

***Wearable* Baseado em Internet das Coisas para Monitoramento de Sonâmbulos**

Darlan Domingos Candioto¹, Luciano Antunes¹

¹Ciência da Computação – Universidade do Extremo Sul Catarinense (UNESC)
Av. Universitária, 1105 - CEP: 88806-000 – Criciúma – SC – Brasil

darlan.candioto98@gmail.com, luc@unesc.net

Abstract. *Sleepwalking is more dangerous than it seems and several incidents have been reported where people do dangerous activities when it happens. The goal of this research is to apply the concept of Internet of Things using a bracelet integrated with a mobile application to manage the displacement of sleepwalkers. The application can detect the displacement of a sleepwalker who is using the bracelet and alert about it on a mobile device through a server, and so a close person assist that person. Inertial sensors were used to monitor the sleepwalker's movements and identify their displacement. The results show that the application performs well in rooms over 9m².*

Resumo. *O sonambulismo é mais perigoso do que parece, vários incidentes foram relatados em que as pessoas praticam atividades perigosas durante os episódios. Esta pesquisa tem como objetivo aplicar o conceito de Internet das Coisas na integração de uma pulseira com um aplicativo móvel para gerenciar o deslocamento de sonâmbulos. A aplicação pode detectar o deslocamento de um sonâmbulo por meio de uma pulseira e alertar o evento em um dispositivo móvel mediante um servidor, a fim de que uma pessoa próxima possa auxiliá-lo. Para monitorar os movimentos do sonâmbulo e identificar seus deslocamentos foram usados sensores inerciais. Os resultados apontam que a aplicação tem bom desempenho em quartos com mais de 9m².*

1. Introdução

O sonambulismo é um distúrbio do sono que atinge homens, mulheres e crianças, não há restrição de idade nem sexo. Este fenômeno consiste em uma série de comportamentos que se inicia durante o sono lento, sua principal característica é o caminhar noturno. O indivíduo pode apresentar durante o episódio um comportamento agitado, envolvendo o sujeito em situações perigosas como, sair de casa, subir em janelas, sacadas, provocando quedas e lesões em locais desprotegidos [Nunes 2002]. O distúrbio além de ser perigoso para o próprio sonâmbulo pode ser também para pessoas próximas, o que torna necessário o monitoramento contínuo do paciente. Porém, o monitoramento contínuo, principalmente à noite, é difícil para os familiares. Por mais que existam alguns medicamentos disponíveis no mercado não é aconselhado usá-los, pois apresentam efeitos colaterais [Singhal e Jain 2015].

Com o advento e o avanço da Internet das Coisas (IoT) o problema pode ser visto com uma perspectiva diferente. Esta tecnologia vem inovando diversas áreas do conhecimento e introduzindo técnicas modernas na resolução de problemas. A IoT possibilita conectar diversos objetos, agregando funcionalidades e expandindo novas tecnologias, como os *wearables*, que são formados por dispositivos eletrônicos

(sensores) conectados aos seres humanos. Estes equipamentos geralmente são usados pelas pessoas como acessórios (pulseiras) que auxiliam em suas tarefas do dia a dia. Além disso, podem ser usados para monitorar as atividades das pessoas [Verma, Rajnish e Fatima 2017; Dian, Vahidnia e Rahmati 2020].

O reconhecimento da atividade humana tornou-se um dos tópicos mais estudados devido ao crescimento de tecnologias como *Machine Learning* (ML), Inteligência Artificial (IA) e Internet das Coisas (IoT) [Jobanputra, Bavishi e Doshi 2019]. Dispositivos vestíveis compostos por sensores inerciais são amplamente usados para reconhecer as atividades das pessoas [Badawi, Al-Kabanny e Shaban 2020].

Os sensores inerciais são dispositivos eletromecânicos tridimensionais (3D) que integram um acelerômetro de três eixos e um giroscópio de três eixos. O acelerômetro é capaz de medir uma aceleração dinâmica e uma estática em cada eixo x, y e z presente no equipamento. A aceleração dinâmica é gerada quando o sensor se move para alguma direção, enquanto a estática é uma aceleração de inércia, esta é conhecida como força gravitacional (G), $1G \approx 9.8 \text{ m/s}^2$ [Kurniawan, Hermawan e Purnama 2016]. O giroscópio é capaz de medir a velocidade angular presente em cada eixo x, y e z do sensor. Uma funcionalidade bem aplicada no uso desses sensores é sua fusão, conhecida popularmente como fusão de sensores. Esta funcionalidade é usada para minimizar erros que tanto o giroscópio quanto o acelerômetro sofrem de ruídos externos. Uma das técnicas mais conhecidas é o uso do filtro de Kalman, que visa por meio de um algoritmo preditivo diminuir os ruídos e entregar informações mais próximas das reais [Oliveira e Gonçalves 2017].

Conforme o levantamento bibliográfico foi encontrado poucas pesquisas com a finalidade de monitorar sonâmbulos, em contrapartida, diversas para classificar atividades humanas. O estudo publicado por Singhal e Jain (2015) desenvolveu uma aplicação que visou monitorar sonâmbulos usando um sensor de presença (infravermelho) e uma campanha a fim de detectar os eventos e alertá-los, um acelerômetro foi posicionado na cintura dos indivíduos para identificar suas quedas. O sistema realizava ligação e enviava mensagem quando identificado um evento. Os resultados em laboratório foram promissores, demonstrando que o sistema estava pronto para ser aplicado em eventos reais. A pesquisa de Li, Hou e Huang (2013) permitiu usar um acelerômetro no tórax ou na cintura para classificar atividades humanas como: caminhar, correr, ficar parado e detectar quedas. O algoritmo de reconhecimento das atividades era executado em um *smartphone*, que recebia os dados do sensor por meio do *Bluetooth*. Os resultados obtiveram uma precisão acima de 90% na classificação das atividades. O estudo realizado por Nguyen, Fan e Shahabi (2015) propôs usar uma base de dados com informações de um acelerômetro aplicado no pulso esquerdo e direito de três indivíduos para reconhecer suas atividades de deambulação. As principais atividades dos usuários avaliadas foram: caminhar, ficar em pé, sentar e deitar. Para auxiliar na classificação dessas atividades usaram-se alguns algoritmos com *Machine Learning* (ML). Os resultados foram positivos, demonstrando que é possível reconhecer atividades de deambulação com um acelerômetro no pulso. O trabalho desenvolvido por Mano et al. (2016) usou um modelo de solução IoT que permitiu monitorar as atividades diárias de idosos (andando, correndo, sentado, deitado e em pé) por meio de um acelerômetro e um giroscópio aplicados no antebraço a fim de identificar quedas e notificá-las a um dispositivo móvel. O sistema apresentou um bom desempenho, chegando a uma precisão de 96.62% no reconhecimento das atividades.

De acordo com os estudos realizados encontram-se poucas aplicações que sejam capazes de auxiliar indivíduos com sonambulismo, principalmente no setor de *wearables*. Portanto, esta pesquisa tem por objetivo aplicar o conceito de Internet das Coisas na integração de uma pulseira eletrônica com um aplicativo móvel para gerenciar o deslocamento de sonâmbulos.

Os objetivos específicos desta pesquisa consistem em: identificar as características de pessoas que possuem sonambulismo; identificar o modelo de acelerômetro e giroscópio a ser usado na supervisão do indivíduo; desenvolver um protótipo de pulseira para identificação do sonâmbulo; desenvolver um protótipo de sistema móvel para gerenciar o comportamento do sonâmbulo; integrar o aplicativo desenvolvido com a pulseira de identificação por meio de um servidor *web*; avaliar os resultados obtidos com a aplicação.

2. Materiais e Métodos

Esta pesquisa é aplicada, de base tecnológica e explicativa. Desenvolveu-se uma aplicação que empregou os conceitos de IoT e *wearables*, a fim de monitorar a movimentação de indivíduos sonâmbulos por meio de sensores inerciais e de notificar o seu deslocamento em um aplicativo móvel. Esta aplicação é constituída de três módulos, são eles: sensor, servidor e móvel (Figura 1). Foi usado o NoSQL *Realtime Database*, fornecido pela plataforma Firebase de forma gratuita com capacidade máxima de 1GB de dados, para armazenar as informações referentes aos deslocamentos do usuário.

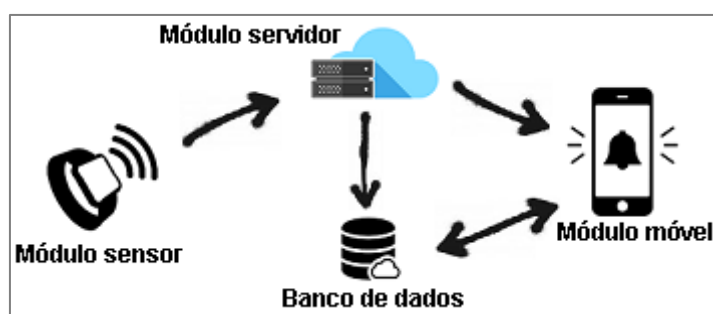


Figura 1. Módulos da aplicação

O módulo sensor consiste em um protótipo de pulseira capaz de monitorar as movimentações do sonâmbulo. O módulo móvel é composto por um aplicativo que notifica, em tempo real, o deslocamento capturado pelo módulo sensor, ao usuário de um dispositivo móvel o qual pode ser um familiar ou uma pessoa próxima ao sonâmbulo. O módulo servidor conecta em tempo real os módulos citados anteriormente por meio de um serviço que trabalha com o protocolo de aplicação *Message Queue Telemetry Transport* (MQTT). Este módulo é o encarregado por armazenar a informação do deslocamento no banco de dados para o uso do aplicativo. A conexão entre os módulos é realizada mediante o modelo de rede sem fio *Wi-Fi*.

2.1. Desenvolvimento do Módulo Servidor

O servidor foi desenvolvido em linguagem JavaScript (JS) no ambiente de programação Visual Studio Code, versão 1.50.1. Para executar o código JS em *backend* foi usado o Node.js, versão 12.16.1. Foram usadas quatro bibliotecas para auxiliar no desenvolvimento da aplicação, são elas: Dotenv versão 8.2.0, Moment versão 2.27.0,

Firestore versão 7.18.0 e Mosca versão 2.8.3. As bibliotecas foram disponibilizadas de forma gratuita pelo gerenciador de pacotes Node Package Manager (NPM), versão 6.13.4.

A biblioteca Dotenv permitiu que fossem configuradas algumas variáveis de ambiente para facilitar o desenvolvimento do código. A Moment disponibilizou várias funções que permitiram manipular variáveis vinculadas às datas. A conexão e a inserção de registros no banco de dados foram feitas pela Firebase. Por meio da biblioteca Mosca foi possível criar e dispor de um serviço baseado no protocolo de aplicação MQTT.

Quando executada a aplicação por meio do Node.js, é iniciado o serviço na porta 1883 mediante a biblioteca Mosca. Para conectar as aplicações em pontos distintos da rede criam-se tópicos de comunicações, estes são gerenciados por um elemento interno da biblioteca conhecido como *broker*. Os tópicos possibilitam enviar e receber mensagens entre os módulos. Aplicações que enviam mensagens por meio desses tópicos são classificadas como editoras e aplicações que ficam escutando as mensagens dos tópicos são classificadas como assinantes.

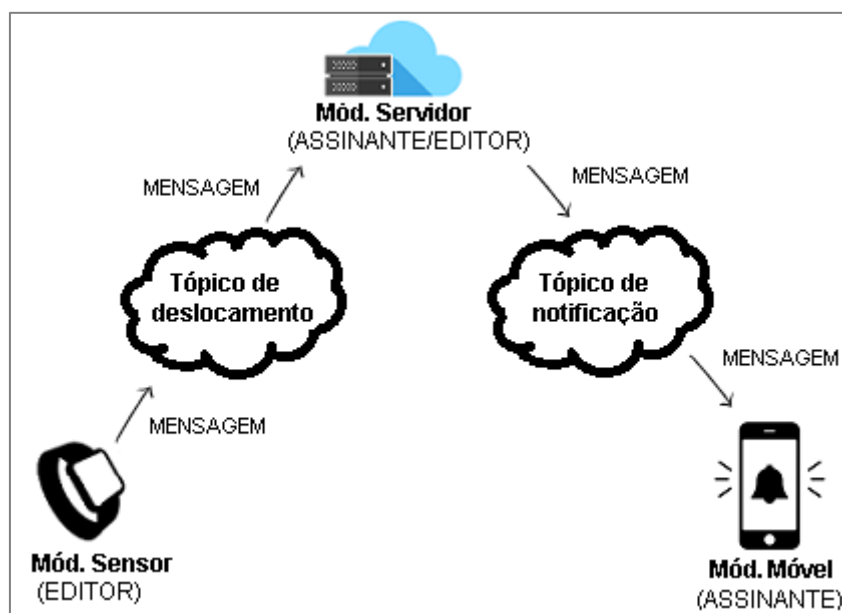


Figura 2. Arquitetura do protocolo de mensagens

Como demonstrado na Figura 2 a solução proposta usa dois tópicos a fim de conectar os módulos, são eles: tópico de deslocamento e tópico de notificação. Os nomes dos tópicos foram criptografados em um *hash* por um serviço online conhecido como SHA256 a fim de criar nomes únicos para cada tópico e aumentar sua segurança. O tópico de deslocamento como o próprio nome sugere, está vinculado à mensagem enviada do módulo sensor ao identificar um novo deslocamento. Portanto, este módulo é um editor do tópico de deslocamento. Para comunicar o aplicativo sobre o evento de deslocamento o módulo servidor assina o tópico citado anteriormente a fim de identificar uma nova mensagem. Quando o módulo servidor recebe uma nova mensagem por meio do tópico de deslocamento, ele imediatamente cria uma mensagem e a envia ao tópico de notificação, e como este tópico tem como assinante o módulo móvel, ao identificar uma nova mensagem este módulo dispara um alerta. Portanto, é possível conectar todos os módulos por meio do envio de mensagens.

```
account: "9eRtQ2PKp6flq9tX80BUKTjgs4F3"  
date: "2020-10-28 19:45:03"  
message: "Evento de sonambulismo detectado às 19:45."
```

Figura 3. Objeto de um deslocamento detectado

A mensagem recebida no módulo servidor por meio do tópico de deslocamento contém o ID da conta conectada no aplicativo. Este ID é configurado de forma estática no módulo sensor. Antes de enviar uma nova mensagem ao tópico de notificação o servidor cria um objeto (Figura 3) com o ID da conta recebida, a data do episódio e um texto com a descrição sobre o horário do evento. A mensagem é enviada ao tópico de notificação com um protocolo que concatena por meio de um & o ID da conta recebida e a descrição do horário. Por fim, o servidor salva o objeto no banco de dados. Estes dados salvos são usados posteriormente pelo aplicativo para gerar relatórios dos eventos identificados.

2.2. Desenvolvimento do Módulo Móvel

O aplicativo voltado para o sistema operacional Android foi desenvolvido em linguagem Java versão 8, a codificação ocorreu por meio da IDE Android Studio versão 3.6.1. As principais bibliotecas usadas para construir o app e anexá-lo dentro do ecossistema IoT foram: Firebase Auth versão 19.3.2, Firebase Database versão 19.3.1, MPAndroidChart versão 2.1.6 e Paho Android Service versão 1.0.2. Todas as bibliotecas mencionadas foram usadas de forma gratuita pelo gerenciador de pacotes do Android Studio.

A biblioteca Firebase Auth permitiu criar contas de acesso ao app e autenticá-las durante o login na aplicação. Por meio da biblioteca Firebase Database foi possível acessar o banco de dados NoSQL *Realtime Database* e suas funções de leitura e escrita. O MPAndroidChart foi usado para criar relatórios gráficos que estavam vinculados a dados salvos pelo servidor. Por fim, foi usado o Paho Android Service, que permitiu estabelecer conexão ao serviço disponibilizado pelo servidor e trabalhar com o protocolo de aplicação MQTT.

No desenvolvimento do aplicativo foi criada uma tela de cadastro, no qual foram requisitados dados básicos do usuário, os registros são realizados por meio de um serviço gratuito do Firebase conhecido como *Authentication*, que salva os dados principais para entrar na aplicação, como e-mail e senha, as outras informações do usuário são salvas no próprio banco de dados. Foi desenvolvida uma tela de login para autenticação do usuário, a conta é autenticada usando-se o serviço *Authentication*. Portanto, logo após o cadastro, o usuário é levado à tela de acesso para entrar na aplicação. Foi desenvolvida uma tela principal para monitorar as mensagens enviadas ao tópico de notificação e para mostrar informações de eventos já capturados. Quando a autenticação é válida o usuário tem permissão para acessar a tela principal do aplicativo, esta tela busca no banco os dados salvos pelo servidor e produz relatórios gráficos referentes aos períodos dos eventos capturados. Estão presentes nesses relatórios a quantidade de eventos detectados, quantos detectados por dia do mês (1, 2, 3...), por dia da semana (Segunda, Terça, Quarta...) e por horário (00:00h, 01:00h, 02:00h...). Estes relatórios estão vinculados ao mês e ao ano, selecionados no topo da aplicação na forma de um filtro.

Durante o acesso a tela principal, o aplicativo cria um serviço de monitoramento por meio do objeto nativo *Service* para executar em *background* e assinar o tópico de notificação. Portanto, é possível receber novas mensagens quando a aplicação está em segundo plano e até mesmo quando o dispositivo móvel estiver bloqueado. Para manter este serviço ativo e evitar que o sistema operacional do android o elimine, é criado um processo com os objetos nativos *Handler* e *Runnable*, que fica executando em um intervalo de 1.5 minutos, neste processo uma mensagem é enviada a um tópico auxiliar do servidor chamado de *alive* (nome do tópico foi criptografado em um *hash* pelo SHA256). É notificado sobre o serviço de monitoramento, pois o sistema operacional android não aconselha executar atividades em *background* sem que o usuário tenha conhecimento.



Figura 4. Notificações (a) e informações sobre o evento de sonambulismo (b)

Ao receber uma nova mensagem, o aplicativo usa o protocolo do & para obter as duas informações enviadas pelo servidor, que são: o ID da conta usada no módulo sensor e a descrição sobre o horário do evento capturado. Posteriormente, é avaliada se a conta é a mesma conectada no aplicativo, caso a condição seja satisfeita, é criada uma notificação por meio dos objetos nativos *NotificationCompat*, *NotificationManager* e *NotificationChannel*, e emitido um som por meio do objeto nativo *MediaPlayer*, o som é semelhante a um despertador, a fim de chamar a atenção do usuário. Só é possível remover a notificação clicando sobre a mesma, e ao clicá-la o usuário é levado a uma tela que foi desenvolvida para exibir algumas informações sobre o evento, como o horário, obtido pela mensagem enviada pelo servidor, e os nomes, informados durante o cadastro da conta, estes nomes referem-se ao usuário do aplicativo e da pulseira, respectivamente. É possível visualizar as notificações e as informações mencionadas na Figura 4.

2.3. Desenvolvimento do Módulo Sensor

No desenvolvimento do protótipo da pulseira, estudaram-se, na parte de *hardware*, as tecnologias associadas aos equipamentos empregados e a montagem do circuito eletrônico. Na parte de *software* foi preciso aprender a forma de estabelecer a comunicação com o *hardware*, por isso este módulo foi feito por último e levou-se um tempo a mais no seu desenvolvimento. O custo total para montagem do equipamento aproximou-se de R\$ 120,00. Todos os equipamentos eletrônicos selecionados foram de pequeno porte para tornar a pulseira a mais compacta possível, já que quanto menor mais confortável para o usuário.

2.3.1. Montagem da Pulseira

No desenvolvimento da pulseira, na parte de *hardware*, foi usada uma bateria de lítio da SONY do tipo N, um controlador de carga TP4056, uma placa controladora Wemos D1 Mini Nodemcu Esp8266 com chip *Wi-Fi* nativo, um sensor MPU-6050 da InvenSense com acelerômetro e giroscópio integrados, quatro resistores com valores de 1 ohm (1 Watt), 6.8 ohms (1 Watt), 68 ohms (1 Watt) e 82 ohms (1/8 Watt), um diodo zener 1N4732, um diodo 1N4148, um botão de retenção, um botão de pulso e um led verde.

Para acoplar os componentes usou-se uma caixa de plástico com as seguintes dimensões 520x420x270mm. Os componentes foram presos no interior da caixa por meio de cola quente. Para fixar a caixa no pulso do usuário foi usada uma pulseira de elástico com velcro. A caixa foi presa ao elástico por meio de cola quente e pequenos parafusos. É possível visualizar a montagem final da pulseira na Figura 5.



Figura 5. Pulseira eletrônica

2.3.2. Montagem do Circuito

No desenvolvimento lógico do circuito eletrônico utilizou-se uma placa controladora Wemos D1 Mini Nodemcu Esp8266 com uma alimentação entre 3.3 volts (V) a 5V e uma dimensão de 39x25x3mm. Para monitorar os movimentos do usuário foi empregue um sensor MPU-6050 com uma alimentação de 3V à 5V e um tamanho de 21x17x3mm, que integra um acelerômetro com três eixos (x, y e z) e um giroscópio com três eixos (x, y e z), suas faixas escalares são ± 2 , ± 4 , ± 8 , $\pm 16g$ e ± 250 , ± 500 , ± 1000 , $\pm 2000^\circ/s$ para o acelerômetro e giroscópio, respectivamente.

A bateria de lítio da SONY aplicada dispõe de 3.6V e 630 miliampère-hora (mAh) para alimentar o circuito. O botão de retenção usado no circuito permite controlar a alimentação na Wemos D1 Mini Nodemcu Esp8266 e no led para indicar o status (ligada/desligada) da pulseira. Empregou-se um botão de pulso para o controle de

tensão sobre o MPU-6050. Usou-se um controlador de carga, três resistores, um diodo e um diodo zener para o controle de tensão e corrente sobre alguns equipamentos. O circuito eletrônico do protótipo é mostrado na Figura 6.

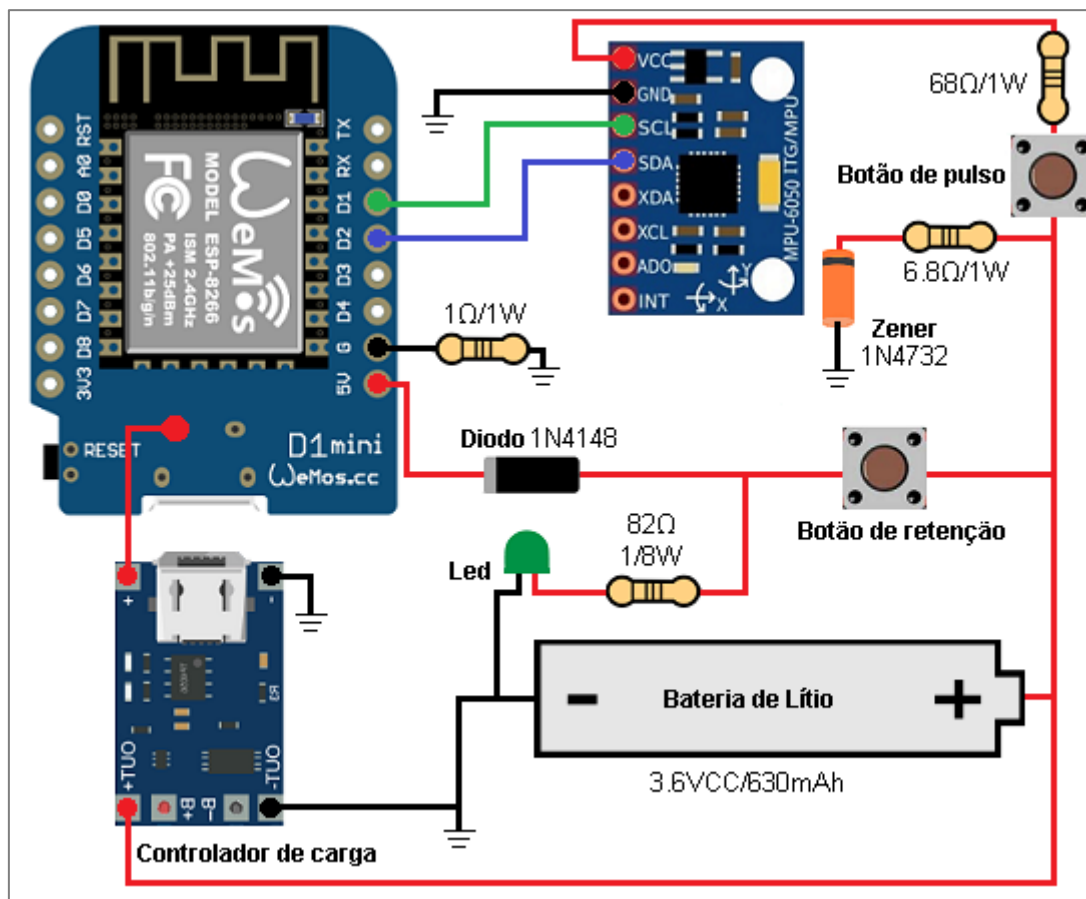


Figura 6. Circuito eletrônico do protótipo

A Wemos D1 Mini Nodemcu Esp8266 e o MPU-6050 comunicam-se mediante a interface *Inter-Integrated Circuit* (I²C). Para estabelecer essa comunicação conectaram-se os pinos SCL e SDA do sensor aos pinos D1 e D2 da placa controladora, respectivamente, que são as portas usadas para comunicar os equipamentos conforme a interface supracitada.

Com a presença da bateria de lítio é possível recarregar o sistema eletrônico apenas energizando-o. Para auxiliar no processo de energização usou-se a entrada micro USB da placa controladora, sendo possível recarregar o circuito com um carregador de 5V e um cabo micro USB. Foi aplicado um controlador de carga para manter o nível de tensão estável sobre a bateria durante o carregamento, evitando que ocorra uma sobrecarga e a danifique. O controlador de carga foi colocado entre a placa controladora e a bateria, conectando-se o *Ground* (GND) de entrada e o OUT- (B-) do controlador ao GND da bateria, a alimentação de entrada do controlador ao USB da placa controladora, e o OUT+ (B+) do controlador na Tensão Corrente Contínua (VCC) da bateria.

Para fazer o controle de alimentação na Wemos D1 Mini Nodemcu Esp8266 foi aplicado um botão de retenção e um led para indicar o status (ligado/desligado) do equipamento, em série com o led foi colocado um resistor de 82ohms (Ω) para controle de tensão e corrente sobre o mesmo, que trabalha com uma alimentação próxima de 2V

e uma corrente de 20 miliampére (mA). Foi posto um diodo 1N4148 inversamente polarizado com o VCC da placa controladora para que durante o carregamento da bateria o led não fique aceso caso o botão de retenção esteja desligado.

O MPU-6050 recebe uma alimentação direta da bateria, por isso, para o controle da corrente e tensão sobre o mesmo foi aplicado um resistor de 68Ω . O GND do sensor foi ligado diretamente ao GND da bateria. O botão de pulso presente no circuito por padrão fica fechado, foi aplicado para cortar a energia e dar *reset* no sensor quando necessário, desta forma, é possível limpar os dados salvos no equipamento de forma manual. Foi ligado com a alimentação do sensor um resistor de 6.8Ω e um diodo zener 1N4732 de 4.7V inversamente polarizado com o VCC da bateria a fim de controlar a tensão sobre o MPU-6050, principalmente quando a pulseira está sendo carregada, pois a tensão no circuito pode ultrapassar 5V e acabar danificando o sensor.

Durante uma conexão USB de algum equipamento (*notebook/computador*) com a placa controladora pode existir uma diferença de potencial entre a bateria de lítio e o dispositivo. Por isso, foi usado um resistor de 1Ω entre o GND da placa controladora e da bateria. Portanto, é possível evitar que o equipamento conectado não se danifique.

2.3.3. Montagem do Software

No desenvolvimento do *software* foi usado o Arduino IDE em sua versão 1.8.13, com a linguagem de programação C++. Para auxiliar no desenvolvimento da aplicação foram implementadas cinco bibliotecas, são elas: Wire.h, ESP8266WiFi.h, MPU6050.h versão 0.1.0, Kalman.h versão 1.0.2 e PubSubClient.h versão 2.8.0. As bibliotecas Wire.h e ESP8266WiFi.h foram disponibilizadas pelo próprio Arduino IDE, por isso, não foi possível identificar a versão. Porém, MPU6050.h, Kalman.h e PubSubClient.h foram obtidas de forma gratuita pelo Github mediante os autores mantenedores do código que eram EletronicCats, TKJEletronics e Knolleary, respectivamente.

A biblioteca Wire.h foi usada para estabelecer a comunicação da placa controladora com o sensor via interface I²C e também para escrever e ler valores dos registradores. Usou-se a MPU6050.h para calibrar o sensor e armazenar os valores de *offsets* calculados durante a calibração, pois não foi possível por meio da Wire.h registrar os *offsets* calculados. Kalman.h foi utilizada para aplicar um filtro recursivo preditivo nas medidas ruidosas do sensor a fim de gerar dados mais próximos dos reais. A ESP8266WiFi.h foi usada para estabelecer conexão a uma rede *Wi-Fi*. Por fim, usou-se a PubSubClient.h para conectar-se ao servidor e trabalhar com o protocolo de aplicação MQTT.

O código do sistema pode ser separado em duas etapas, são elas: Configuração (*setup*) e Execução (*loop*). Durante a etapa de configuração é feita a conexão a uma rede *Wi-Fi*, posteriormente a conexão ao servidor mediante o seu endereço, em seguida são configurados os dados necessários para o funcionamento do sensor, como sua faixa escalar, definida em $\pm 2g$ para o acelerômetro e $\pm 250^\circ/s$ ao giroscópio, esta e outras configurações aplicadas no sensor estão disponíveis na Tabela 1. Logo após a configuração do MPU-6050 é realizada sua calibração para melhorar a precisão de seus valores com um algoritmo de Luis Ródenas. Durante o processo de calibração o sensor deve ficar com o sentido dos eixos iguais aos da Figura 7 e parado em uma superfície horizontal plana, sem nenhuma perturbação próxima, pois durante este processo é removida a aceleração gravitacional ($1G \approx 9.8m/s^2$) do eixo Z, a fim de calcular os

offsets que aproximem à leitura dos valores de zero, aumentando a precisão dos dados. Por fim, são configuradas as informações iniciais e obrigatórias da biblioteca de Kalman, que se referem a ângulos de rotações calculados por meio dos dados do acelerômetro e velocidades angulares obtidas do giroscópio.

Tabela 1. Configurações do sensor MPU-6050

	Acelerômetro	Giroscópio
Taxa de amostragem limite	1kHz	1kHz
Largura de banda	260Hz	256Hz
Delay	0ms	0.98ms
Faixa escalar	±2g	±250°/s

A etapa de execução voltou-se à manipulação dos dados e à lógica estabelecida para identificar um deslocamento. Para monitorar a movimentação do usuário a fim de gerenciar seu deslocamento são usados os valores de aceleração e orientação do sensor. Os valores de aceleração são obtidos do próprio acelerômetro e a orientação é obtida mediante fórmulas matemáticas e algoritmo preditivo. Como demonstrado na Figura 7 é possível obter a orientação do equipamento por meio dos ângulos de rotações sobre os eixos x e y do acelerômetro, conhecidos como *roll* e *pitch*, respectivamente. A equação (1) e (2) referem-se às expressões matemáticas usadas para calcular *roll* e *pitch*, respectivamente [Kurniawan, Hermawan e Purnama 2016].

$$roll = \tan^{-1} \left(\frac{ay}{\sqrt{ax^2 + az^2}} \right) \quad (1)$$

$$pitch = \tan^{-1} \left(\frac{ax}{\sqrt{ay^2 + az^2}} \right) \quad (2)$$

Os valores de ax , ay e az referem-se às acelerações obtidas durante a leitura dos eixos x, y e z respectivamente do sensor. Para calcular a tangente inversa no código foi usada a função atan. Por mais que calibre-se, o acelerômetro ainda sofre de sinais ruidosos, por esse motivo *roll* e *pitch* sofrem interferências diretas em seus valores. Por isso, é aplicado o filtro de Kalman, que usa um método matemático de predição para obter ângulos mais precisos por um longo período de tempo. O algoritmo de predição disponibilizado pela biblioteca usa separadamente os valores calculados em *roll* e *pitch* com os da velocidade angular em torno dos eixos x e y respectivamente, estas velocidades são obtidas por meio do giroscópio e convertidas para a unidade de graus por segundo (°/s), assim como os ângulos calculados em *roll* e *pitch* são convertidos para a unidade de graus (°). Portanto, emprega-se a técnica da fusão de sensores. Por meio desta técnica é possível obter ângulos (°) de rotações mais próximos dos reais. Kalman precisa de valores iniciais para executar seu algoritmo, como já mencionado, estes valores foram definidos na etapa de configuração do código. Além disso, ao usar o método é preciso informar o intervalo de tempo entre as execuções. Logo após a leitura dos valores e a estimativa dos ângulos a aceleração é convertida para a unidade de m/s^2 e é estabelecida a lógica para identificar um deslocamento. Somente durante a calibração que não é usada a aceleração gravitacional, no decorrer do *software* todas as acelerações são afetadas pela gravidade ($1G \approx 9.8m/s^2$).

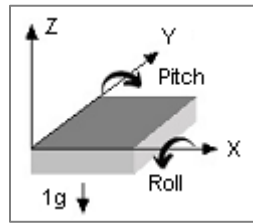


Figura 7. Ângulo de rotação de pitch e roll

2.3.3.1. Algoritmo para Detectar Deslocamento

Como pode ser observado na Figura 8 durante uma caminhada o corpo realiza um ciclo de movimentos para deslocar-se de um espaço a outro. Durante esses movimentos os braços movem-se de trás para frente e vice versa a fim de manter o equilíbrio corporal e conservar energia [Bruijn et al. 2010; Meyns, Bruijn e Duysens 2013].

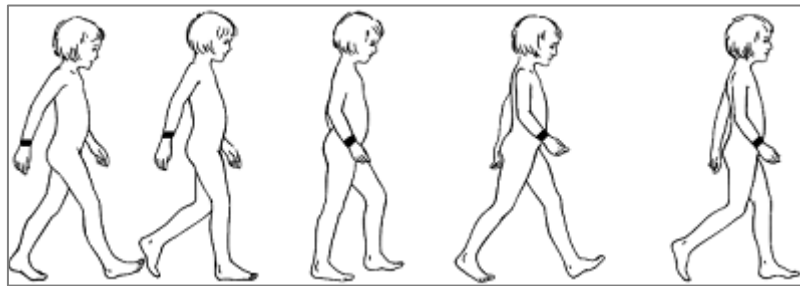


Figura 8. Ciclo da marcha humana
Fonte: Adaptado de Abu-Faraj et al. (2015)

Para gerenciar o deslocamento do usuário por meio de uma pulseira é preciso identificar um padrão de movimentação do braço durante uma caminhada. Por isso, a pulseira deve ser usada com o sentido dos eixos do sensor (MPU-6050) iguais ao da Figura 9 (é possível visualizar na montagem da pulseira que os eixos do sensor foram representados sobre a caixa de plástico que acoplou os equipamentos), pois com o sentido dos eixos conhecidos é possível estabelecer uma lógica para trabalhar com valores isolados e reconhecer um padrão de informação para avaliar um deslocamento.

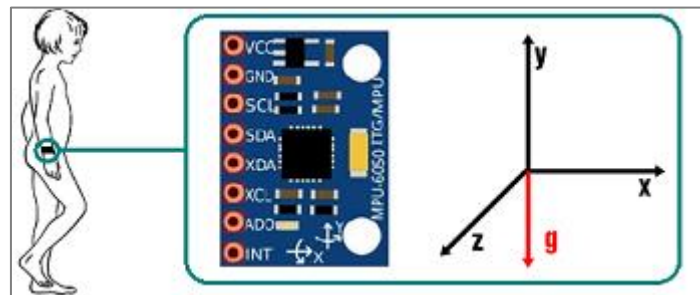


Figura 9. Orientação do sensor no pulso do usuário
Fonte: Adaptado de Abu-Faraj et al. (2015)

Para auxiliar na identificação de movimentação do braço é usada a fórmula do *Signal Vector Magnitude* (SVM) que permite obter a aceleração resultante dos três eixos presentes no acelerômetro [Fan et al. 2020]. O SVM (3) pode ser calculado da seguinte forma:

$$SVM = \sqrt{ax^2 + ay^2 + az^2} \quad (3)$$

No qual ax , ay e az referem-se aos valores x , y e z respectivamente do acelerômetro. Portanto, com o valor da aceleração resultante obtido é criado outro cálculo nomeado como *Signal Vector Magnitude Delta* (SVM Δ), que permite obter a variação de aceleração que existe dentro de um intervalo de tempo na movimentação do braço, como mostrado na Figura 10. Para calcular o SVM Δ é preciso subtrair o SVM atual pelo SVM anterior (armazenado em uma variável). É usado somente o módulo do SVM Δ .

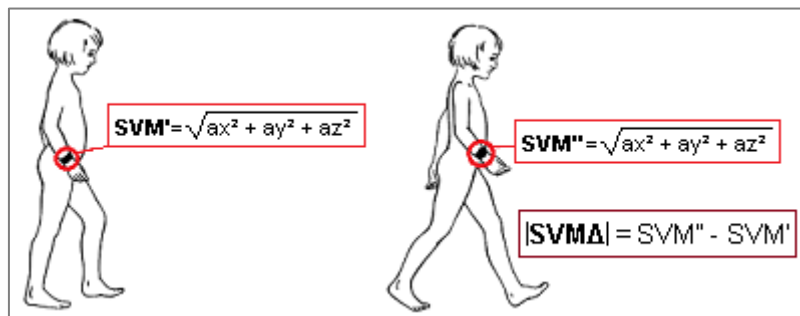


Figura 10. Cálculo do SVM Δ durante uma caminhada
Fonte: Adaptado de Abu-Faraj et al. (2015)

Para evitar que qualquer movimentação produza uma variação de aceleração (SVM Δ) e seja detectado um falso deslocamento é avaliada também a orientação do braço. Para isso, usam-se os sentidos e valores dos eixos do acelerômetro, bem como os ângulos de rotações em x e y obtidos por meio de Kalman. É possível visualizar na Figura 11 que ao movimentar o braço para frente do corpo todos os eixos do acelerômetro ficam positivos, isto se deve ao sentido dos eixos do sensor. Além disso, é possível perceber que ocorrem rotações nos eixos do sensor durante a movimentação do braço.

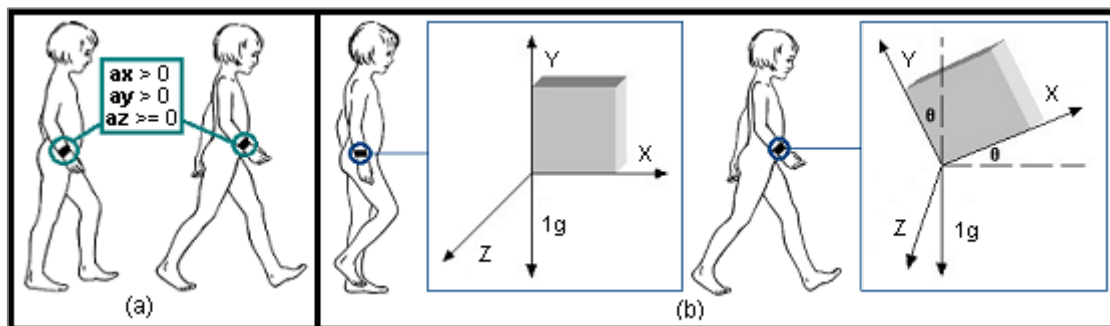


Figura 11. Valores dos eixos do acelerômetro (a) e ângulos de rotações durante uma caminhada (b)
Fonte: Adaptado de Abu-Faraj et al. (2015)

Foi criada uma técnica de intervalos para avaliar somente a caminhada do usuário e evitar que um falso deslocamento seja detectado quando o mesmo estiver deitado. Como demonstrado na Tabela 2 para aplicar esta técnica foram definidos intervalos de valores para os eixos x , y e z do acelerômetro e para os ângulos de rotações em x e y obtidos por meio de Kalman. Além disso, foi criado um valor de

referência nomeado como SVMΔ_REFERÊNCIA para avaliar a variação da aceleração. Este valor de referência foi definido como 3m/s².

Tabela 2. Intervalos definidos para detectar um deslocamento

Variável	Intervalo
ACELERAÇÃO_X (m/s ²)	0 < ACELERAÇÃO_X < 8
ACELERAÇÃO_Y (m/s ²)	6 < ACELERAÇÃO_Y < 13
ACELERAÇÃO_Z (m/s ²)	0 <= ACELERAÇÃO_Z < 4
KALMAN_ÂNGULO_X (°)	50 < KALMAN_ÂNGULO_X < 80
KALMAN_ÂNGULO_Y (°)	0 < KALMAN_ÂNGULO_Y < 30

Portanto, para detectar um deslocamento primeiramente é avaliado se SVMΔ_ATUAL (calculado no momento) e o SVMΔ_ANTERIOR (calculado anteriormente e armazenado em uma variável) são maiores que zero, pois caso não sejam significa que não existe uma variação de aceleração, portanto, o usuário está parado. Se o SVMΔ_ATUAL e o SVMΔ_ANTERIOR forem maiores que zero, são avaliados os valores isolados dos eixos x, y e z do acelerômetro, os valores devem estar dentro dos intervalos definidos, como demonstrados na Tabela 2. Posteriormente, são analisados os valores referentes aos ângulos de rotações de x e y, estes devem estar dentro dos intervalos definidos na Tabela 2. Por fim, é avaliado se o SVMΔ_ATUAL é maior ou igual ao SVMΔ_REFERÊNCIA (3m/s²) e se o SVMΔ_ANTERIOR é menor. É possível visualizar a lógica estabelecida na figura 12.

```

if((SVMΔ_ATUAL > 0)&&(SVMΔ_ANTERIOR > 0)){
//Usuário movimentando-se!
if((0 <= ACELERAÇÃO_Z < 4)&&(6 < ACELERAÇÃO_Y < 13)&&(0 < ACELERAÇÃO_X < 8)){
if((0 < KALMAN_ÂNGULO_Y < 30)&&(50 < KALMAN_ÂNGULO_X < 80)){
if ((SVMΔ_ATUAL >= 3)&&(SVMΔ_ANTERIOR < 3)){
//Deslocamento detectado!
}
}
}
} else {
//Usuário parado!
}

```

Figura 12. Algoritmo para identificar um deslocamento

Quando todos os valores estão dentro da lógica estabelecida, o deslocamento é detectado. Porém, antes de enviar uma nova mensagem ao tópico de deslocamento é avaliada se a última mensagem foi enviada há mais de um minuto ou se nenhuma foi enviada ainda, caso uma das condições seja satisfeita, a aplicação envia a nova mensagem com o ID da conta configurada de forma estática em uma variável no próprio código. O algoritmo usado na etapa de execução (*loop*) tem um *delay* de 100 milissegundos.

2.4. Medições Experimentais

Nesta pesquisa realizaram-se medições experimentais, portanto submeteu-se o projeto ao Comitê de Ética em Pesquisa da Universidade do Extremo Sul Catarinense (UNESC) tendo sido aprovado pelo parecer n° 4.242.538 (Anexo A). Para realizar os experimentos foi necessário calcular a amostra de participantes da pesquisa. O cálculo

do tamanho mínimo da amostra foi realizado por meio da fórmula (4) proposta por Barbetta, Reis e Bornia (2010):

$$n_0 = \frac{z_y^2 \times \sigma^2}{E_0^2} \quad (4)$$

Em que $y(1,96)$ é o nível de significância atrelado ao valor $z(1,96)$ bilateral da distribuição normal padronizada; σ é o desvio padrão obtido de forma estimada por uma amostra piloto, que resultou em 1,20 metros; E_0 é o erro amostral máximo tolerável pelo pesquisador, que é uma distância de 1,50 metros; e n_0 trata-se do tamanho mínimo da amostra, que resultou em três indivíduos.

Considerando-se que o sonambulismo é um evento que não pode ser previsto, simularam-se alguns episódios junto aos voluntários da pesquisa. Os critérios de inclusão destes voluntários consistiram em: ter mais de 18 anos, ser aluno da UNESC, acadêmico do curso de Ciência da Computação e estar cursando a disciplina de Trabalho de Conclusão de Curso (TCC) III. O espaço usado para testar a aplicação foi disponibilizado pelas clínicas integradas da própria universidade (UNESC). Cada voluntário participou da pesquisa executando quatro movimentações do corpo. Inicialmente o participante estava deitado em um objeto similar a uma cama, disponibilizado pela clínica de fisioterapia da universidade, posteriormente ficou em pé e começou a caminhar, por fim parou, simulando um episódio de sonambulismo. Foram realizadas cinco simulações por participante. Durante a execução dessas atividades o voluntário estava com a pulseira que monitorava sua movimentação. Ao identificar o deslocamento a pulseira comunicou-se com o servidor por meio do protocolo de aplicação MQTT e de comunicação sem fio *Wi-Fi*. Por fim, o servidor recebendo o evento salvou essa informação no banco de dados e notificou o aplicativo com a finalidade de alertar o deslocamento.

A conexão entre pulseira, servidor e aplicativo ocorreu por meio de uma rede local sem fio, criada com um *smartphone* Motorola, modelo *One*. Para possibilitar o acesso a Internet e salvar as informações dos deslocamentos no banco de dados (*NoSQL Realtime Database*) usou-se o 4G disponibilizado pela operadora telefônica. Criou-se esta rede, pois a disponibilizada pelo espaço só permitia acessar a Internet por meio da autenticação de um usuário vinculado a universidade, porém a conexão da pulseira com o *Wi-Fi* não permite fazer essa autenticação, somente são configurados nome e senha da rede. Portanto, com a rede local criada garantiu-se que todas as aplicações tivessem conexão com a Internet. Durante os testes o servidor ficou executando em um notebook Dell, modelo Inspiron 15 Série 5000, e usou-se um *smartphone* Motorola, modelo *Moto G4 plus*, para alertar os deslocamentos.

A avaliação do desempenho da aplicação baseou-se no tempo e na distância que foram obtidas para detectar um deslocamento. Para isso, foi usado o cronômetro do próprio *smartphone* que alertava o evento e duas trenas com 5 metros cada uma. O tempo começou a ser cronometrado a partir do instante em que o participante levantou-se e iniciou sua caminhada, e parou quando foi identificado um deslocamento, neste momento, por meio das trenas foi medida a distância alcançada. A Figura 13 mostra todos os passos realizados durante cada teste. Cada participante estava vinculado a uma conta distinta no aplicativo, por isso, para mudar a variável da conta no *script*, foi preciso executar o *sketch* (*script* do módulo sensor) a cada troca de voluntário, consequentemente, foi calibrado o sensor durante as mudanças.



Figura 13. Etapas do processo de coleta dos dados

Os dados coletados foram expressos por meio da mediana e amplitude (mínimo - máximo). As análises inferenciais foram realizadas com um nível de significância $\alpha = 0,05$ e, portanto, confiança de 95%. A investigação da distribuição das variáveis quantitativas (tempo e distância) quanto à normalidade por meio da aplicação do teste de Shapiro-Wilk. A comparação dos valores médios obtidos de tempo e distância entre os participantes da pesquisa foi realizada por meio da aplicação dos testes H de Kruskal-Wallis. O *software* usado para auxiliar nos cálculos foi o IBM Statistical Package for the Social Sciences (SPSS), versão 21.0.

3. Resultados e Discussões

O aplicativo demonstrou ser um sistema de monitoramento eficaz. As bibliotecas do Firebase facilitaram todo o processo de desenvolvimento, principalmente o serviço de autenticação dos usuários. Além disso, o banco (disponibilizado pela plataforma Firebase) selecionado permitiu trabalhar com a atualização dos dados em tempo real. No desenvolvimento da aplicação identificou-se um problema com a biblioteca Paho Android Service, pois quando o servidor não está ativo, gera-se um *exception* que não é possível tratar com o bloco de código *try/catch*, isso fecha a aplicação instantaneamente. Outro impasse que surgiu foi à forma de notificar o evento, pois o objeto nativo de *NotificationCompat* por padrão não emite som quando o dispositivo móvel está com a opção de não perturbe, por isso, foi usado o *MediaPlayer* que somente com a configuração de silêncio total o aplicativo não alerta o evento.

No desenvolvimento da pulseira a ideia era construir um equipamento mais perto do comercial possível, e como o sensor selecionado perde seus dados de configuração ao ser desligado, ele foi alimentado diretamente pela bateria e não esteve vinculado no circuito ao botão de retenção, a fim de que ao ligar e desligar a pulseira não fosse preciso calibrar e registrar os *offsets* novamente. Porém, aplicando esta técnica houve um problema vinculado ao registro dos dados de configuração no MPU-6050, pois se percebeu que quando alimentado durante muito tempo o sensor não permitia realizar os registros novamente mediante o *sketch*, por isso como destacado melhor na montagem do circuito, foi aplicado um botão de pulso para cortar a alimentação e fazer o *reset* do equipamento de forma manual, a fim de que se surgisse este problema durante os testes houvesse uma opção de limpar os dados. Posteriormente, percebeu-se que esta técnica de manter o sensor ligado diretamente com a bateria foi um esforço desnecessário, pois quando a bateria se esgota o sensor acaba perdendo seus dados.

O circuito eletrônico da aplicação tem aproximadamente 99mA de consumo quando a pulseira está ligada. Considerando os 630mAh fornecidos pela bateria e a potência que existe no circuito, a pulseira tem capacidade de manter-se ligada por um

período de cinco horas e trinta minutos. Para um serviço de monitoramento não é um resultado satisfatório, pois o ideal seria no mínimo um período de doze horas.

O sistema comportou-se bem durante os testes realizados. O servidor trabalhou de forma eficiente na conexão entre as aplicações, não houve nenhuma falha na troca de mensagens, o protocolo de aplicação selecionado além de ser fácil de empregar, funcionou corretamente. O aplicativo alertou todos os deslocamentos detectados, porém era perceptível que como a notificação e o som são objetos nativos diferentes, existia um pequeno *delay* entre a visualização e a escuta do alerta. A rede local sem fio com acesso à Internet por meio do 4G não influenciou no desempenho da aplicação, pois as operações de acesso ao banco de dados demonstraram serem leves. Durante a troca de voluntários a pulseira não apresentou nenhum problema na execução do *sketch*, portanto, não foi preciso usar o botão de pulso.

Os dados coletados estão representados na Tabela 3. É possível visualizar que o tempo máximo para detectar um evento entre todos os voluntários foi de 8,23s (voluntário 3) e a distância máxima foi de 5,70m (voluntário 2). Durante os testes a única restrição que foi imposta aos participantes era que não corressem, mas que praticassem uma caminhada do seu dia a dia.

O estudo de Bohannon e Williams Andrews (2011) aplicou uma meta-análise em uma base de dados com velocidades de caminhada de vinte e três mil indivíduos saudáveis de quarenta e um estudos diferentes. Mediante a metodologia estatística empregue foi possível calcular e agrupar a velocidade média de uma marcha conforme o sexo e idade dos indivíduos. O grupo de voluntários da presente pesquisa contou somente com pessoas do sexo masculino e com faixa etária entre 20 a 29 anos. Conforme a pesquisa de Bohannon e Williams Andrews (2011) a velocidade média de deambulação desses participantes seria de 1,35m/s. O intervalo de velocidade estimada pelo estudo seria entre 1,27m/s a 1,44m/s. Avaliando-se as velocidades (Tabela 3) é possível visualizar que em geral os deslocamentos foram detectados antes dos indivíduos aproximarem-se da velocidade média supracitada.

Tabela 3. Resultados da coleta dos dados

	Tempo(s)	Distância(m)	Velocidade(m/s)
Voluntário 1			
1ª Simulação	2,06	1,96	0,95
2ª Simulação	2,17	2,24	1,03
3ª Simulação	4,73	3,67	0,77
4ª Simulação	3,28	2,41	0,73
5ª Simulação	4,17	2,99	0,71
Voluntário 2			
1ª Simulação	6,64	5,70	0,85
2ª Simulação	1,71	2,17	1,26
3ª Simulação	3,39	4,05	1,19
4ª Simulação	2,95	3,21	1,08
5ª Simulação	5,33	4,42	0,82
Voluntário 3			
1ª Simulação	3,53	3,40	0,96
2ª Simulação	3,51	2,75	0,78
3ª Simulação	3,46	2,40	0,69

4ª Simulação	3,60	2,25	0,62
5ª Simulação	8,23	5,53	0,67

Fonte: Dados da pesquisa, 2020.

Foram realizados os cálculos de mediana e amplitude nos dados coletados (Tabela 4), é possível perceber que o valor-p do teste aplicado ficou maior que 0,05, o que indica que não houve diferenças estatisticamente significativas entre os voluntários, tanto no tempo, quanto na distância, mostrando homogeneidade quanto a essas variáveis. É possível perceber que a mediana de tempo entre os voluntários aproximaram-se, em contrapartida, a mediana da distância do voluntário 2 apresentou uma diferença significativa em relação aos voluntários 1 e 3.

Segundo os resultados alcançados com os dados coletados e o erro amostral (1,50 metros) há evidências que a aplicação tem bons números em quartos tradicionais com mais de 9m², no qual é possível estimar 1,50m da cama até a porta e 1,50m da cama até a janela, resultando em uma parede com 3m de comprimento.

Tabela 4. Resultados de mediana e amplitude dos dados coletados

	Mediana (Mínimo - Máximo)	Valor-p
Tempo(s)		
Voluntário 1	3,28 (2,06 – 4,73)	0,512
Voluntário 2	3,39 (1,71 – 6,64)	
Voluntário 3	3,53 (3,46 – 8,23)	
Distância(m)		
Voluntário 1	2,41 (1,96 – 3,67)	0,298
Voluntário 2	4,05 (2,17 – 5,70)	
Voluntário 3	2,75 (2,25 – 5,53)	

Fonte: Dados da pesquisa, 2020.

O algoritmo empregue nesta pesquisa demonstrou capacidade de monitorar a movimentação do usuário e detectar seu deslocamento por meio de uma pulseira. Porém, a técnica de intervalos usada diminuiu a possibilidade de identificar outra movimentação do corpo, como por exemplo, quando o usuário está correndo, já que o movimento do braço difere-se de uma caminhada. Os resultados obtidos confirmam que durante uma caminhada as pessoas movem seus braços para frente e para trás, e vice versa, pois, foi possível detectar todos os deslocamentos. Durante as simulações não houve nenhum obstáculo para dificultar a deambulação do usuário, como geralmente existe nas casas (porta do quarto), isto pode ter influenciado positivamente nos resultados.

Foi possível perceber durante o desenvolvimento do algoritmo que o pulso é um local do corpo que sofre muita movimentação, portanto, é um local complexo de se trabalhar na monitoração de movimentos do corpo humano. O trabalho de Nguyen, Fan e Shahabi (2015) usou uma base de dados retirada da *Activity Recognition Challenge* que continha dados de um acelerômetro aplicado no pulso esquerdo e direito de três indivíduos realizando suas atividades matinais. O objetivo da pesquisa era avaliar essa base de dados por meio de algoritmos com *Machine Learning* (ML) para reconhecer as deambulações dos usuários. Os algoritmos de ML aplicados permitiram além de reconhecer as caminhadas ampliar o grau de classificação das movimentações do corpo em: ficar em pé, sentar e deitar, classificações que se necessárias o algoritmo empregue na presente pesquisa não seria capaz de fazer de forma prática e precisa. Os principais

algoritmos de ML usados no trabalho de Nguyen, Fan e Shahabi (2015) foram: *Naive Bayes*, *Support vector machine*, *Decision Tree*, *Multilayer Perceptron*, k-NN e *Random Forest*. WEKA foi a ferramenta usada para empregar os algoritmos na base de dados. Os resultados demonstraram que o *Random Forest* atingiu 91,92% de precisão na classificação das movimentações do corpo.

O algoritmo empregue na pesquisa de Li, Hou e Huang (2013) para classificar atividades humanas em: caminhar, correr, ficar parado e detectar quedas, usou o filtro de Kalman e o *Signal Vector Magnitude* (SVM) assim como a presente pesquisa, porém com técnicas diferentes, demonstrando que o filtro e o SVM podem ser aplicados em diversos cenários. O algoritmo não usou ML para classificar as atividades humanas, porém como se desenvolveu um código mais complexo seu processamento foi feito em um *smartphone*. Os dados eram enviados ao dispositivo móvel por meio do modelo de comunicação sem fio *Bluetooth*. Comparando-se o trabalho de Li, Hou e Huang (2013) com a presente pesquisa, percebe-se que a localização do equipamento no corpo humano influencia na seleção do sensor e nas técnicas estabelecidas para monitorar as movimentações do corpo, como no trabalho de Li, Hou e Huang (2013) o dispositivo poderia ser posicionado no tórax ou na cintura, lugares que não sofrem tanta movimentação quanto o pulso, usaram somente um acelerômetro, diferente da presente pesquisa que além do acelerômetro precisou-se de um giroscópio para auxiliar na avaliação da orientação do braço. Por mais que tenham sido empregados diferentes métodos para conhecer as atividades do corpo, comparando-se os resultados é possível visualizar que ações como caminhar e ficar parado foram reconhecidas em ambos os trabalhos.

O artigo publicado por Mano et al. (2016) aplicou sensores inerciais (acelerômetro e giroscópio) no antebraço de pessoas idosas a fim de monitorar suas movimentações (andando, correndo, sentado, deitado e em pé) e identificar possíveis quedas, notificando-as em um dispositivo móvel. Similar a presente pesquisa foram definidos três módulos, são eles: sensor, servidor e móvel. Porém, a estrutura da aplicação diferenciou-se da presente pesquisa, não se usou o protocolo MQTT, pois o processamento dos dados era realizado no módulo sensor somente quando a carga de trabalho no módulo era baixa, caso a carga fosse alta os dados eram enviados ao módulo servidor para serem avaliados, por isso foi usado o protocolo *Network File System* (NFS), a fim de que os dados do sensor fossem transmitidos ao módulo servidor. Se a carga de trabalho continuasse alta no módulo servidor, os dados eram enviados por meio do protocolo *SSH File Transfer Protocol* (SFTP) a um serviço *web* para serem avaliados. O módulo sensor era formado por um *RaspBerry Pi*, o que permitiu trabalhar com o protocolo NFS. É possível perceber que a forma e o local onde são processados os dados interferem diretamente na seleção do protocolo de comunicação. Para a presente pesquisa foi selecionado o protocolo MQTT, pois é leve e rápido, comportando-se bem no transporte das mensagens desta pesquisa, pois são enviadas mensagens simples e em longos intervalos de tempo, não existe uma grande massa de dados sendo enviada continuamente como da aplicação de Mano et al. (2016), por isso não se pode afirmar que o MQTT se comportaria bem em um cenário como o de Mano et al. (2016).

O trabalho de Singhal e Jain (2015) apresenta o mesmo propósito da presente pesquisa, que é apoiar sonâmbulos. Para isso eles usaram um sensor de presença (infravermelho) que aciona uma campainha ao detectar uma aproximação, realizando

ligação e enviando mensagem para números cadastrados, além disso, usaram um acelerômetro na cintura dos indivíduos para identificar suas quedas. Uma das poucas comparações que é possível fazer entre a presente pesquisa e o trabalho de Singhal e Jain (2015) são os caminhos distintos que podem ser percorridos para solucionar um problema em comum, é possível demonstrar que os sensores inerciais e o sensor de presença que são equipamentos totalmente diferentes podem ser usados para finalidades semelhantes.

4. Conclusão

Este trabalho aplicou os conceitos de Internet das Coisas em um protótipo de pulseira que possibilita gerenciar o deslocamento de sonâmbulos em um dispositivo móvel. A detecção e notificação em tempo real dos episódios de sonambulismo pode permitir um rápido atendimento ao sonâmbulo, aumentando a sensação de segurança e reduzindo a possibilidade de acidentes. A aplicação demonstrou ser um sistema flexível e prático de usar, bastando apenas de uma rede *Wi-Fi* para conectar todos os dispositivos, além disso, possui baixa complexidade computacional. Os resultados alcançados confirmaram a eficácia no uso de sensores inerciais para monitorar os movimentos humanos. Por meio de variáveis como tempo (segundos), distância (metros) e erro amostral (metros) evidenciou-se que a aplicação apresentou resultados satisfatórios em quartos tradicionais com mais de 9m².

O algoritmo empregue demonstrou ser capaz de monitorar pessoas a fim de identificar seu caminhar e conseqüentemente seu deslocamento. Porém, a técnica aplicada limitou o reconhecimento de outras movimentações do usuário (correndo, sentado, em pé e deitado). O pulso comprovou ser um local complexo de trabalhar com sensores inerciais.

Embora os resultados tenham sido positivos na detecção de deslocamentos mediante caminhadas, para aplicar esta pesquisa em cenários reais novas medições experimentais devem ser feitas a fim de avaliar o comportamento do algoritmo quando o usuário permanece deitado por um longo período de tempo.

Com base nos conhecimentos adquiridos, bem como nos resultados obtidos, propõe-se para futuros trabalhos: utilizar novas plataformas de prototipagem e sensores inerciais; compactar mais a pulseira; usar uma bateria com maior valor de corrente; empregar algoritmos de classificação com *Machine Learning* para reconhecer as movimentações do corpo a fim de gerenciar o deslocamento do usuário; aplicar filtros complementares nos dados do sensor para aumentar sua precisão; realizar experimentos com obstáculos no percurso dos indivíduos; aplicar experimentos com os participantes deitados em uma cama para avaliar o desempenho do algoritmo.

Referências

- Abu-Faraj, Ziad O., Harris, Gerald F., Smith, Peter A., Hassani, Sahar. (2015). Human Gait and Clinical Movement Analysis.
- Badawi, Abeer A., Al-Kabbany, Ahmad., Shaban, Heba A. (2020). Sensor Type, Axis, and Position-Based Fusion and Feature Selection for Multimodal Human Daily Activity Recognition in Wearable Body Sensor Networks. *Journal of Healthcare Engineering*, v. 2020.

- Barbetta, Pedro Alberto., Reis, Marcelo Menezes., Bornia, Antonio Cezar. (2010). Estatística: para cursos de engenharia e informática.
- Bohannon, Richard. W. e Williams Andrews, A. (2011). Normal walking speed: A descriptive meta-analysis. *Physiotherapy*, v. 97, n. 3, p. 182–189.
- Bruijn, Sjoerd M., Meijer, Onno G., Beek, Perter J., Van Dieen, Jaap H. (2010). The effects of arm swing on human gait stability. *Journal of Experimental Biology*, v. 213, n. 23, p. 3945–3952.
- Dian, F. John., Vahidnia, Reza., Rahmati, Alireza. (2020). Wearables and the Internet of Things (IoT), Applications, Opportunities, and Challenges: A Survey. *IEEE Access*, v.8, p.69200-69211.
- Fan, Yimeng., Jin, Hao., Ge, Yongzhe., Wang, Nan. (2020). Wearable Motion Attitude Detection and Data Analysis Based on Internet of Things. *IEEE Access*, v. 8, p.1327–1338.
- Jobanputra, Charmi., Bavishi, Jatna., Doshi, Nishant. (2019). Human activity recognition: A survey. *Procedia Computer Science*, v. 155, p. 698–703.
- Kurniawan, A., Hermawan, A. R., Purnama, I.K.E. (2016). A wearable device for fall detection elderly people using tri dimensional accelerometer. *Proceeding - 2016 International Seminar on Intelligent Technology and Its Application, ISITIA 2016: Recent Trends in Intelligent Computational Technologies for Sustainable Energy*, p. 671–674.
- Li, Na., Hou, Yibin., Huang, Zhangqin. (2013). Implementation of a Real-Time Human Activity Classifier Using a Triaxial Accelerometer and Smartphone. *International Journal of Advancements in Computing Technology*, v. 5, n. 4, p. 234–242.
- Mano, Leandro Yukio., Funes, Marcio Maestralo., Volpato, Tiago., Torres Neto, José Rodrigues. (2016). Explorando tecnologias de IoT no contexto de Health Smart Home: uma abordagem para detecção de quedas em pessoas idosas. *Journal on Advances in Theoretical and Applied Informatics*, v. 2, n. 1, p. 46-57.
- Meyns, Pieter., Bruijn, Sjoerd M., Duysens, Jacques. (2013). The how and why of arm swing during human walking. *Gait and Posture*.
- Nguyen, Minh., Fan, Liyue., Shahabi, Cyrus. (2015). Activity Recognition Using Wrist-Worn Sensors for Human Performance Evaluation. *Proceedings - 15th IEEE International Conference on Data Mining Workshop, ICDMW 2015*, p. 164–169.
- Nunes, Magda Lahorgue. (2002). Distúrbios do sono. *Jornal de Pediatria*, p. 63-72.
- Oliveira, Waldri Dos Santos. e Gonçalves, Eduardo Nunes. (2017). Implementação em c: filtro de kalman, fusão de sensores para determinação de ângulos. *ForScience*, v. 5, n. 3, p. 1–18.
- Singhal, Smriiti. e Jain, P. C. (2015). Wireless health monitoring system for sleepwalking patients. *Proceedings of the 2015 39th National Systems Conference, NSC 2015*.
- Verma, Parul., Rajnish, Ranjana., Fatima, Shahnaz. (2017). Challenges: Wearable Computing for Internet of Things (IoT). *International Journal of Science and Research (IJSR)*, v. 6, p. 1306-1311.

ANEXO A - Parecer do Comitê de Ética em Pesquisa de Seres Humanos



CEP

COMITÊ DE ÉTICA EM PESQUISA
DE SERES HUMANOS



RESOLUÇÃO

O Comitê de Ética em Pesquisa da UNESC, reconhecido pela Comissão Nacional de Ética em Pesquisa (CONEP)/ Ministério da Saúde analisou o projeto abaixo:

Parecer n.: 4.242.538

CAAE: 36686920.8.0000.0119

Pesquisador(a) Responsável: Luciano Antunes

Pesquisador(a): DARLAN DOMINGOS CANDIOTTO

Título: "WEARABLE BASEADO EM INTERNET DAS COISAS PARA MONITORAMENTO DE SONÂMBULOS".

Este projeto foi aprovado em seus aspectos éticos e metodológicos, de acordo com as Diretrizes e Normas Internacionais e Nacionais. Todas e qualquer alteração do Projeto deverá ser comunicada ao CEP. Os membros do CEP não participaram do processo de avaliação dos projetos onde constam como pesquisadores.

Criciúma, 27 de agosto de 2020.


Marco Antônio da Silva
Coordenador do CEP

Av. Universitária, 1.105 – Bairro Universitário – CEP: 88.806-000 – Criciúma / SC
Bloco Administrativo – Sala 21 | Fone (48) 3431 2606 | cetica@unesc.net | www.unesc.net/cep
Horário de funcionamento do CEP: de segunda a sexta-feira, das 08h às 12h e das 13h30 às 17h30.