

UNIVERSIDADE FEDERAL DE ALAGOAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO
EM MODELAGEM COMPUTACIONAL DE CONHECIMENTO

GUSTAVO JOSÉ BARBOSA SILVA

**Reconhecimento de Atividades Humanas
Através de um Smartphone**

**Maceió
2014**

Gustavo José Barbosa Silva

Reconhecimento de Atividades Humanas Através de um Smartphone

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Modelagem Computacional de Conhecimento do Instituto de Computação da Universidade Federal de Alagoas.

Orientador: Prof. Dr. Leandro Dias da Silva

Maceió
2014

Catálogo na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico
Bibliotecário Responsável:

S586r Silva, Gustavo José.
Reconhecimento de atividade humana através de um Smartphone / Gustavo José
Barbosa Silva. – 2014.
61 f. : il.

Orientador: Leandro Dias da Silva.
Dissertação (Mestrado em Modelagem Computacional de Conhecimento) –
Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2014.

Bibliografia: f. 54-56.
Apêndices: f. 57-61.

1. Interação homem-máquina. 2. Smartphone. 3. Aprendizado do computador.
I. Título.

CDU: 004.5



UNIVERSIDADE FEDERAL DE ALAGOAS/UFAL
Programa de Pós-Graduação em Modelagem Computacional de Conhecimento
Avenida Lourival Melo Mota, Km 14, Bloco 09, Cidade Universitária
CEP 57.072-900 – Maceió – AL – Brasil
Telefone: (082) 3214-1364



Membros da Comissão Julgadora da Dissertação de Mestrado de Gustavo José Barbosa Silva, intitulada: “Reconhecimento de atividades humanas através de um smartphone”, apresentada ao Programa de Pós-Graduação em Modelagem Computacional de Conhecimento da Universidade Federal de Alagoas em 10 de março de 2014, às 13h00min, no auditório do Instituto de Computação da UFAL.

COMISSÃO JULGADORA



Prof. Dr. Leandro Dias da Silva

UFAL – Instituto de Computação

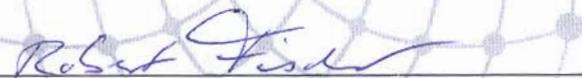
Orientador



Prof. Dr. Aydano Pamponet Machado

UFAL – Instituto de Computação

Examinador



Prof. Dr. Robert Fischer

UFAL – Campus Sertão

Examinador



Prof. Dr. Kyller Costa Gorgônio

UFCG – Unidade Acadêmica de Sistemas e Computação

Examinador

Maceió, março de 2014.

A minha mãe, Nasidy, que sempre me incentivou a ser cada vez melhor e a minha esposa, Amanda, por sua paciência e compreensão, dedico este trabalho.

AGRADECIMENTOS

A Deus, primeiramente, por toda alegria e saúde que ele proporciona a mim e a minha família.

Aos voluntários, pela ajuda gratuita e paciência durante os experimentos.

Ao Prof. Dr. Leandro Dias da Silva, pelo apoio, motivação e tempo dedicado ao meu trabalho.

Ao amigo Leonardo Medeiros, pela ajuda durante toda a elaboração desta dissertação.

Aos meus familiares e amigos, pelo incentivo incondicional.

E a todos que contribuíram direta ou indiretamente na elaboração deste trabalho.

RESUMO

O reconhecimento de atividades humanas é uma área de pesquisa em expansão (ZHANG; SAWCHUK, 2013) e tem como objetivo capturar o estado do usuário e do seu ambiente utilizando sensores heterogêneos (DAVIES; SIEWIOREK; SUKTHANKAR, 2008). Através de um monitoramento não-intrusivo das atividades de um indivíduo, pode-se aferir, por exemplo, se o mesmo está levando uma vida saudável, com prática frequente de atividades dinâmicas (andar, correr, subir e descer escadas), ou se está levando uma vida sedentária, na qual o indivíduo passa a grande maioria do seu tempo executando atividades estáticas (sentar, deitar ou ficar em pé). Considerando-se o viés médico, a assistência pervasiva tem um potencial significativo para aumentar a eficiência dos prestadores de assistência médica, mas tem como um de seus principais problemas o reconhecimento automático de atividades humanas diárias (ORWAT; GRAEFE; FAULWASSER, 2008). Além disso, um histórico com informações sobre as atividades executadas por um indivíduo durante um dado período pode ajudá-lo a ter uma vida mais saudável e menos sedentária. Propõe-se, neste trabalho, a criação de uma prova de conceito que consiga identificar, com um grau de certeza elevado, atividades comuns sendo executadas pelo usuário, a partir de dados coletados de um *smartphone* qualquer. Almeja-se, também, permitir a criação de alertas para o usuário, ou para um responsável indicado, quando certas condições customizáveis forem atendidas. Por fim, disponibilizar um histórico de todas as atividades executadas, inclusive com informações da posição geográfica do usuário, quando cada atividade foi identificada. Antes do desenvolvimento propriamente dito, criou-se um conjunto de dados com 10 voluntários que realizaram um circuito de atividades pré-definido. Em seguida, três abordagens foram utilizadas para gerar um modelo confiável de aprendizagem de máquina: impessoal, que usa dados de nove usuários para treinamento e de um para teste, que teve 89.4% de acurácia; pessoal, que é voltada para um único indivíduo, treinando e testando o modelo com dados distintos do mesmo, e que teve a melhor acurácia com 98.5%; e híbrida, que utiliza dados distintos de todos os dez voluntários para treinamento e para teste, e que obteve 98.16% de acurácia. Uma vez gerado o modelo, desenvolveu-se a prova de conceito proposta que foi testada por um voluntário no seu dia-a-dia apresentando resultados satisfatórios.

Palavras-chaves: Reconhecimento de atividades. *Smartphone*. Aprendizagem de máquina.

ABSTRACT

Human activity recognition is an expanding research area (ZHANG; SAWCHUK, 2013) and aims to capture the user state and its environment using heterogeneous sensors (DAVIES; SIEWIOREK; SUKTHANKAR, 2008). Through non-intrusive activities monitoring of an individual, we might infer, for instance, if he is leading a healthy lifestyle, practicing frequently dynamic activities (like walking, running, climbing and descending stairs), or if he is leading a sedentary one, spending the majority of his time static (sitting, lying or standing). Considering the health bias, the pervasive care has significant potential to increase the efficiency of health care providers, but it has as one of its main problems the automatic recognition of daily human activities (ORWAT; GRAEFE; FAULWASSER, 2008). Moreover, a report with information on the activities performed by an individual during a given period can help him have a healthier and less sedentary life. It is proposed the construction of a proof of concept that can identify, with a high accuracy, common activities being performed by the user, based on data collected from a smartphone. It is also desired the generation of alerts to the user, or to other person defined by him, when certain customizable conditions are met. And, finally, provide a history of all activities performed, including geographical information from the user, when each activity was identified. Before the development itself, a data set was created with 10 volunteers who performed a pre-defined activities circuit. Three approaches were used to generate a reliable model of machine learning: impersonal, which uses data from nine users for training and one for testing, which achieved 89.4% accuracy; personal, that is focused on a single individual, training and testing the model with different data from him, it had the best accuracy, 98.5%; and hybrid, which uses different data from all ten volunteers for training and testing, and obtained 98.16% accuracy. Once generated the model, the proposed proof of concept was developed, and then tested by a volunteer in his everyday, and overall achieved satisfactory results.

Keywords: Activity recognition. Smartphone. Machine learning.

LISTA DE ILUSTRAÇÕES

Figura 1	– Exemplo de dois conjuntos de dados que são utilizados para reconhecimento de atividades humanas a partir de uma abordagem baseada em visão.	16
Figura 2	– Exemplo de uma casa equipada com 14 sensores em uma abordagem ambiental.	16
Figura 3	– Interface “ <i>Explorer</i> ” do software Weka com os dados brutos gerados que serão aferidos com a abordagem híbrida descrita em 3.4.	19
Figura 4	– Em 5(a), tem-se a os itens necessários à realização dos experimentos, um iPhone e uma <i>armband</i> alongada. Ao lado, a Figura 5(b) mostra uma voluntária utilizando o sistema com o iPhone na cintura.	25
Figura 5	– Tela do iPhone demonstrando a página Web desenvolvida para captura de dados brutos para geração do conjunto de dados. Em 6(a), tem-se a tela principal da página onde o usuário deve preencher os dados para poder iniciar a captura. A Figura 6(b), retrata a captura sendo realizada.	26
Figura 6	– Características de todos os voluntários reduzidas para duas dimensões utilizando PCA.	28
Figura 7	– Comparação entre as atividades esperadas e as atividades reconhecidas referentes à abordagem impessoal.	30
Figura 8	– Características dos três voluntários reduzidas para duas dimensões utilizando PCA.	32
Figura 9	– Comparação entre as atividades esperadas e as atividades reconhecidas referentes ao indivíduo 1.	33
Figura 10	– Comparação entre as atividades esperadas e as atividades reconhecidas referentes ao indivíduo 2.	34
Figura 11	– Comparação entre as atividades esperadas e as atividades reconhecidas referentes ao indivíduo 3.	36
Figura 12	– Comparação entre as atividades esperadas e as atividades reconhecidas referentes à abordagem híbrida.	38
Figura 13	– Comparação entre quatro algoritmos de reconhecimento (<i>IBk</i> , <i>SVM</i> , <i>J48</i> , e <i>Logistic</i>) utilizando os dados coletados.	39
Figura 14	– Arquitetura conceitual do sistema proposto.	40
Figura 15	– Tela do iPhone demonstrando a página Web principal de uso do sistema onde o usuário deve selecionar com qual frequência as atividades devem ser reconhecidas	41
Figura 16	– Diagrama de atividade da função principal de uso do sistema.	42
Figura 17	– Tela do iPhone mostrando os botões “Parar” e “Emergência”.	43

Figura 18 – Gráfico comparando as atividades reais e a atividades reconhecidas pelo modelo para o indivíduo 1. O mesmo foi subdividido em atividades estáticas, 19(a), e atividades dinâmicas, 19(b).	44
Figura 19 – Comparação entre as atividades executadas e as atividades identificadas pelo modelo para o indivíduo 2. Novamente, o gráfico foi subdividido em atividades estáticas, 20(a), e atividades dinâmicas, 20(b).	45
Figura 20 – Tela do iPhone demonstrando a página Web desenvolvida para cadastro de regras customizáveis que serão verificadas a cada nova atividade reconhecida. Em 21(a), tem-se a tela de cadastro referente ao tipo de regra <i>1 - Tempo máximo em uma atividade</i> , e em 21(b), demonstra-se a tela relacionada ao tipo de regra <i>2 - Transição entre atividades</i>	46
Figura 21 – Gráfico ilustrando as atividades reconhecidas num exemplo prático de uso do sistema proposto neste trabalho durante 2 horas e 15 minutos.	49
Figura 22 – Quatro exemplos mostrando a posição geográfica do indivíduo no momento em que atividade foi reconhecida.	50

LISTA DE TABELAS

Tabela 1 – Características no domínio do tempo utilizadas.	20
Tabela 2 – Exemplo de uma matriz de confusão. Abordagem impessoal discutida na Seção 3.2	22
Tabela 3 – Informações sobre os indivíduos que participaram dos experimentos.	24
Tabela 4 – Dados brutos gerados.	27
Tabela 5 – O número de exemplos por atividade em todo o conjunto de dados, coletado de 10 usuários.	27
Tabela 6 – Acurácia por indivíduo na abordagem impessoal.	31
Tabela 7 – Matriz de confusão - Abordagem Impessoal	31
Tabela 8 – O número de exemplos por atividade no conjunto de dados coletados do indivíduo 1.	33
Tabela 9 – Matriz de confusão - Abordagem Pessoal - Indivíduo 1	34
Tabela 10 – O número de exemplos por atividade no conjunto de dados coletados do indivíduo 2.	35
Tabela 11 – Matriz de confusão - Abordagem Pessoal - Indivíduo 2	35
Tabela 12 – O número de exemplos por atividade no conjunto de dados coletados do indivíduo 3.	36
Tabela 13 – Matriz de confusão - Abordagem Pessoal - Indivíduo 3	36
Tabela 14 – Matriz de confusão - Abordagem Híbrida	37
Tabela 15 – Atividades reconhecidas pelo modelo em um uso prático durante 2 horas e 45 minutos.	48

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Histórico e Motivação	12
1.2	Problemática	12
1.3	Objetivos	13
1.4	Organização do Texto	13
2	REFERENCIAL TEÓRICO	15
2.1	Reconhecimento de Atividades Humanas	15
2.2	Aprendizagem de Máquina	18
2.2.1	Preprocessamento	19
2.2.2	Extração de características	20
2.2.3	Classificação	20
2.2.4	Avaliação	22
2.3	HTML5/JavaScript	22
2.4	Octave	23
3	EXPERIMENTOS E CONSTRUÇÃO DO MODELO	24
3.1	Gerando o conjunto de dados	24
3.2	Abordagem Impessoal	29
3.3	Abordagem Pessoal	31
3.3.1	Resultados do Indivíduo 1	33
3.3.2	Resultados do Indivíduo 2	34
3.3.3	Resultados do Indivíduo 3	35
3.3.4	Conclusões sobre a abordagem pessoal	37
3.4	Abordagem Híbrida	37
3.5	Comparação entre SVM e outros algoritmos	38
4	IMPLEMENTAÇÃO DA PROVA DE CONCEITO	40
4.1	Página principal de uso	41
4.1.1	Uso do sistema pelo indivíduo 1	44
4.1.2	Uso da sistema pelo indivíduo 2	45
4.2	Criação de regras	46
4.3	Acompanhamento das atividades realizadas	48
5	CONCLUSÕES E TRABALHOS FUTUROS	51
5.1	Trabalhos futuros	52

	Referências	54
	APÊNDICE A – CÓDIGOS EM OCTAVE	57
A.1	Calcular velocidade angular	57
A.2	Gerar características	57
A.3	Normalizar características	59
A.4	Gerar modelo	60
A.5	Reconhecer atividade	61

1 INTRODUÇÃO

O reconhecimento de atividades humanas (RAH) é uma área de pesquisa em expansão (ZHANG; SAWCHUK, 2013) e tem como objetivo capturar o estado do usuário e do seu ambiente utilizando sensores heterogêneos (DAVIES; SIEWIOREK; SUKTHANKAR, 2008). Cada vez mais, existe a necessidade de uma monitoração não-intrusiva de um indivíduo para fins médicos, no caso de um responsável por um idoso que deseja saber se o indivíduo está conseguindo ter uma vida normal, por exemplo, ou para fins pessoais, quando um usuário deseja obter um histórico de suas atividades durante o dia. A possibilidade de gravar e reconhecer atividades individuais é essencial para determinar o grau de desempenho funcional e o nível de atividade de uma pessoa (KARANTONIS et al., 2006).

Considerando-se o viés médico, a assistência pervasiva, ou seja, o uso de tecnologias computacionais para integrar o monitoramento da saúde e intervir transparentemente no dia a dia das pessoas, tem um potencial significativo para aumentar a eficiência dos prestadores de assistência médica, mas tem como um de seus principais problemas o reconhecimento automático de atividades humanas diárias (ORWAT; GRAEFE; FAULWASSER, 2008).

De acordo com o Instituto Brasileiro de Geografia e Estatística (IBGE) a quantidade de brasileiros com 60 anos ou mais cresceu 55% entre 2001 e 2011. Neste ano, já são mais de 23.5 milhões de idosos que representam 12% da população total do Brasil. Além disso, a população de idosos morando sozinhos vem aumentando (IBGE, 2012). Faz-se necessário, então, um cuidado maior com esta parcela devido às suas limitações e fragilidades intrínsecas.

Pesquisas mostram que idosos temem mais casas de repouso e a perda da independência do que a morte propriamente dita, e a metade dos indivíduos pesquisados estão dispostos a usar novas tecnologias para permitir sua independência (CLARITY, 2007). Ainda neste campo, a análise a longo prazo de atividades humanas pode ser útil na prevenção antecipada de doenças (CZABKE; MARSCH; LUETH, 2011).

Considerando-se, também, o viés pessoal, o monitoramento de atividades humanas é importante para ajudar os indivíduos a manter uma vida ativa e saudável. Há um mercado vasto de aplicativos que disponibilizam aos usuários um histórico de suas atividades realizadas.

Devido a sua miniaturização, baixo custo e capacidade de gravação de sinais de movimento com sistemas usáveis e não-intrusivos, sensores de movimento se tornaram muito importantes nas aplicações de reconhecimento de atividades humanas (SILVA, 2013). Acelerômetros e giroscópios também foram utilizados em muitos estudos sobre monitoração de atividades cotidianas (KARANTONIS et al., 2006; CZABKE; MARSCH; LUETH,

2011; ANGUITA et al., 2012; WEISS; LOCKHART, 2012).

A classificação de informação de movimento, coletada através de sensores, em classes de atividades é normalmente feita utilizando técnicas de aprendizagem de máquina que requerem a extração de parâmetros dos dados (características) para treinar um classificador que irá prever para novos dados qual atividade está associada com os mesmos. Os classificadores devem ser genéricos o suficiente para abstrair as diferenças entre indivíduos distintos realizando a mesma atividade cotidiana.

1.1 Histórico e Motivação

Esta dissertação está inserida no campo de mineração de dados, referente à gravação e à classificação de atividades humanas usando sensores de movimento (acelerômetros e giroscópios) inseridos em *Smartphones*.

Com o envelhecimento progressivo da população e o investimento limitado na saúde pública, uma atenção maior é voltada para o monitoramento de atividades diárias, aumentando a assistência aos pacientes e ajudando-os a tomar conta de si próprios, reduzindo, assim, a assistência médica convencional e caminhando para uma assistência remota (KARANTONIS et al., 2006).

Neste contexto, aplicações para monitoramento de atividades humanas estão surgindo tanto para aqueles que querem atender aos seus parentes, quanto para as pessoas que querem verificar o estado de suas atividades físicas e melhorá-las, caso necessário.

A motivação por trás desse projeto é a criação de uma prova de conceito na forma de um sistema de monitoração de baixo custo que poderia ajudar as pessoas a rastrear suas atividades físicas cotidianas e ajudar os cuidadores na assistência a idosos, inclusive possibilitando aos últimos o contato com os primeiros em caso de uma emergência.

1.2 Problemática

A grande maioria dos estudos que tratam o tema de reconhecimento de atividades humanas usando sensores inerciais usáveis são realizados em ambientes controlados e utilizam sensores específicos que restringem a mobilidade dos usuários (MAURER et al., 2006).

Deseja-se construir, como prova de conceito, um sistema Web que permita ao um usuário em sua cidade a partir de qualquer dispositivo móvel com acelerômetro, giroscópio e GPS embutido (um *smartphone* simples, por exemplo), a identificação da atividade sendo executada pelo mesmo, entre as sete opções seguintes: andar; correr; subir escadas; descer escadas; em pé; sentado; e deitado.

Além da identificação com uma taxa de probabilidade, deseja-se que um histórico de todas as atividades reconhecidas esteja disponível ao usuário, e, também, que alertas sejam gerados para os responsáveis cadastrados previamente de acordo com algumas regras

customizáveis, como, por exemplo, caso o usuário esteja no estado *Andando* e, abruptamente, passe e permaneça no estado *Deitado* em uma posição geográfica diferente da sua cama, pode-se inferir que aconteceu uma queda, então um alerta contendo informações pertinentes será gerado para que o responsável possa tomar alguma medida.

1.3 Objetivos

Construir-se-á, como prova de conceito, um sistema Web para identificação de atividades humanas tendo em mente os seguintes objetivos:

- Ter baixo custo, pois precisará somente de um servidor Web e de um *smartphone* comum;
- Exigir o mínimo possível de processamento tanto no dispositivo móvel, quanto no servidor Web;
- Poderá ser acessado através de qualquer dispositivo móvel que possua os requisitos mínimos (acelerômetro, giroscópio e GPS), não sendo necessária a aquisição de hardware proprietário ou software específico;
- Identificará entre sete possíveis atividades comuns em qualquer ambiente, não somente na casa do indivíduo: “ANDAR”, “CORRER”, “SUBIR ESCADAS”, “DESCER ESCADAS”, “EM PÉ”, “SENTADO”, “DEITADO”;
- Gerará alertas para o responsável designado pelo usuário de acordo com regras customizáveis, incluindo, entre outras informações, a atividade identificada com uma certa probabilidade e a posição geográfica do indivíduo;
- Disponibilizará um histórico das atividades identificadas com tempo médio de cada durante um período de tempo pré-definido: dia, semana ou mês;
- Conterá um botão de emergência para casos em que o usuário precisa contatar seu responsável urgentemente.

O usuário do sistema poderá escolher se deseja que o responsável tenha acesso aos seus dados de posição geográfica ou não.

1.4 Organização do Texto

O restante deste trabalho está organizado da seguinte forma:

- No Capítulo 2 é apresentado o referencial teórico necessário para que os objetivos sejam alcançados. Este contém informações gerais sobre aprendizagem de máquina, e algoritmos utilizados, além de um resumo sobre as tecnologias utilizadas, como HTML5/JavaScript e Octave;

- No Capítulo 3 são disponibilizadas informações sobre os experimentos realizados e as abordagens utilizadas para gerar os modelos;
- No Capítulo 4 estão disponíveis as informações de implementação da prova de conceito e resultados do uso prático da mesma;
- No Capítulo 5 estão contempladas as conclusões e trabalhos futuros.
- E no Apêndice A estão detalhados alguns códigos em Octave utilizados no sistema proposto.

2 REFERENCIAL TEÓRICO

Neste capítulo serão descritos os métodos e tecnologias utilizadas para que os objetivos deste trabalho sejam alcançados.

2.1 Reconhecimento de Atividades Humanas

O reconhecimento correto de atividades humanas utilizando sensores vem se tornando cada vez mais importante ao longo dos anos, e tem como objetivo criar ou melhorar sistemas de auxílio no cuidado de idosos, monitoramento de saúde e de atividades físicas, e na assistência a pessoas com distúrbios cognitivos (LOPES; MENDES-MOREIRA; GAMA, 2012).

Analisando as atividades de um usuário ao longo de um período de tempo, podem-se identificar tendências ou hábitos frequentes e, com base nestas informações, disponibilizar um conteúdo específico voltado para este usuário que pode agregar valor a sua atividade, como por exemplo, indicar a um indivíduo que pratica corrida frequentemente em um mesmo local, onde há um bebedouro mais próximo (FIGO et al., 2010).

O processo de reconhecimento de atividades começa com a coleta dos dados brutos, várias abordagens já foram utilizadas e podem ser classificadas em três categorias principais (ABBATE et al., 2010):

- Baseadas em visão;
- Ambientais;
- Usáveis.

As abordagens baseadas em visão utilizam câmeras fixas que gravam continuamente o movimento do indivíduo. Os dados obtidos são submetidos a algoritmos de imagem específicos que conseguem reconhecer padrões pré-definidos. Suas desvantagens são o longo tempo e custo de instalação, o espaço limitado de atuação (somente onde há câmeras instaladas) e a violação de privacidade. Na Figura 1 temos um exemplo de dois conjuntos de dados que são utilizados para o reconhecimento de atividades humanas a partir de uma abordagem baseada em visão.

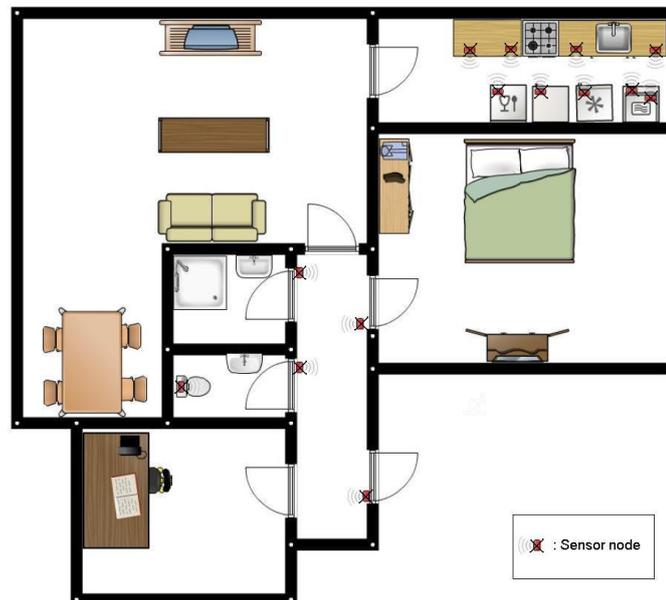
O uso de dispositivos de ambiente, abordagem ambiental, é baseado na instalação de sensores nos lugares a serem monitorados. Quando os indivíduos interagem com o ambiente, dados são gerados e alarmes pré-definidos podem ser ativados. No exemplo desta abordagem na Figura 2, os autores (KASTEREN et al., 2008) instalaram 14 sensores na casa de um homem de 26 anos e os deixaram ativos por 28 dias o que gerou 2120 eventos dos sensores e 245 instâncias de atividades.

Figura 1 – Exemplo de dois conjuntos de dados que são utilizados para reconhecimento de atividades humanas a partir de uma abordagem baseada em visão.



Fonte: (YU; KIM; CIPOLLA, 2010).

Figura 2 – Exemplo de uma casa equipada com 14 sensores em uma abordagem ambiental.



Fonte: (KASTEREN et al., 2008).

Na abordagem utilizada neste trabalho, usável, um ou mais dispositivos são usados pelo indivíduo. Geralmente são equipados com sensores de movimento, como acelerômetros e giroscópio, cujos valores transmitidos via rádio são analisados. Entre as vantagens deste método estão o custo baixo de instalação, tamanho pequeno, e a possibilidade de utilização tanto em um ambiente controlado (laboratório de pesquisa ou hospital), como em um ambiente externo qualquer (casa do indivíduo).

Vários trabalhos anteriores também utilizaram acelerômetros e giroscópios para o monitoramento de atividades diárias, mas nenhum tem as mesmas características da prova de conceito proposta:

- (ZHANG; SAWCHUK, 2013) em Los Angeles, nos Estados Unidos, conseguiram uma acurácia de 96.1% identificando em 14 indivíduos, 09 atividades do cotidiano,

mas, para isso, além do usuário utilizar o dispositivo proprietário *MotionNode* na cintura, o mesmo também precisava carregar um *notebook* todo o tempo, limitando sua usabilidade;

- (MAURER et al., 2006) em Munique, na Alemanha, conseguiram diferentes acurácias dependendo da posição onde o *eWatch*, dispositivo proprietário, estava posicionado. Seis destes dispositivos foram colocados ao mesmo tempo em posições diversas no corpo dos indivíduos para identificar 6 atividades possíveis, também limitando sua usabilidade no dia a dia.
- Também em Munique, (CZABKE; MARSCH; LUETH, 2011) implementaram um novo algoritmo em um microcontrolador proprietário, *Motionlogger*, e obtiveram uma acurácia significativa maior que 90% na identificação de 4 atividades possíveis, em 10 indivíduos carregando o dispositivo no bolso. Este trabalho obteve um resultado expressivo, mas também depende de um hardware proprietário;
- (ANGUITA et al., 2012) em Gênova, na Itália, criaram um conjunto de dados a partir de um *smartphone* Samsung Galaxy S2 e obtiveram resultados satisfatórios identificando 6 atividades com 89% de acurácia, com uma abordagem diferenciada voltada para dispositivos com pouco poder computacional, mas restringiram-se à criação do conjunto de dados. Não expandiram a implementação da solução para possibilitar a obtenção de dados de atividades executadas pelo indivíduo em tempo real.
- (WEISS; LOCKHART, 2012) em Nova York, nos Estados Unidos, a partir de uma visão semelhante à utilizada neste trabalho, compararam três abordagens de construção de modelos, impessoal, pessoal e híbrida, e obtiveram conclusões semelhantes, mas os resultados de sua abordagem impessoal ficaram muito aquém dos encontrados neste trabalho, muito provavelmente porque o experimento com alguns indivíduos específicos pode ter prejudicado o resultado como um todo. Os autores também não aplicaram a solução em um ambiente em tempo real.

Depois dos dados coletados pelos sensores, estes devem ser classificados com os rótulos de cada atividade executada. Para isto, utilizam-se técnicas de aprendizagem de máquina que requerem a extração de métricas dos dados de movimento para treinar um classificador para prever qual atividade executada a partir dos novos dados adquiridos. Classificadores devem ser genéricos o suficiente para entender as variações de pessoa a pessoa para cada atividade executada.

2.2 Aprendizagem de Máquina

A predição de atividades humanas é normalmente tratada como um problema de classificação. Uma vez que o conjunto de dados está pronto, um algoritmo de classificação precisa ser implementado. Em 1959, Arthur Samuel definiu que aprendizagem de máquina, ou *machine learning*, “é o campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados” (SIMON, 2013). Em termos mais práticos, Tom Mitchell definiu que “um programa de computador *aprende* de uma experiência E referente a uma tarefa T e com uma performance P, se sua performance em T, medida por P, melhorar com a experiência E” (MITCHELL, 1997).

A base da aprendizagem de máquina, como Samuel a descreveu, é criar um modelo e um classificador capazes de aprender utilizando novos dados. O modelo representa as instâncias dos dados (normalmente cada instância representa uma janela de dados com tamanho fixo) e funções dessas instâncias nos dados de treinamento que serão usadas pelo classificador para aferir a atividade reconhecida em novos dados (SILVA, 2013). Aprendizagem de máquina é uma área da ciência da computação preocupada com problemas de indução nos quais um modelo subjacente para propósitos de predição ou descrição precisa ser descoberto, baseado em propriedades conhecidas aprendidas dos dados de teste (WILDE, 2011).

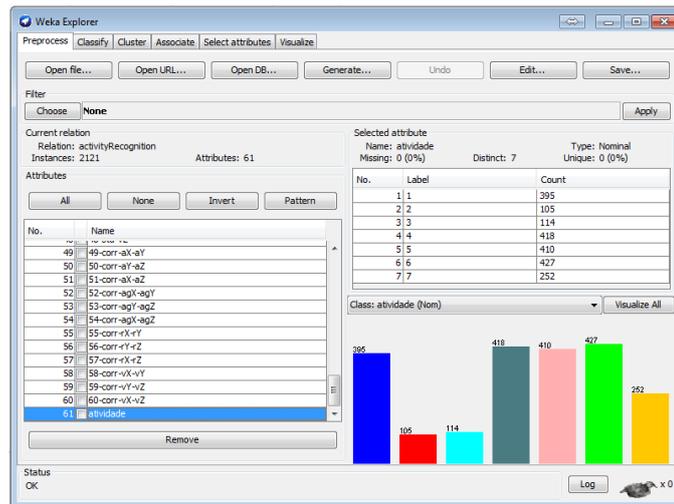
Existem duas grandes categorias de algoritmos de aprendizagem de máquina:

- **Aprendizagem supervisionada** utiliza dados rotulados com cada referida classe para treinar um algoritmo, que, então, consegue classificar dados sem rótulo (WILDE, 2011). O processo de classificação pode ser implementado de maneira *offline*, ou seja, coletam-se e armazenam-se dados brutos e posteriormente é realizada a classificação, ou de maneira *online*, quando os dados são classificados imediatamente depois que são coletados. Neste trabalho, as duas formas de classificação foram implementadas, a primeira para gerar o conjunto de dados, e a segunda que é utilizada na aplicação prática proposta;
- **Aprendizagem não-supervisionada** é um método que tenta construir modelos de reconhecimento diretamente de dados não rotulados. Esta abordagem utiliza métodos de estimação de densidade para descobrir grupos de exemplos similares e, assim, criar modelos de aprendizagem (WILDE, 2010).

Cinco passos básicos foram utilizados para conseguir um modelo confiável para reconhecer atividades humanas: coleta dos dados (Seção 3.1); pré-processamento (Subseção 2.2.1); extração de características (Subseção 2.2.2); classificação (Subseção 2.2.3); e avaliação (Subseção 2.2.4). Para implementá-los, utilizaram-se as ferramentas de aprendizagem de máquina compostas na coleção de algoritmos do software de código aberto Weka (*Waikato Environment for Knowledge Analysis*) (HALL et al., 2009), Figura 3. Criado em

1992, o Weka é reconhecido hoje como um sistema de referência em mineração de dados e aprendizagem de máquina (PIATETSKY-SHAPIRO, 2005). Esta coleção *open-source* inclui ferramentas para pré-processamento, classificação, regressão, clusterização, regras de associação, e visualização, bem como ferramentas para avaliação de algoritmos (WILDE, 2010).

Figura 3 – Interface “*Explorer*” do software Weka com os dados brutos gerados que serão aferidos com a abordagem híbrida descrita em 3.4.



Fonte: elaborado pelo autor.

2.2.1 Pré-processamento

Os dados brutos geralmente precisam ser pré-processados para serem fornecidos ao kit de ferramentas Weka que opera com arquivos no formato ARFF, faz-se necessário, então, que todos os arquivos tenham este formato.

Dados brutos de acelerômetros e giroscópios precisam ser divididos, sequencialmente, em janelas, para serem pré-processados. Utilizou-se, neste trabalho, uma abordagem comum de janelas deslizantes, na qual o sinal é dividido em janelas sem intervalos, com 50% de sobreposição, ou seja, cada janela é dividida em 50% de dados da janela anterior e 50% de novos dados. Desta forma, as transições entre as janelas acontecerão mais suavemente.

Dados brutos precisam ser separados em um conjunto de treinamento e um de teste. O conjunto de treinamento será utilizado para treinar o algoritmo de reconhecimento e o de teste será então usado para avaliá-lo depois do treinamento. É muito importante não usar amostras do conjunto de teste para treinar o classificador para não gerar resultados enviesados e garantir que o classificador será avaliado utilizando novos dados.

O kit de ferramentas Weka tem algumas estratégias para separar os dados, como a utilizada neste trabalho, validação cruzada. Nesta uma porção pré-definida (o usuário escolhe quantas dobras, ou *folds*, quer que seus dados sejam divididos) do conjunto de

dados é utilizada para treinamento e o resto para teste, evitando-se, assim, uma partição com classes mais representadas no conjunto de treinamento (WILDE, 2011). Outra estratégia é a separação por percentual de divisão, na qual geralmente um terço dos dados é separado para teste e os outros dois terços para treinamento.

2.2.2 Extração de características

Para cada janela, um conjunto de características é extraído para categorizar os sinais. Estas características, ou *features*, são utilizadas como entrada para os algoritmos de reconhecimento que associam cada janela com uma atividade (WILDE, 2011). Vários tipos de características podem ser extraídas dos dados de movimento, mas, neste trabalho, devido à restrição de se conseguir resultados satisfatórios no reconhecimento de atividades exigindo o mínimo possível de processamento nos dispositivos envolvidos, restringiu-se a um conjunto de características no domínio do tempo, que são métricas matemáticas e estatísticas simples e de fácil cálculo. Na Tabela 1 estão as características utilizadas.

Tabela 1 – Características no domínio do tempo utilizadas.

Característica	Fórmula	Aplicação
Mean	Separa os dados em duas metades	Suavização dos dados, calibração dos eixos Discriminar posturas diferentes
Min, Max	Valores máximos e mínimos	Discriminar entre correr e andar
Desvio padrão	Raiz quadrada da variância	Estabilidade do sinal
Correlação	$\rho_{x,y} = \frac{cov(x,y)}{\sigma_x \sigma_y}$	Mede a força e a direção de um relacionamento linear entre dois sinais
Velocidade angular	Ângulo entre o eixo de aceleração e gravidade nos 3 eixos	Determinar a orientação, detectar quedas

Fonte: (FIGO et al., 2010).

De acordo com (FIGO et al., 2010), quase todas as características do domínio do tempo são propícias para dispositivos móveis, levando em conta que operações de correlação têm um razoável custo computacional.

Para melhor visualização das características calculadas, utilizou-se PCA (*Principal Components Analysis*), que provê uma sequência das melhores aproximações lineares desses dados (HASTIE; TIBSHIRANI; FRIEDMAN, 2009), para reduzir a dimensão dos mesmos e, assim, conseguir identificar, visualmente, as diferenças entre as classes. Foram selecionadas as duas primeiras *principal components* para gerar uma visualização em duas dimensões dos dados coletados.

2.2.3 Classificação

Depois de extraídas as características dos sinais, foram aplicadas técnicas de aprendizagem de máquina para construir um classificador. É possível usar a suíte Weka para

implementar estes algoritmos de reconhecimento. Weka divide os classificadores em *lazy methods* (k-NN), *decision tree learners* (C4.5), *function-based learners* (SVM), além de métodos “*Bayesian*” e *miscellaneous*.

No algoritmo k-NN, ou k-Vizinhos mais próximos, a classe associada a um vetor x é a classe com o maior número de votos coletados de k amostras próximas ao vetor x . Estes classificadores são baseados em memória e não necessitam de um modelo para se ajustar (CAMPILHO, 2013). Para encontrar as amostras mais próximas, o algoritmo pode utilizar a distância Euclidiana. A classe majoritária dos k vizinhos mais próximos encontrada é atribuída para a instância de teste. Este é um algoritmo rápido e sua complexidade independe do tamanho de classes. No Weka, k-NN é implementado usando *Instance-Based k*, ou IBk.

Árvores de decisão, ou *decision trees*, utilizam uma árvore na qual para cada atributo, um galho é gerado para cada possível resultado de um teste. O algoritmo para quando encontra uma folha, que representa um classe (WILDE, 2011), e utiliza uma abordagem dividir-e-conquistar (depois da árvore treinada, uma nova instância é classificada começando na raiz e seguindo o caminho definido pelos resultados de testes para a mesma). A implementação do Weka desse algoritmo é o J48.

Regressão logística multinomial, ou *multinomial logistic regression*, aplica a máxima verossimilhança depois de transformar as características, utilizando a função logística (inversa da sigmóide), para se obter uma classe como predição. É implementada no Weka pelo algoritmo *Logistic*.

SVM são classificadores lineares que localizam um hiperplano separado no espaço das classes e classifica os pontos no mesmo. O objetivo é encontrar o maior hiperplano na margem separando duas classes. As instâncias com a menor distância ao hiperplano são definidas como vetores de suporte. O custo computacional aumenta quando desejam-se separar mais classes. Os algoritmos SVM definem uma função núcleo, ou *kernel*, que é responsável por um papel importante no produto de pontos no espaço das classes. Há vários algoritmos SVM disponíveis no Weka, como por exemplo SMO e LibSVM. Devido à aceitação expressiva na comunidade acadêmica, escolheu-se, neste trabalho, utilizar o núcleo RBF (*Radial Basis Function*) e o algoritmo LibSVM, que é uma biblioteca popular para SVMs, criada em 2000 e mantida até hoje, desenvolvida em C e C++ que suporta classificação e regressão, e está presente em diversos softwares de aprendizagem de máquina, além de ser suportada em linguagens como R e Octave (CHANG; LIN, 2011).

Além dos SVMs, foram comparados o desempenho de outros três algoritmos implementados no Weka (IBk, J48, e Logistic) com parâmetros padrões, com exceção do IBk no qual foram conseguidos melhores resultados com $k=3$. Os resultados desta comparação foram detalhados em 3.5.

2.2.4 Avaliação

Há várias formas diferentes para avaliar o desempenho de um algoritmo de reconhecimento e as mais presentes na literatura, que foram utilizadas neste trabalho, são as matrizes de confusão, a acurácia, a precisão e a sensibilidade (SILVA, 2013).

Acurácia é a taxa de sucesso global, que mede o desempenho como um todo do algoritmo no que diz respeito a decisões corretas (MENDONCA, 2009). Quando lidando com um problema multi-classe (cada atividade representa uma classe) o resultado da classificação pode ser representado utilizando uma matriz $n \times n$ para n classes chamada matriz de confusão. Como visto na Tabela 2, cada número representa a quantidade de instâncias pertencentes à classe i classificadas como uma classe j . O resultado ideal é uma matriz diagonal onde $i=j$ para cada uma das n classes.

A precisão mede a fração de exemplos classificados como positivos que são realmente positivos, enquanto que a sensibilidade mede a proporção de casos positivos que foram corretamente identificados (DAVIS; GOADRICH, 2006).

Tabela 2 – Exemplo de uma matriz de confusão. Abordagem impessoal discutida na Seção 3.2

Real \ Pred.	Andando	S. Escadas	D. Escadas	Sentado	Em Pé	Deitado	Correndo	Total	Sensitividade
Andando	371	11	5	0	0	0	8	395	93.9%
S. Escadas	20	74	9	0	0	0	2	105	70.4%
D. Escadas	12	3	84	0	1	0	14	114	73.7%
Sentado	0	0	0	363	55	0	0	418	86.8%
Em pé	0	0	0	20	399	0	1	420	95.0%
Deitado	0	0	0	1	32	378	6	417	90.6%
Correndo	1	0	1	0	2	0	248	252	98.4%
Total	404	88	99	384	489	378	279	2121	
Precisão	91.8%	84.1%	84.8%	94.5%	81.6%	100%	88.8%		

Fonte: elaborado pelo autor.

2.3 HTML5/JavaScript

Desejou-se, neste trabalho, que a prova de conceito desenvolvida funcionasse em qualquer dispositivo com os requisitos mínimos (acelerômetro, giroscópio e GPS). Com este propósito em mente, a mesma foi desenvolvida utilizando HTML5 em conjunto com JavaScript, pois, desta forma, abrange uma gama maior de dispositivos móveis com diferentes sistemas operacionais.

Foi utilizado neste trabalho, semelhante à abordagem usada por (HAUBER et al., 2013), o evento *DeviceMotion* (W3, 2012a) que provê a aceleração do centro de massa do dispositivo, representada em coordenadas Cartesianas (*acceleration* e *accelerationIncludingGravity* nos eixos x, y e z) e expressada em m/s^2 , e, também, a taxa de rotação do mesmo (*rotationRate: alpha, beta, e gamma*), correspondente aos ângulos de Euler (ϕ, θ, ψ) (WEISSTEIN, 2010) expressados em graus/s. Além destas informações, o evento tam-

bém disponibiliza o atributo *interval* que é uma constante expressada em milissegundos contendo o intervalo no qual os dados foram obtidos do dispositivo.

Para conseguir obter um histórico indicando onde o usuário estava quando cada atividade foi reconhecida, utilizou-se a API *Geolocation* (W3, 2012b) que define uma interface de alto nível que provê acesso a informações relacionadas à posição geográfica do referido dispositivo no formato do sistema de coordenadas WGS84 (IMAGERY; AGENCY, 2000). Foram utilizados os atributos latitude e longitude que são coordenadas geográficas especificadas em graus decimais.

2.4 Octave

Para fins do reconhecimento em tempo real da atividade sendo executada pelo indivíduo, foi utilizada a linguagem interpretada *open-source* de alto nível GNU Octave (OCTAVE COMMUNITY, 2014) que é voltada, principalmente, para computações matemáticas, equivalente ao *Matlab*.

Em conjunto com a referida linguagem, utilizou-se a biblioteca LibSVM, 2.2.3, para gerar o modelo e prever qual a atividade executada em tempo real pelo usuário.

A interface entre a linguagem de programação Java, utilizada na construção da aplicação Web, e o Octave, que realizou os cálculos do modelo, deu-se através da biblioteca JavaOctave (Hansen, Kim, 2012).

Alguns códigos em Octave utilizados no sistema proposto estão detalhados no Apêndice A.

3 EXPERIMENTOS E CONSTRUÇÃO DO MODELO

O primeiro passo para gerar um modelo de aprendizagem de máquina confiável que consiga reconhecer as atividades executadas pelos indivíduos é capturar os dados brutos através de experimentos. Estes foram realizados com 10 voluntários, seis do sexo masculino e quatro do sexo feminino, com idade variando entre 25 e 32 anos (média 28.1 anos e desvio padrão 2.13 anos) e altura variando entre 1,54 e 1,90m (média 1,72m e desvio padrão 0,12m), de acordo com a Tabela 3.

Tabela 3 – Informações sobre os indivíduos que participaram dos experimentos.

Indivíduo	Altura	Idade	Sexo
1	1.74	25	M
2	1.54	26	F
3	1.79	32	M
4	1.67	30	F
5	1.70	27	M
6	1.64	27	F
7	1.90	27	M
8	1.72	29	M
9	1.88	28	M
10	1.60	30	F

Fonte: elaborado pelo autor.

Os indivíduos utilizaram um dispositivo móvel iPhone 4 com iOS 7.04 acoplado na cintura com ajuda de um *armband* alongado. Definiu-se que os indivíduos usariam o dispositivo na cintura porque, desta forma, o celular estaria mais próximo do centro de gravidade dos mesmos, além de que vários trabalhos na literatura também seguiram esta definição, como citeanguita-1, por exemplo. Na Figura 4, notam-se os itens necessários à realização dos experimentos, 5(a), e uma voluntária utilizando o sistema com o iPhone na cintura, 5(b).

3.1 Gerando o conjunto de dados

Para gerar o conjunto de dados cada voluntário realizou um circuito contendo as 7 atividades nas seguintes especificações:

1. Atividade: “SENTADO”; Tempo: 60s;
2. Atividade: “EM PÉ”; Tempo: 60s;
3. Atividade: “DEITADO”; Tempo: 60s;

Figura 4 – Em 5(a), tem-se a os itens necessários à realização dos experimentos, um iPhone e uma *armband* alongada. Ao lado, a Figura 5(b) mostra uma voluntária utilizando o sistema com o iPhone na cintura.



(a) iPhone e *armband* alongado



(b) Indivíduo com iPhone na cintura

Fonte: elaborado pelo autor.

4. Atividade: “ANDANDO”; Tempo: 60s;
5. Atividade: “SUBINDO ESCADAS”; Tempo: 15s;
6. Atividade: “DESCENDO ESCADAS”; Tempo: 20s;
7. Atividade: “CORRENDO”; Tempo: 60s;

Desenvolveu-se uma página Web, Figura 5, na qual, para cada item acima, eram selecionados no dispositivo três opções, de acordo com a Figura 6(a): a *atividade* a ser realizada; o *indivíduo* que faria o experimento; e o *tempo*. Depois de clicar no botão “Iniciar”, um intervalo de 3 segundos é disparado para que o indivíduo se prepare para realizar a atividade. Após este tempo, um alerta sonoro é emitido pelo dispositivo informando ao usuário que os dados começaram a ser capturados, em seguida, a tela referenciada na Figura 6(b) aparece disponibilizando um botão de fácil acesso que permita ao usuário parar a captura em qualquer momento, além de informá-lo que a captura está acontecendo (neste exemplo a quinta captura de dados brutos foi realizada com sucesso). Findado o tempo pré-definido, outro aviso sonoro é emitido e a captura é finalizada.

Os dados foram capturados a uma frequência de 20Hz (variável “*Interval* = 0.05”), definiu-se, então, com base no estudo de (ANGUITA et al., 2012) que seria utilizada uma janela de 1,3 segundos equivalente a 26 resultados com 50% de sobreposição, ou seja, cada janela é composta dos 26 resultados da janela anterior mais 26 novos resultados.

A cada iteração completa, além dos vetores contendo o código do indivíduo realizando o experimento, a atividade a ser realizada, e a data e hora no formato *epoch* (quantidade de milissegundos desde 01/01/1970), nove vetores são gerados com um tamanho

Figura 5 – Tela do iPhone demonstrando a página Web desenvolvida para captura de dados brutos para geração do conjunto de dados. Em 6(a), tem-se a tela principal da página onde o usuário deve preencher os dados para poder iniciar a captura. A Figura 6(b), retrata a captura sendo realizada.



Fonte: elaborado pelo autor.

fixo de 52 resultados: $acceleration(x,y,z)$; $accelerationIncludingGravity(x,y,z)$; $rotationRate(alpha, beta, gamma)$.

A partir dos ângulos de Euler capturados, calculou-se a velocidade angular de cada ponto de acordo com a transformação linear descrita na Equação 3.1.

$$\begin{aligned}
 \begin{bmatrix} \text{VelocidadeAngularX} \\ \text{VelocidadeAngularY} \\ \text{VelocidadeAngularZ} \end{bmatrix} &= \begin{bmatrix} \phi \\ 0 \\ 0 \end{bmatrix} + \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} * \begin{bmatrix} \theta \\ 0 \\ 0 \end{bmatrix} \right) + \\
 &\left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} * \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} d * \begin{bmatrix} 0 \\ 0 \\ \psi \end{bmatrix} \right). \tag{3.1}
 \end{aligned}$$

Na Tabela 4 são definidos todos os dados brutos gerados que são utilizados para construir as características.

Para cada um dos 12 vetores gerados a partir de cada iteração completa, cinco funções simples foram utilizadas para gerar as características que (FIGO et al., 2010) conside-

Tabela 4 – Dados brutos gerados.

Nome do Vetor	Funcionalidade
<i>Subject</i>	Indivíduo que realizou o experimento
Atividade	Código da atividade realizada
Data/Hora	Vetor com data e hora de cada resultado
<i>Acceleration</i> (x,y,z)	Aceleração nos três eixos
<i>AccelerationIncludingGravity</i> (x,y,z)	Aceleração incluindo a gravidade nos três eixos
<i>RotationRate</i> (ϕ, θ, ψ)	Ângulos de Euler
<i>VelocidadeAngular</i> (x,y,z)	Velocidade angular calculada a partir dos ângulos de Euler

Fonte: elaborado pelo autor.

raram de baixo custo computacional, pois executam somente operações simples, como adição, subtração, multiplicação, divisão e, ocasionalmente, raiz quadrada ou logaritmo:

- $\min(x)$ - retorna o valor mínimo do vetor x;
- $\max(x)$ - retorna o valor máximo do vetor x;
- $\text{mean}(x)$ - retorna média aritmética dos valores do vetor x;
- $\text{std}(x)$ - retorna o desvio padrão dos valores do vetor x;
- $\text{corr}(x,y)$ - retorna os coeficientes da matriz de correlação entre x e y;

Tem-se, então, um vetor com 60 características a cada iteração. A Tabela 5 mostra o número e a distribuição dos exemplos já transformados em atividades. Em um total de 2121 exemplos por atividade, temos que *Deitado* é a atividade mais frequente, com 20.13% dos resultados, porque é a mais fácil de se executar, foi reduzido o tempo executando as atividades *Subindo Escadas* e *Descendo Escadas*, 4.95% e 5.37% de resultados, respectivamente, devido ao esforço gerado para os usuários em cada execução.

Tabela 5 – O número de exemplos por atividade em todo o conjunto de dados, coletado de 10 usuários.

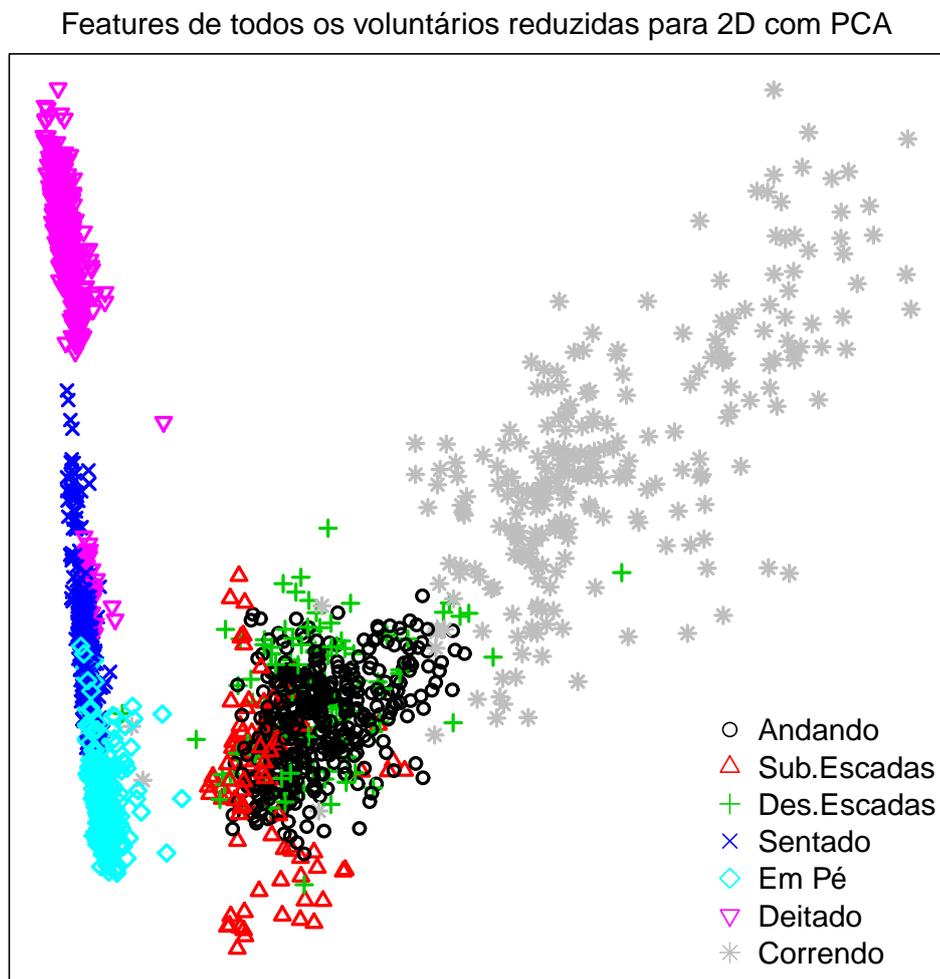
	Andando	S. Escadas	D. Escadas	Sentado	Em Pé	Deitado	Correndo	Total
Total	395	105	114	418	410	427	252	2161
%	18.6	4.95	5.37	19.7	19.3	20.13	11.9	100

Fonte: elaborado pelo autor.

Haja vista que a visualização dos dados com 60 características não é trivial, utilizou-se PCA para reduzir este conjunto para somente duas dimensões. Em seguida, foi gerado um gráfico com os dados em 2D, Figura 6, destacando como cada atividade está disposta em relação às outras. Percebe-se uma separação bem definida entre o grupo de atividades

estáticas (*Sentado*, *Em Pé* e *Deitado*) e o grupo de atividades dinâmicas (*Andando*, *Subindo Escadas*, *Descendo Escadas* e *Correndo*), mas há uma leve sobreposição entre as primeiras, e uma forte entre as segundas. Nota-se que a maior sobreposição ocorre entre as atividades dinâmicas, principalmente as três primeiras (*Andando*, *Subindo Escadas* e *Descendo*). Analisando o gráfico, espera-se que os modelos gerados não tenham maiores problemas para reconhecer as atividades estáticas e a atividade dinâmica *Correndo*, mas provavelmente terá dificuldade para distinguir entre as outras três atividades que estão mais sobrepostas.

Figura 6 – Características de todos os voluntários reduzidas para duas dimensões utilizando PCA.



Fonte: elaborado pelo autor.

A partir desses dados coletados, analogamente ao método utilizado por (WEISS; LOCKHART, 2012), três abordagens foram utilizadas para criação de modelos para reconhecimento de atividades humanas: uma impessoal que utiliza dados brutos de usuários diferentes para treinamento e teste; uma pessoal que utiliza dados de um mesmo usuário, mas de exemplos diferentes, para treinamento e teste; e uma abordagem híbrida que é

um conjunto das outras duas, ou seja, dados de todos os usuários são utilizados para treinamento e teste, mas de exemplos diferentes.

Como foi definido, desde o início deste projeto, que o algoritmo utilizado para treinar e aferir os modelos seria o SVM, todas as abordagens, descritas nas seções a seguir, foram avaliadas usando o mesmo, e uma comparação com outros algoritmos será descrita em 3.5.

3.2 Abordagem Impessoal

Nesta abordagem, o modelo foi gerado utilizando dados de treinamento de um conjunto de usuários que não irão subsequentemente utilizar o modelo, ou seja, o conjunto de treinamento e o de teste não terão usuários em comum. Esta foi escolhida para a geração do modelo que foi utilizado na aplicação prática deste trabalho porque ela é universal, ou seja, treina-se o modelo com um conjunto de usuários e este pode ser utilizado com uma margem de erro aceitável para aferir resultados de novos usuários que não fizeram parte da etapa de criação do mesmo. Desta forma, um público bem mais vasto que o inicial pode se beneficiar do modelo já existente sem a necessidade de novas marcações e aquisição de novos dados para treinamento (WEISS; LOCKHART, 2012).

A partir dos 2121 resultados gerados, foram calculadas, então, as características de acordo com a Seção 3.1. Em seguida, antes de usar estes dados, os mesmos tiveram que ser normalizados. Para isso, utilizou-se o cálculo de normalização da média, Equação 3.2, para cada uma das 60 características.

$$MeanNormalization(feature) = \frac{feature - mean(feature)}{std(feature)} \quad (3.2)$$

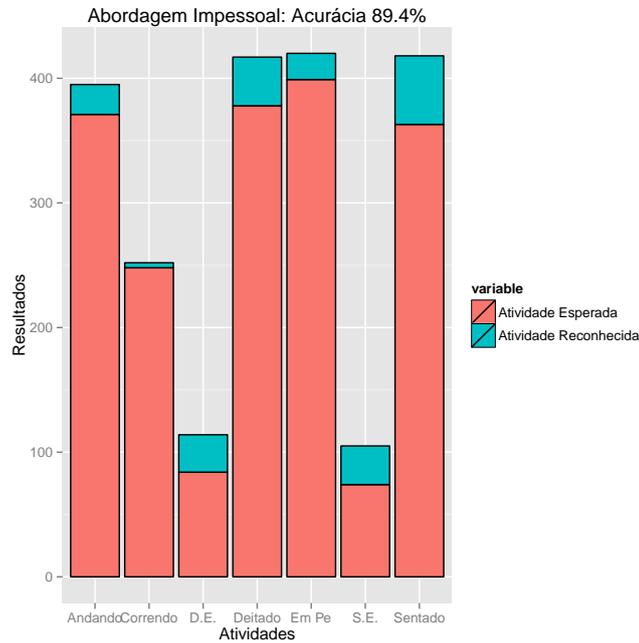
Os parâmetros $mean(feature)$ e $std(feature)$, utilizados para normalizar cada característica, foram guardados em um arquivo para uso na normalização da média dos dados de teste.

Como há 10 indivíduos no experimento, foi treinada uma SVM, utilizando o *Weka* (HALL et al., 2009) e sua função *LibSVM* (CHANG; LIN, 2011) como classificador com parâmetros $C = 1.0$ e $\gamma = 0.025$, que recebeu como entrada os dados de 09 indivíduos, reservando o décimo para testar o modelo. Repetiu-se este procedimento dez vezes, cada vez com um indivíduo diferente utilizado para teste.

Obteve-se uma acurácia média de 89.4% de acordo com a matriz de confusão, apresentada na Tabela 7. As seguintes características, refletidas na Figura 7, podem ser percebidas:

- A atividade *Correndo* foi a mais facilmente reconhecida pelo modelo, na qual o mesmo só errou 4 dos 252 resultados (98.4% de sensibilidade com 88% de precisão), seguida da atividade *Andando*, com 93.9% de sensibilidade, 14 erros em 395 resultados, e 91.8% de precisão;

Figura 7 – Comparação entre as atividades esperadas e as atividades reconhecidas referentes à abordagem impessoal.



Fonte: elaborado pelo autor.

- As outras duas atividades dinâmicas, *Subindo Escadas* e *Descendo Escadas*, tiveram as piores taxas de sensibilidade, 70.4% e 73.7%, com precisão 84.1% e 84.8% respectivamente, o que pode estar associado ao fato de que as mesmas são as que possuem menos resultados de todas as estudadas;
- Finalmente, nota-se que, por mais que o modelo tenha se comportado de maneira satisfatória identificando as atividades estáticas *Em Pé*, *Sentado* e *Deitado*, com 95%, 86.8% e 90.6% de sensibilidade e com precisão 81.6%, 94.5% e 100%, respectivamente, o mesmo ainda confunde uma quantidade significativa de resultados entre estas três atividades.

Analisando-se mais a fundo os dados de cada um dos indivíduos que ficou de fora em suas respectivas rodadas, Tabela 6, percebe-se que os resultados do indivíduo 8, acurácia de somente 48.41%, em discrepância com todos os outros resultados que ficaram acima de 88.87%, fizeram com que a acurácia de todo o modelo diminuísse em quase 4%. Neste caso, pode ter ocorrido algum problema durante a execução do experimento com o respectivo voluntário o que sugere que, caso houvesse disponibilidade, seria de muita valia repetir o circuito com este usuário para melhorar o modelo utilizado.

Os resultados indicam uma boa acurácia total, mas também sugerem que em abordagens futuras deve-se analisar a possibilidade de se adicionarem ao modelo novas características que tenham baixo custo computacional e, assim, melhorar o reconhecimento entre as atividades estáticas. Além disso, coletar mais dados referentes às atividades de

Tabela 6 – Acurácia por indivíduo na abordagem impessoal.

Indivíduo	1	2	3	4	5	6	7	8	9	10
Acerto(%)	94.31	91.94	97.1	95.93	96.46	89.34	94.78	48.31	96.96	88.87

Fonte: elaborado pelo autor.

subir e descer escadas, o que provavelmente deve melhorar a capacidade do modelo em reconhecê-las.

Tabela 7 – Matriz de confusão - Abordagem Impessoal

Real \ Pred.	Andando	S. Escadas	D. Escadas	Sentado	Em Pé	Deitado	Correndo	Total	Sensitividade
Andando	371	11	5	0	0	0	8	395	93.9%
S. Escadas	20	74	9	0	0	0	2	105	70.4%
D. Escadas	12	3	84	0	1	0	14	114	73.7%
Sentado	0	0	0	363	55	0	0	418	86.8%
Em pé	0	0	0	20	399	0	1	420	95.0%
Deitado	0	0	0	1	32	378	6	417	90.6%
Correndo	1	0	1	0	2	0	248	252	98.4%
Total	404	88	99	384	489	378	279	2121	
Precisão	91.8%	84.1%	84.8%	94.5%	81.6%	100%	88.8%		

Fonte: elaborado pelo autor.

Desde o início das pesquisas, era esperado que a abordagem impessoal tivesse o pior resultado entre as três definidas, o que realmente ocorreu, mas com uma acurácia que excedeu as expectativas. Nas próximas seções, descrever-se-ão as outras abordagens utilizadas.

3.3 Abordagem Pessoal

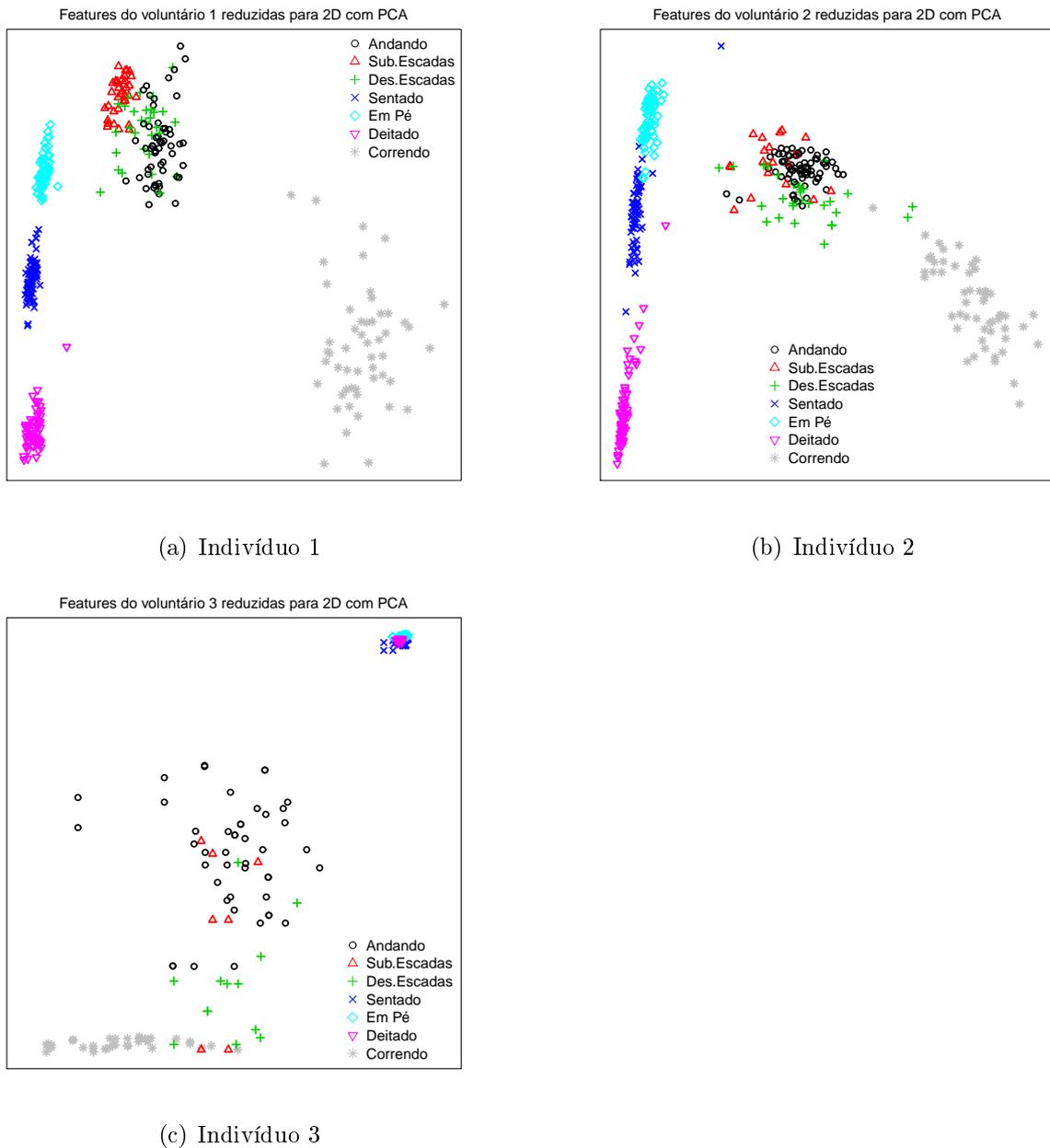
Nesta abordagem, considerou-se treinar a aplicação para uso exclusivo de um mesmo indivíduo para aferir como os resultados se comportariam em comparação com o modelo impessoal. Espera-se que um modelo treinado exclusivamente com dados de um usuário e tendo que classificar mais dados com as mesmas características (mesmo usuário, mas diferentes exemplos) comporte-se de maneira muito melhor do que um impessoal, por exemplo.

Foram selecionados, aleatoriamente, três indivíduos dos dez que participaram dos experimentos. Em seguida, semelhante à abordagem anterior, foram calculadas as características utilizando todos os dados de cada voluntário, foram normalizadas utilizando o cálculo de normalização da média, Equação 3.1, e foi treinada uma SVM com uma validação cruzada de 10 dobras, utilizando novamente o *Weka* e sua função *LibSVM* como classificador com parâmetros $C = 1.0$ e $\gamma = 0.025$.

Analogamente ao que foi realizado com os dados de todos os voluntários, Figura 6, utilizou-se novamente PCA para reduzir a dimensão dos dados de cada um dos três indiví-

duos e conseguir uma visualização melhor destes. Percebe-se uma separação bem definida entre as atividades de cada grupo distinto (atividades estáticas e atividades dinâmicas). Para os indivíduos 1 e 2, Figuras 9(a) e 9(b), respectivamente, não há sobreposição significativa entre os dados do grupo de atividades estáticas, mas esta é visível entre as atividades dinâmicas. Em contrapartida, para o indivíduo 3, Figura 9(c), há uma sobreposição maior entre as atividades estáticas em relação às dinâmicas.

Figura 8 – Características dos três voluntários reduzidas para duas dimensões utilizando PCA.



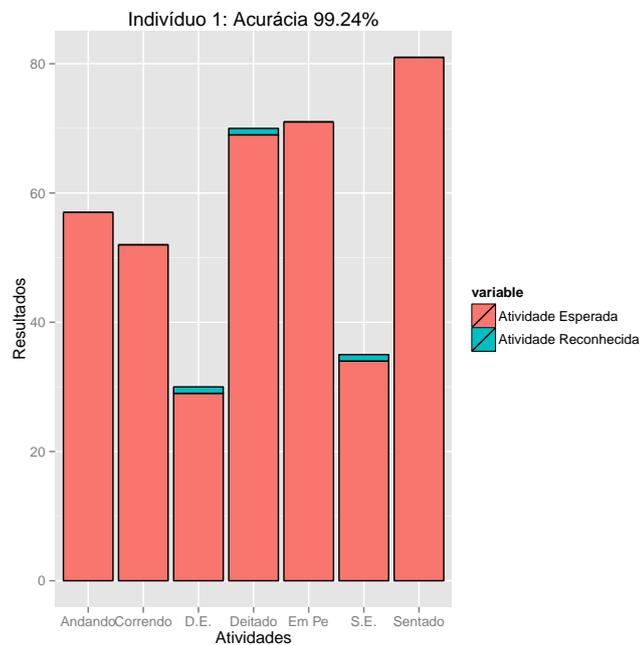
Fonte: elaborado pelo autor.

Os resultados encontrados referentes à cada indivíduo serão descritos detalhadamente a seguir.

3.3.1 Resultados do Indivíduo 1

O indivíduo 1 teve um total de 396 resultados registrados de acordo com a Tabela 8 com predominância das atividades estáticas, 56.1% dos resultados. O modelo gerado para o referido voluntário obteve uma acurácia significativa de 99.24% de acordo com a Figura 9. Percebe-se, pela matriz de confusão detalhada na Tabela 9, que só houve erros do modelo em 3 resultados, um na atividade *Subindo Escadas*, outro na atividade *Descendo Escadas* e, finalmente, um na atividade *Deitado*. Além disso, com exceção das atividades *Andando* e *Correndo*, com 98.2% e 96.3% de precisão, as outras obtiveram 100% neste quesito.

Figura 9 – Comparação entre as atividades esperadas e as atividades reconhecidas referentes ao indivíduo 1.



Fonte: elaborado pelo autor.

Tabela 8 – O número de exemplos por atividade no conjunto de dados coletados do indivíduo 1.

	Andando	S. Escadas	D. Escadas	Sentado	Em Pé	Deitado	Correndo	Total
Total	57	35	30	81	71	70	52	2161
%	14.4	8.8	7.6	20.5	17.9	17.7	13.1	100

Fonte: elaborado pelo autor.

Tabela 9 – Matriz de confusão - Abordagem Pessoal - Indivíduo 1

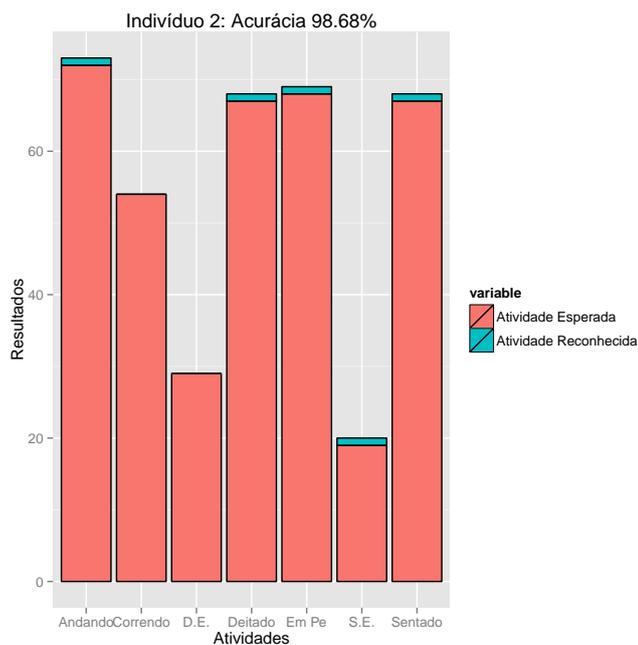
Real \ Pred.	Andando	S. Escadas	D. Escadas	Sentado	Em Pé	Deitado	Correndo	Total	Sensitividade
Andando	57	0	0	0	0	0	0	57	100%
S. Escadas	1	34	0	0	0	0	0	35	97.1%
D. Escadas	0	0	29	0	0	0	1	30	96.7%
Sentado	0	0	0	81	0	0	0	81	100%
Em Pé	0	0	0	0	71	0	0	71	100%
Deitado	0	0	0	0	0	69	1	70	98.6
Correndo	0	0	0	0	0	0	52	52	100%
Total	58	34	29	81	71	69	54	396	
Precisão	98.2%	100%	100%	100%	100%	100%	96.3%		

Fonte: elaborado pelo autor.

3.3.2 Resultados do Indivíduo 2

O indivíduo 2 teve um total de 381 resultados registrados de acordo com a Tabela 10, com predominância da atividade *Andando*, com 19.1% dos resultados. Analogamente ao indivíduo 1, o modelo gerado para o voluntário 2 obteve uma acurácia significativa de 98.68% de acordo com a Figura 10. Nota-se, pela matriz de confusão detalhada na Tabela 11, que no reconhecimento das atividades *Descendo Escadas* e *Correndo*, o modelo obteve 100% de sensibilidade, com 96.7% e 96.4% de precisão, enquanto que nas outras cinco atividades, houve um erro em cada, dos quais dois tiveram a predição da atividade *Correndo*, destaque para as atividades *Andando* e *Em Pé* que tiveram precisão de 100%.

Figura 10 – Comparação entre as atividades esperadas e as atividades reconhecidas referentes ao indivíduo 2.



Fonte: elaborado pelo autor.

Tabela 10 – O número de exemplos por atividade no conjunto de dados coletados do indivíduo 2.

	Andando	S. Escadas	D. Escadas	Sentado	Em Pé	Deitado	Correndo	Total
Total	73	20	29	68	69	68	54	381
%	19.1	5.25	7.6	17.8	18.1	17.8	14.2	100

Fonte: elaborado pelo autor.

Tabela 11 – Matriz de confusão - Abordagem Pessoal - Indivíduo 2

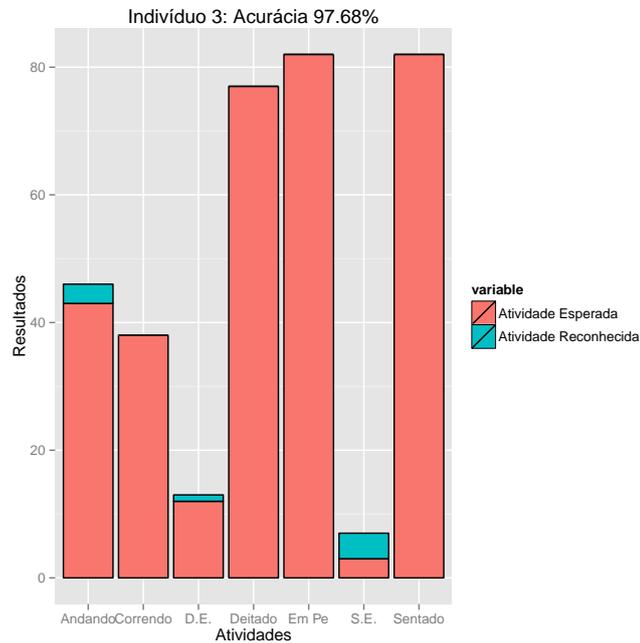
Real \ Pred.	Andando	S. Escadas	D. Escadas	Sentado	Em Pé	Deitado	Correndo	Total	Sensitividade
Andando	72	1	0	0	0	0	0	73	98.6%
S. Escadas	0	19	1	0	0	0	0	20	95%
D. Escadas	0	0	29	0	0	0	0	29	100%
Sentado	0	0	0	67	0	0	1	68	98.5%
Em Pé	0	0	0	1	68	0	0	69	98.5%
Deitado	0	0	0	0	0	67	1	68	98.5%
Correndo	0	0	0	0	0	0	54	54	100%
Total	72	20	30	68	68	67	56	381	
Precisão	100%	95%	96.7%	98.5%	100%	98.5%	96.4%		

Fonte: elaborado pelo autor.

3.3.3 Resultados do Indivíduo 3

Analisando os resultados do indivíduo 3, este obteve um total de 345 resultados registrados de acordo com a Tabela 12, com predominância explícita das atividades estáticas, que juntas têm 69,7% dos resultados. Comparando-se aos modelos gerados para os indivíduos 1 e 2, o novo modelo gerado para o indivíduo 3 também teve uma acurácia significativa, 97.68%, mesmo sendo a pior dos três, de acordo com Figura 11. A partir da matriz de confusão detalhada na Tabela 11, percebe-se que as três atividades estáticas, *Em Pé*, *Sentado* e *Deitado*, e a atividade dinâmica *Correndo* tiveram 100% de acerto, além disso, houve 3 erros no reconhecimento da atividade *Andando*, 4 erros na identificação da atividade *Subindo Escadas*, e 1 erro na predição da atividade *Descendo Escadas*. Além disso, as atividades *Descendo Escadas*, *Sentado*, *Em Pé*, e *Deitado* tiveram 100% de precisão.

Figura 11 – Comparação entre as atividades esperadas e as atividades reconhecidas referentes ao indivíduo 3.



Fonte: elaborado pelo autor.

Tabela 12 – O número de exemplos por atividade no conjunto de dados coletados do indivíduo 3.

	Andando	S. Escadas	D. Escadas	Sentado	Em Pé	Deitado	Correndo	Total
Total	46	7	13	82	82	77	38	345
%	13.3	2	3.7	23.7	23.7	22.3	11	100

Fonte: elaborado pelo autor.

Tabela 13 – Matriz de confusão - Abordagem Pessoal - Indivíduo 3

Real \ Pred.	Andando	S. Escadas	D. Escadas	Sentado	Em Pé	Deitado	Correndo	Total	Sensitividade
Andando	43	2	0	0	0	0	1	46	93.5%
S. Escadas	4	3	0	0	0	0	0	7	42.8%
D. Escadas	0	0	12	0	0	0	1	13	92.3%
Sentado	0	0	0	82	0	0	0	82	100%
Em Pé	0	0	0	0	82	0	0	82	100%
Deitado	0	0	0	0	0	77	0	77	100%
Correndo	0	0	0	0	0	0	38	38	100%
Total	47	5	12	82	82	77	40	345	
Precisão	91.5%	60%	100%	100%	100%	100%	95%		

Fonte: elaborado pelo autor.

3.3.4 Conclusões sobre a abordagem pessoal

Pode-se concluir, pela análise dos dados apresentados nas subseções anteriores, que, mesmo tendo relativamente poucos exemplos, a abordagem pessoal é superior à impessoal, foi obtida uma acurácia média de 98.5%, mais de 9% superior à primeira, o que mostra que quando há a necessidade de uma acurácia muito grande e há tempo e recursos para treinamento, um modelo personalizado para um usuário específico e treinado com suas características é a melhor opção.

3.4 Abordagem Híbrida

A última abordagem utilizada com o conjunto de dados criado é uma junção das duas anteriores, na qual exemplos distintos de todos os indivíduos que participaram do experimento são utilizados tanto para treinar o modelo quanto para testá-lo. Esperava-se obter resultados melhores que a abordagem impessoal, já que configurações de dados distintas foram utilizadas para aferir o modelo, mas um pouco piores do que os resultados apresentados pela abordagem pessoal, pois, nesta última, o modelo é treinado e testado com dados de características muito semelhantes.

Analogamente à abordagem impessoal, todos os dados brutos dos usuários foram consolidados em arquivos únicos, foram calculadas as características que serviram de entrada para uma SVM com validação cruzada com 10 dobras, que foi treinada utilizando o *Weka* e sua função *LibSVM* como classificador com parâmetros $C = 1.0$ e $\gamma = 0.025$. O modelo gerado obteve uma acurácia bastante significativa de 98.16%, de acordo com Figura 12.

Tabela 14 – Matriz de confusão - Abordagem Híbrida

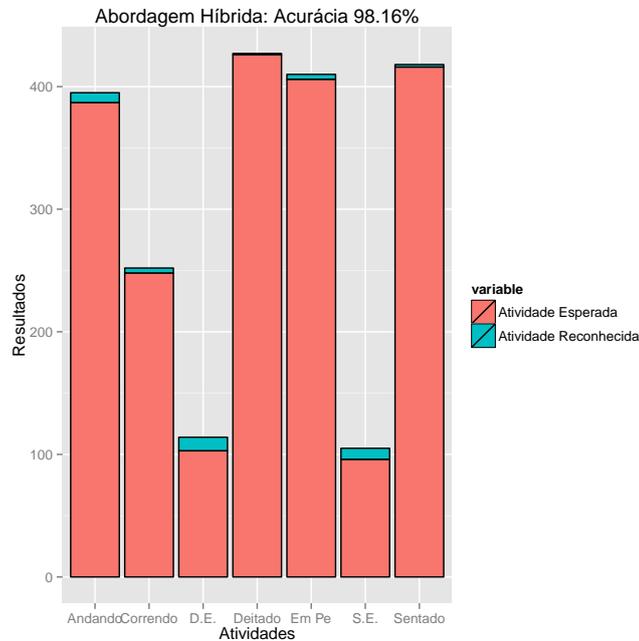
Real \ Pred.	Andando	S. Escadas	D. Escadas	Sentado	Em Pé	Deitado	Correndo	Total	Sensitividade
Andando	387	6	1	0	0	0	1	395	98%
S. Escadas	7	96	1	0	0	0	1	105	91.4%
D. Escadas	2	0	103	1	0	0	8	114	90.4%
Sentado	0	0	0	416	2	0	0	418	99.5%
Em Pé	0	0	0	4	406	0	0	410	99%
Deitado	0	0	0	0	0	426	1	427	99.8%
Correndo	2	0	0	0	2	0	248	252	98.4%
Total	398	102	105	420	411	426	259	2121	
Precisão	97.2%	94.1%	98.1%	99.1%	98.8%	100%	95.8%		

Fonte: elaborado pelo autor.

Observa-se na matriz de confusão na Tabela 14, que as maiores taxas de acerto do modelo ocorreram para as atividades estáticas, *Deitado*, *Sentado* e *Em Pé*, com, 99.76%, 99.52% e 99% de resultados acertados, e 100%, 99.1% e 98.8% de precisão, respectivamente. Nas atividades dinâmicas, a maior taxa foi de 98.41% de sensibilidade e 95.8% de precisão da atividade *Correndo*, seguida da atividade *Andando* com 97.97% de sensibilidade e 97.2% de precisão, *Subindo Escadas* com 91.42% de sensibilidade e 94.1%

de precisão e, finalmente, *Descendo Escadas* com 90.35% de sensibilidade e 98.1% de precisão.

Figura 12 – Comparação entre as atividades esperadas e as atividades reconhecidas referentes à abordagem híbrida.



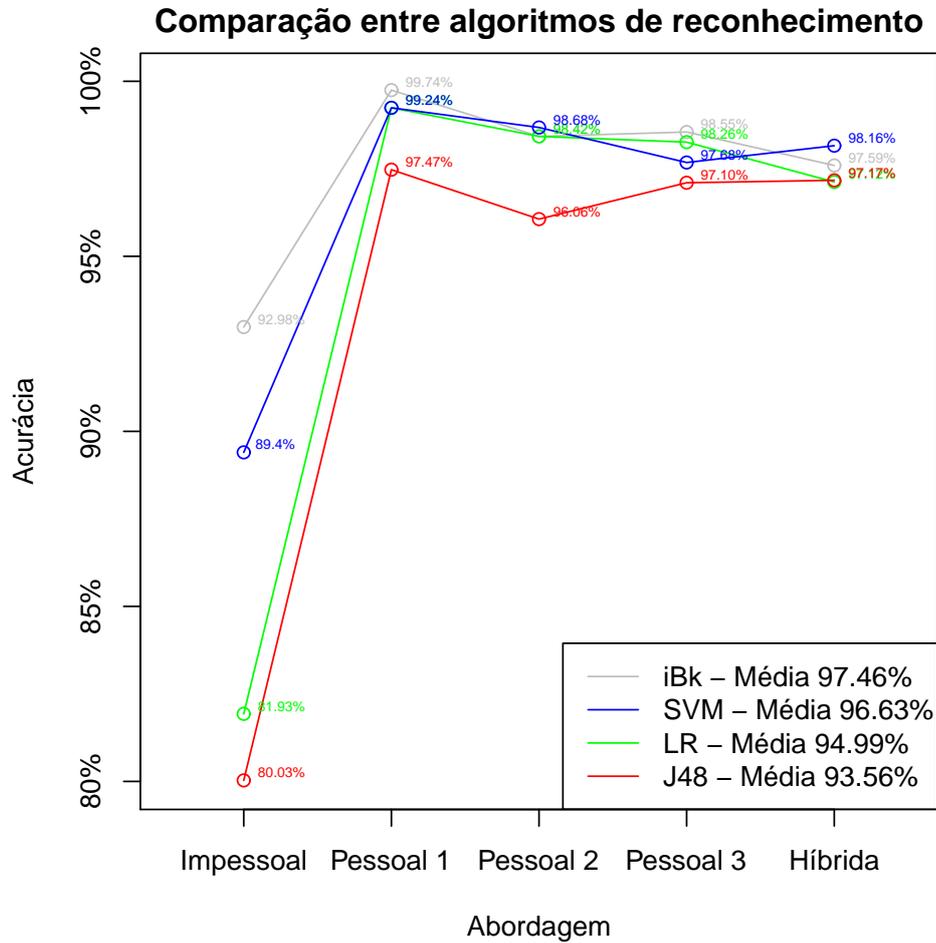
Fonte: elaborado pelo autor.

3.5 Comparação entre SVM e outros algoritmos

Mesmo com a definição inicial do projeto em utilizar SVM como o algoritmo de reconhecimento de atividades, considerou-se válido aferir como modelos com outros algoritmos se comportariam com os dados gerados.

Na Figura 13 tem-se uma comparação entre quatro algoritmos: *IBk*, *SVM*, *J48*, e *Logistic*. Percebe-se que o algoritmo *IBk*, implementação do Weka para o algoritmo k-Vizinhos mais próximos, obteve a melhor acurácia média, 97.46%, seguido do próprio SVM, com 96.63%, em seguida aparece a implementação do Weka do algoritmo de Regressão Logística com 94.99%, *Logistic*, e, com a menor acurácia média entre os quatro, o algoritmo *J48* que é a implementação do Weka para a árvore de decisão (C4.5).

Figura 13 – Comparação entre quatro algoritmos de reconhecimento (*IBk*, *SVM*, *J48*, e *Logistic*) utilizando os dados coletados.



Fonte: elaborado pelo autor.

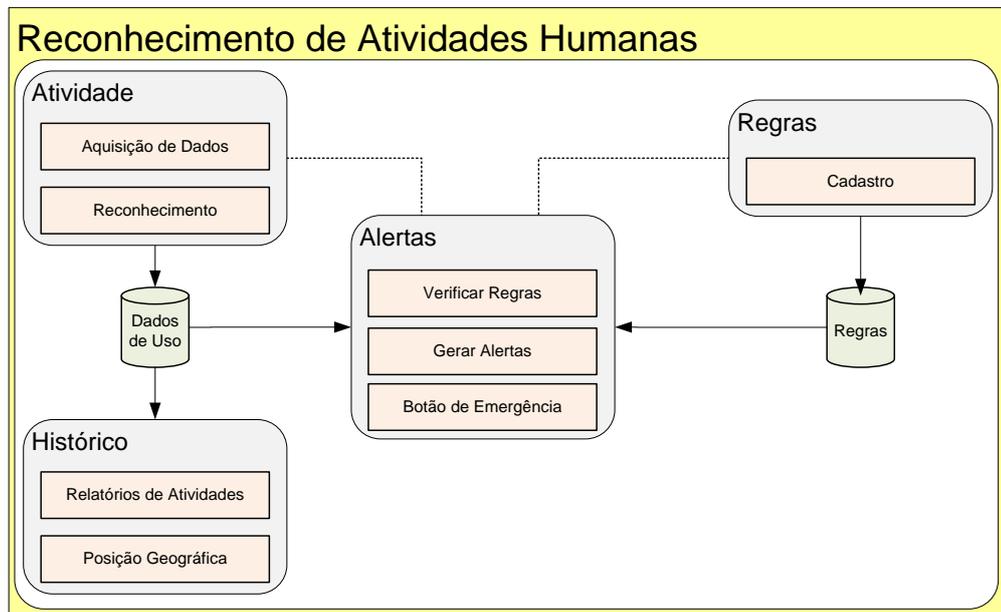
Nota-se que os quatro algoritmos utilizados obtiveram uma acurácia média bastante significativa, com destaque para os algoritmos *IBk* e o *SVM*. Independentemente da abordagem ou algoritmo escolhidos, mostrou-se que o conjunto de dados gerado é propício para o reconhecimento de atividades humanas a partir de sensores encontrados em *smartphones*. No próximo capítulo, descrever-se-á como, a partir de um modelo gerado, pode-se desenvolver uma aplicação para identificação e monitoramento das referidas atividades.

4 IMPLEMENTAÇÃO DA PROVA DE CONCEITO

Depois de criado um modelo confiável para identificação das atividades humanas, via a abordagem impessoal descrita em 3.2, foi necessário disponibilizar uma infraestrutura de uso prático para o mesmo, de forma que qualquer pessoa possa utilizá-lo. Para isso, implementou-se uma prova de conceito através de uma página Web simples utilizando várias tecnologias descritas no capítulo 2, entre essas, HTML5/Javascript e Octave.

A arquitetura conceitual do sistema está descrita na Figura 14. Quatro módulos distintos podem ser identificados: o módulo principal *Atividade*, responsável pela aquisição dos dados brutos e reconhecimento da atividade sendo executada; o módulo *Regras*, responsável pelo cadastro e manutenção das regras; o módulo *Alertas* que faz a interface entre os outros dois, verificando regras, gerando alertas e atuando no caso de uma emergência quando o respectivo botão for acionado; e, finalmente, o módulo *Histórico*, responsável pelos relatórios de atividades e pelos mapas de posição geográfica do usuário.

Figura 14 – Arquitetura conceitual do sistema proposto.



Fonte: elaborado pelo autor.

Na prática, três páginas Web foram criadas e serão descritas nas seções seguintes: uma página principal de uso do sistema; uma página para criação de regras customizáveis; e uma página de acompanhamento para que o usuário ou seu responsável acompanhe em tempo real as atividades que estão sendo realizadas.

4.1 Página principal de uso

Para iniciar o uso do sistema, o indivíduo deve selecionar com qual frequência (tempo em segundos) as atividades devem ser reconhecidas, e iniciar a captura, Figura 15.

Figura 15 – Tela do iPhone demonstrando a página Web principal de uso do sistema onde o usuário deve selecionar com qual frequência as atividades devem ser reconhecidas

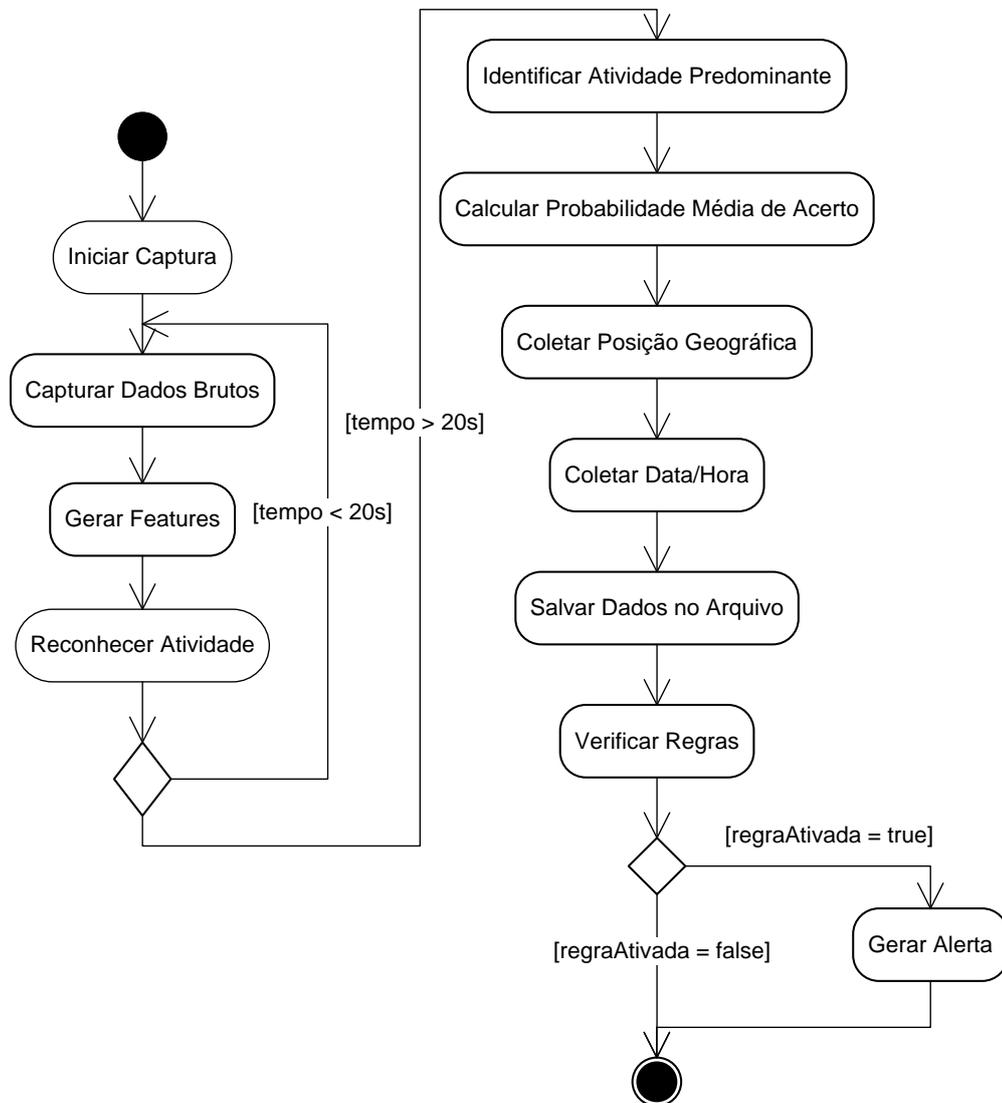


Fonte: elaborado pelo autor.

Depois de iniciada a captura, é seguido um fluxo de acordo com o diagrama representado na Figura 16. A cada Δt segundos, o sistema passará 20 segundos capturando os dados brutos, convertendo-os em características e identificando a atividade que está sendo realizada. Findado este tempo, o servidor identifica qual atividade reconhecida predominou, quantas vezes esta atividade aconteceu e com qual probabilidade de acerto foi identificada em cada iteração, calcula, então, a probabilidade média de acerto, captura a posição geográfica atual do indivíduo, a data e a hora no formato *epoch*, e grava esses dados em um arquivo de texto que será utilizado tanto para a verificação das regras quanto para gerar o histórico de uso do sistema. Após a gravação no arquivo, o sistema invoca um método para verificar se alguma regra previamente cadastrada foi ativada e, caso positivo, gera e envia o alerta referente à mesma, conforme a Seção 4.2.

Segue na Listagem 4.1 um trecho do arquivo de texto gerado para um indivíduo utilizando o sistema proposto. Percebe-se que nas duas primeiras linhas a atividade 4 (*Sentado*) foi identificada com mais de 99% de certeza média, seguida da atividade

Figura 16 – Diagrama de atividade da função principal de uso do sistema.



Fonte: elaborado pelo autor.

1 (*Andando*), nas duas linhas seguintes, com 94% de certeza média, e, finalmente, da atividade 5 (*Em Pé*), com 84% de certeza. Além disso, foram coletados dados da posição geográfica do usuário que escolheu que novos dados fossem coletados a cada 30 segundos.

Listing 4.1 – Trecho do arquivo de uso gerado pelo sistema proposto.

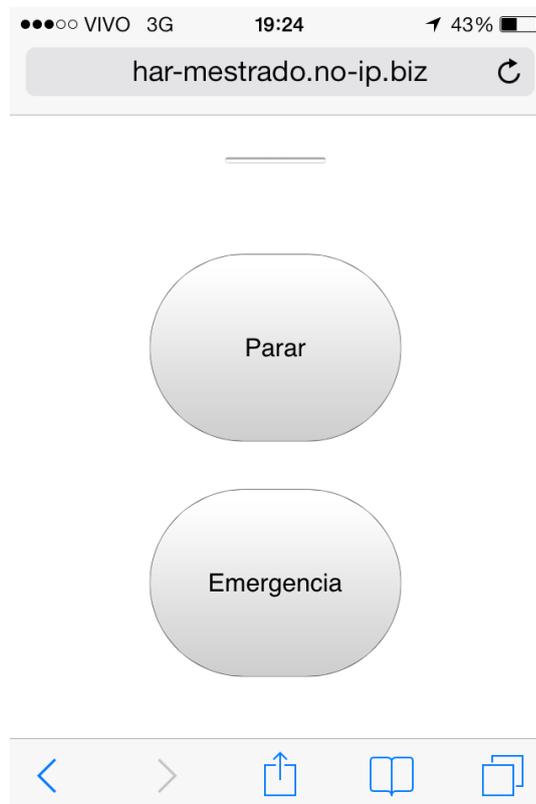
```

Indivíduo, Atividade Reconhecida, Probabilidade, Data/Hora, Latitude,
  Longitude, Frequência
...
1,4,0.991201162354,1391421200647,-22.91095921423772,-43.218732286662686,30
1,4,0.99834440474,1391421233628,-22.910958619238386,-43.218732717707496,30
1,1,0.939444555109,1391421291201,-22.910958371284316,-43.21873289733645,30
1,1,0.946503588795,1391421336034,-22.91084452761327,-43.22165099141855,30
1,5,0.845154606932,1391421426779,-22.91192704924099,-43.22152747821062,30
...

```

Além disso, implementou-se, como definido nos objetivos do trabalho, um botão de emergência para ser usado em caso de necessidade pelo usuário do sistema e serve como um contato direto entre o indivíduo que o está utilizando e um responsável previamente cadastrado pelo mesmo. Depois de iniciada a captura dos dados, dois botões ficam visíveis ao usuário, Figura 17, o primeiro, “Parar”, que quando ativado para imediatamente o processo de captura de dados, e o segundo, “Emergência”, que, quando clicado duas vezes em menos de 2 segundos, envia um aviso ao responsável cadastrado.

Figura 17 – Tela do iPhone mostrando os botões “Parar” e “Emergência”.



Fonte: elaborado pelo autor.

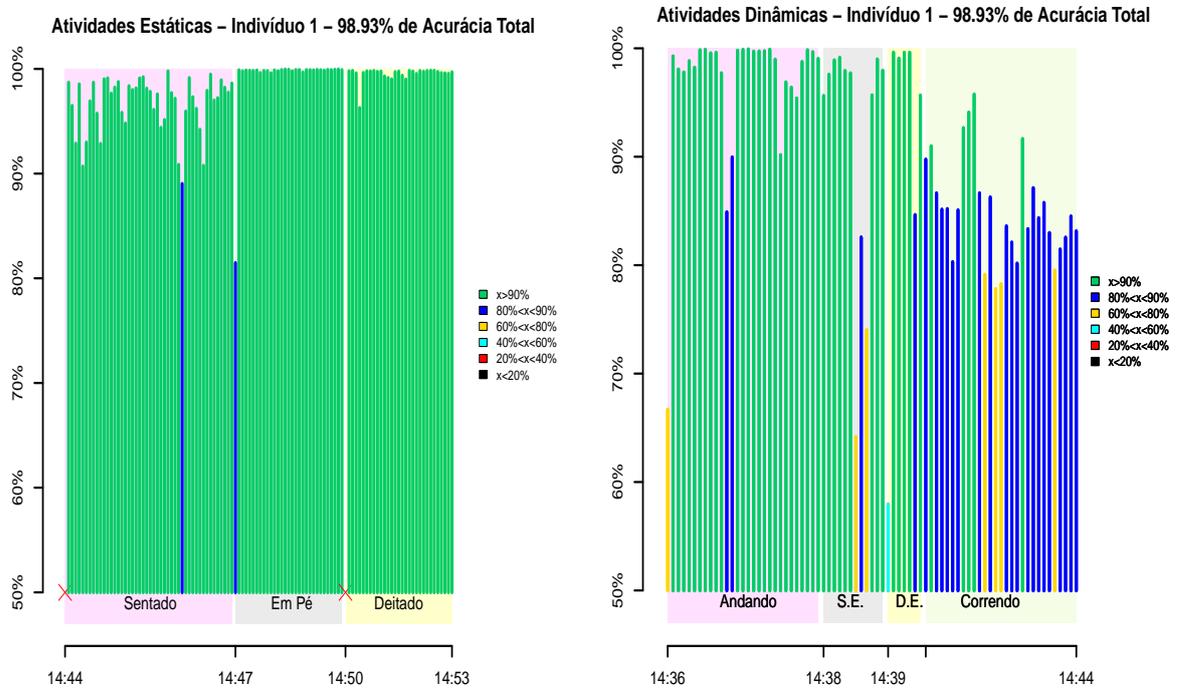
Para aferir esta funcionalidade principal como um todo, recriou-se o circuito detalhado na Seção 3.1 com dois indivíduos já utilizando o modelo gerado, e os resultados serão

descritos nas subseções seguintes.

4.1.1 Uso do sistema pelo indivíduo 1

O indivíduo 1 gerou um total de 187 resultados que, depois de utilizados como entrada para o modelo gerado, foram reconhecidos com uma acurácia de 98.93% de acordo com a Figura 18. Esta compara as atividades reais executadas pelo indivíduo com as atividades reconhecidas pelo modelo, a mesma foi dividida em atividades estáticas, 19(a), e atividades dinâmicas, 19(b).

Figura 18 – Gráfico comparando as atividades reais e a atividades reconhecidas pelo modelo para o indivíduo 1. O mesmo foi subdividido em atividades estáticas, 19(a), e atividades dinâmicas, 19(b).



(a) Atividades Estáticas

(b) Atividades Dinâmicas

Fonte: elaborado pelo autor.

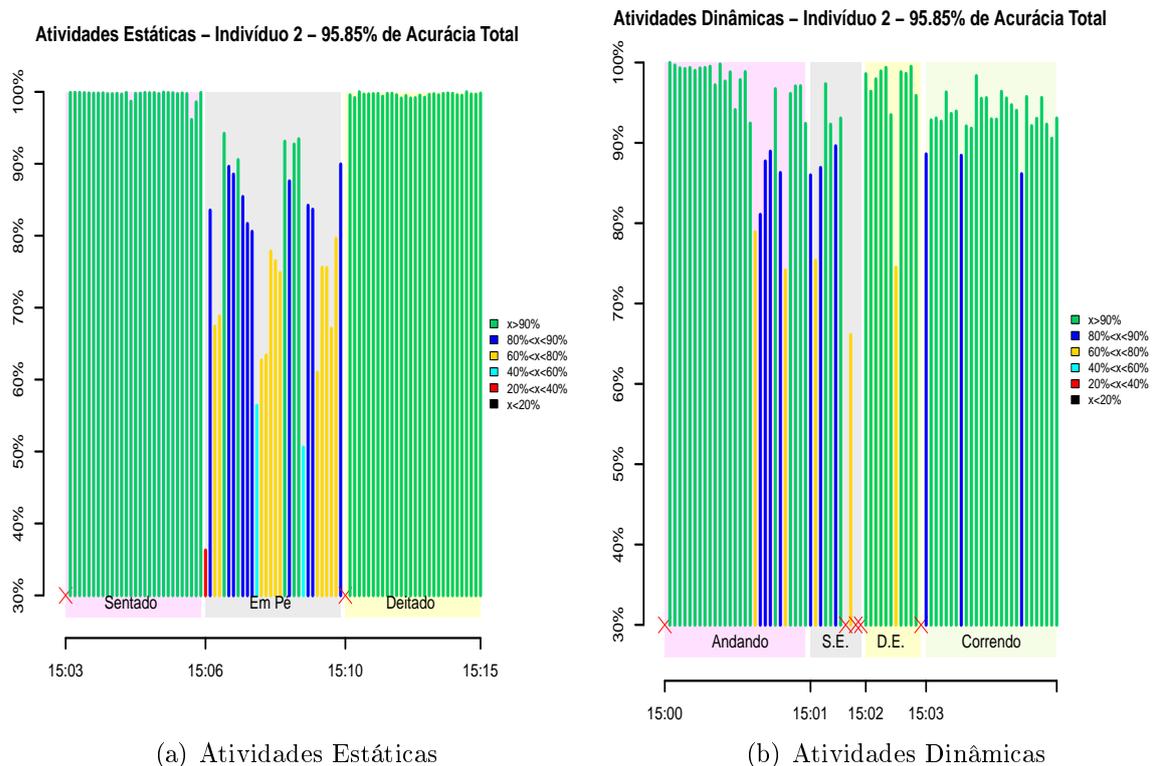
Percebe-se que houve somente dois erros na identificação das atividades, um na atividade *Andando* e outro na atividade *Sentado*. Além disso, a mesma Figura 18 mostra a segurança na qual o modelo acertou as atividades, nota-se, facilmente, a predominância de acertos com certeza acima de 90% (cor verde), seguidos de vários com certeza entre 80 e 90% (cor azul). Somente na atividade *Deitado* o modelo reconheceu uma atividade com menos de 60% de certeza.

Conclui-se, então, que o modelo se comportou muito bem na identificação das atividades executadas pelo indivíduo 1.

4.1.2 Uso da sistema pelo indivíduo 2

O indivíduo 2 gerou um total de 169 resultados que foram reconhecidos pelo modelo com uma acurácia de 95.85% de acordo com a comparação entre as atividades realizadas e as reconhecidas pelo modelo, explicitada na Figura 19. Analogamente ao indivíduo 1, a mesma foi dividida em atividades estáticas, 20(a), e atividades dinâmicas, 20(b).

Figura 19 – Comparação entre as atividades executadas e as atividades identificadas pelo modelo para o indivíduo 2. Novamente, o gráfico foi subdividido em atividades estáticas, 20(a), e atividades dinâmicas, 20(b).



Fonte: elaborado pelo autor.

Comparando-se as atividades reais com as reconhecidas pelo modelo, percebe-se que 07 erros foram encontrados sendo, 05 nas atividades estáticas 20(a) na qual somente a atividade *Deitado* teve três erros, e 02 nas atividades dinâmicas, um na atividade *Andando* e outro na atividade *Correndo*.

Diferentemente do que foi previamente visualizado para o indivíduo 1, várias atividades reconhecidas para o indivíduo 2 tiveram uma taxa de certeza abaixo de 80% e acima de 60% (cor amarela), principalmente nas atividades *Subindo Escadas* e *Descendo Escadas* que ainda tiveram dois resultados com taxa de certeza entre 60 e 40% (cor laranja) e até um resultado com certeza abaixo de 40% (cor vermelha). Identifica-se, também, que algumas atividades foram reconhecidas sem maiores problemas, com uma grande predominância de taxa de certeza acima de 90%, foram os casos das atividades *Andando*, *Sentado* e *Correndo*.

Conclui-se que, mesmo com algumas atividades não se comportando com uma taxa de certeza tão alta, o modelo aferiu de maneira satisfatória todas as atividades, conseguindo uma acurácia média de 97.39%.

4.2 Criação de regras

Um dos objetivos do sistema é gerar alertas para um responsável indicado pelo usuário (que pode ser o próprio) de acordo com regras customizáveis. Criou-se, então, uma página Web para o cadastro de regras, vide a Figura (20), que possui os seguintes campos:

Figura 20 – Tela do iPhone demonstrando a página Web desenvolvida para cadastro de regras customizáveis que serão verificadas a cada nova atividade reconhecida. Em 21(a), tem-se a tela de cadastro referente ao tipo de regra 1 - *Tempo máximo em uma atividade*, e em 21(b), demonstra-se a tela relacionada ao tipo de regra 2 - *Transição entre atividades*.

Cadastrar Regras

Tipo: 1-Tempo Máx em uma atividade

Subject: Indivíduo 1

Atividade Inicial: 4-Sentado

GPS: Carragar Posição Atual
Latitude: -22.9083401
Longitude: -43.2218680

Tempo máximo na atividade: 3600

Cadastrar

(a) Regra de tipo 1

Cadastrar Regras

Tipo: 2-Transição entre atividades

Subject: Indivíduo 3

Atividade Inicial: 1-Andando

Atividade Final: 6-Deitado

GPS: Carragar Posição Atual
Latitude: -22.9117093
Longitude: -43.2217482

Cadastrar

(b) Regra de tipo 2

Fonte: elaborado pelo autor.

- **Tipo:** neste campo o usuário seleciona qual o tipo de regra a ser cadastrada entre 1 - *Tempo máximo em uma atividade* e 2 - *Transição entre atividades*;
- **Atividade Inicial:** usuário deve selecionar a atividade inicial que a referida regra faz referência, caso o tipo 1 seja selecionado, somente esta atividade estará disponível;

- **Atividade Final:** usuário deve selecionar a atividade final que a regra esta relacionada, este campo só está disponível quando o tipo 2 foi selecionado;
- **Latitude:** campo que o usuário deve preencher com a latitude do GPS a ser utilizada na regra;
- **Longitude:** campo que o usuário deve preencher com a longitude do GPS a ser utilizada na regra;
- **Tempo máximo:** disponível somente para o tipo 1, o usuário deve selecionar o tempo máximo que a atividade será executada até que o alerta seja gerado;

Há dois tipos de regras possíveis. A regra de tipo 1 - *Tempo máximo em uma atividade* refere-se ao alerta em que o usuário deseja que seja enviado caso uma atividade seja executada por um tempo maior do que o máximo definido, como por exemplo: gerar um alerta caso o usuário passe mais de uma hora sentado, Figura 21(a); ou alertar caso o usuário fique deitado por mais de 10 horas; entre outros. A regra de tipo (2 - *Transição entre atividades*), por outro lado, refere-se ao alerta gerado quando acontece uma transição entre duas atividades selecionadas (*Atividade Inicial* e *Atividade Final*), como por exemplo: gerar uma alerta quando o usuário que estava andando ou correndo, e passou abruptamente para o estado deitado, o mesmo pode ter sofrido uma queda, Figura 21(b); ou, de maneira análoga, quando o indivíduo estava subindo ou descendo escadas e passou para o estado sentado; entre outros.

Caso o usuário deseje que a regra leve em conta a posição geográfica do mesmo, este deve preencher os campos de GPS no cadastro de regras. Há um botão (*Carregar Posição Atual*) que completa os campos *Latitude* e *Longitude* com os valores da posição atual do usuário. Um uso possível da posição geográfica seria criar uma regra do tipo 1 indicando que caso o usuário esteja no estado *Deitado* em uma posição geográfica diferente da sua cama, por exemplo, gere um alerta para o responsável, pois pode indicar que o indivíduo está em uma situação de emergência.

Na Listagem 4.2, foi disponibilizado um exemplo de um arquivo de texto contendo regras geradas pelo sistema para um indivíduo. Na primeira linha, foi criada uma regra do tipo 1 - *Tempo máximo em uma atividade* que será ativada quando o usuário executar a atividade 4 (*Sentado*) por mais de 1h (3600s). Na linha seguinte, um alerta será gerado quando o indivíduo passar mais de 8h (28800s) na atividade (*Deitado*). As próximas duas linhas contêm regras do tipo 2 - *Transição entre atividades*. A terceira regra será ativada quando o indivíduo realizar a transição entre as atividades 1 (*Andando*) e 6 (*Deitado*) em uma posição geográfica distinta da cadastrada. Finalmente, na quarta regra deseja que um alerta seja gerado quando, independentemente da posição do usuário, este realizar uma transição entre as atividades 3 (*Descendo Escadas*) e 6 (*Deitado*), pois o mesmo pode ter sofrido uma queda, por exemplo.

Listing 4.2 – Trecho do arquivo de regras gerado pelo sistema proposto.

```

Indivíduo, Tipo de Regra, Atividade Inicial, Atividade Final, Latitude,
  Longitude, Tempo Máximo
...
4,1,4,0,0,0,3600
4,1,6,0,0,0,28800
4,2,1,6,-22.91095921423772,-43.21873228666268,0
4,2,3,6,-22.91095837128431,-43.21873289733645,0
...

```

A cada nova atividade reconhecida durante o uso do sistema, todas as regras cadastradas são verificadas e, caso as condições de alguma regra sejam atendidas, o alerta respectivo será gerado para o responsável indicado que, inclusive, pode ser o próprio indivíduo no caso de uso do sistema para aferir o *status* de suas atividades durante o dia.

4.3 Acompanhamento das atividades realizadas

Para que o usuário ou seu responsável tenha um acompanhamento em tempo real das atividades realizadas, criou-se uma página Web que lê, a cada 1 minuto, os dados de atividade reconhecidos pelo modelo para um indivíduo escolhido e mostra um histórico das atividades executadas pelo mesmo.

Para testar esta funcionalidade, e o sistema como um todo, selecionou-se um indivíduo que utilizou um iPhone acoplado na cintura por um período de 2 horas e 45 minutos com a frequência configurada para que novos dados fossem obtidos a cada 30 segundos. O indivíduo foi instruído a seguir com suas atividades diárias normais. No total, 330 atividades foram reconhecidas de acordo com a Tabela 15.

Tabela 15 – Atividades reconhecidas pelo modelo em um uso prático durante 2 horas e 45 minutos.

Atividade	Quantidade	Tempo	%
Andando	43	21,5 min	13
S. Escadas	1	0,5 min	0.3
D. Escadas	3	1,5 min	0.9
Sentado	237	1h 58,5 min	71.8
Em Pé	33	16,5 min	10
Deitado	2	1 min	0.6
Correndo	11	5,5 min	3.3
Total	330	2h 45 min	100

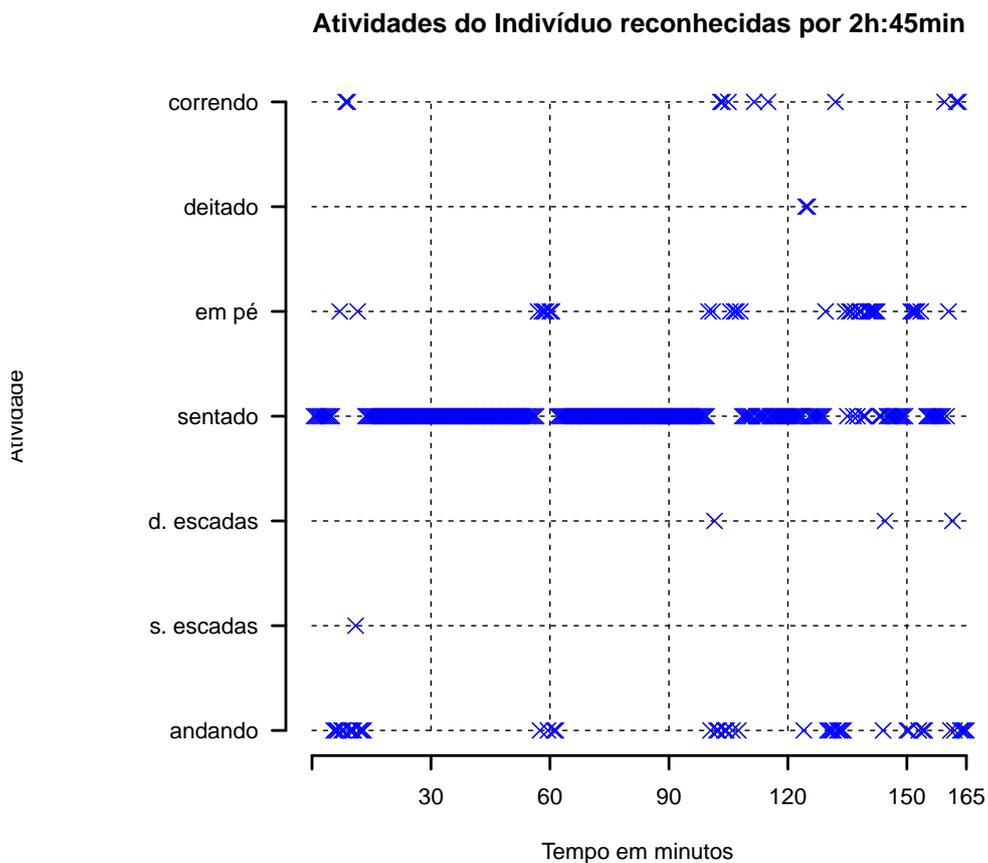
Fonte: elaborado pelo autor.

Analisando a Figura 21 é possível identificar que a atividade predominante, durante as 2 horas e 45 minutos de uso do sistema pelo indivíduo escolhido, foi a *Sentado* com quase 2 horas, ou 72% do tempo total de captura dos dados, seguida da atividade *Andando*, com 21 minutos e 30 segundos de execução, ou 13%. Com 16 minutos e 30 segundos, ou

10% do tempo total, aparece a atividade *Em Pé*. As outras 4 atividades somam juntas somente 8 minutos e 30 segundos, ou 5.1% do tempo total de captura.

Percebeu-se que, devido ao uso extensivo da funcionalidade de posição geográfica do indivíduo, houve uma degradação significativa do nível da bateria, o qual diminuiu 50% durante o tempo de uso da solução.

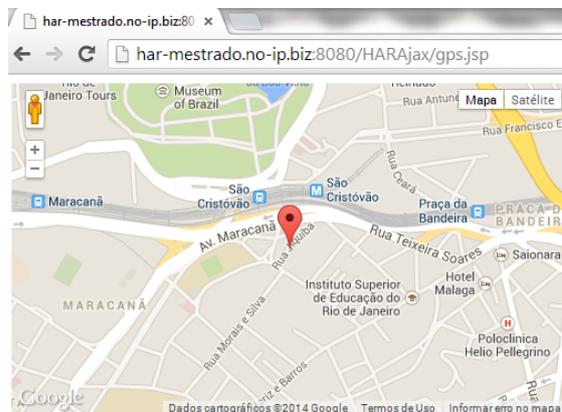
Figura 21 – Gráfico ilustrando as atividades reconhecidas num exemplo prático de uso do sistema proposto neste trabalho durante 2 horas e 15 minutos.



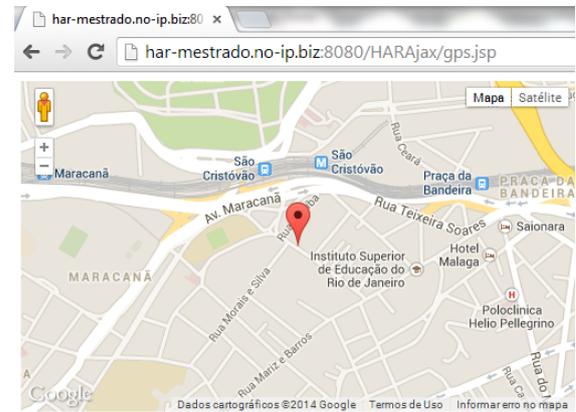
Fonte: elaborado pelo autor.

Finalmente, além do histórico das atividades executadas, deseja-se que informações da posição geográfica do indivíduo sejam mostrados. Na Figura 22 há quatro exemplos da posição do indivíduo quando executou quatro atividades distintas. A captura de dados inicia-se com o indivíduo sentado (23(a)), depois de um período curto, uma nova atividade é reconhecida, *Andando*, e a posição geográfica do mesmo muda (23(b)). No terceiro exemplo, reconheceu-se a atividade do usuário como *Em Pé* (23(c)) e, finalmente, no último exemplo o indivíduo estava realizando a atividades *Subir Escadas* (23(d)).

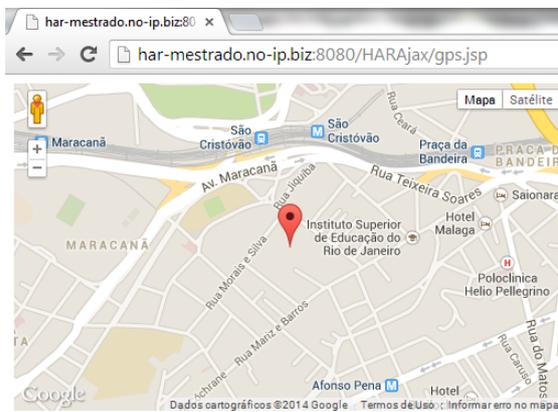
Figura 22 – Quatro exemplos mostrando a posição geográfica do indivíduo no momento em que atividade foi reconhecida.



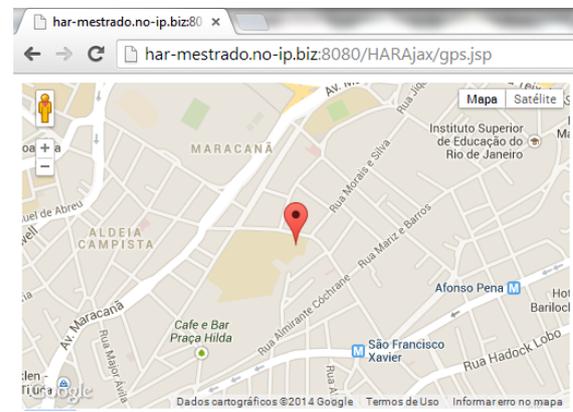
(a) Sentado



(b) Andando



(c) Em Pé



(d) Subindo Escadas

Fonte: elaborado pelo autor.

5 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, foi desenvolvida uma prova de conceito através de uma aplicação Web para reconhecer as atividades humanas executadas por um indivíduo portando um *smartphone* simples em sua cintura, que, além da função principal de reconhecimento, disponibiliza uma página para configuração de regras customizáveis que são verificadas a cada novo resultado identificado. Foi disponibilizado, também, um histórico das atividades reconhecidas durante um período de tempo pré-definido com informações sobre a posição geográfica do dispositivo, caso desejado pelo usuário. Desenvolveu-se, ainda, um botão de emergência que coloca o usuário em contato direto com o responsável em situações de necessidade.

Com o intuito de criar o referido modelo, foi gerado um novo conjunto de dados a partir de experimentos realizados com 10 voluntários que, utilizando um *smartphone* acoplado na cintura, foram instruídos a realizar um circuito de atividades pré-definidas, nas quais os dados dos sensores inerciais do dispositivo móvel (acelerômetro e giroscópio) foram coletados e armazenados separadamente para cada indivíduo.

Em seguida, os dados brutos colhidos foram normalizados para posterior treinamento de um modelo, levando em conta três abordagens: uma impessoal, que considera os dados de nove indivíduos para treinamento e de um indivíduo para teste, alternando-se os papéis até completar o ciclo, e foi obtida uma acurácia média de 89.4%; uma pessoal, que leva em conta as características únicas de um indivíduo através do uso, tanto para treinamento quanto para teste, de dados exclusivos de um mesmo voluntário, atingiu-se uma acurácia média de 98.5%; e uma híbrida, que utilizou resultados distintos de todos os voluntários para treinar e, também, para testar o modelo, foi alcançada uma acurácia de 98.16%.

Com o modelo devidamente treinado com a abordagem escolhida (impessoal), dois voluntários foram convidados a refazer o circuito de atividades para que novos dados fossem gerados e para que o algoritmo fosse, de fato, testado. Para o primeiro indivíduo, foi obtida uma acurácia de 98.93% e uma taxa de certeza média acima de 94%, ou seja, notou-se que, além do modelo conseguir reconhecer corretamente a atividade que estava sendo executada, o mesmo o fez com uma confiança muito alta. Os resultados para o segundo indivíduo, com uma acurácia de 95.85%, foram tão satisfatórios quanto, mas a confiança do modelo para este voluntário, 91.2%, foi um pouco menor do que para o anterior.

A partir do modelo fechado, foi implementada a aplicação Web proposta utilizando algumas tecnologias consolidadas, como *HTML5/JavaScript*, *Java* e *Octave*, por exemplo.

Finalmente, selecionou-se um voluntário para utilizar o sistema completo durante o seu dia-a-dia por um período de quase três horas (2h45min). Notou-se que a aplicação se comportou muito bem, executando os objetivos para os quais a mesma foi proposta, e que

esta tem várias características importantes no contexto de reconhecimento de atividades humanas a partir de sensores inerciais embutidos em um *smartphone* simples, como por exemplo:

- Portabilidade: este aspecto foi fundamental para escolha do desenvolvimento da aplicação em *HTML5*, pois, desta forma, qualquer *smartphone* com os requisitos mínimos (acelerômetro, giroscópio e GPS) pode utilizar o sistema independentemente de sistema operacional (iOS, Android, etc.);
- Independência de ambiente: a aplicação foi disponibilizada na Web para que o reconhecimento das atividades não seja restrito a um ambiente específico, ou seja, através de uma rede de dados em qualquer lugar do mundo o indivíduo pode utilizar o sistema livremente;
- Baixo custo computacional: tendo em mente que a aplicação foi desenvolvida para ser executada em um servidor Web simples e acessada por um dispositivo móvel com recursos computacionais limitados, tentou-se reduzir o consumo excessivo destes recursos.
- Método não intrusivo: diferentemente de vários métodos propostos na literatura, o presente sistema não exige que vários sensores acoplados ao corpo do indivíduo, como em (ROGGEN et al., 2010; MAURER et al., 2006), mas somente o uso do *smartphone* confortavelmente em sua cintura.

O maior problema encontrado com a implementação da prova de conceito foi a degradação do nível de bateria durante o uso. O mesmo diminuiu de maneira significativa devido ao uso extensivo de funções que consomem muitos recursos.

Mesmo com este ponto de atenção, conclui-se que todos os objetivos propostos neste trabalho foram alcançados satisfatoriamente e que a prova de conceito desenvolvida tem uma aplicação prática importante no contexto em que se insere. Na Seção 5.1, discutir-se-ão novos rumos possíveis à mesma.

5.1 Trabalhos futuros

O reconhecimento de atividades humanas é uma área vasta de pesquisas e de aplicações diversas que tendem a crescer cada vez mais a cada ano com o surgimento de novas tecnologias e novas formas de se coletar dados das pessoas. Demonstrou-se, neste trabalho, que é possível reconhecer, com um bom grau de certeza, a atividade sendo realizada por um indivíduo com um *smartphone* simples acoplado em sua cintura. Deseja-se, no futuro, expandir o sistema proposto, inclusive utilizando outros algoritmos de reconhecimento. Haja vista que a abordagem pessoal teve a maior acurácia entre as pesquisadas, acredita-se que seja válido disponibilizar ao usuário um método automático que o instrua

a realizar algumas atividades e, transparentemente, treinar o modelo de acordo com suas características.

Além disso, espera-se aferir a possibilidade de criação de um *cache* no dispositivo móvel para que, mesmo em ambientes sem conectividade, os dados continuem sendo coletados e, quando o indivíduo voltar para uma área com cobertura de alguma rede de dados, estes sejam transmitidos ao servidor. Desta forma, o nível de bateria consumido seria menor, mitigando os efeitos do maior problema encontrado.

Outra ideia possível de expansão baseia-se em criar perfis distintos para uso do sistema: um modo para pessoas que necessitam de cuidados, no qual é mandatório o cadastro de um responsável diferente do próprio usuário do sistema, além de algumas regras já habilitadas por padrão, como as que podem identificar quedas, por exemplo, e um modo para pessoas que querem ter uma vida mais saudável adicionando atividades dinâmicas a sua rotina diária, neste perfil o sistema poderia priorizar regras que gerem alertas identificando muito tempo de inatividade.

Outras aplicações podem também utilizar o modelo proposto para disponibilizar ao usuário conteúdo específico de acordo com o contexto da atividade reconhecida, ou seja, quando o indivíduo iniciar uma corrida, por exemplo, a aplicação poderia automaticamente iniciar um cronômetro para marcar o tempo daquela corrida. Além disso, a mesma poderia mostrar o histórico das últimas vezes em que o usuário realizou esta mesma atividade próximo à posição geográfica atual, incluindo o tempo de cada corrida anterior.

Finalmente, uma tecnologia promissora a ser analisada com mais detalhes é o coprocessador de movimento M7 da Apple presente no iPhone 5S que foi desenvolvido com o intuito exclusivo de tratar os dados coletados nos sensores inerciais do aparelho, o que acontece sempre, mesmo com o dispositivo em *stand-by*, com gastos mínimos de bateria (ESTES, 2013). Isto significa que o usuário não precisará mais lembrar de ativar a aplicação de coleta de dados, o dispositivo estará continuamente gravando os dados transparentemente, o que facilitará muito a vida dos desenvolvedores.

REFERÊNCIAS

- ABBATE, S. et al. Monitoring of Human Movements for Fall Detection and Activities Recognition in Elderly Care Using Wireless Sensor Network: a Survey. In: _____. *Wireless Sensor Networks: Application-Centric Design*. [S.l.]: InTech, 2010. cap. 9, p. 147–166. ISBN 978-953-307-321-7.
- ANGUITA, D. et al. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In: BRAVO, J.; HERVAS, R.; RODRIGUEZ, M. (Ed.). *Ambient Assisted Living and Home Care*. Springer Berlin Heidelberg, 2012, (Lecture Notes in Computer Science, v. 7657). p. 216–223. ISBN 978-3-642-35394-9. Disponível em: <http://dx.doi.org/10.1007/978-3-642-35395-6_30>.
- CAMPILHO, A. *Nonparametric learning*. 2013. Acessado em 10/02/2014. Disponível em: <<http://repositorio-aberto.up.pt/bitstream/10451/25648/1/157242.pdf>>.
- CHANG, C.-C.; LIN, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, v. 2, p. 27:1–27:27, 2011. Software available at <<http://www.csie.ntu.edu.tw/~cjlin/libsvm>>.
- CLARITY. *Aging in Place in America*. 2007. Acessado em 06/02/2014. Disponível em: <<http://www.slideshare.net/clarityproducts/clarity-2007-aging-in-place-in-america-2836029>>.
- CZABKE, A.; MARSCH, S.; LUETH, T. Accelerometer based real-time activity analysis on a microcontroller. In: *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on*. [S.l.: s.n.], 2011. p. 40–46.
- DAVIES, N.; SIEWIOREK, D.; SUKTHANKAR, R. Activity-based computing. *Pervasive Computing, IEEE*, v. 7, n. 2, p. 20–21, April 2008. ISSN 1536-1268.
- DAVIS, J.; GOADRICH, M. The relationship between precision-recall and roc curves. In: *Proceedings of the 23rd International Conference on Machine Learning*. New York, NY, USA: ACM, 2006. (ICML '06), p. 233–240. ISBN 1-59593-383-2. Disponível em: <<http://doi.acm.org/10.1145/1143844.1143874>>.
- ESTES, A. *How Apple's M7 Chip Makes the iPhone 5S the Ultimate Tracking Device*. 2013. Acessado em 06/02/2014. Disponível em: <<http://gizmodo.com/how-apples-m7-chip-makes-the-iphone-5s-the-ultimate-tr-1286594287>>.
- FIGO, D. et al. Preprocessing techniques for context recognition from accelerometer data. *Personal Ubiquitous Comput.*, Springer-Verlag, London, UK, UK, v. 14, n. 7, p. 645–662, out. 2010. ISSN 1617-4909. Disponível em: <<http://dx.doi.org/10.1007/s00779-010-0293-9>>.
- HALL, M. et al. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, ACM, New York, NY, USA, v. 11, n. 1, p. 10–18, 2009. ISSN 1931-0145. Disponível em: <<http://dx.doi.org/10.1145/1656274.1656278>>.
- Hansen, Kim. *JavaOctave: A bridge from Java to Octave*. 2012. Acessado em 05/02/2014. Disponível em: <<https://kenai.com/projects/javaoctave/pages/Home>>.

- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning*. [S.l.]: Springer, New York, 2009.
- HAUBER, M. et al. jactivity: Supporting mobile web developers with html5/javascript based human activity recognition. In: *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*. New York, NY, USA: ACM, 2013. (MUM '13), p. 45:1–45:2. ISBN 978-1-4503-2648-3. Disponível em: <<http://doi.acm.org/10.1145/2541831.2541873>>.
- IBGE. *Pesquisa Nacional por Amostra de Domicílios*. 2012. Acessado em 29/01/2014. Disponível em: <http://www.ibge.gov.br/home/estatistica/populacao/trabalhoerendimento/pnad2011/default_brasil.shtm>.
- IMAGERY, N.; AGENCY, M. *National Imagery and Mapping Agency Technical Report 8350.2, Third Edition*. 2000. Acessado em 05/02/2014. Disponível em: <<http://earth-info.nga.mil/GandG/publications/tr8350.2/wgs84fin.pdf>>.
- KARANTONIS, D. et al. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *Information Technology in Biomedicine, IEEE Transactions on*, v. 10, n. 1, p. 156–167, Jan 2006. ISSN 1089-7771.
- KASTEREN, T. van et al. Accurate activity recognition in a home setting. In: *Proceedings of the 10th International Conference on Ubiquitous Computing*. New York, NY, USA: ACM, 2008. (UbiComp '08), p. 1–9. ISBN 978-1-60558-136-1. Disponível em: <<http://doi.acm.org/10.1145/1409635.1409637>>.
- LOPES, A.; MENDES-MOREIRA, J.; GAMA, J. *Semi-supervised learning: predicting activities in Android environment*. Montpellier, França: [s.n.], 2012. 38-42 p. Ubiquitous Data Mining (UDM) Workshop in conjunction with the 20th European Conference on Artificial Intelligence - ECAI 2012.
- MAURER, U. et al. Activity recognition and monitoring using multiple sensors on different body positions. In: *Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on*. [S.l.: s.n.], 2006. p. 4 pp.–116.
- MENDONCA, A. M. *Evaluation and Interpretation of Computer Aided Diagnosis Systems*. 2009. Acessado em 06/02/2014.
- MITCHELL, T. M. *Machine learning*. New York, NY [u.a.: McGraw-Hill, 1997. ISBN 0071154671 9780071154673. Disponível em: <http://www.worldcat.org/search?qt=worldcat_org_all&q=9780071154673>.
- OCTAVE COMMUNITY. *GNU Octave 3.8*. 2014. Disponível em: <www.gnu.org/software/octave/>.
- ORWAT, C.; GRAEFE, A.; FAULWASSER, T. Towards pervasive computing in health care a literature review. *BMC Medical Informatics and Decision Making*, BioMed Central, v. 8, n. 1, p. 1–18, 2008. Disponível em: <<http://dx.doi.org/10.1186/1472-6947-8-26>>.
- PIATETSKY-SHAPIRO, G. *KDnuggets news on SIGKDD service award*. 2005. Acessado em 05/02/2014. Disponível em: <<http://www.kdnuggets.com/news/2005/n13/2i.html>>.

ROGGEN, D. et al. An educational and research kit for activity and context recognition from on-body sensors. In: *BSN*. IEEE, 2010. p. 277–282. ISBN 978-1-4244-5817-2. Disponível em: <<http://dblp.uni-trier.de/db/conf/bsn/bsn2010.html#RoggenBSHLTWMG10>>.

SILVA, J. *Smartphone Based Human Activity Prediction*. 2013. Acessado em 06/02/2014. Disponível em: <<http://repositorio-aberto.up.pt/bitstream/10216/67649/2/954235.pdf>>.

SIMON, P. *Too Big to Ignore: The Business Case for Big Data*. Wiley, 2013. 89 p. (Wiley and SAS Business Series). ISBN 9781118642108. Disponível em: <<http://books.google.gr/books?id=Dn-Gdoh66sgC>>.

W3. *DeviceOrientation Event Specification*. 2012. Acessado em 27/01/2014. Disponível em: <<http://dev.w3.org/geo/api/spec-source-orientation.html>>.

W3. *Geolocation API Specification*. 2012. Acessado em 05/02/2014. Disponível em: <<http://dev.w3.org/geo/api/spec-source.html>>.

WEISS, G.; LOCKHART, J. *The Impact of Personalization on Smartphone-Based Activity Recognition*. 2012. Disponível em: <<https://www.aaai.org/ocs/index.php/WS/AAAIW12/paper/view/5203>>.

WEISSTEIN, E. W. *Euler Angles*. 2010. Acessado em 27/01/2014. Disponível em: <<http://mathworld.wolfram.com/EulerAngles.html>>.

WILDE, A. *An overview of human activity detection technologies for pervasive systems*. 2010. Disponível em: <https://diuf.unifr.ch/main/pai/sites/diuf.unifr.ch.main.pai/files/education_seminar_activityrecognition_awilde.pdf>.

WILDE, A. *Activity recognition for motion-aware pervasive systems*. 2011. Disponível em: <<http://eprints.soton.ac.uk/272433/>>.

YU, T.-H.; KIM, T.-K.; CIPOLLA, R. Real-time action recognition by spatiotemporal semantic and structural forest. In: *Proceedings of the British Machine Vision Conference*. [S.l.]: BMVA Press, 2010. p. 52.1–52.12. ISBN 1-901725-40-5. Doi:10.5244/C.24.52.

ZHANG, M.; SAWCHUK, A. Human daily activity recognition with sparse representation using wearable sensors. *Biomedical and Health Informatics, IEEE Journal of*, v. 17, n. 3, p. 553–560, May 2013. ISSN 2168-2194.

APÊNDICE A – CÓDIGOS EM OCTAVE

Neste apêndice estão detalhados alguns códigos implementados em Octave e utilizados no sistema proposto.

A.1 Calcular velocidade angular

Na Listagem A.1, está detalhada a função que, a partir dos *ângulos de Euler* (capturados pela variável *rotationRate*), utiliza a Equação 3.1 para retornar a velocidade angular nos três eixos.

Listing A.1 – Função para calcular a velocidade angular a partir dos *ângulos de Euler*.

```
function [velAngX, velAngY, velAngZ] = calcularVelocidadeAngular(alphaV,
    betaV, gammaV)
    velAngX = zeros(size(alphaV));
    velAngY = zeros(size(betaV));
    velAngZ = zeros(size(gammaV));
    for (j=1:length(alphaV))
        alpha = alphaV(j);
        beta = betaV(j);
        gamma = gammaV(j);

        a = [alpha 0 0]' + ([1 0 0; 0 cos(alpha) sin(alpha); 0 -sin(
            alpha) cos(alpha)] * [0 beta 0]') + ([1 0 0; 0 cos(
            alpha) sin(alpha); 0 -sin(alpha) cos(alpha)] * [cos(beta
            ) 0 -sin(beta); 0 1 0; sin(beta) 0 cos(beta)] * [0 0
            gamma]');

        velAngX(j) = a(1);
        velAngY(j) = a(2);
        velAngZ(j) = a(3);
    end
end
```

A.2 Gerar características

Na Listagem A.2, detalhou-se a função que, a partir dos dados brutos capturados em cada janela, calcula as características que serão utilizadas como entrada tanto para gerar o modelo, quanto para o algoritmo de reconhecimento. Inicialmente a função calcula o vetor de velocidade angular, de acordo com a Listagem A.1, em seguida calcula as cinco funções simples descritas na Seção 3.1, e, finalmente, retorna um vetor-coluna com as características.

Listing A.2 – Função para calcular as características a partir dos dados brutos.

```

function [features] = gerarFeatures(aX, aY, aZ, agX, agY, agZ, rX, rY, rZ)

[vX, vY, vZ] = calcularVelocidadeAngular(rX,rY,rZ);

f1 = min(aX);
f2 = min(aY);
f3 = min(aZ);
f4 = min(agX);
f5 = min(agY);
f6 = min(agZ);
f7 = min(rX);
f8 = min(rY);
f9 = min(rZ);
f10 = min(vX);
f11 = min(vY);
f12 = min(vZ);

f13 = max(aX);
f14 = max(aY);
f15 = max(aZ);
f16 = max(agX);
f17 = max(agY);
f18 = max(agZ);
f19 = max(rX);
f20 = max(rY);
f21 = max(rZ);
f22 = max(vX);
f23 = max(vY);
f24 = max(vZ);

f25 = mean(aX);
f26 = mean(aY);
f27 = mean(aZ);
f28 = mean(agX);
f29 = mean(agY);
f30 = mean(agZ);
f31 = mean(rX);
f32 = mean(rY);
f33 = mean(rZ);
f34 = mean(vX);
f35 = mean(vY);
f36 = mean(vZ);

f37 = std(aX);
f38 = std(aY);
f39 = std(aZ);

```

```

f40 = std(agX);
f41 = std(agY);
f42 = std(agZ);
f43 = std(rX);
f44 = std(rY);
f45 = std(rZ);
f46 = std(vX);
f47 = std(vY);
f48 = std(vZ);

f49 = corr(aX, aY);
f50 = corr(aY, aZ);
f51 = corr(aX, aZ);
f52 = corr(agX, agY);
f53 = corr(agY, agZ);
f54 = corr(agX, agZ);
f55 = corr(rX, rY);
f56 = corr(rY, rZ);
f57 = corr(rX, rZ);
f58 = corr(vX, vY);
f59 = corr(vY, vZ);
f60 = corr(vX, vZ);

features = [f1, f2,      f3,      f4,      f5,      f6,      f7,      f8,      f9,
            f10, f11, f12, f13, f14, f15, f16, f17, f18, f19, f20, f21, f22,
            f23, f24, f25, f26, f27, f28, f29, f30, f31, f32, f33, f34, f35, f36,
            f37, f38, f39, f40, f41, f42, f43, f44, f45, f46, f47, f48, f49, f50,
            f51, f52, f53, f54, f55, f56, f57, f58, f59, f60]';

endfunction

```

A.3 Normalizar características

Depois de calculadas, as características devem ser normalizadas. Na Listagem A.3 está detalhada a função que calcula os parâmetros a serem usados na normalização da média, que são a média e o desvio padrão de cada característica.

Listing A.3 – Função para calcular os parâmetros a serem usados na normalização da média.

```

function [parametros] = gerarParametrosMeanNormalization(matriz)
    colunas = size(matriz, 2);
    parametros = zeros(2, colunas);
    for (i = 1: colunas)
        vetor = matriz(:, i);
        parametros(1, i) = mean(vetor);
        parametros(2, i) = std(vetor);
    end

```

```
end
```

Em seguida, estes parâmetros servem como entrada para a função que normaliza de fato as características, detalhada na Listagem A.4, de acordo com a Equação 3.2.

Listing A.4 – Função para calcular a normalização da média das características de acordo com a Equação 3.2.

```
function [featuresNormalizadas] = meanNormalization(features , parametros)
    featuresNormalizadas = zeros(size(features));
    colunas = size(features ,2);
    for (i = 1: colunas)
        vetor = features (: , i);
        n1 = (vetor - parametros(1 , i))/parametros(2 , i);
        featuresNormalizadas (: , i) = n1;
    end
end
```

A.4 Gerar modelo

Os passos descritos na Listagem A.5 foram seguidos para geração do modelo a ser utilizado. Primeiro foram geradas as características (Listagem A.2), depois os parâmetros para normalização foram salvos (Listagem A.3). Em seguida, as características foram normalizadas (Listagem A.4) que serviram como entrada, junto com um vetor-coluna contendo o rótulo de cada atividade, para a função *svmtrain* da biblioteca LibSVM. Finalmente, o modelo foi salvo.

Listing A.5 – Função para reconhecer qual atividade está sendo executada.

```

features = gerarFeatures(accX, accY, accZ, accGravityX, accGravityY,
    accGravityZ, rotX, rotY, rotZ);

parametros = gerarParametrosMeanNormalization(features)
save parametrosMN.txt parametros

featuresNormalizadas = meanNormalization(features, parametros)

options = "-nr_classes_7-c_1-g_0.025-b_1"
modelo1 = svmtrain(atividade, featuresNormalizadas, options)

save -binary modelo1.binary modelo1

```

A.5 Reconhecer atividade

Finalmente, para reconhecer a atividade sendo executada foi criada uma função que recebe como entrada as características e retorna o código da atividade sendo executada. A mesma está detalhada na Listagem A.6.

Primeiro a função carrega os parâmetros da normalização, de acordo com a Listagem A.3. Em seguida, normaliza as características, Listagem A.4. O próximo passo é carregar o modelo previamente gerado para reconhecer a atividade. Finalmente, a função *svmpredict* da biblioteca LibSVM recebe como entrada as características normalizadas e o modelo (Listagem A.5), e tem como saída a atividade identificada com maior probabilidade.

Listing A.6 – Função para reconhecer qual atividade está sendo executada.

```

function [resultado, dec_values_L] = predictFeatures(features)

parametros = load("parametrosMN.txt");
featuresComMeanNormalization = meanNormalization(features, parametros);

load ('modelo1.binary');

[predict_label_L, accuracy_L, dec_values_L] = svmpredict(0,
    featuresComMeanNormalization, modelo1, '-b_1');
resultado = predict_label_L;

endfunction

```