuário do aparelho. Múltiplos *Services* podem ser iniciados a partir de uma única aplicação, o que caracteriza um cenário de multi-tarefa.

2.1.3 *Android BroadcastReceiver*

Um *BroadcastReceiver* é uma classe capaz de receber eventos de tipos variados, oriundos do sistema operacional, e executar uma breve ação. Um evento do sistema pode ser, por exemplo, um aviso de desligamento de tela, ou ainda um horário de alarme assinado para uma classe específica que estenda desse *BroadcastReceiver*. No dois casos, o *BroadcastReceiver* é chamado pelo sistema operacional e executa algumas operações de curta duração. A duração deve ser breve, pois o intuito dessa classe é executar operações simples ou então delegar uma função mais complexa a um *Service*. O SO do *Android* só disponibiliza uma curta fatia do processador para esse tipo de classe e, se a operação não for finalizada a tempo, o processo é terminado.

2.2 *Machine Learning*

Machine Learning, ou aprendizado de máquina¹, é a definição de como uma máquina pode aprender ou aperfeiçoar tarefas de uma maneira mais autônoma e eficiente. Talvez uma das definições mais elegantes e resumidas para *machine learning* é descrita por Tom M. Mitchell em seu livro (Mitchell 1997):

"Um programa de computador aprende com uma experiência **E** relativa a uma classe de tarefas **T** e medida de performance **P**, se sua performance nas tarefas **T**, medida por **P**, é aumentada pela experiência com **E**."¹

De uma maneira genérica, pode-se dizer que, se um programa é capaz de melhorar os resultados de suas saídas de dados a partir de certos dados de entrada, este programa aprendeu com experiências passadas. A noção do aprendizado de máquina parece bastante óbvia. No entanto, a tarefa de aprendizado é bastante complexa, pois os computadores, e a programação dos mesmos, é determinística. Um exemplo de como um programa pode aprender, é o que será usado neste trabalho; uma árvore de decisão (Figura 2.1).

¹Tradução livre

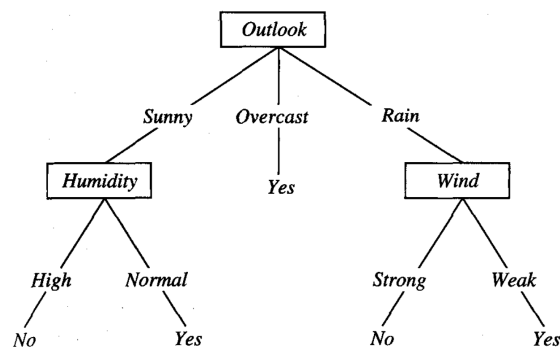


Figura 2.1: Árvore de decisão para verificar se devemos jogar tênis (imagem retirada do livro (Mitchell 1997)).

O aprendizado por árvore de decisão é um método para aproximação de funções de valores discretos que são representadas por árvores de decisão, sendo um método bastante usado para aprendizado de máquina. As árvores de decisão classificam uma instância percorrendo a árvore da raiz até que se chegue em uma folha que corresponda à classe da instância. Cada nodo da árvore especifica um teste para um atributo da instância a ser testada - instância essa que pode ter vários atributos.

Antes da etapa de classificação de instâncias, contudo, a árvore tem de ser construída. A fase de treinamento se constitui em adquirir uma série de instâncias, cada uma com diversos atributos. Para cada instância sabe-se qual a classe à qual ela pertence. No caso do presente trabalho, cada instância é composta por diversos atributos que são os cálculos heurísticos, descritos na subseção 2.3.3, que fazemos com base nos dados coletados de um acelerômetro, e por um *label* que diz qual a sua classe - correndo, caminhando ou parado.

Além das árvores de decisão, outros tipos de algoritmos, que serão comentados no decorrer deste trabalho, são comumente utilizados para realizar inferência de classes em conjuntos de instâncias. A seguir há uma listagem e breve explicação sobre como alguns desses algoritmos funcionam, inclusive o que foi utilizado no desenvolvimento do *Detector*.

- **C4.5:** este algoritmo é uma árvore de decisão, treinada a partir de um conjunto de instâncias, cada uma com atributos diversos e uma classe que as identifica. Cada nodo da árvore seleciona um dos atributos de maior significância para dividir o conjunto de acordo com suas classes. A árvore, uma vez construída, pode ser usada para classificar novas instâncias que não fazem parte do conjunto original.
- **Multilayer Perceptron:** é uma rede neural de múltiplas camadas, em que usualmente cada neurônio de uma camada tem conexão com neurônios da camada posterior.
- **Plurality Voting:** é um sistema de votação. As classes disponíveis são votadas de acordo com um vetor de votação e uma instância de entrada. A classe que recebe mais votos – não necessariamente mais da metade deles – é a classe que corresponde à instância.

- **J48**: esta é uma implementação *open source*, desenvolvida na linguagem Java, do algoritmo C4.5.

O **WEKA** é um conjunto de ferramentas que contém diversos algoritmos conhecidos para *machine learning* e mineração de dados. A aplicação é escrita em Java e possui uma interface gráfica para fácil manuseio. **WEKA** é *open-source* e a sua biblioteca de ferramentas pode ser utilizada para fins como o deste trabalho. Como a biblioteca é escrita em Java, ela pôde ser adaptada para uso no celular com *Android* – excluindo-se a parte de interface visual.

2.3 Acelerômetro

O acelerômetro (Figura 2.2) é um pequeno dispositivo capaz de medir a aceleração absoluta atuando sobre ele. Os acelerômetros mais comuns encontrados atualmente no mercado geralmente medem a aceleração absoluta exercida nos seus eixos relativos cartesianos X, Y e Z. Significa que, se o dispositivo está em queda livre, em direção ao solo, a aceleração medida nos seus três eixos deverá ser próxima a zero, descartando-se erros de leitura.

Normalmente a unidade de medida expressa pela leitura dos valores de saída é um valor de aceleração. Então, se há um acelerômetro em repouso em cima de uma mesa, o valor de sua saída deverá ser aproximadamente 0,0,9.8 para os eixos cartesianos X, Y, Z respectivamente, provido que o eixo Z esteja apontando diretamente para o centro da terra. Teremos **1g** de aceleração no eixo Z, pois há uma magnitude de gravidade, ou aproximadamente $9.8m/s^2$, atuando sobre aquele eixo.

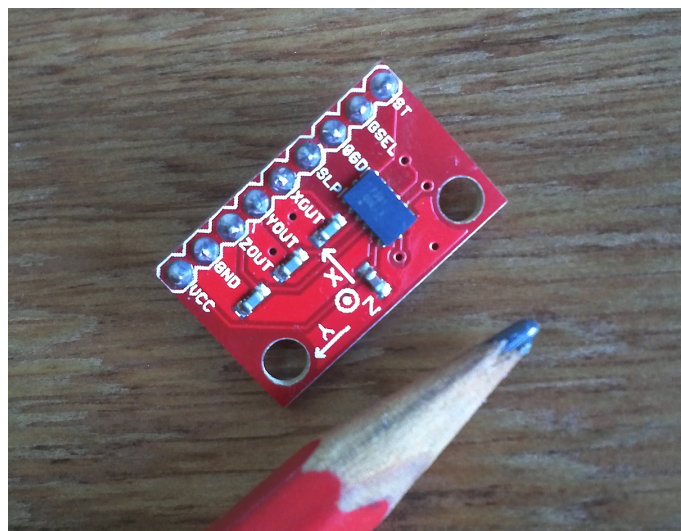


Figura 2.2: Acelerômetro soldado em placa de circuito impresso

O aperfeiçoamento e barateamento de sensores eletrônicos relativamente complexos como acelerômetros permitiu que eles se popularizassem como ferramenta adicional aos recursos de *Smartphones* como *Iphone* (Apple Inc.) e celulares baseados no sistema operacional *Android* (Android Mobile Platform). O uso mais comum deste tipo de hardware

é permitir a identificação da posição do celular em relação ao solo. O próprio sistema operacional pode, então, rotacionar a *Home screen* para que sempre fique orientada corretamente em relação ao usuário, dependendo da inclinação em que o aparelho se encontra.

Podemos citar outros usos deste tipo de *hardware*, como sistemas de direcionamento de foguetes e detectores de queda em *notebooks*. Essa última aplicação é bastante interessante: o *software* que monitora os valores do acelerômetro identifica quando os três eixos cartesianos X, Y, e Z registram um valor muito próximo a zero, significando que o computador está em queda livre. Este sistema então recolhe o mais prontamente possível a agulha do disco rígido, evitando danos mais graves ao aparelho e consequente perda de dados.

2.3.1 Identificação em tempo real de atividade realizada por pessoas utilizando acelerômetros

É natural sermos levados a pensar que diversas das nossas atividades físicas oferecem uma diferente *assinatura* digital. O impacto exercido em nossos membros inferiores, em termos de força *g*, é diferente quando caminhamos, quando corremos ou estamos sentados. Quando sentados, estamos teoricamente num estado de repouso, sem movimentação aparente. Um acelerômetro, que é capaz de medir esses impactos em termos de aceleração da massa, com um devido algoritmo, pode diferenciar essas atividades desenvolvidas com um certo grau de acerto.

A captura de movimentos, como citada na introdução deste trabalho, não é ideia nova. Uma quantidade relativamente grande de trabalhos científicos aborda a questão do monitoramento de pessoas utilizando-se *hardware* com acelerômetros, e há uma grande convergência entre eles. A maioria provê uma detecção do tipo de atividade sendo realizada pelo indivíduo que carrega os sensores.

Muitos dos trabalhos existentes são bastante recentes, dado que somente nos últimos anos o poder computacional e o tamanho dos sistemas embarcados viabilizasse uma solução para esse tipo de monitoramento. O sistema deveria ser computacionalmente poderoso para coletar e processar uma generosa quantidade de dados analógicos, além de eficiente em questões de duração da bateria. Além disso, suficientemente pequeno para que não fosse um fardo para quem o carregasse. Grande parte dos *Smartphones* atuais cobrem uma boa parcela, senão a totalidade, desses requisitos.

Pesquisas como de (Choudhury et al. 2008), de (Ravi et al. 2005) e de (Kwapisz et al. 2010) utilizam acelerômetros e hardware embarcado na tarefa de coleta e decisão de que tipo de atividade está sendo executada pelo indivíduo. Dos dados do acelerômetro são retiradas algumas informações, explicadas posteriormente, que são encaminhadas para um algoritmo de decisão. O algoritmo de decisão é previamente treinado, utilizando-se técnicas de *machine learning*, com dados coletados de diferentes pessoas e diferentes atividades. Os dados coletados de cada atividade, na etapa de treino, são marcados com *labels* que servem de referência futura.

2.3.2 Hardware para captura de dados de acelerômetros

O hardware de captura de dados pode ser construído de diversas formas. Dependendo de como ele é construído, o sistema se torna mais invasivo, conforme observamos em

(Ravi et al. 2005) e (Bao e Intille 2004). Idealmente, queremos um hardware que possa adquirir dados da movimentação de uma pessoa e identificar a atividade que ela está realizando, sem que ela perceba ou se sinta incomodada pelo sistema.

Um ou mais acelerômetros são usados para medir os movimentos corporais de uma pessoa, que deve carregá-los junto ao seu corpo. Quanto mais acelerômetros existirem, melhor será a precisão do sistema, mas também maior será o poder de processamento necessário para analisar os dados. Segundo (Bao e Intille 2004), membros superiores e inferiores como braços e pernas podem ser submetidos a **+12g** de força. A região da cintura pode experimentar forças próximas a **+6g**. Os acelerômetros utilizados em (Bao e Intille 2004) possuem amplitude de medição de **+10g**, embora acelerômetros com amplitude de medição de **+2g** tenham se mostrado suficientemente eficientes para realizar a tarefa (Kwapisz et al. 2010).

Em trabalho mais recente, (Kwapisz et al. 2010) utilizam um *Smartphone* para monitorar pessoas. Ele cita que existem diversas vantagens em se utilizar um aparelho celular para realizar a tarefa. Uma delas é que o celular está cada vez mais presente no cotidiano das pessoas, sendo um objeto útil e não obstrutivo a elas. Outro fator importante é a existência do *SDK* e do *plugin* para a *IDE* do Eclipse, citada na seção 2.1. Além disso, a diversidade de opções quanto a diferentes formas de se conectar com a rede mundial de computadores foi outro motivo de sua escolha.

2.3.3 Análise dos dados de acelerômetro

Após adquiridos os dados do acelerômetro, vem a etapa de interpretação dos dados. Todos os artigos pesquisados utilizam *machine learning* como técnica de reconhecimento de atividade, variando apenas os algoritmos de avaliação. Usar *machine learning* implica treinar os algoritmos com dados que sabidamente pertencem a um tipo de atividade.

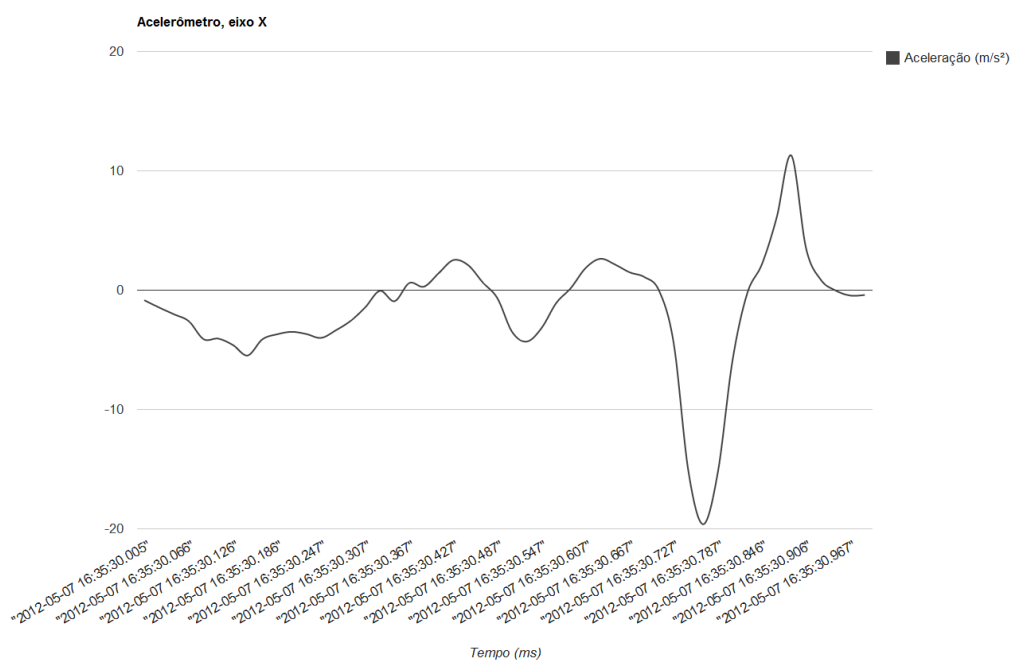


Figura 2.3: Gráfico de aceleração do eixo X do acelerômetro capturado no *Detector*.

Para processar as ondas analógicas (Figura 2.3) geradas por cada eixo do acelerômetro, os artigos descrevem variadas heurísticas para pré-processar os dados analógicos e extrair informação que pode ser usada no algoritmo de inteligência artificial. Os dados devem ser pré-processados pois os algoritmos de aprendizado de máquina geralmente não funcionam tão bem com séries cruas de dados. Abaixo citamos alguns dos cálculos utilizados:

- Média aritmética do valor de cada um dos eixos do acelerômetro: o cálculo da média aritmética para cada eixo é útil para verificar a posição do aparelho em relação ao solo. Quando se corre ou caminha, teoricamente o celular encontra-se de pé dentro do bolso da calça. Ao sentar, normalmente sua posição é deitada, e a média identifica estes comportamentos.
- Tempo entre picos de ondas (Kwapisz et al. 2010): atividades como caminhadas ou corridas tendem a gerar uma onda repetitiva. O artigo descreve a heurística utilizada para detectar os picos, para obter uma média entre as distâncias. Esse dado foi importante para diferenciar atividades como corrida e caminhada, que tem tempos entre picos de onda diferentes.
- Desvio padrão (Ravi et al. 2005): o desvio padrão é um bom indicativo da amplitude da variação ocorrida no sinal e é diretamente relacionado à intensidade da atividade. Quando se está em repouso o desvio padrão é comparativamente muito menor que o desvio padrão do sinal proveniente de uma amostra de corrida. O cálculo para esta característica é o que segue:

$$\sigma_x = \sqrt{\frac{1}{N} \left(\sum_{i=1}^N x_i^2 - \frac{(\sum_{i=1}^N x_i)^2}{N} \right)}$$

em que N é o número de amostras de uma onda do acelerômetro, x_i são os valores das amostras e σ_x é o desvio padrão da onda.

- Energia da onda (Ravi et al. 2005): o cálculo da energia da onda se faz útil para analisar a intensidade do sinal. Para calcular a energia da onda primeiro calculamos a transformada de *Fourier* para os pontos do período capturado; após isto, fazemos um somatório dos quadrados das componentes da transformada, obtendo assim a energia da mesma.
- Correlação entre ondas (Ravi et al. 2005): a correlação entre duas ondas identifica, em termos numéricos, quão parecidas elas são entre si. Se uma delas aumenta de valor, a outra também deve aumentar para que as duas sejam relacionadas. Por exemplo: em uma caminhada os valores dos eixos x e y do acelerômetro podem ser negativamente correlacionados; enquanto a onda oriunda do eixo x aumenta de valor, a onda de y diminui, caracterizando correlação negativa. Já para quando não há movimento, os valores das ondas tendem a não registrar variações; sem variação, as duas ondas tem um comportamento fortemente correlacionado, ou positivo, de acordo com a fórmula. Esta característica da correlação pode então ser usada para discriminar dois tipos diferentes de atividades. Para o cálculo da correlação temos a seguinte fórmula para duas ondas X e Y :

$$\text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_x \sigma_y}, \text{ dado que } \text{cov}(X, Y) = \overline{(xy)} - \bar{x}\bar{y}$$

X e Y correspondem respectivamente a dois intervalos de onda, $cov(X, Y)$ é a covariância de X e Y , \overline{xy} é a média da multiplicação dos valores das ondas X e Y , $\overline{x}\overline{y}$ é a multiplicação das médias dos valores de X e Y e $\sigma_x\sigma_y$ é a multiplicação dos desvios padrões de X e Y .

Para uma caminhada, exaustivamente coletam-se vários intervalos de x segundos com gravação do acelerômetro contendo movimento característico da atividade. x é um intervalo suficientemente grande para que a atividade seja reconhecida o mais facilmente possível. Além de pesquisar o tempo ideal de medição x , um dos focos dos artigos como (Ravi et al. 2005), (Bao e Intille 2004) e (Kwapisz et al. 2010) é avaliar o quão precisos são os diferentes algoritmos usados para detectar atividades usando *machine learning*.

Para que o algoritmo de inteligência artificial possa identificar em tempo real as atividades representadas pela análise dos dados dos acelerômetros, deve haver um treinamento inicial do algoritmo escolhido. Nesta fase de treinamento, as atividades que devem ser reconhecidas pela aplicação são executadas e os dados dos acelerômetros capturados são pré-processados usando-se cálculos heurísticos, alguns apresentados acima, para extrair informação útil para o algoritmo de machine learning. Cada uma das atividades, juntamente com os dados processados, recebem um *label* que diz qual atividade estava sendo realizada.

De posse destes dados, pode-se criar uma instância de um avaliador, que será usado na detecção em tempo real de atividades. Os vários artigos pesquisados no presente trabalho discutem a eficiência de diversos tipos de avaliadores utilizados. Em (Bao e Intille 2004) por exemplo, temos que o classificador do tipo C4.5 é o que mais corretamente classifica as atividades. De modo geral, em (Kwapisz et al. 2010) o classificador com melhores resultados foi o *multilayer perceptron*. Já em (Ravi et al. 2005), *Plurality Voting* obteve resultados mais acurados.

2.4 Aparelhos médicos domésticos

A evolução da tecnologia proporcionou o surgimento de pequenos aparelhos capazes de medir ou monitorar diversos sinais do corpo humano. Pacientes que antes necessitavam visitar um médico para verificar seus níveis de glicose no caso da diabetes, agora podem fazê-lo em casa (Figura 2.4). Sensores de batimento cardíaco são comumente encontrados em bicicletas ergométricas e utilizados para medir o gasto calórico da atividade física. Tênis providos de sensores de passada, também chamados pedômetros, se comunicam por meio de *Bluetooth* com *Smartphones* para registrar os dados e poder estimar a quantidade de exercício realizada e consumo calórico do corpo.



Figura 2.4: Medidor de nível glicêmico

Em pesquisa preliminar para realização deste trabalho, levantou-se a ideia de implementar a comunicação de alguns destes dispositivos com um *Smartphone*. Seria usado um microcontrolador *Arduino* (Arduino Team), que se comunicaria com o celular por meio de *Bluetooth*. No final, os dados estariam disponíveis em um *website*, que permitiria ao médico acompanhar, em tempo real, a evolução ou regressão de um quadro médico a partir dos dados coletados pelo aparelho. Com um pouco mais de pesquisa, verificou-se que, para tal integração, já existe um padrão de comunicação para este uso específico, estabelecido pela *Institute of Electrical and Electronic Engineers* (IEEE), chamado ***Bluetooth Health Device Profile*** (Bluetooth SIG).

2.4.1 Bluetooth Health Device Profile

A seguinte subseção é um resumo do ***BHDP***. Como o trabalho não utiliza este protocolo, fica aqui somente para referência futura, podendo ser utilizado para aperfeiçoamento do presente trabalho.

O ***Bluetooth Health Device Profile***, ou ***BHDP***, especificado por um grupo de membros e empresas para o avanço da tecnologia *Bluetooth* (Bluetooth SIG), é um protocolo de camada de aplicação que usa como base o protocolo ***MCAP*** (*Multi Channel Adaptation Layer*). ***MCAP***, por sua vez, é um protocolo de rede de transporte ou sessão passível de múltiplas conexões. De acordo com a especificação encontrada em (Bluetooth SIG), existem *Sources* (fontes)¹ e *Sinks* (drenos)¹ de dados. Um dispositivo de medição de batimentos cardíacos, por exemplo, é um *Source*.

Já um *Smartphone* que se conecta com o dispositivo pode ser considerado tanto como *Sink*, pois está recebendo dados de um dispositivo médico, quanto como *Source*, pois ele próprio pode enviar dados para, por exemplo, um servidor remoto. O servidor remoto do tipo *Sink* pode ser um computador ao qual um médico tem acesso para verificar os dados do paciente. A Figura 2.5 exibe de forma esquemática o funcionamento descrito.

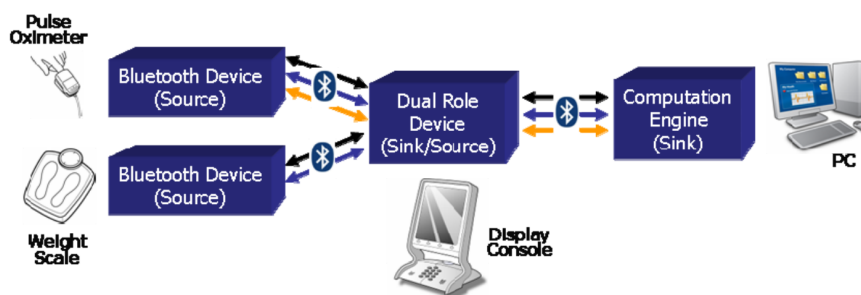


Figura 2.5: Funcionamento do protocolo *BHDP*

O sistema operacional ***Android*** mais recente disponível no mercado, o *Ice Cream Sandwich* (Android ICS), pertencente à quarta geração do sistema da *Google*, introduziu o suporte a tais dispositivos. Para maiores informações sobre o protocolo implementado no ***Android***, consultar (Android HDP). Para detalhes sobre o protocolo BHDP, consulte a norma *11073-20601 da IEEE* ou folheto que se encontra em (Bluetooth SIG).

¹Tradução livre

2.5 Trabalhos anteriores

Dentre os trabalhos relacionados, todos têm em comum o fato de utilizarem aprendizado de máquina para detecção de atividades. O que variam são os algoritmos de decisão utilizados, onde a classificação é executada, o *hardware* utilizado e a forma como são extraídas as informações de interesse oriundas das ondas originais dos acelerômetros.

A forma de apresentação das atividades detectadas, nos trabalhos relacionados, se resume basicamente a algumas ferramentas de plotagem de gráficos de forma linear e gráficos de coloração, não acessíveis remotamente. São usadas também tabelas de confusão com as porcentagens de acerto dos algoritmos de classificação. As tabelas 2.1 e 2.2 apresentam as diferenças entre as funcionalidades encontradas em alguns dos trabalhos pesquisados.

2.5.1 Reconhecimento de atividades usando dados anotados de acelerômetros

No trabalho de (Bao e Intille 2004) é apresentado um sistema que utiliza acelerômetros para medir a aceleração de diversos membros do corpo, para posterior análise da atividade realizada. O sistema compõe-se basicamente de 5 acelerômetros que se conectam fisicamente por meio de fios a um receptor anexado à cintura. Os acelerômetros são fixados à perna, braços, joelhos e cintura, este último no mesmo pacote onde está o receptor. Os dados são armazenados em um cartão de memória.

Posteriormente à gravação, os dados presentes no cartão são transferidos a um computador e analisados, utilizando-se a ferramenta **WEKA**, com diversos tipos de classificadores. O trabalho de (Bao e Intille 2004) tem como vantagem uma grande precisão na detecção de variados tipos de atividades, visto que são usados vários acelerômetros acoplados a diversas partes do corpo da pessoa que está sendo monitorada. Contudo, a mobilidade da pessoa que carrega os sensores é prejudicada, portanto o sistema é bastante invasivo.

2.5.2 Utilizando acelerômetros para reconhecimento de atividades

No artigo de (Ravi et al. 2005), a detecção de atividades é feita utilizando-se um hardware composto. O primeiro consiste numa placa contendo um acelerômetro e um rádio *Bluetooth* que transmite os dados para um segundo dispositivo. Este segundo dispositivo é um iPAQ, um PDA, também chamado de computador de mão, que grava os dados capturados para análise posterior em um computador.

Os dois aparelhos ficam junto ao corpo da pessoa. O sistema é moderadamente invasivo, no sentido de que além do iPAQ, a placa contendo o acelerômetro, junto com o transmissor *bluetooth*, é um equipamento extra a ser carregado. Os dados devidamente pré-processados são classificados utilizando-se também a ferramenta **WEKA**.

2.5.3 Plataforma de sensores móvel

Na pesquisa de (Choudhury et al. 2008) são desenvolvidas duas soluções para a detecção de atividade. Na primeira foi criado um hardware embarcado que coletava as informações de diversos sensores; um acelerômetro, um barômetro, microfone, giroscópio, sensor de infravermelho e de luz visível e um compasso digital. Os dados coletados

eram inicialmente enviados por *Bluetooth*, para um iPAQ, que então inferia a atividade que estava sendo realizada pelo indivíduo.

A transmissão por *Bluetooth*, no entanto, tinha um grande problema: não era eficiente e confiável o bastante para fazer o streaming em tempo real dos dados coletados. O hardware embarcado então foi conectado diretamente por meio de um cabo ao iPAQ. Isto fez com que o conjunto ficasse prejudicado no quesito mobilidade e praticidade. Nos dois casos, com e sem cabo, a vida da bateria do iPAQ era drasticamente reduzida, visto que este tipo de aparelho é planejado para processamentos esporádicos, não constantes como no caso apresentado.

Na segunda versão do equipamento, o hardware embarcado tinha autonomia e poder de processamento suficiente para que fosse capaz de adquirir, processar, armazenar e, eventualmente, enviar as atividades já classificadas para um computador por meio de *Bluetooth* ou *ZigBee*. Tanto a primeira versão do hardware como a segunda utilizam software proprietário desenvolvido especialmente para identificar atividades. O software também utiliza aprendizado de máquina.

2.5.4 Reconhecimento de atividades utilizando celular

A equipe de (Kwapisz et al. 2010) utiliza como hardware um celular *Android* para capturar e armazenar dados capturados do acelerômetro presente no telefone. Os registros capturados são transferidos por meio de comunicação *Wireless* ou por cabo USB para um computador e processados com a ferramenta **WEKA** para inferir as atividades realizadas.

2.5.5 Comentários finais

Em todos os trabalhos, o **WEKA** foi uma ferramenta decisiva na detecção de atividades, e também para o *Detector* será fundamental neste quesito. Os principais problemas apresentados nos artigos são a eficiência de seus métodos de detecção, a duração da bateria e quão invasivos são, fato que também deverá ser analisado no desenvolvimento deste trabalho. A pesquisa de (Kwapisz et al. 2010) é particularmente interessante pois pode ser aprimorada para que se possa detectar as atividades no próprio celular, sem que haja necessidade de pós-processar os dados analógicos dos sensores em outro computador.

Tabela 2.1: Comparativo das características dos trabalhos relacionados ao *Detector*

Artigo/Característica	Acelerômetros	Invasivo	Outros sensores	Visualização WEB
(Bao e Intille 2004)	5	sim	não	não
(Ravi et al. 2005)	1	sim	não	não
(Choudhury et al. 2008)	1	não	sim	não
(Kwapisz et al. 2010)	1	não	não	não
<i>Detector</i>	1	não ¹	sim ²	sim

¹Provido que o *Smartphone* faça parte do cotidiano da pessoa

²Através de sensor infravermelho presente na residência da pessoa

Tabela 2.2: Comparativo das características dos trabalhos relacionados ao *Detector* (continuação)

Artigo/Característica	Upload pela internet	Detecção de atividade no dispositivo
(Bao e Intille 2004)	não	não
(Ravi et al. 2005)	não	não
(Choudhury et al. 2008)	sim ¹	sim
(Kwapisz et al. 2010)	sim	não
<i>Detector</i>	sim	sim

¹ Através de conexão *Bluetooth* ou *ZigBee*

3 **DETECTOR: UM SISTEMA DE MONITORAMENTO DE PESSOAS**

Este capítulo explica a implementação do *Detector*, um sistema de monitoramento não invasivo que permite identificar atividades simples executadas por uma pessoa.

3.1 **Visão geral do *Detector***

Um sistema de monitoramento, conforme idéia apresentada na introdução deste trabalho, pode ajudar uma pessoa a conquistar uma independência maior, ou servir de ferramenta de informação sobre o estado de saúde dela. O monitoramento serve, por exemplo, para verificar se ela se exercita regularmente, o que pode ser útil para que o médico verifique se ela possui um estilo de vida saudável. No caso dos idosos ou pessoas com mobilidade reduzida, por exemplo, o sistema poderia identificar acidentes vasculares, quedas ou outras ocorrências que ponham em risco as suas vidas.

O *Detector* foi criado para ser capaz de identificar algumas atividades pré-determinadas – caminhar, correr, estar parado – realizadas por um indivíduo, e oferece também uma ferramenta visual para acompanhamento dessas atividades. O sistema é composto basicamente por dois módulos, cada qual não necessariamente operado pela mesma pessoa.

3.1.1 **Módulo de detecção de atividades**

O primeiro módulo é o de detecção de atividades, e é subdividido em duas partes. A primeira parte realiza uma detecção mais precisa das atividades através de um *Smartphone*, identificando diferentes tipos de atividades por meio do uso de algoritmos de aprendizado de máquina.

Para o sistema ser não invasivo, esforços foram concentrados em quais tipos de hardware poderiam ser utilizados para o módulo de detecção das atividades. O *Smartphone* é um excelente aliado nesta tarefa, visto que atualmente é computacionalmente poderoso e oferece um número generoso de recursos. Além disso, o *Smartphone* costuma estar dentro do bolso do seu dono, tornando-o perfeito para esta aplicação. Por meio dos acelerômetros do aparelho, detectam-se os movimentos exercidos pela pessoa e, de posse desses dados, pode-se inferir qual atividade está sendo realizada por ela em tempo real.



Figura 3.1: Celular LG P990 com Android

O celular que será utilizado é um *Smartphone* modelo P990 da *LG* (Figura 3.1), que possui um processador *dual core* Nvidia Tegra 2, com 1Ghz de velocidade e 512 Mb de memória RAM. O sistema operacional é uma modificação *Open source* baseada no *Android Gingerbread* versão 2.3.7, chamada CyanogenMod versão 7.1 RC2. Este SO é conhecido por ser mais otimizado em termos de processamento, gastos de bateria e de memória RAM do que o software original presente no telefone da *LG*. Maiores detalhes de como a identificação de atividades é realizada podem ser encontrados na seção 3.2.4.

A segunda parte do módulo de detecção de atividades identifica simplesmente se houve ou não movimentação de uma pessoa em um recinto, usando-se para isto um sensor de infravermelho (Figura 3.2). O uso deste tipo de sensor se justifica pois o proprietário, por vezes, esquece de manter o celular em seu bolso, deixando-o em cima de uma mesa em sua casa – o que evita a detecção de atividades pelo aparelho. Um ou mais destes sensores podem ser fixados em diferentes cômodos da casa. Quanto mais destes dispositivos há, melhor será a cobertura de detecção de movimentação.



Figura 3.2: Sensor de infravermelho

O infravermelho é acoplado a um coletor de dados digitais da empresa Autech (Figura 3.3), que é um hardware embarcado contendo um microprocessador Z180 com 4Mhz de *clock* e acesso à rede ethernet. A conectividade possibilita a gravação, no servidor, dos *timestamps* de quando houve movimentação em um recinto. Detalhes da implementação e uso do sensor de infravermelho e do coletor são apresentados na seção 3.3.



Figura 3.3: Coletor de dados da Autech

3.1.2 Módulo de visualização

Para o módulo de visualização, um *website* com gráficos das atividades e movimentação foi construído para que os dados pudessem ser visualizados remotamente. Este módulo pode não somente ser acessado pela própria pessoa que foi monitorada, mas também por familiares ou médicos interessados na rotina de um paciente.

O *website* foi implementado utilizando-se o **GWT** – *Google Web Toolkit* (Google), um *SDK* para programação WEB criado pelo *Google* em que a programação é feita utilizando-se somente Java. A interface visual é traduzida em tempo de compilação para Javascript. Uma explanação de como foi elaborado o *website* pode ser encontrada na seção 3.4.

3.1.3 Funcionamento geral

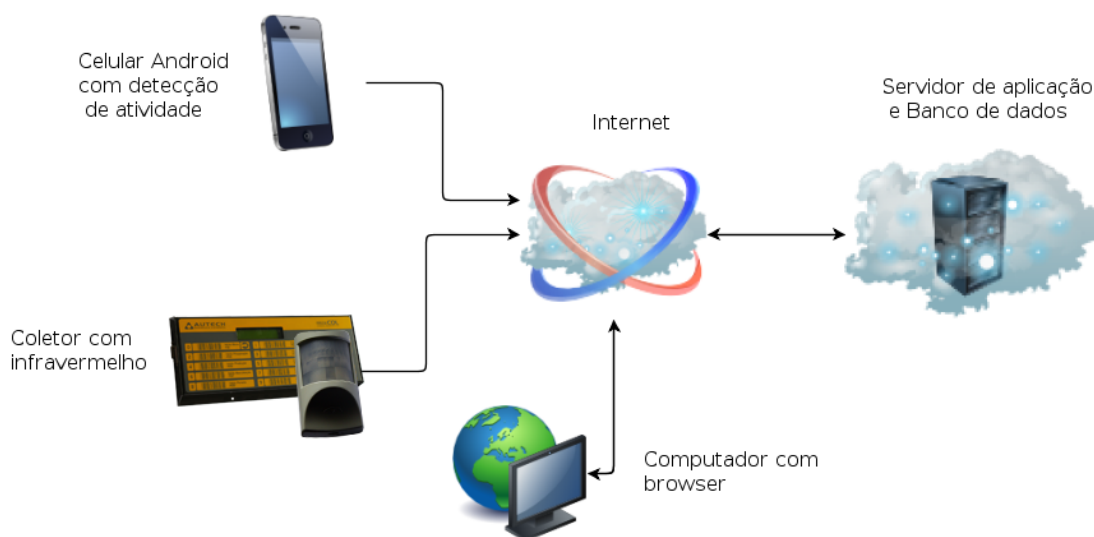


Figura 3.4: Diagrama de funcionamento do sistema

Cada dispositivo alimenta um banco de dados remoto, e o *website* pode então acessar os dados nele armazenados. O servidor onde são executados o banco de dados, a aplicação WEB e o extrator de dados do coletor, é uma máquina virtual com sistema operacional Linux, hospedada nos servidores da *Amazon*. A vantagem deste servidor é que pode-se

acessá-lo em modo *root*, o que permite a instalação de qualquer tipo de programa ou pacote necessário.

O *Smartphone* regularmente envia as atividades capturadas ao longo do dia, conectando-se com o banco de dados que executa no servidor remoto. Neste servidor, há também um programa implementado em Java, que a cada 3 minutos extrai as informações armazenadas no coletor de dados, salvando-os no mesmo banco de dados. Finalmente, a aplicação WEB, que também está no mesmo ambiente remoto, acessa as informações armazenadas no banco. O *website* é acessível por qualquer pessoa que possua um browser e as devidas credenciais de autenticação. O sistema completo funciona de acordo com o diagrama da Figura 3.4.

3.2 A detecção de movimento no celular

No início deste trabalho, a ideia de detecção de tipos de atividades ainda não tinha sido estabelecida. A primeira implementação consistia numa simples captura dos dados analógicos do acelerômetro do celular e posterior gravação em um banco de dados hospedado em um servidor remoto. Em pouco tempo notou-se que a quantidade de dados trafegados e armazenados remotamente era bastante grande. Isto porque o celular gravava localmente os valores para cada eixo do acelerômetro a uma taxa de amostragem de 20Hz. A cada 15 minutos os enviava ao servidor. Realizando-se as contas, totalizavam-se aproximadamente 18000 registros gravados no banco de dados em apenas 15 minutos. Para um dia inteiro de armazenamento, somariam-se 1728000.

Além da grande quantidade de informação armazenada, dispunha-se somente de dados analógicos, sem uma informação de contexto. Sabia-se que havia movimentação, mas não era possível verificar mais precisamente o que havia ocorrido em um certo intervalo de tempo. A posterior pesquisa sobre o assunto resultou em uma solução que detectasse tipos de atividades.

Para permitir a detecção de atividades, é necessário primeiro capturar os dados analógicos do acelerômetro correspondentes a cada atividade que queremos posteriormente identificar. De posse dos dados, juntamente com sua classificação – *label* – que diz a qual tipo de atividade eles pertencem, agrupa-se uma quantidade desses dados, referentes a um intervalo de tempo, e extrai-se informações – características – que serão utilizadas para treinar uma instância de um algoritmo classificador. A instância do classificador é capaz de inferir posteriormente, em tempo real, a qual classe pertence um agrupamento de dados de acelerômetro para o qual não temos um *label*.

A seguir é explicado como funciona o aplicativo para captura inicial anotada dos dados; como se retira as características importantes dos dados do acelerômetro; como se constrói um classificador, de posse dos dados anotados e pré-processados; e finalmente, é apresentado o aplicativo que é capaz de detectar as atividades feitas por uma pessoa.

3.2.1 Captura de dados anotados para treinamento do classificador

Um aplicativo simples foi programado para o *Android*, capaz de armazenar as ondas do acelerômetro características a cada atividade física (Figura 3.5). A captura de dados foi feita utilizando-se o celular da Figura 3.1. Este foi colocado dentro do bolso da calça

de um usuário, enquanto ele realizava algumas atividades como correr ou ficar parado.

A captura dos dados é implementada em uma classe que estende de *Service*. A classe *Service* permite que a aplicação possa continuar executando em plano de fundo mesmo que a interface visual do aplicativo seja fechada e voltemos à tela inicial – *Home Screen* do *Android*. Para que mais tarde fosse possível identificar quais dados pertenciam a qual atividade, cada instância de dado recebeu um *label*, em forma numérica, que indicava a que tipo de atividade correspondiam aqueles valores de acelerômetro.



Figura 3.5: Interface do aplicativo para captura de dados

Os dados analógicos, com uma marca de tempo do instante em que foram capturados, e o seus respectivos *labels*, foram gravados localmente no celular usando-se a biblioteca de banco de dados *SQLite* (Hwaci), disponível no *SDK* do *Android*. Gravar cada valor do acelerômetro no banco de dados local tão logo ele seja adquirido é impraticável, pois o *throughput* é muito baixo.

A primeira versão do aplicativo utilizava um *buffer* que armazenava uma quantidade de registros na memória e depois os gravava em lote no banco. O uso de somente um *buffer* era ineficiente, pois enquanto os valores de um *buffer* cheio eram gravados, novos valores do acelerômetro eram perdidos, pois o tempo de gravação dos dados era superior à taxa de amostragem do sensor. A solução foi utilizar dois *buffers* em memória. Quando um dos *buffers* está cheio, ele é passado para um método que o grava no banco de dados e o outro *buffer* passa a ser usado para armazenar os novos valores medidos. Deste modo, não são perdidos eventos novos.

Inicialmente os dados foram capturados a uma taxa de amostragem de 20Hz. Esta frequência mostrou-se visualmente suficiente (conforme Figura 3.6). Entretanto, quando o movimento do usuário é muito intenso, por exemplo no caso da corrida, a resolução da amostragem eventualmente poderia se mostrar insuficiente para capturar toda característica da onda analógica. Por experimentação, chegou-se em uma taxa de amostragem de aproximadamente 50Hz (Figura 3.7), o que resultou numa maior margem de segurança.

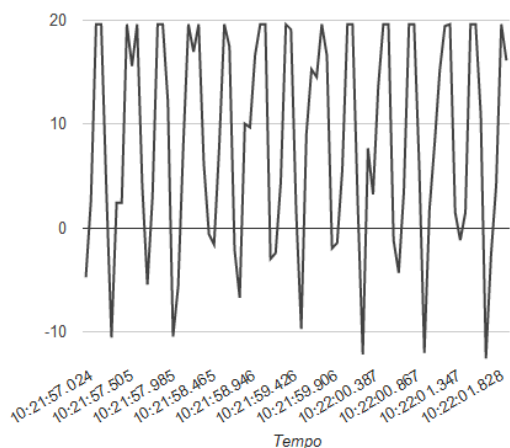


Figura 3.6: Dados do acelerômetro (eixo X) capturados a uma taxa de 20Hz

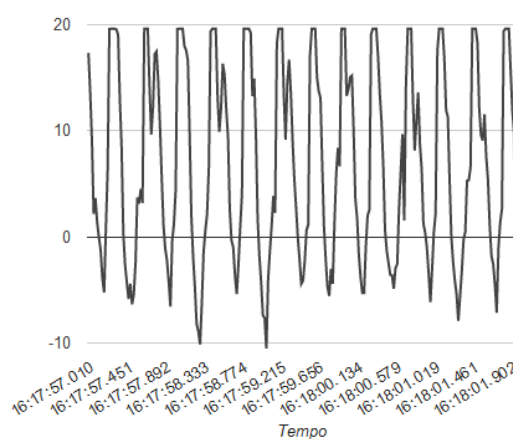


Figura 3.7: Dados do acelerômetro (eixo X) capturados a uma taxa de 50Hz

Um controle de Wii, ou *Wiimote* (Figura 3.8), foi utilizado para iniciar e parar a captura de dados. O *Wiimote* se conecta por meio de conexão *Bluetooth* com o celular e, no *Android*, se comporta como um teclado sem fio.



Figura 3.8: *Wiimote*

A decisão de usar este controle como forma de início e fim de coleta de dados de treinamento remete à necessidade do celular já estar posicionado dentro do bolso da calça quando a captura começa. Se o início for determinado por meio de pressionamento de um botão na interface do aplicativo, os dados capturados no intervalo em que o aparelho está na mão do usuário até o momento que o celular é posicionado dentro do bolso precisam ser descartados, pois não correspondem à atividade escolhida. Da forma como o aplicativo para captura foi implementado, utilizando o controle, os dados já estão livres de classificação incorreta, e não há necessidade de se analisar intervalos para descarte.

A captura, utilizando o aplicativo, ocorre da seguinte forma:

- Conecta-se o *Wiimote* no celular.

- O tipo de atividade que será executada é escolhida – no caso da Figura 3.5, caminhada está selecionada no *drop down*.
- O *Smartphone* é então posicionado dentro do bolso da calça e inicia-se a atividade – a caminhada.
- O pressionamento do botão 1 do *Wimote* inicia a captura, e o botão 2 a interrompe.

Foram coletados dados para as seguintes atividades: caminhando, correndo e parado. Para esta última atividade, duas situações foram contempladas: quando o autor estava sentado e também parado em pé.

Após um período de captura de aproximadamente 34 minutos de caminhada, 6 minutos de corrida e 4 minutos parado, os dados foram enviados, sem serem processados, diretamente para um banco de dados remoto. Os dados originais do acelerômetro foram gravados pois, caso os algoritmos de decisão mudassem ou o modo de se avaliar os dados mudassem, não seria necessário capturar novamente as mesmas atividades.

3.2.2 Extração de características das ondas

Após a captura de dados, um processamento sobre eles se faz necessário para que mais tarde seja possível usar um classificador que irá identificar atividades. Como explicado na contextualização deste trabalho, é preciso extrair *features* – características – em forma numérica das ondas geradas pelo acelerômetro, que poderão ser utilizadas pelo algoritmo de decisão.

O cálculo dessas informações é realizado a partir de um período de amostragem. Para o presente trabalho, o período é de pouco mais de 5 segundos, mais precisamente 256 amostras do acelerômetro a uma taxa de captura de 50HZ. A escolha do período de 5 segundos é justificada com base nos trabalhos pesquisados e por fornecer tempo suficiente para capturar alguns ciclos de repetição que caracterizam uma caminhada.

Os cálculos realizados sobre os dados capturados – analogamente aos cálculos encontrados em (Ravi et al. 2005) – são os que seguem:

- Média de cada eixo;
- Desvio padrão;
- Correlação entre ondas;
- Energia de cada componente do acelerômetro;

3.2.3 Construção da instância classificadora de atividades

Para esta etapa foi construído um programa que faz o cálculo das *features* (Figura 3.9), conforme explicado na seção 3.2.2, sobre os dados que foram capturados de acordo com a explicação da seção 3.2.1. Depois da extração é criada uma instância de um classificador.

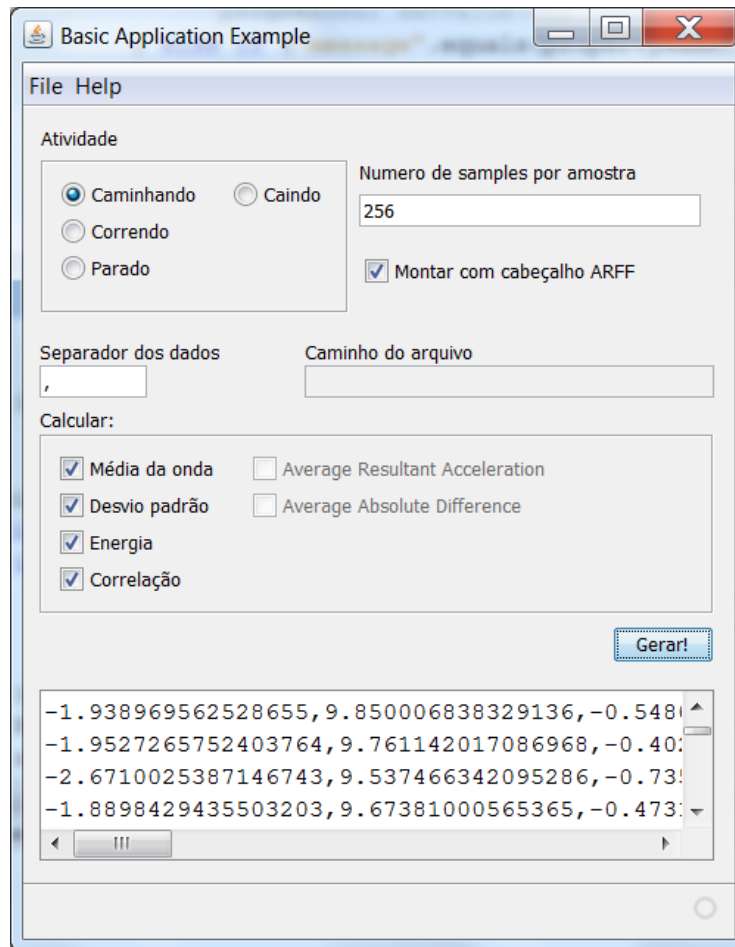


Figura 3.9: Etapa de pré-processamento

O programa se conecta com o banco de dados *Postgre*, onde estão armazenados os valores não processados do acelerômetro, busca os dados e então calcula as *features* para cada um dos tipos de atividade. A instância abaixo é resultado do cálculo de uma amostra de 5 segundos pertencente à caminhada. Tem-se 12 valores numéricos e o *label*, proveniente da captura descrita na seção 3.2.1, identificando a atividade.

-1.9527265752403764, 9.761142017086968, -0.40255490988784004,
 3.771768239454675, 3.6197894798773094, 3.388391004180807,
 1820.9292805759806, 1677.031350182861, 1469.2795092013323,
 -0.3030147398393791, 0.4371421347337076, -0.28457810203065714, caminhando

Em ordem, dado o intervalo de 5 segundos, temos: \bar{x} , \bar{y} e \bar{z} , as médias de cada eixo do acelerômetro; σ_x , σ_y e σ_z , o desvio padrão dos valores de cada eixo; E_x , E_y e E_z , as energias das ondas de cada eixo; finalmente, $corr(x, y)$, $corr(x, z)$ e $corr(y, z)$ são as correlações entre ondas dos diferentes eixos do acelerômetro: x correlacionado a y , x correlacionado a z e y correlacionado a z .

Constrói-se então um arquivo de texto (Figura 3.10) que contém várias instâncias com suas respectivas características – atributos – e *labels* – ou classes. Neste arquivo há um cabeçalho especificando quais os atributos que definem uma instância e, na continuação, as instâncias propriamente ditas.

```

@RELATION activity
@ATTRIBUTE mediaX NUMERIC
@ATTRIBUTE mediaY NUMERIC
@ATTRIBUTE mediaZ NUMERIC
@ATTRIBUTE desvioX NUMERIC
@ATTRIBUTE desvioY NUMERIC
@ATTRIBUTE desvioZ NUMERIC
@ATTRIBUTE energiaX NUMERIC
@ATTRIBUTE energiaY NUMERIC
@ATTRIBUTE energiaZ NUMERIC
@ATTRIBUTE correlacaoXY NUMERIC
@ATTRIBUTE correlacaoXZ NUMERIC
@ATTRIBUTE correlacaoYZ NUMERIC
@ATTRIBUTE class {caminhando,correndo,parado,caindo}

@DATA
-1.0626223165309057,10.019229332916439,-0.07043273396266159,2.7644144867264573,3.050154;
-2.1057779090915574,9.883843652904034,-0.6937126450211508,3.4449193495955908,3.65859300;
-2.2100091642787447,9.767426536418498,-0.4460484449955402,3.723896320628569,3.503221346;
-2.025995006042649,9.9234668770805,-0.355229308232083,3.9948097913037657,3.480667225426;
-1.038060562528655,0.850006838320136,-0.5480571402003080,1.1807062517300535,3.840768575;

```

Figura 3.10: Parte do arquivo "dados.ARF", contendo instâncias do trabalho

Uma vez que se tenha um conjunto de instâncias com características bem definidas e um *label* que as identifica, basta treinar um classificador para as atividades. Neste trabalho o algoritmo de *machine learning* **J48**, um classificador do tipo de árvore de decisão, foi escolhido por ter alcançado bons resultados na inferência de novas instâncias (Kwapisz et al. 2010).

O **WEKA** dispõe de ferramentas muito úteis para construção de um classificador. O arquivo de texto gerado acima é lido diretamente pela biblioteca e usado para treinar a árvore de decisão (Figura 3.11). Para evitar que o treinamento seja feito todas as vezes que o classificador for usado, a linguagem Java possibilita salvar uma instância serializada deste objeto, pois este implementa a classe *Serializable*. Além disso, como a programação no *Smartphone Android* é feita utilizando-se Java, isto permite que esta instância seja importada no projeto da aplicação e usada diretamente – fazendo a desserialização do objeto uma vez que se necessite usá-lo.

```

314 //constroi o Classificador e grava uma instancia serializada do
315 //mesmo no disco
316 BufferedReader reader =
317     new BufferedReader(
318         new FileReader(this.fileNameField.getText()));
319 Instances data = new Instances(reader);
320 reader.close();
321 data.setClassIndex(data.numAttributes() - 1);
322 tree = new J48(); // new instance of tree
323 tree.buildClassifier(data); // build classifier
324 weka.core.SerializationHelper.write(
325     "E:\\Material TCC\\textos-dados\\activity.model", tree);
326

```

Figura 3.11: Código fonte da construção do classificador J48

3.2.4 Detecção em tempo real no *Smartphone*

Com a instância do classificador de atividades pronta para uso, uma aplicação foi criada para o *Android* para a detecção de atividades. O funcionamento básico é explicado

pelo diagrama da Figura 3.12.

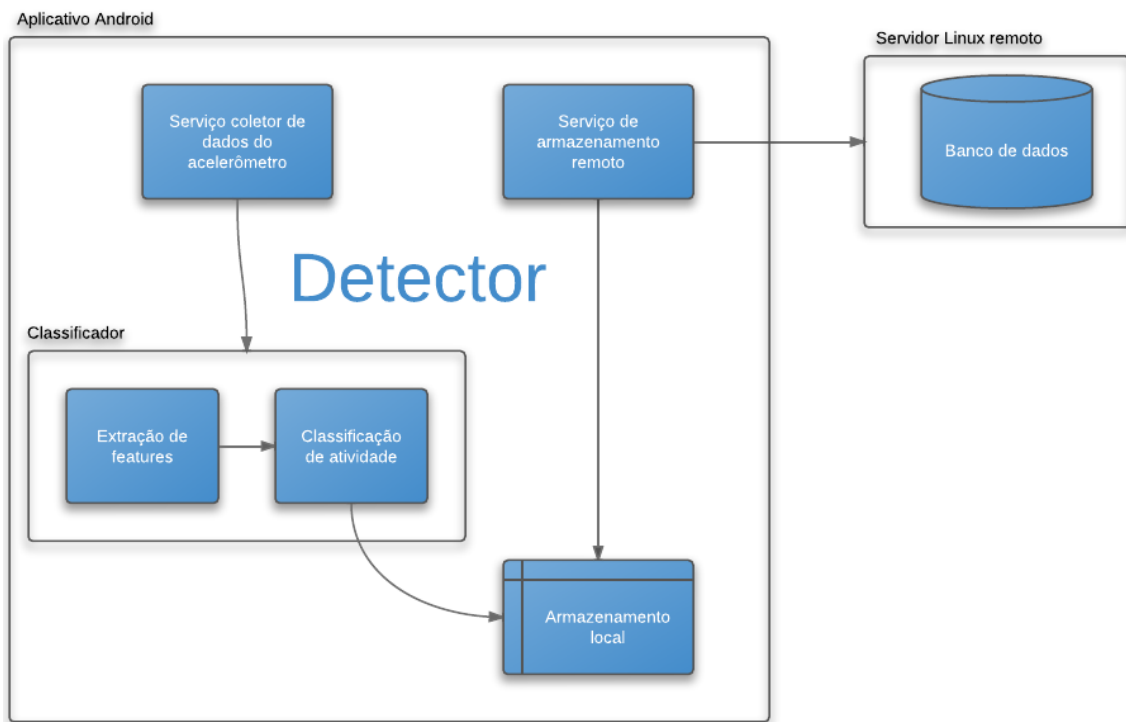


Figura 3.12: Diagrama de funcionamento do *Detector*

A coleta de dados do acelerômetro é implementada em uma classe Java que estende de *Service*. Esta classe pode ser executada em *background*, permitindo que a interface visual do aplicativo seja fechada, enquanto a coleta continua. A amostragem do acelerômetro, analogamente à seção 3.2.1, é feita a uma taxa de 50Hz. Da mesma forma como no aplicativo de coleta de dados para a criação do classificador, existem dois *buffers*, cada um com 256 posições, o que totaliza em torno de cinco segundos de captura para cada *buffer*. Cada objeto armazenado no buffer guarda os valores X, Y e Z do acelerômetro, bem como o instante em que o evento ocorreu.

Ao invés de gravar os dados sem processamento em um banco de dados local, como visto na seção 3.2.1, os dados são classificados previamente neste aplicativo. Quando um dos *buffers* é preenchido, os dados são repassados para uma *Thread* Java (Figura 3.13, linha 160) que é encarregada de classificar os dados.

A *Thread* primeiro extrai as características das ondas analógicas, exatamente como descrito na seção 3.2.2. Essa *Thread* possui uma instância desserializada do classificador que foi construído conforme a explicação da seção anterior, e utiliza as *features* extraídas para determinar qual atividade foi executada no período correspondente aos dados do *buffer*.

Logo após a identificação, a instância classificada é gravada juntamente com um *timestamp* – o tempo médio entre o primeiro e o último dado do *buffer* – em um banco de dados local SQLite. Os dados são gravados localmente por questões de economia de

energia e por nem sempre haver conexão disponível com a internet para que se possa enviá-los a outro local – o servidor remoto.

```

144
145 @Override
146 public void onSensorChanged(SensorEvent event) {
147     if (event.sensor != this.accelerometer)
148         return;
149
150     //sincronizado para nao dar overrun do buffer sendo utilizado.
151     synchronized (this.bufferFullness) {
152         if (bufferFullness == TAMANHODABUFFERIZACAO) {
153
154             final AccelerometerType bufferAGravar[] =
155                 bufferSendoUtilizado;
156
157             //classifica os dados bufferizados e
158             //salva no banco de dados local.
159             this.threadClassificadora.setBuffer(bufferAGravar);
160             new Thread(threadClassificadora).start();
161
162             //swap dos buffers
163             if (this.bufferSendoUtilizado == bufferSensor1) {
164                 this.bufferSendoUtilizado = bufferSensor2;
165             } else
166                 this.bufferSendoUtilizado = bufferSensor1;
167
168             bufferFullness = 0;
169         }
170
171         this.bufferSendoUtilizado[bufferFullness] =
172             new AccelerometerType( event.values[0],
173                                   event.values[1],
174                                   event.values[2],
175                                   0);
176
177         bufferFullness++;
178     }
179 }
180

```

Figura 3.13: Serviço de captura dos dados

Posteriormente, as atividades detectadas são enviadas ao servidor. Antes disso, porém, deve-se configurar o nome do usuário do aparelho e o intervalo de tempo entre cada *upload* dos dados. Este tempo varia de 1 minuto a uma hora. Uma classe que estende de *BroadcastReceiver* foi construída para iniciar um *Service*, dado o intervalo, que executa o *upload* dos dados locais junto com o nome do usuário. Se o *upload* é feito com mais frequência, temos uma atualização mais constante dos dados exibidos no *website*, contudo o consumo da bateria é maior. A transferência dos dados é realizada através de uma conexão e posterior inserção das atividades em um banco de dados *Postgre* configurado no servidor remoto.

3.3 Captura de dados através de sensor infra-vermelho

A detecção de atividades por meio de um *Smartphone* parte do princípio que o celular deve estar no bolso para que seja possível saber o que a pessoa está fazendo. No entanto, quando se está em casa, frequentemente o aparelho fica parado em cima de um móvel, como uma mesa por exemplo, impedindo a detecção. Já que não se pode identificar o que a pessoa está fazendo, pode-se ao menos verificar se há movimentação na casa.

Um sistema simples de detecção de movimento utilizando um sensor de infravermelho, comumente usado para sistemas de alarmes, foi usado em conjunto com um coletor

de dados, apresentados no início deste capítulo, nas Figuras 3.2 e 3.3. O sensor de infravermelho funciona como uma chave liga-desliga e quando há movimento uma luz liga, indicando que esta chave está eletricamente aberta. Por questões de praticidade e para isolar o comportamento do autor do presente trabalho, o sensor foi fixado em uma parede do seu quarto, próximo à mesa do computador. Assim, mesmo com o celular fora do bolso, foi possível detectar o seu movimento.

O coletor de dados é usado para detectar o momento de abertura e fechamento do sensor, gravando na sua memória interna o horário em que esses eventos ocorrem. Uma limitação deste coletor é que a resolução de tempo entre abertura e fechamento do sensor não deve ser de menos de 1 segundo. Para o presente trabalho, este intervalo é bastante aceitável, levando-se em conta que movimentos de tão curta duração não nos interessam. Além disso, o sensor infravermelho especificamente usado para o trabalho permanece geralmente mais de um segundo ligado, mesmo quando a duração do movimento é inferior a esse intervalo de tempo, tornando a limitação irrelevante.

A memória interna do coletor armazena 128 trocas de estado – de chave fechada para aberta ou vice-versa – com o instante em que ocorreram. Como o trabalho já utiliza um banco de dados, bastou criar uma nova tabela que armazenasse os valores capturados. O aparelho tem interfaceamento *Ethernet* para que se possa acessá-los mais tarde. Uma vez que foram acessados, eles são removidos da memória do dispositivo.

Para coleta dos instantes em que há movimentação, foi criado um aplicativo Java, que é executado no mesmo servidor do banco de dados remoto. O servidor está constantemente ligado, portanto pôde ser reaproveitado não somente como servidor WEB e de banco de dados, mas também como forma de se comunicar com o coletor de dados. O programa se conecta ao dispositivo a cada intervalo de três minutos, extraíndo a informação armazenada em seus *buffers* internos. Então, insere os dados no mesmo banco de dados onde estão armazenadas as informações de atividade.

3.4 Ferramenta de visualização WEB

A identificação das atividades demanda também uma forma de visualizá-los pelo usuário, bem como por familiares ou pessoas que estejam interessadas em saber um pouco mais da rotina do indivíduo. Um aplicativo que é passível de instalação não é uma boa solução, já que cada usuário tem uma configuração específica de sistema e pode não ser possível utilizar a ferramenta. Analisando a situação por outro ângulo, a internet e a programação de *websites* convergem para um padrão mais homogêneo, possibilitando que diversos sistemas proprietários, bem como abertos, possam oferecer uma solução de software o mais universal possível, bastando-se possuir um browser para tal.

Pensando nisto, um *website* foi desenvolvido para facilitar o acompanhamento das atividades de uma pessoa. O *website* do *Detector* é construído utilizando-se um framework para construção de páginas WEB chamado GWT (Google), desenvolvido por uma equipe de programadores do *Google*. Com o *GWT*, programa-se utilizando a linguagem Java. A parte do código que deve executar no servidor – que deve ter acesso a banco de dados e outras bibliotecas que necessitem dessa infraestrutura – é compilada para bytecode, como *servlets* Java normalmente encontrados em aplicações *JSP*. A parte do cliente, contudo,

que também é programada em Java, é compilada para Javascript, para permitir interação visual utilizando o *Browser*.

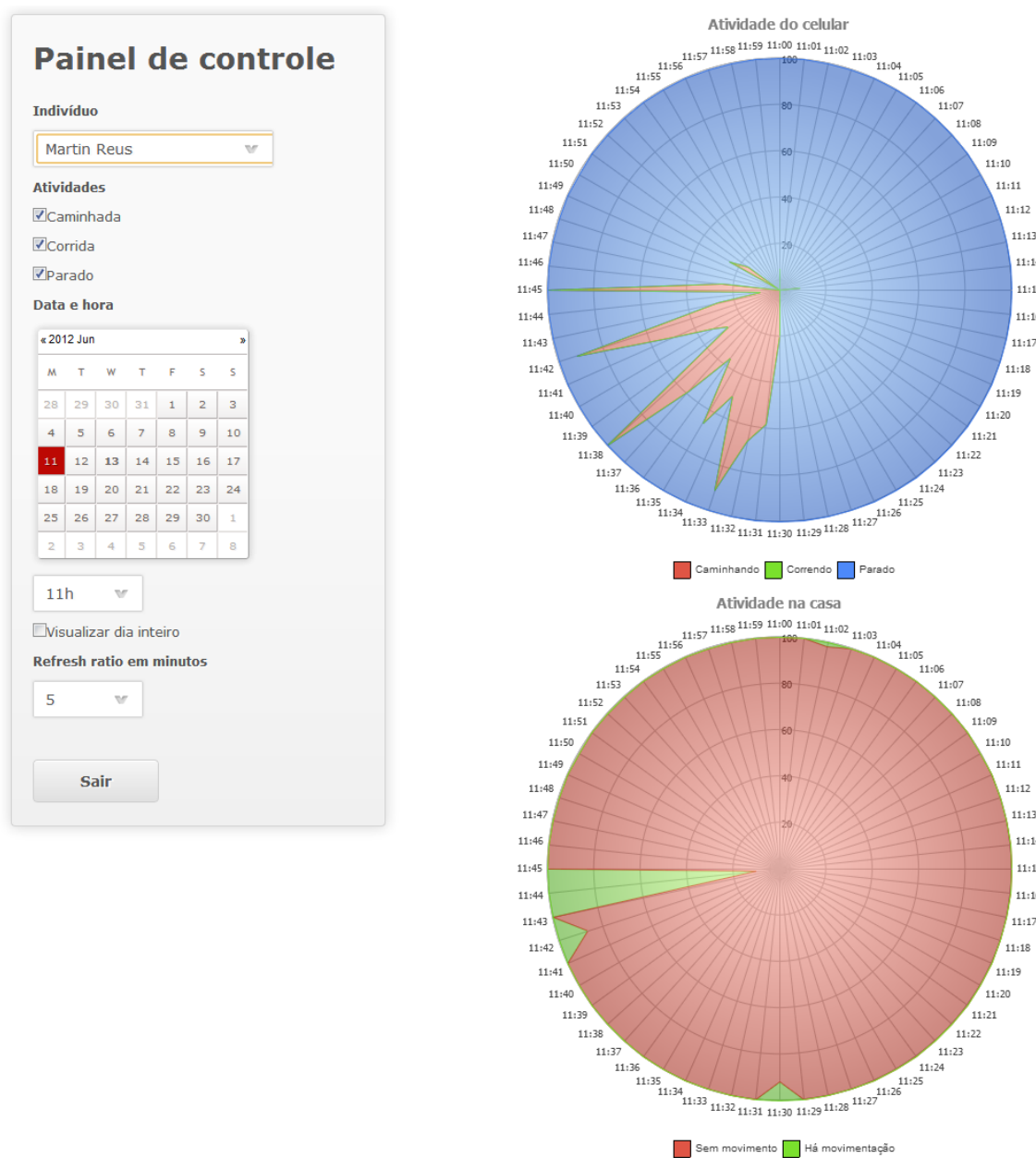


Figura 3.14: Captura da tela do *website*

A compilação em Javascript permite que a aplicação se comunique através de chamadas *Ajax* com o servidor. A grande vantagem de se desenvolver aplicativos utilizando o *GWT* é justamente a transmissão de dados cliente/servidor. Para o programador, a interação entre cliente e servidor é transparente no sentido de que *objetos* Java podem ser transmitidos ao servidor sem que seja necessário saber a infraestrutura de *POST* e *GET http* ou chamadas *Ajax*, comumente encontradas nas diversas linguagens de programação

WEB. O *framework* se encarrega de serializar objetos Javascript do cliente e transmiti-los para o servidor, desserializando-os na outra ponta como objetos Java, assim como ocorre com o caminho inverso.

O *GWT* ainda facilita o trabalho de criação de interface visual. É sabido que cada um dos diversos *Browsers* atualmente existentes no mercado possuem peculiaridades na forma como a linguagem Javascript é executada. É por isso que o código Java da parte do cliente é compilado para Javascript com diversas pequenas diferenças, de maneira que seja executado da forma mais compatível possível.

Para os gráficos que foram utilizados para mostrar as atividades, a biblioteca *SmartGWT*, com licença comercial de teste, serviu a este propósito. A escolha desta biblioteca reside no fato de que o *SmartGWT* possui forte integração com o *GWT* e possui diversos tipos de gráficos.

3.4.1 Funcionamento do *website*

Um painel de login possibilita que apenas pessoas com autorização visualizem os dados compartilhados. Após efetuada a devida autenticação, o usuário pode interagir com o *website*. O painel à esquerda na Figura 3.14 permite que se escolha a pessoa que foi monitorada; uma vez que alguém é selecionado, são visualizados os gráficos referentes à sua movimentação. É possível escolher também o dia e a hora dos dados que foram capturados. Além disso, pode-se ter uma visão mais ampla do cenário de movimentação da pessoa, escolhendo, para tal, se queremos visualizar um dia inteiro, ou apenas uma hora específica do dia escolhido.

O primeiro gráfico visível é o das atividades realizadas e identificadas através do celular. As atividades devem ser agrupadas antes de serem exibidas, pois cada registro corresponde a um intervalo de 5 segundos. De acordo com os parâmetros do painel esquerdo, é decidido se um agrupamento por minuto ou por hora é necessário. Ao fazer o *request* ao servidor, se o *checkbox* "Visualizar dia inteiro" estiver marcado, este irá selecionar os dados no banco de dados correspondentes a um dia inteiro – o dia selecionado no calendário. Os dados são então agrupados por seu timestamp e por tipo de atividade. Para cada hora é somado o total de atividades realizadas. Com este total, mais o somatório de cada atividade, pode-se extrair a porcentagem de tempo, no intervalo de uma hora, em que a pessoa caminhou, correu ou ficou parada.

O segundo gráfico se aplica ao sensor de infravermelho, quando existente. Da mesma forma que no gráfico de atividades, aqui os dados também devem ser agrupados. Contudo, o agrupamento é um pouco diferente. Os dados armazenados representam trocas de estado, de quando o sensor não estava detectando movimento, para quando começou a detectar, e o instante em que isto ocorreu. Para fazer o agrupamento, separam-se os dados nos intervalos correspondentes, dependendo se queremos agrupar por hora ou minuto. Depois soma-se o total de tempo em que o sensor permaneceu ligado. Com o cálculo podemos determinar qual a porcentagem de tempo, em uma hora, em que houve movimentação.

4 RESULTADOS E ANÁLISE DO *DETECTOR*

O seguinte capítulo apresenta uma análise dos resultados obtidos com a implementação e posterior teste do *Detector*.

4.1 Metodologia de avaliação de resultados

Ao total, o aplicativo do *Detector*, desenvolvido para *Android*, foi testado em 5 aparelhos: LG P990, com versão customizada do *Android Gingerbread 2.3.7*; Samsung Captivate SGH i89, com *Android Gingerbread 2.3.5*; Samsung Galaxy i5801 com *Android Froyo 2.2*; Samsung Galaxy S2 com *Android Gingerbread 2.3*; Motorola Atrix com *Android Froyo 2.2*. Com os dois primeiros celulares, foram realizados testes preliminares de duração da bateria. Para os últimos dois celulares, somente foi verificado se a aplicação era funcional.

O sensor infravermelho somente foi testado com o autor do trabalho, sendo possível verificar que estava funcionando através dos gráficos observados no *website*. Esta ferramenta de visualização, por sua vez, foi testada por sete pessoas, com idades entre 23 e 57 anos. Duas pessoas deste mesmo grupo testaram também a aplicação para o celular, além do autor deste trabalho.

Além disso, um questionário sobre a vontade de usar ou não o sistema foi respondido por nove pessoas, com idades variando entre 23 e 82 anos, as sete pessoas que já haviam analisado o *website*, mais duas senhoras, de 81 e 82 anos. Essas duas últimas responderam somente o questionário sobre o uso do sistema.

Os resultados do questionário referentes ao uso do *website* e uso do aplicativo, são apresentados nos próximos subcapítulos.

4.2 A detecção de atividades utilizando *Smartphones*

A etapa de detecção de atividades por meio do celular, desenvolvida neste trabalho, tem muitos aspectos em comum com as pesquisas relacionadas apresentadas no capítulo 2. Alguns detalhes como duração da bateria, comunicação e armazenamento também são avaliados. *Detector* foi testado por alguns alunos de graduação e mestrado que possuíam um aparelho compatível, ajudando na validação dos resultados.

Um dos problemas identificados no decorrer das implementações do software para o

celular foi um dos fatores decisivos para a duração da bateria. Cada fabricante de celular implementa os *drivers* do sistema operacional para o *hardware* específico de cada aparelho. O Google fornece apenas *headers* de código fonte para as bibliotecas requeridas pelo *Android*, enquanto a equipe encarregada do software do aparelho desenvolve o código de acesso ao *hardware*.

Decorrente das diferentes implementações, vários acelerômetros de celulares de diferentes marcas e diferentes versões do sistema operacional *Android* são eletronicamente desligados, através dos *drivers*, quando a tela do celular é posta em modo *sleep*, de economia de energia. Isto impede que novos eventos de mudança de orientação do acelerômetro sejam detectados e enviados ao aplicativo que os está usando. Para que funcione, o aplicativo então deve ficar com a tela ligada, em modo *dimmed*, que corresponde à tela ligada com mínimo de retro-iluminação.

Infelizmente, para o celular utilizado no desenvolvimento deste trabalho, os acelerômetros só registram novos eventos quando a tela é mantida ligada. Este fato acarreta outro problema: incondicionalmente, ou acidentalmente, o usuário do celular por vezes desliga a tela, prevenindo a captura de dados. Para verificar que o problema não era da aplicação desenvolvida e sim do conjunto *hardware-software* do celular, o programa foi testado com sucesso em um outro celular, um *Samsung i897 Captivate*. Este registrou corretamente as atividades, mesmo com a tela desligada.

4.2.1 Duração da bateria

Durante os testes da aplicação, utilizando-se o celular LG P990 apresentado na seção 3.1.1, em que a aplicação funciona somente com a tela ligada, o consumo de bateria foi superior, se comparado ao uso cotidiano. Entretanto, mesmo para este cenário de uso, a bateria de 1500mAh foi suficiente para atingir 11 horas e 20 minutos de autonomia no primeiro teste, restando, ao final, 7% de carga na bateria. Os dados foram coletados ininterruptamente, sem que se fizesse o *upload* dos mesmos, ou que o aparelho estivesse conectado à internet.

Para o segundo teste utilizou-se o mesmo celular, configurado para que enviasse os dados a cada 15 minutos por meio de conexão *Wifi*. Nessa situação, a captura começou com uma carga de bateria em 100%. 9 horas e 50 minutos após, restaram 3% de carga.

Em um terceiro teste, para um celular da marca *Samsung*, modelo *Captivate SGH i897*, em que é possível executar a aplicação com a tela em modo de economia de bateria (desligada), a autonomia chegou a 12 horas e 40 minutos. A verificação iniciou com o nível de carga da bateria de 1500mAh em 100%, chegando-se ao final em 5%, sendo que os dados foram gravados no servidor remoto somente quando havia conexão *WiFi*.

Um gráfico dos resultados pode ser visualizado na Figura 4.1. Para fins comparativos, os dados foram normalizados, como se a carga inicial da bateria estivesse em 100% e a final em 0%.

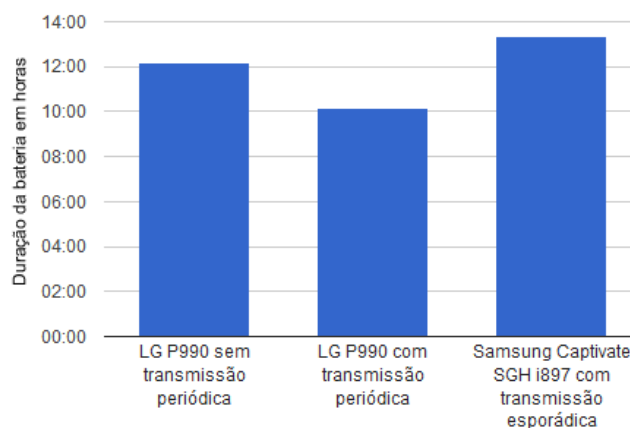


Figura 4.1: Gráfico dos testes de duração da bateria

4.2.2 Acuidade da detecção

O foco deste trabalho não era prover o método mais correto e confiável de detecção de atividades, mas sim ser uma ferramenta em que fosse possível capturar os dados, analisá-los de forma automatizada e fornecer uma interface de visualização. Mesmo assim, o objetivo de detecção de diferentes tipos de atividades no celular foi alcançada, apesar de o treinamento da árvore de decisão ter sido realizado com relativamente poucos dados, se comparado aos trabalhos usados como referência. Além disso, o treinamento da mesma tomou como base somente um indivíduo, o autor deste trabalho.

Três estudantes da universidade, incluindo o autor, testaram a aplicação, com seus respectivos celulares. Para o celular LG P990, o aplicativo foi testado por quatro dias, o que, excluídos os períodos de inatividade, totalizou 43 horas de captura. O celular Samsung Captivate SGH i897 executou o aplicativo para coleta de dois dias de atividades, totalizando 18 horas de dados. Para o *Smartphone* Samsung Galaxy i5801, o aplicativo coletou, por 26 dias, 406 horas de atividades realizadas por seu usuário.

O usuário do Samsung Captivate SGH i897, ao analisar seus dados através da ferramenta de visualização, mencionou que não havia corrido em um dos dias, apesar de no gráfico constar tal atividade. Uma explicação seria a identificação errada de uma de suas atividades físicas. Atividades como subir ou descer escadas podem, por vezes, imitar o comportamento de uma corrida e, já que subir ou descer escadas não está previsto pelo algoritmo de detecção, o classificador escolhe a classe de atividade que mais se assemelha aos dados capturados, no caso, corrida. Outra situação de falsa detecção é quando o aplicativo acusa que o usuário caminhou, mesmo que ele estivesse sentado com o aparelho na mão, escrevendo uma mensagem. Estes problemas podem ser resolvidos com um intenso treino do classificador, utilizando-se mais pessoas e abrangendo-se mais tipos de atividades.

Um outro fator que impede que a detecção de atividades seja mais exata é o problema já comentado do acelerômetro que é fisicamente desligado. Muitos dos gráficos obtidos dos testes de campo mostraram intermitência na coleta dos dados. Isto prejudica a qualidade da detecção, pois não há continuidade na obtenção. Entretanto, o objetivo da detecção foi alcançado, já que primariamente a função do sistema é verificar se houve

movimentação. Mesmo que houvesse períodos não registrados, diferentes atividades puderam ser observadas quando houve o registro.

4.2.3 Pesquisa preliminar sobre uso do *Detector*

Uma pesquisa foi realizada com nove pessoas, como apresentado na metodologia de avaliação, com idades variando entre 23 e 82 anos, sobre o uso do *Detector*. Dessas nove pessoas, sete utilizaram o *website* e, dessas sete, duas também testaram o aplicativo para o celular.

Para a pergunta "Você usaria um sistema de monitoramento para que, caso você sofresse um acidente estando sozinho em casa, alguém soubesse do ocorrido?", seis pessoas responderam que sim, um dos quais com 23 anos, com ressalvas de que o sistema seria um exagero de precaução, mas que para pessoas idosas talvez fosse vantagem. Uma das entrevistadas, de 81 anos, citou que não usaria o aparelho celular para medir as atividades, exceto se realmente necessário, por às vezes simplesmente não ter onde colocá-lo, uma vez que usa vestidos sem bolsos. As pessoas mais jovens foram mais receptivas quanto ao uso do sistema, pelo fato do celular já fazer parte de seu cotidiano.

Para a pergunta "Você se sentiria incomodado por um sistema de monitoramento? Por quê?", cinco pessoas responderam que não se importariam com o monitoramento, sendo que a principal preocupação do uso era relativa à segurança dos dados, que deveriam ser acessados estritamente por quem tivesse a devida autorização. O restante das respostas, dos que se importariam com o monitoramento, eram justificadas principalmente pela questão da privacidade, de que não gostariam de ter alguém monitorando toda a sua rotina. Uma sugestão de um entrevistado que fazia parte do grupo de pessoas que não gostaria de ser monitorado, referente à privacidade, foi de que apenas deveria monitorar-se situações de risco da pessoa, como seria o caso das quedas.

Na última pergunta, "Você se importaria se o sistema de monitoramento permitisse seu médico acompanhar sua rotina de exercícios ou atividade física?", sete pessoas responderam que não se importariam. Quatro destas pessoas mencionaram que o uso do *Detector* seria um incentivo à prática de exercícios ou de uma vida mais saudável. As duas respostas afirmativas foram feitas por duas senhoras, de 81 e 82 anos. A primeira justificou que não usaria o sistema, a não ser que fosse exigência médica e a outra justificou que não usaria o sistema pois não está acostumada com celulares.

A análise das respostas evidencia que a adoção do sistema por pessoas idosas irá depender de quão atualizadas com questões tecnológicas elas estão. Além disso, mesmo para pessoas mais jovens, segundo as respostas dadas, o sistema deve trazer benefícios para o usuário sem que a privacidade e a segurança sejam violadas.

4.2.4 Privacidade e invasividade

O celular, por sua própria finalidade, não é fardo extra para quem está acostumado com seu uso. Contanto que mantido dentro do bolso, o celular pode ser considerado como não invasivo. Entretanto, para um público-alvo de mais idade, algumas questões simples, como o fato de a pessoa não utilizar o celular regularmente, podem representar uma resistência no uso do *Detector*, como visto na seção 4.2.3.

Além da invasividade do *Smartphone*, deve ser levada em conta também a invasividade do aplicativo quando instalado no aparelho. Uma questão levantada por um usuário do aplicativo foi o fato de as notificações da aplicação serem um pouco exageradas. Notificações, no *Android*, são mensagens mostradas ao usuário para deixá-lo ciente do estado em que a aplicação se encontra, de eventuais erros ou outras comunicações pertinentes. Para a aplicação criada havia duas notificações, uma indicando que a coleta de dados estava ativa, outra para cada upload efetivado com sucesso ou falha, o que se mostrou excessivo.

Ainda analisando a invasividade, pode-se considerar o consumo de memória. Para esta aplicação, houve um uso aproximado de cinco a seis MB de memória RAM. Para o celular *LG P990* isto representa aproximadamente 2% da memória disponível para o usuário, o que configura um uso significativamente baixo deste recurso.

4.3 O sensor infravermelho

Para o caso do celular estar fora do bolso, em cima de uma mesa, o sensor de infravermelho supriu a carência de informação, indicando que houve algum tipo de movimento, mesmo que a detecção no *Smartphone* apontasse o contrário. Um dos aspectos percebidos para este dispositivo é que o sensor não identifica quem o ativou, ou quantas pessoas o fizeram. Isso não descaracteriza sua utilidade, uma vez que ele ainda é útil quando só há uma pessoa em casa e o celular está em repouso fora do bolso.

O sensor infravermelho foi testado somente com o autor. O autor avalia que algumas pessoas eventualmente se sintam vigiadas por estarem cientes de que tal monitoramento esteja ocorrendo. Outro detalhe que denunciaria o monitoramento seria o ativamento da luz do sensor em caso de movimento. Uma vantagem deste sistema é que os sensores infravermelhos já estão muitas vezes presentes no domicílio, pois a finalidade original dos mesmos é fazer parte de sistemas de alarme, podendo estes ser compartilhados com o *Detector*.

4.4 Análise do *website*

A disponibilização de gráficos de fácil entendimento das atividades e movimentação realizadas pelo usuário são peça-chave do trabalho desenvolvido. Uma breve discussão sobre os métodos utilizados, o resultado da visualização dos dados, bem como suas vantagens ou deficiências, como observadas no decorrer dos testes, são apresentadas a seguir.

Normalmente as atividades que esperamos identificar mais vezes em uma pessoa que possui uma rotina sem muita movimentação, são as que envolvem nenhum movimento expressivo, ou seja, quando se está parado. Isto se confirmou em grande parte dos gráficos visíveis no *website*. O exemplo da Figura 4.2 mostra que grande parte do dia o autor - dono das atividades demonstradas no gráfico - permaneceu sentado.

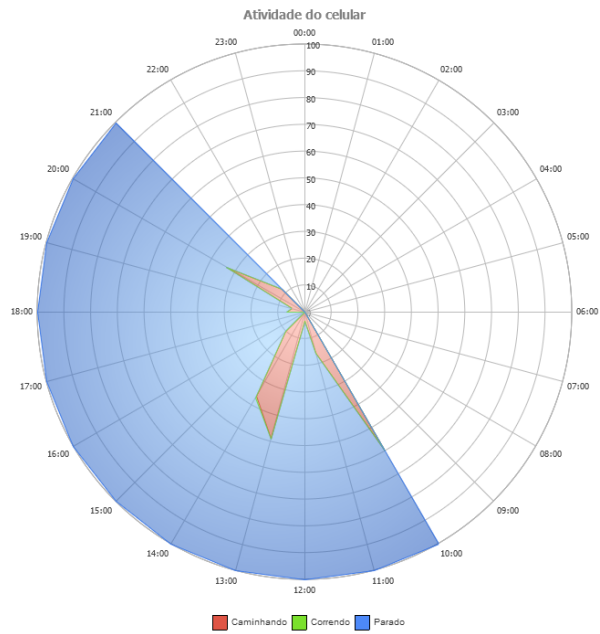


Figura 4.2: Gráfico de um dia inteiro de captura de dados do *Smartphone*

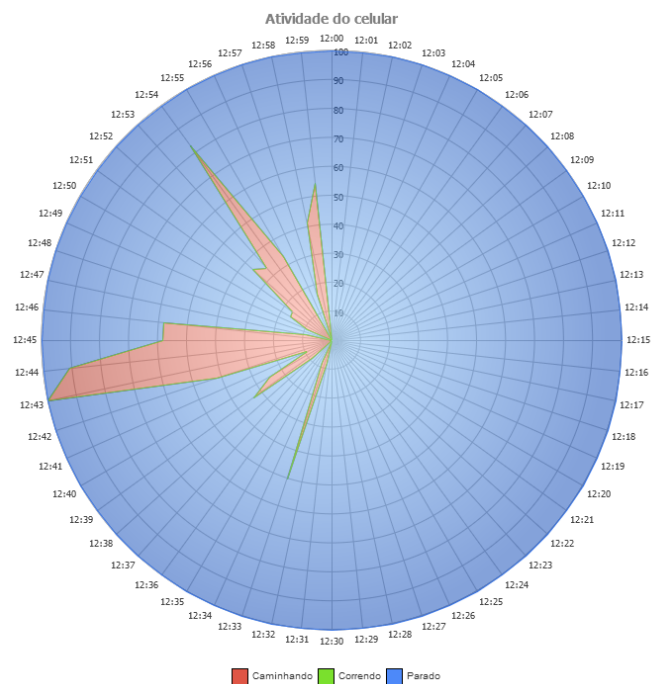


Figura 4.3: Gráfico de uma hora de captura de dados do *Smartphone*

Aqui, a atividade *parado* é caracterizada pela cor azul, e predomina no gráfico. Ao utilizarmos um gráfico do tipo *radar*, criamos a impressão de um relógio, com cada hora representada por uma porcentagem da quantidade de certo tipo de atividade desenvolvida. Isto é mais visível na Figura 4.3, em que o gráfico é dividido em partes iguais de um minuto. O gráfico de hora oferece mais granularidade na informação, fornecendo maiores detalhes de como as atividades foram distribuídas num menor espaço de tempo, se comparado com o gráfico diário.

O gráfico do tipo *radar* foi utilizado como gráfico padrão, para que lembrasse o formato de um relógio e, embora seja visualmente atraente, ele cria um efeito inusitado na apresentação. Como o gráfico empilha as atividades por sua ordem de exibição, as atividades posicionadas mais próximas à borda têm um tamanho exagerado, se comparadas ao núcleo.

No caso do gráfico das atividades capturadas no celular, o fato é agravado pela natureza das atividades. A atividade "parado" é a última que aparece, estando ela mais próxima do fim do gráfico. Como na maior parte do tempo se está parado, é passada uma impressão errônea de que não houve tanta movimentação como ocorreu na realidade. Além disso, atividades que deveriam ser mais visíveis, como quando houve um momento de corrida, ficam comprimidas no centro do gráfico, impedindo uma visualização mais clara.

A situação inversa também pode ser constatada, quando analisamos o gráfico criado para o sensor de infravermelho (Figura 4.4). Neste, o fato de que houve movimento em um recinto é exagerado, pois a ordem de disposição das atividades no gráfico faz com que a movimentação seja visualizável nas bordas do mesmo. Neste caso, o exagero é benéfico, uma vez que não há tantos registros de movimentação, se comparado ao gráfico do celular, o que torna a visualização mais clara. Com relação a essa inversão houve um comentário de um testador, para o *website*, sugerindo a troca da ordem de exibição, de forma que o gráfico ficasse consistente com o gráfico das atividades do celular - com a atividade "parado" aparecendo mais próxima à borda.

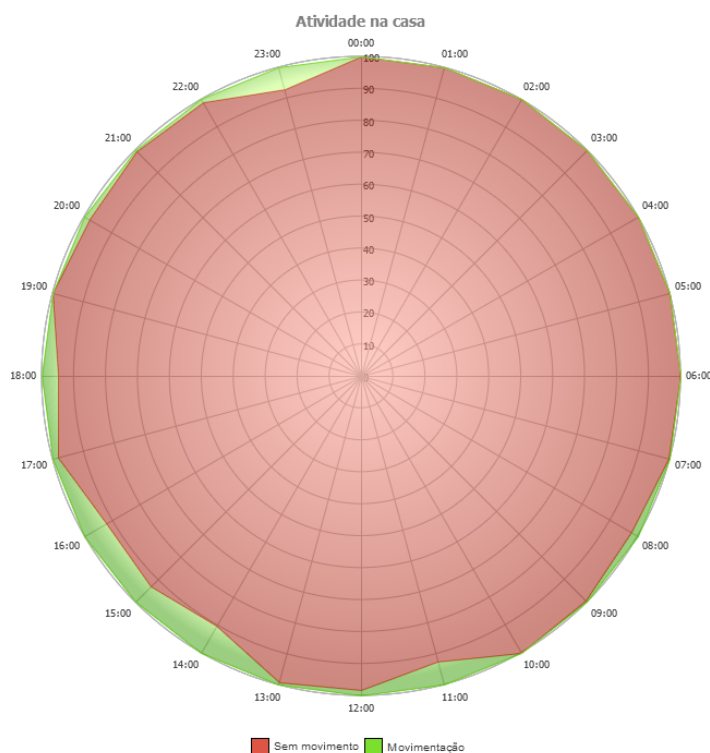


Figura 4.4: Gráfico de um dia de captura de dados do sensor infravermelho

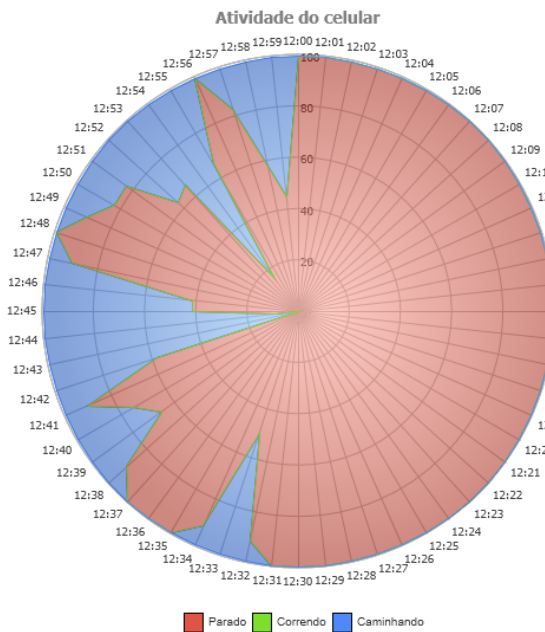


Figura 4.5: Gráfico de uma hora de captura de dados do *Smartphone*, com dados apresentados na ordem invertida

Um experimento de inversão da ordem de apresentação dos dados das atividades do celular pode ser observado na Figura 4.5, para o mesmo horário do gráfico da Figura 4.3. Desta forma, a caminhada é mais evidenciada.

Como a biblioteca de gráfico usada no *website* contém outros gráficos além do tipo *radar*, é fácil obter-se uma visualização diferente para os dados. Em especial, o gráfico do tipo *area* é útil para se obter uma linearidade na visualização dos dados. O gráfico *area* é um gráfico do tipo cartesiano, em que as atividades são empilhadas da mesma forma que no *radar*, mas em que não há compressão da visualização, devido à sua forma horizontal. Um exemplo dos mesmos dados visualizados na forma de gráfico *radar*, das Figuras 4.2 e 4.4, podem ser observados na forma de gráfico *area*, nas Figuras 4.6 e 4.7.

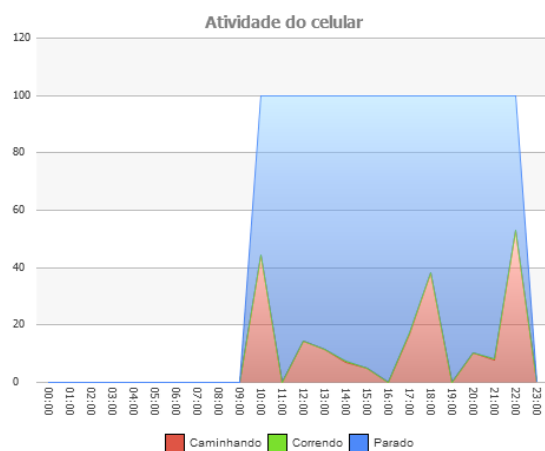


Figura 4.6: Gráfico de área de um dia de captura de dados do *Smartphone*

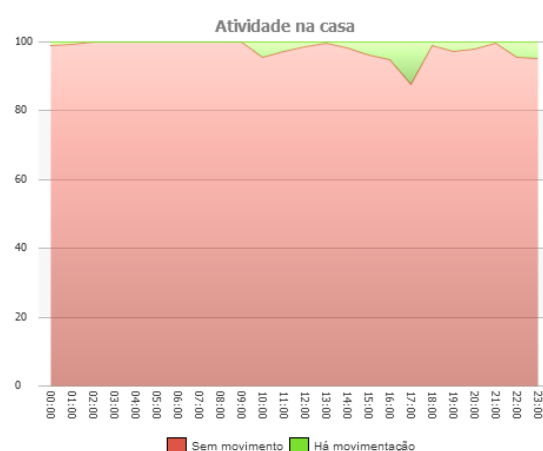


Figura 4.7: Gráfico de área de um dia de captura de dados do sensor infravermelho

4.4.1 Pesquisa preliminar sobre uso do *website*

Os gráficos, bem como o *website*, foram testados e avaliados por sete pessoas – seis das quais com pelo menos uma graduação em computação em andamento ou concluída – quanto à sua facilidade de interação e clareza dos dados. Os resultados de preferência relativa aos tipos de gráfico foi bastante heterogênea, dependendo do gosto de cada pessoa. Houve também a requisição, por um usuário, de que os gráficos fossem visíveis na aplicação do celular.

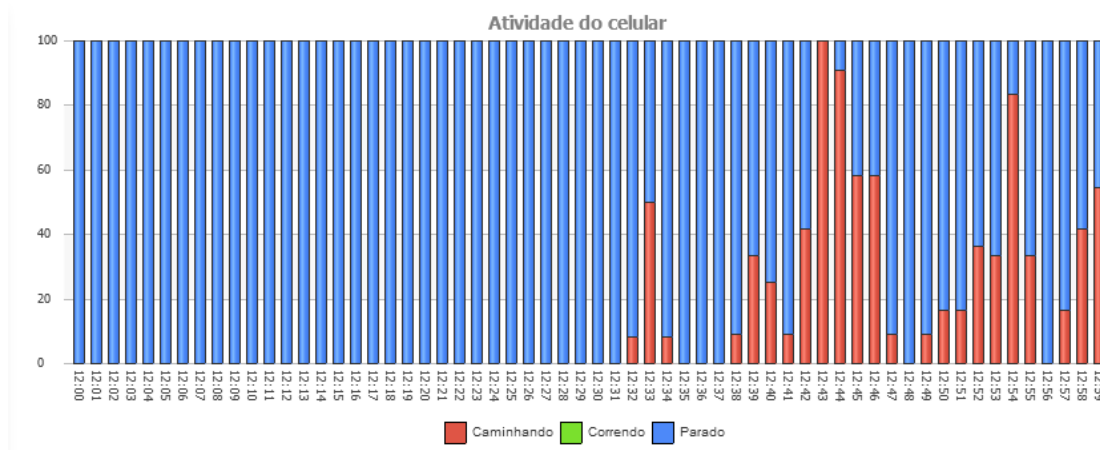


Figura 4.8: Gráfico do tipo *column*

A preferência dos gráficos no *website* foram as seguintes: o gráfico do tipo *radar*, a exemplo da Figura 4.3, ganhou dois votos; o do tipo *column* (Figura 4.8), em que a representação dos dados é feita por colunas empilhadas das atividades para cada minuto ou hora, também ganhou dois. A justificativa para o uso do gráfico tipo *column*, explicada por um dos usuários, é que se consegue visualizar melhor o instante das atividades. O gráfico *area* (Figura 4.6) recebeu dois votos, e *line* (Figura 4.9), que é um gráfico comum de linhas, sem empilhamento das atividades, ganhou um voto.

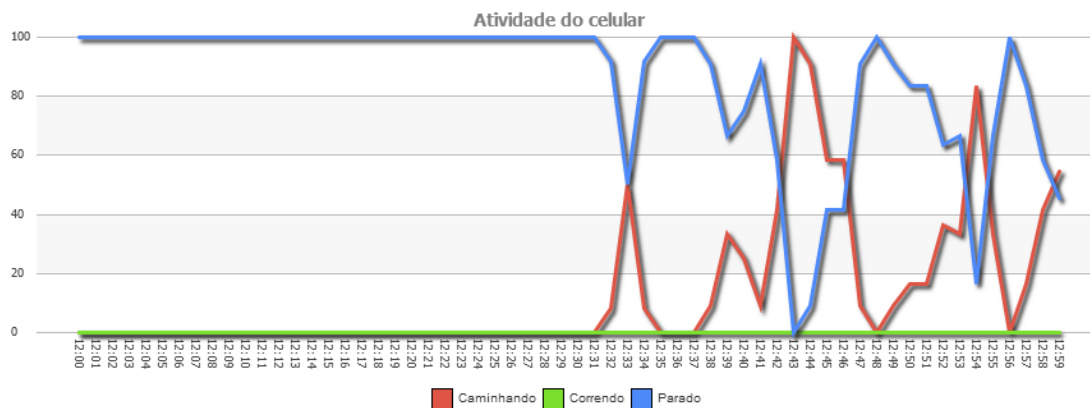


Figura 4.9: Gráfico do tipo *line*

A usabilidade do *website* foi reportada pelos usuários como de fácil entendimento.

Entretanto, o fácil entendimento compreende que a função toda do *Detector* seja explicada a eles, dando sentido aos controles e gráficos existentes.

4.4.2 Correlação entre gráficos

A correlação entre os gráficos agrega outras informações. Quando há registro de movimentação para os dois gráficos no mesmo instante, há uma grande chance de que a pessoa tenha realizado a atividade em um determinado recinto, descartando-se os casos em que há mais de um indivíduo na casa. Além disso, é possível verificar quando esta saiu ou entrou em seu quarto, como foi observado durante os testes. Na Figura 4.10 pode ser observado que, por volta das 11h30min o autor saiu de seu quarto e possivelmente foi preparar um café. Às 11h45min, depois de muita dificuldade no preparo, sentou-se novamente em frente ao computador para terminar seu trabalho de graduação.

A ideia de localidade das ações executadas pode ser útil quando a atividade inferida pelo *Smartphone* é algo nocivo à pessoa monitorada, como na eventualidade de uma queda. Neste caso, saber onde se encontra a pessoa que sofreu o acidente pode acelerar a questão do atendimento. Um outro cenário imaginável é o de uma pessoa que assiste muita televisão – verificável com a existência de um sensor de infravermelho na sua sala – o que indicaria um sedentarismo excessivo.

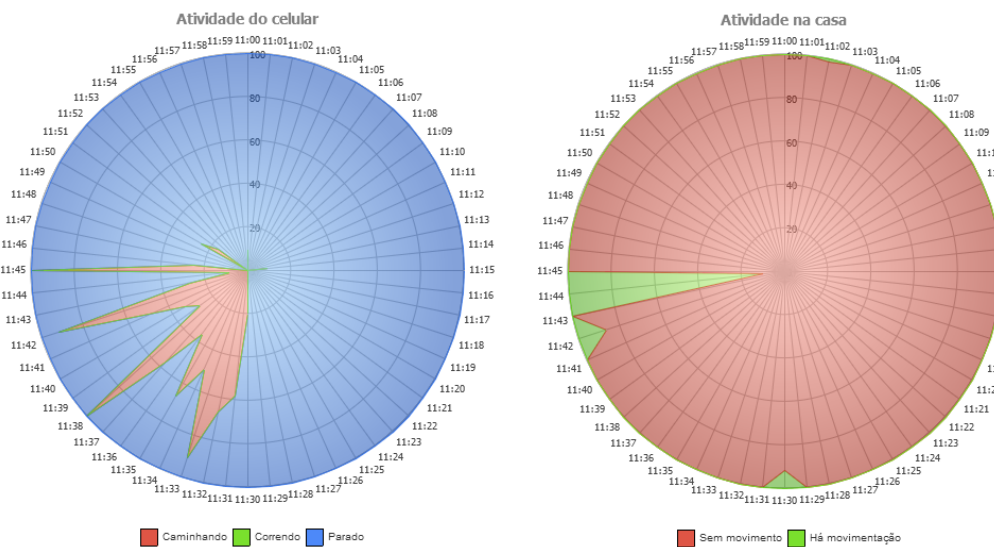


Figura 4.10: Gráfico que mostra a correlação entre os dados capturados

Além da localização para o caso dos acidentes, saber onde a pessoa passa a maior parte do seu tempo pode configurar ou não um quadro de ociosidade. Se, por exemplo, *Detector* identifica que o usuário está parado por muito tempo em sua sala, o que pode ser verificado pelo sensor de infravermelho, pode ser que a pessoa esteja vendo televisão por tempo excessivo. O médico, ao ver dados de um paciente nesta situação, pode aconselhar a pessoa a ter uma vida mais saudável.

4.4.3 Continuidade dos dados

Um dos problemas decorrentes de alguns celulares não registrarem novos eventos do acelerômetro com a tela desligada, foi o fato de que a captação de dados era intermitente,

pois alguns dos celulares tiveram a tela desligada acidental ou inconscientemente pelos seus usuários. Quando visualizamos o gráfico diário, tem-se uma falsa impressão de continuidade da informação, que é desmentida pelo gráfico por hora, como visto nas Figuras 4.11 e 4.12. No gráfico horário, há informação fragmentada, em que o somatório das atividades pode não corresponder ao somatório da atividade real executada pelo usuário, possivelmente com uma subestimação ou superestimação de uma atividade específica.

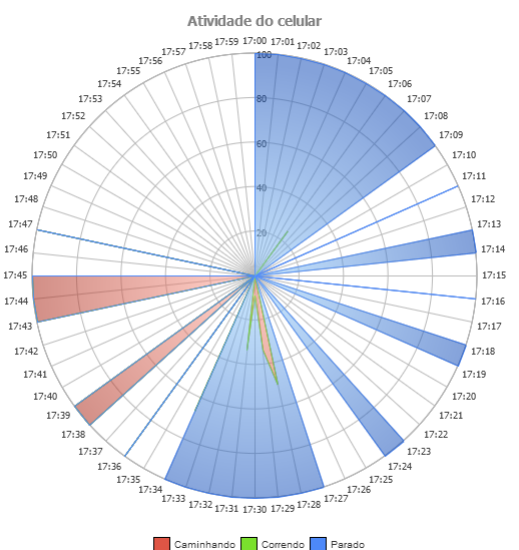


Figura 4.11: Gráfico com dados incompletos em uma hora

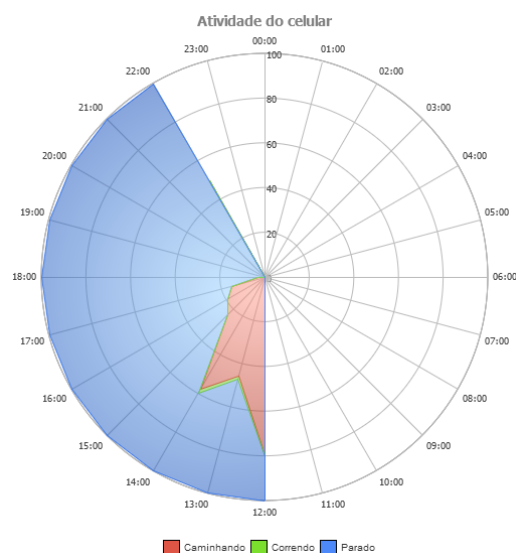


Figura 4.12: Gráfico do dia aparentemente completo

A vantagem de existir um sensor infravermelho é que a coleta não é interrompida por desligamentos de tela ou por falta de carga de bateria, pois o equipamento fica ligado constantemente na rede elétrica. Na eventualidade da falta de energia, pode-se arquitetar um sistema de *back-up*. A complexidade da informação é mais baixa, não sendo possível a inferência da atividade, mas o fato de a coleta ser ininterrupta é vantagem quando os dados do *Smartphone* não estão disponíveis.

4.5 Considerações sobre a utilização do *Detector*

Das opiniões de quem testou o *website*, quem avalia positivamente a questão do monitoramento o faz por razões médicas ou para sua própria proteção. Inúmeros casos de uso para este sistema podem ser pensados. Pelo fato da interface de visualização dos dados ser disponibilizada de forma a possibilitar acesso remoto, o primeiro uso lógico é o monitoramento da pessoa de interesse para que se possa garantir sua segurança ou integridade física. Desta forma, o indivíduo pode viver normalmente em sua casa, mesmo sozinho, podendo ser acompanhado, obviamente com seu consentimento, por familiares ou amigos, de forma a aumentar sua independência.

Com o acompanhamento por membros da família, podem ser identificados períodos de ociosidade excessiva, indicando que a pessoa possa estar com problemas de saúde, deitada em sua cama. Muitas vezes a negligência da pessoa consigo mesma pode piorar a sua situação de saúde, impedindo uma melhora mais rápida e, se uma intervenção é feita

com antecedência, o tempo de recuperação pode ser menor. Se o *Smartphone* é capaz de detectar quedas, um pronto atendimento pode significar uma drástica diferença na forma de atendimento.

Um outro uso para o *Detector* seria acompanhamento médico. Para pacientes que estão acima do peso, por exemplo, o sistema pode ser usado para verificar se a pessoa está realizando exercícios de forma regular. Muitas vezes os pacientes tendem a supervalorizar a quantidade de exercícios que fazem (Choudhury et al. 2008), tornando a informação disponível para o médico incompleta ou em desacordo com a realidade. Se o médico tem gráficos que refletem melhor as atividades diárias desse indivíduo, pode aconselhar melhor sua dieta e forma de perda de calorias.

Analogamente ao acompanhamento por um médico, a própria pessoa pode observar o seu desempenho em atividades físicas. A ferramenta, neste caso, é útil para reforço de metas pessoais.

5 CONCLUSÃO E TRABALHOS FUTUROS

O desenvolvimento do *Detector* permitiu a detecção de atividades realizadas por uma pessoa que carregasse um *Smartphone* por meio de seus acelerômetros. Os resultados mostraram que a detecção de atividades em tempo real em um celular moderno é possível, já que seu poder de processamento é bastante elevado. A constante alimentação de um banco de dados, com as atividades já classificadas e o instante em que ocorreram, são de muita utilidade se temos uma ferramenta propícia de visualização, que também foi apresentada neste trabalho.

A adição de um sensor de infravermelho foi a forma encontrada para suprir a carência de informação quando o celular não estava ligado, ou não estava registrando as atividades da pessoa. Não somente para indicar que há movimento em um recinto, pode-se usar a informação do sensor para complementar a informação das atividades detectadas no celular, podendo-se extrair, por exemplo, a localização do indivíduo. A aplicação, como um todo, pode ser uma maneira eficaz de verificar se a rotina da pessoa é normal e saudável.

O objetivo da construção de um sistema completo de monitoramento foi alcançado. Porém, vários quesitos do *Detector* podem ser aperfeiçoados.

5.1 Detecção de atividades

A inferência de atividades realizada foi demasiadamente simples. Para que o classificador de atividades atinja um nível superior na qualidade das decisões, diversas modificações no aplicativo do celular são necessárias. A primeira delas é a quantidade de atividades detectadas. A que foi mais comentada durante a apresentação deste trabalho é a detecção de quedas, que é muito útil para proteger pessoas idosas ou com dificuldade de locomoção. Para detectar quedas, um treinamento intensivo do algoritmo de *machine learning* deve ser feito neste sentido, preferencialmente com um ator profissional que consiga mimetizar esta ação o mais próximo possível da realidade.

A detecção de quedas possui uma característica particular que faz com que seu treinamento seja mais difícil. Para criar uma instância de *features* que reflitam a queda, a captura do acelerômetro deve ser iniciada no exato momento do início da queda, e finalizada antes de que os cálculos das *features* de uma nova instância do classificador seja iniciada. Isto não é necessário para o treinamento de uma caminhada, por exemplo, pois a captura dos dados neste caso é circular.

A identificação de novas atividades é importante para que se evite a falsa identificação de atividades. Casos apresentados neste trabalho, como identificação de corrida quando na realidade se estava subindo escadas, podem ser evitadas com adição dessas atividades. Além disso, mais atividades identificáveis implicam mais informação contextual para ser vista no *website*.

Outro ponto importante na acuidade da detecção é a quantidade de pessoas que realizam o treinamento do classificador. Cada pessoa tem uma assinatura digital, por assim dizer, de como realiza suas atividades. Quanto mais variações existirem nos dados, mais adaptável o algoritmo de decisão será para que a classificação se aproxime da realidade.

Quanto ao classificador utilizado, a árvore de decisão **J48** talvez não seja o algoritmo que traga os melhores resultados. Apesar de ter oferecido resultados condizentes com a realidade, uma pesquisa mais aprofundada pode ser feita no sentido de se aperfeiçoar a relação entre o algoritmo, as *features* empregadas e a quantidade de dados necessários para a maximização da correta classificação de atividades.

5.2 Outras melhorias

A página *WEB* para visualização pode ser melhorada no sentido de não somente exibir os dados, mas fornecer outras informações úteis à pessoa que está sendo monitorada. Uma estimativa para o gasto calórico poderia ser utilizável tanto pelo médico, para um acompanhamento mais preciso de uma dieta, quanto pela pessoa. A vantagem de haver detecção de diferentes tipos de atividades, é que a contabilização dos gastos calóricos pode ser mais precisa pois cada atividade tem um gasto calórico diferente.

Pensando ainda no gasto calórico, uma melhoria na ferramenta pode ser incorporada para que seja possível estabelecer metas diárias para determinados exercícios. A pessoa poderia programar a quantidade ou intensidade de exercícios desejados. Se há pouco exercício, o celular poderia avisar, por meio de notificação ou sinal sonoro, que a quantidade de exercícios praticada está abaixo do estabelecido. Similarmente a metas pré-estabelecidas, um aviso de alarme que soa para o usuário, indicando que ele está muito tempo sentado, é bom para profissionais de escritório, que trabalham por longos períodos de tempo na mesma posição. Pausas regulares e alongamentos promovem a qualidade de vida e evitam problemas de saúde laborais.

Se o sistema detecta quedas, uma outra melhoria seria informar, o quanto antes, pessoas responsáveis sobre o ocorrido. A grande vantagem em se usar um *Smartphone* como detector de atividades é justamente a sua função celular, que ironicamente muitas vezes é esquecida pelo fato do aparelho possuir tantas outras funções. Uma forma de aviso instantâneo é o envio de mensagens SMS. O SMS é uma forma de levar a informação de forma rápida e direta ao interessado, e não fazer o consumidor da informação procurá-la, como no caso do *website*.

Um ponto que necessita atenção é a questão da adesão ao sistema. Um questionamento a um público maior, de diferentes faixas etárias, sobre como o sistema se tornaria mais natural e útil, é fundamental para que possa ser usado de forma transparente e que não prejudique as atividades diárias dessas pessoas.

REFERÊNCIAS

- [Android HDP]ANDROID HDP. Acesso em abril de 2012. Disponível em: <<http://developer.android.com/guide/topics/wireless/bluetooth.html#HDP>>.
- [Android ICS]ANDROID ICS. Acesso em abril de 2012. Disponível em: <<http://www.android.com/about/ice-cream-sandwich/>>.
- [Android Mobile Platform]ANDROID Mobile Platform. Acesso em abril de 2012. Disponível em: <<http://www.android.com/>>.
- [Apple Inc.]APPLE INC. *Apple Iphone*. Acesso em abril de 2012. Disponível em: <www.apple.com>.
- [Arduino Team]ARDUINO TEAM. *Arduino*. Acesso em julho de 2012. Disponível em: <<http://www.arduino.cc/>>.
- [Bao e Intille 2004]BAO, L.; INTILLE, S. S. Activity recognition from user-annotated acceleration data. In: *Pervasive*. [S.l.: s.n.], 2004. p. 1–17.
- [Bersch 2012]BERSCH, R. *Introdução a TA*. Porto Alegre, RS, 2012. Acesso em abril de 2012. Disponível em: <www.assistiva.com.br/Introducao%20TA%20Rita%20Bersch.pdf>.
- [Bluetooth SIG]BLUETOOTH SIG. *Health Device Profile*. Acesso em abril de 2012. Disponível em: <https://developer.bluetooth.org/KnowledgeCenter/TechnologyOverview/Documents/HDP_SPEC.pdf>.
- [Choudhury et al. 2008]CHOUDHURY, T. et al. The mobile sensing platform: An embedded activity recognition system. *IEEE Pervasive Computing*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 7, n. 2, p. 32–41, 2008. ISSN 1536-1268. Disponível em: <<http://dx.doi.org/10.1109/MPRV.2008.39>>.
- [Google]GOOGLE. *Google Web Toolkit*. Acesso em julho de 2012. Disponível em: <<https://developers.google.com/web-toolkit/>>.
- [Hwaci]HWACI. *SQLite database*. Acesso em junho de 2012. Disponível em: <<http://www.sqlite.org>>.
- [Kwapisz et al. 2010]KWAPISZ, J. R. et al. Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.*, ACM, New York, NY, USA, v. 12, n. 2, p. 74–82, 2010. ISSN 1931-0145. Disponível em: <<http://doi.acm.org/10.1145/1964897.1964918>>.

[Mitchell 1997]MITCHELL, T. M. *Machine learning*. [S.l.]: McGraw-Hill, 1997. I-XVII, 1-414 p. (McGraw Hill series in computer science). ISBN 978-0-07-042807-2.

[Ravi et al. 2005]RAVI, N. et al. Activity recognition from accelerometer data. In: *Proceedings of the 17th conference on Innovative applications of artificial intelligence - Volume 3*. AAAI Press, 2005. (IAAI'05), p. 1541–1546. ISBN 1-57735-236-x. Disponível em: <<http://dl.acm.org/citation.cfm?id=1620092.1620107>>.