

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA
GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO



**SOLUÇÃO PARA DETECÇÃO E ALERTA DE QUEDAS
COM HARDWARE DE BAIXO CUSTO**

WELLINGTON ALVES DOS SANTOS

GOIÂNIA
2022

WELLINGTON ALVES DOS SANTOS

**SOLUÇÃO PARA DETECÇÃO E ALERTA DE QUEDAS
COM HARDWARE DE BAIXO CUSTO**

Trabalho de Conclusão de Curso apresentado à Escola Politécnica, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador(a):

Profª. Ma. Ludmilla Reis Pinheiro dos Santos

Banca examinadora:

Prof. Me. Geraldo Valeriano Ribeiro

Prof. Me. Rafael Leal Martins

GOIÂNIA

2022

WELLINGTON ALVES DOS SANTOS

**SOLUÇÃO PARA DETECÇÃO E ALERTA DE QUEDAS
COM HARDWARE DE BAIXO CUSTO**

Trabalho de Conclusão de Curso aprovado em sua forma final pela Escola Politécnica, da Pontifícia Universidade Católica de Goiás, para obtenção do título de Bacharel em Engenharia de Computação, em ____/____/_____.

Profª. Ma. Ludmilla Reis Pinheiro dos Santos
Coordenador(a) de Trabalho de Conclusão de Curso

Banca examinadora:

Orientador(a): Ma. Ludmilla Reis Pinheiro dos Santos

Prof. Me. Geraldo Valeriano Ribeiro

Prof. Me. Rafael Leal Martins

GOIÂNIA

2022

A Deus pela minha vida e oportunidades.

A minha família que me apoiaram e tornaram possíveis meus estudos.

A todos os professores que me ensinaram e orientaram para o caminho certo.

AGRADECIMENTOS

A Professora Ludmilla Reis Pinheiro dos Santos, orientadora acadêmica, pelo apoio e confiança depositada.

À Coordenação da Escola Politécnica, da Pontifícia Universidade Católica de Goiás por ajudar, de forma inestimável, na execução deste projeto.

Aos meus pais que me apoiaram e tornaram possíveis meus estudos.

Aos meus colegas Gabriel Teixeira, Gabriel Mendes, Gabriel Cardoso, João Xavier e Walcy Santos pelas discussões técnicas e inestimáveis sugestões.

A todos que direta ou indiretamente colaboraram para materialização deste trabalho.

“Motivação é a arte de fazer as pessoas fazerem o que
você quer que elas façam porque elas o querem fazer.”

Dwight Eisenhower

RESUMO

Estudos indicam que uma em cada três pessoas com mais de 65 anos de idade, caem a cada ano. As estatísticas mostram que as consequências mais graves não são resultado direto da queda, mas a falta de assistência e tratamento rápido. Este trabalho tem como objetivo criar um protótipo de baixo custo capaz de realizar leituras de aceleração do usuário e comunicar, através de um *smartphone*, possíveis eventos que caracterize uma queda. Se uma queda for detectada, o *smartphone* através de uma aplicação Flutter, envia uma mensagem de alerta para os contatos de emergência, via e-mail e SMS, por fim uma chamada de voz é realizada para o primeiro contato da lista. O conteúdo das mensagens inclui nome do usuário, hora e data da queda e instruções para que entre em contato ou solicite socorro imediatamente. Para criar o protótipo foi utilizado um microcontrolador ESP32 juntamente com um sensor acelerômetro MPU-6050. O algoritmo de detecção de quedas foi desenvolvido através de análises empíricas das leituras de aceleração de várias simulações de quedas em diferentes direções, utilizando como base a soma vetorial dos três eixos de aceleração do sensor. Para mitigar a ocorrência de falsos positivos, outras atividades como caminhar, correr e subir escadas foram testadas e não são identificadas como uma queda. O uso do dispositivo permite prestar socorro às vítimas de maneira rápida, aumentando as chances de sobrevivência do público-alvo.

Palavras-Chave: ESP32, MPU-6050, Flutter, Detecção de quedas, Alerta.

ABSTRACT

Studies indicate that one in three people over the age of 65 falls each year. Statistics show that the most serious consequences are not a direct result of the fall, but the lack of assistance and prompt treatment. This work aims to create a low-cost prototype capable of performing user acceleration readings and communicating, through a smartphone, possible events that characterize a fall. If a fall is detected, the smartphone, through a Flutter application, sends an alert message to the emergency contacts, via email and SMS, finally a voice call is made to the first contact on the list. The content of the messages includes the user's name, time and date of the fall, and instructions to contact or request assistance immediately. To create the prototype, an ESP32 microcontroller was used together with an MPU-6050 accelerometer sensor. The fall detection algorithm was developed through empirical analysis of the acceleration readings of several simulations of falls in different directions, based on the vector sum of the sensor's three acceleration axes. To mitigate the occurrence of false positives, other activities such as walking, running, and climbing stairs were tested and are not identified as a fall. The use of the device makes it possible to provide help to victims quickly, increasing the chances of survival of the target audience.

Keywords: *ESP32, MPU-6050, Flutter, Falls, Elderly, Alert, Emergency.*

LISTA DE FIGURAS

Figura 1 - Segmentação de uma pessoa em queda	19
Figura 2 - Fluxograma do projeto	24
Figura 3 - Diagrama de blocos do projeto	25
Figura 4 - MPU-6050	26
Figura 5 – Exemplo de comunicação I2C	27
Figura 6 – ESP-32	28
Figura 7 – Pinagem ESP-32	29
Figura 8 – Esquema de comunicação ESP-32 – MPU-6050	30
Figura 9 – Protótipo	30
Figura 10 – Tela inicial	32
Figura 11 – Menu de configurações	33
Figura 12 – Configurações <i>Bluetooth</i>	34
Figura 13 – Configurações de E-mail	35
Figura 14 – Tela de contatos	36
Figura 15 – Tela de alerta	37
Figura 16 – E-mail de alerta	38
Figura 17 – SMS de alerta	39
Figura 18 – Gráfico andando	40
Figura 19 – Gráfico correndo	41
Figura 20 – Gráfico subindo escadas	42
Figura 21 – Gráfico descendo escadas	42
Figura 22 – Ilustração de queda frontal	43
Figura 23 – Gráfico de queda frontal	43
Figura 24 – Ilustração de queda de costas	43

Figura 25 – Gráfico de queda de costas 44

Figura 26 - Visualização da queda do protótipo 44

LISTA DE SIGLAS

BLE	<i>Bluetooth Low Energy</i>
GPIO	<i>General Purpose Input/Output</i>
I2C	<i>Inter-Integrated Circuit</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
IoT	<i>Internet of Things</i>
IP	<i>Internet Protocol</i>
MEMS	<i>Micro-Electro-Mechanical System</i>
MQTT	<i>Message Queue Telemetry Transport</i>
OMS	Organização Mundial de Saúde
PC	Computador pessoal
SCL	<i>Serial Clock</i>
SDA	<i>Serial Data</i>
SDK	<i>Software Development Kit</i>
SMS	<i>Short Message Service</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
SOC	<i>System on a Chip</i>
TCP	<i>Transmission Control Protocol</i>
UI	<i>User Interface</i>
USB	<i>Universal Serial Bus</i>

SUMÁRIO

1	INTRODUÇÃO.....	14
1.1	Objetivos gerais	15
1.2	Objetivos específicos.....	15
1.3	Materiais e métodos.....	15
1.4	Metodologia aplicada e experimental.....	16
1.5	Justificativa.....	17
2	REFERENCIAL TEÓRICO.....	18
2.1	Métodos de detecção de quedas.....	18
2.1.1	Método baseado em câmera	18
2.1.2	Método baseado em acústica	20
2.1.3	Método baseado em cinemática.....	21
2.2	Análise dos trabalhos relacionados.....	22
2.3	Framework Flutter	22
2.4	Bibliotecas Flutter	23
2.4.1	FlutterBlue	23
2.4.2	Mailer	23
2.4.3	Telephony	23
3	DESENVOLVIMENTO.....	24
3.1	Proposta de solução	24
3.2	Componentes de Hardware.....	26
3.2.1	MPU-6050	26
3.2.2	Interface I2C	27
3.2.3	ESP32	28
3.3	Bluetooth Low Energy	28
3.4	Construção do protótipo	29
3.5	Firmware de testes	31
4	APLICAÇÃO MÓVEL	32
4.1	Tela inicial	32
4.2	Tela de Configurações	33
4.2.1	Configuração do Bluetooth.....	34
4.2.2	Configuração de E-mail.....	35
4.3	Tela de Contatos	36
4.4	Tela de Alerta	37

4.5 Mensagem de alerta	38
4.5.1 E-mail	38
4.5.2 SMS	39
5 TESTES E RESULTADOS	40
5.1 Testes do protótipo	40
5.1.1 Andando	40
5.1.2 Correndo	41
5.1.3 Subindo e descendo escadas	41
5.1.4 Queda Frontal	42
5.1.5 Queda de costas	43
5.1.6 Visualização da queda do protótipo.....	44
5.2 Implementação do firmware	45
6 CONSIDERAÇÕES FINAIS	46
6.1 Sugestões de trabalhos futuros	47
REFERÊNCIAS	48
APÊNDICE A – FIRMWARE DE TESTES	51
APÊNDICE B – CÓDIGO PYTHON PARA GERAR OS GRÁFICOS.....	52
APÊNDICE C – TERMO DE AUTORIZAÇÃO DE PUBLICAÇÃO DE PRODUÇÃO ACADEMICA.....	53

1 INTRODUÇÃO

Estudos realizados pelo Instituto Brasileiro de Geografia e Estatística (IBGE), mostraram que no início de 2021 o Brasil possuía uma população de mais de 37 milhões de pessoas com 60 anos ou mais, a tendência é de que esse número aumente de forma significativa nos próximos anos (SOUZA; BOSCO; FIGUEIREDO, 2021).

Segundo informações da pesquisa ConVid - Pesquisa de Comportamentos, organizada pela Fundação Oswaldo Cruz, em parceria com a Universidade Federal de Minas Gerais e a Universidade Estadual de Campinas, no período de distanciamento social/quarentena devido à pandemia em 2020, o distanciamento social total foi adotado por 30,9% dos idosos e 56,9% fizeram distanciamento social intenso (ELENA; RODRIGUES, 2021).

Com a população idosa crescendo ao longo dos últimos anos, é inevitável que pessoas com idade avançada e limitações físicas causadas pelo envelhecimento, tenham que viver sozinhas e sem a companhia diária de uma pessoa mais jovem.

De acordo com a Organização Mundial de Saúde, aproximadamente 28-35% das pessoas com 65 anos ou mais sofrem quedas a cada ano, aumentando para 32-42% dos idosos com 70 anos ou mais, esse número tende a aumentar de forma exponencial com o avanço da idade (OMS, 2007). Entre 5-10% das quedas resultam em fraturas, e 90% das fraturas são resultados de quedas (PEETERS; SCHOOR; LIPS, 2009).

Justifica-se estudar este tema porque, com a análise desse problema social, é importante garantir o bem-estar e a qualidade de vida da população idosa em seu dia a dia, utilizando técnicas e conhecimentos para a criação de dispositivos envolvendo sistemas inteligentes, capazes de monitorar e informar sobre o estado do usuário.

Na ocorrência de uma queda sofrida por uma pessoa idosa, o tempo que se leva para realizar o amparo e o atendimento, é vital para reduzir o risco de sequelas e aumentar drasticamente a chance de sobrevivência da vítima. Soluções para detecção de quedas já existem no mercado, implementadas em dispositivos como o *Apple Watch* ou soluções proprietárias de fornecedores de serviços médicos particulares, porém são soluções de alto custo e que muitas vezes acompanham outros serviços (APPLE BRASIL, 2022)

Com o advento de novas tecnologias embarcadas, juntamente com o avanço da *Internet of Things* (IoT), pesquisas na área de detecção de quedas integrado com um

sistema de alerta para contatos de emergência tem se tornado cada vez mais populares, viabilizando soluções acessíveis e de baixo custo.

1.1 Objetivos gerais

Este trabalho objetiva criar uma solução de baixo custo que viabiliza a detecção e alerta de quedas em conjunto com uma aplicação mobile. Tal solução tem o objetivo de informar os contatos de emergência para realizar a prestação de socorro à vítima, de forma rápida e eficiente, maximizando as chances de sobrevivência e reduzindo o risco de sequelas.

1.2 Objetivos específicos

- Estudar a aplicação de um microcontrolador para monitoramento de sensores com taxas de amostragem de 100 Hz;
- Estudar protocolos de comunicação *Bluetooth Low Energy* (BLE) com dispositivos móveis;
- Estudar a tecnologia de desenvolvimento multiplataformas utilizando o *Framework Flutter*;
- Desenvolver um protótipo de baixo custo para detecção de quedas;
- Desenvolver um aplicativo mobile para realizar a comunicação com o protótipo;
- Enviar um alerta aos contatos de emergência através da aplicação mobile;
- Utilizar um sensor acelerômetro de 3 eixos para leitura de dados do usuário.

1.3 Materiais e métodos

A primeira etapa do trabalho foi encontrar os componentes necessários e estudar diferentes conceitos e teorias relacionados ao mesmo. Além disso, foram realizadas pesquisa em trabalhos relacionados para saber como integrar os diferentes tipos de componentes. O protótipo para detecção de quedas utilizou um microcontrolador ESP32 com Wi-Fi e Bluetooth, integrado a um sensor acelerômetro de 3 eixos MPU-6050. O preço encontrado dos componentes foram R\$35,50 e R\$11,95 respectivamente, totalizando R\$47,45.

1.4 Metodologia aplicada e experimental

As metodologias adotadas neste trabalho foram a Pesquisa aplicada e a Pesquisa experimental. A metodologia da pesquisa aplicada consiste na aplicação prática da ciência para encontrar soluções úteis de problemas do cotidiano, geralmente direcionadas para problemas práticos. Já a metodologia da pesquisa experimental se caracteriza por manipular diretamente variáveis relacionadas com o objeto de estudo e tem como finalidade testar hipóteses que dizem respeito à convicção de quem está pesquisando (TUMELERO, 2019).

Conforme TUMELERO (2019), as características da pesquisa aplicada são:

- Gerar conhecimentos para a aplicação prática dirigida à solução de problemas;
- Obter conhecimento que será usado a curto ou médio prazo;
- É uma investigação original concebida pelo interesse em adquirir novos conhecimentos;
- Envolver conhecimento disponível e sua ampliação aplicada ao conhecimento, visando utilidade econômica e social;
- Buscar o conhecimento pela apropriação do “know-how” e/ou patentes.

Já as características da pesquisa experimental são:

- Manipular pelo menos uma das características dos elementos estudados;
- Introduzir um ou mais controles na situação experimental, sobretudo criando um grupo de controle;
- A designação dos elementos para participar dos grupos experimentais e de controle devem ser feitas aleatoriamente;
- Formulação de hipóteses a serem testadas;
- Executar o experimento e observar a saída.

Enquanto a pesquisa básica é uma pesquisa científica focada na melhoria das teorias científicas, a pesquisa aplicada consiste na realização de trabalhos originais com finalidade de obter novos conhecimentos, dirigida para um determinado fim ou objetivo prático. A pesquisa aplicada consiste em trazer novas ideias, criar um produto ou um processo e encontrar a solução para um problema concreto (TUMELERO, 2019).

1.5 Justificativa

Esse tema foi escolhido devido ao atual cenário de crescimento da população idosa do Brasil, as estatísticas mostram que 28 a 35% das pessoas com 65 anos ou mais sofrem quedas a cada ano, entre 5 e 10% das quedas resultam em fraturas, sendo que 90% das fraturas são resultadas de quedas, portanto ocorrências de quedas e fraturas tendem a ser cada vez mais frequentes.

Este trabalho se justifica na intenção de aumentar a qualidade de vida e as chances de sobrevivência, emitindo um alerta para os contatos de emergência em uma eventual queda do usuário. O trabalho visou construir um dispositivo de baixo custo para realizar a detecção de quedas, utilizando um smartphone para alertar o evento através da internet ou rede celular.

2 REFERENCIAL TEÓRICO

Nesta seção serão abordados os trabalhos relacionados e os conceitos de diversas áreas importantes para dar andamento no desenvolvimento deste trabalho.

Quedas são um dos principais problemas médicos enfrentados por pessoas idosas. As consequências de uma queda podem ser fatais, se não houver atendimento médico imediato. Contratar uma enfermeira ou um cuidador pode ser uma das melhores opções para monitorar e dar suporte constante para as atividades diárias das pessoas idosas, porém em alguns casos, a família não pode contratar tais serviços por falta de condições financeiras.

Uma queda pode ocorrer quando os cuidadores ou familiares não estiverem por perto. Portanto, um sistema de detecção de quedas confiável e de baixo custo pode ser considerado uma opção para monitorar e dar suporte às pessoas idosas.

Atualmente, existem três métodos que são utilizadas para desenhar sistemas de detecção de quedas, que são: método baseado em câmera, método baseado em acústica e o método baseado em cinemática. Cada método possui suas próprias vantagens e desvantagens. Todas essas diferentes abordagens auxiliam na assistência de enfermeiros, cuidadores e familiares.

2.1 Métodos de detecção de quedas

Esta seção apresenta os diferentes métodos de detecção de quedas que são mais aplicados em projetos semelhantes, serão descritos as vantagens e desvantagens de cada método e como o mesmo pode ser abordado.

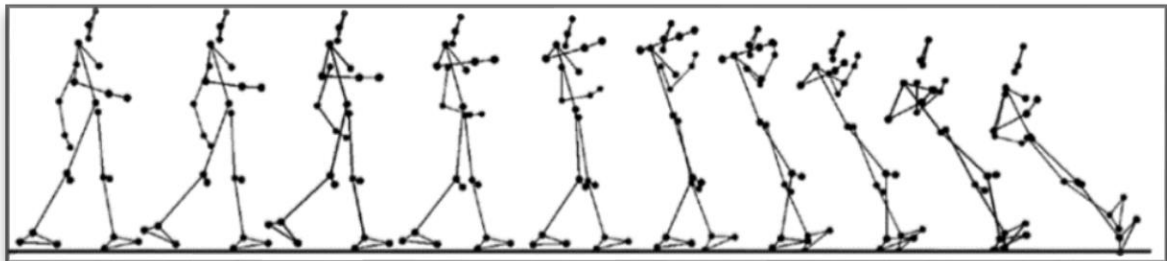
2.1.1 Método baseado em câmera

Esse método utiliza um pequeno conjunto de câmeras e um microcontrolador ou um computador pessoal como servidor dedicado. A ideia por trás desse método é que as câmeras são utilizadas para capturar o vídeo, transferindo as imagens para o servidor, para analisar e processar, segmentando os movimentos das pessoas no ambiente monitorado. Esse método requer somente uma única configuração inicial e pode monitorar múltiplas pessoas ao mesmo tempo.

Diferentes abordagens são utilizadas para criar uma solução com esse método Yu et al. (2012) utilizou o reconhecimento de postura através da segmentação do corpo de uma pessoa,

ilustrado na Figura 1, onde é executado um algoritmo de subtração do fundo sobre o vídeo capturado pelas câmeras e para obter melhores resultados, foi implementado um algoritmo de pós-processamento. Os resultados obtidos através desse método são bastante consistentes com um alto nível de detecção de quedas (97.08%) e com um nível de alarme falso baixíssimo, aumentando a confiabilidade do sistema, onde uma queda que não ocorreu, mas foi detectada, com uma taxa de apenas 0.8%. Uma desvantagem desse método é que quando vários objetos estão se movendo na sala monitorada, o algoritmo de detecção tem dificuldades em identificar o corpo da pessoa no ambiente.

Figura 1 - Segmentação de uma pessoa em queda



Fonte: Yu et al. (2012).

Já Nasution et al. (2007) implementou um sistema de detecção de quedas utilizando uma câmera fixa e um PC dedicado. O primeiro plano foi segmentado utilizando um algoritmo de subtração de fundo da imagem atual. Em seguida, são extraídos recursos das imagens como, o histograma de projeção vertical e horizontal do primeiro plano segmentado, o ângulo entre a última postura em pé em relação ao ângulo da postura atual da pessoa. Embora o sistema seja capaz de reconhecer 90% das quedas, o problema de vários objetos se movendo no ambiente ainda persiste. O ponto cego também é outra grande preocupação, já que é utilizada somente uma câmera fixa nessa abordagem.

As vantagens do método baseado em câmeras são:

- Extremamente confiável, com altas taxas de sucesso na detecção de quedas;
- Aplicação de diferentes algoritmos de processamento de imagens;
- Baixo risco de falso positivos.

Já as desvantagens são:

- Alto custo de implementação;
- Necessita de uma máquina capaz de executar algoritmos de processamento de imagens;
- Sujeito a problemas de oclusão de ambiente (vários objetos se movendo ao mesmo tempo).

2.1.2 Método baseado em acústica

Sensores acústicos e de ambiente são usados neste método para criar uma solução de detecção de quedas. O custo de implementação desse método é relativamente barato em comparação com o método baseado em câmera, devido ao baixo custo de hardware. Nesse método são usados sensores infravermelhos, sensores de vibração e microfones. Esses sensores coletam dados para serem processados por um microcontrolador ou um PC. Uma queda é detectada e alertada somente quando os dados coletados excederem um determinado limite ou condições definidos no algoritmo.

Alwan et al. (2006) criou um sistema de detecção de quedas que monitora a vibração do piso causado por uma queda. Nessa abordagem foi utilizado um sensor piezoelétrico acoplado ao piso para obter os dados a serem analisados. O resultado do experimento demonstra que 100% das quedas foram detectadas com sucesso, com nenhum caso de alarme falso. No entanto, a desvantagem desse método é que somente serão detectadas quedas na área monitorada. Outro fator negativo é que as vibrações não podem ser detectadas em todos os tipos de materiais de piso.

Popescu et al. (2008) propôs um sistema de detecção de quedas baseado em uma série de sensores acústicos. Para detectar uma queda, foram utilizados o volume e altura do som. A etapa mais importante é remover o ruído do ambiente, para isso foi montado dois microfones a 4 metros de distância um do outro. O sistema foi capaz de identificar 70% das quedas sem alarmes falsos. Porém, aumentando a sensibilidade, o sistema consegue detectar 100% das quedas, mas com 5 alarmes falsos a cada hora. O que torna o sistema inconsistente, reduzindo a confiabilidade dos alertas.

2.1.3 Método baseado em cinemática

Esse é um método comumente utilizado em sistemas de detecção de quedas, devido ao baixo custo de hardware, e por usar um sensor de forma vestível, acompanhando o usuário internamente ou externamente de sua residência. Acelerômetros e giroscópios podem ser posicionados em diferentes partes do corpo do usuário, possibilitando a diferenciação de uma queda ou atividades do dia a dia da pessoa monitorada. Os dados coletados pelo acelerômetro e giroscópio são processados em um microcontrolador. Por ser um dispositivo vestível, a região de atuação não é um limitador para esse método.

Ao realizar pesquisas relacionadas ao tema, observou-se diferentes formas de detectar e alertar quedas de pessoas idosas. Torquato et al. (2017) apresentou um trabalho utilizando o acelerômetro do celular e um aplicativo capaz de detectar uma eventual queda analisando dois parâmetros, respectivamente: (a) a aceleração resultante, para analisar uma variação brusca da aceleração que caracteriza uma queda, ou uma atividade comum como uma corrida ou caminhada; e (b) a análise de uma parada dos movimentos do usuário por um período, identificando uma possível queda.

Em sua tese Gomes et al. (2018) apresentou um sistema de detecção de quedas utilizando um sensor acelerômetro MPU6050 e os microcontroladores Arduino Uno e NodeMCU. A detecção de queda foi feita através da análise dos dados do sensor e por cálculos utilizando a equação de Torricelli [1], verificando a velocidade máxima alcançada em queda livre. Ao detectar a queda do usuário, o microcontrolador enviava um sinal de alerta utilizando o protocolo *Message Queue Telemetry Transport* (MQTT) via *Transmission Control Protocol/Internet Protocol* (TCP/IP), e um e-mail para os contatos de emergência utilizando o protocolo *Simple Mail Transfer Protocol* (SMTP).

$$v^2 = v_0^2 + 2 * a * \Delta s \quad [1]$$

No artigo de Hammarstedt (2019) foi utilizado uma abordagem diferente para detectar a queda do usuário, com o uso da plataforma Raspberry Pi juntamente com um sensor Acelerômetro ADXL345 para obter e tratar os dados de aceleração. O algoritmo utilizado para detectar a queda foi construído baseado na soma vetorial da aceleração dos eixos. Em seguida foram definidos parâmetros para comparar os dados armazenados, uma sequência de condições tem que deveriam ser satisfeitas. Para realizar o alerta de queda, foi escolhido um modem

Universal Serial Bus (USB) com capacidade de comunicação com a rede celular *Global System for Mobile Communications* (GSM). Após a detecção da queda, um *Short Message Service* (SMS) era encaminhado para os contatos de emergência.

2.2 Análise dos trabalhos relacionados

Comparando os métodos e projetos desenvolvidos, percebe-se várias maneiras de criar um sistema para monitoramento e detecção de quedas, usando de microcontroladores à *smartphones*. Ambos os métodos, baseado em câmera e acústico, possuem alguns problemas que são a falta de mobilidade, um número de ambientes limitados que podem ser monitorados, e alto ou médio custo. Com um dispositivo compacto, vestível e de baixo custo, todos estes problemas citados são resolvidos.

2.3 Framework Flutter

Flutter é um *framework* gratuito para desenvolvimento em multiplataformas, *open-source* criada pelo Google e lançada em maio de 2017. Ele permite ao desenvolvedor criar aplicações nativas para diversas plataformas, inicialmente com foco em plataformas móveis como Android e iOS, tudo com apenas uma linguagem de programação e um único código (THOMAS, 2019).

O *Framework* Flutter consiste em duas partes principais:

- *Software Development Kit* (SDK), que é uma coleção para auxiliar no desenvolvimento de aplicações, incluindo ferramentas para compilar o código nativo para cada plataforma;
- *Framework*, biblioteca UI baseada em *widgets*, sendo uma coleção de elementos de UI reutilizáveis (botões, entradas de texto, listas, dentre outros) que pode ser utilizado de acordo com a necessidade do desenvolvedor.

Para desenvolver utilizando Flutter, é utilizada a linguagem de programação Dart. Essa linguagem foi criada pela Google em outubro de 2011, mas recebeu diversas melhorias nos últimos anos. Dart é uma linguagem fortemente tipada e orientada a objetos, com uma sintaxe semelhante ao Javascript.

2.4 Bibliotecas Flutter

Essa seção trata sobre as bibliotecas Flutter utilizadas na criação da aplicação móvel e suas aplicações.

2.4.1 *FlutterBlue*

FlutterBlue é um plugin *bluetooth* para Flutter e tem como objetivo oferecer suporte para ambas as plataformas (iOS e Android). Com o uso da biblioteca *FlutterBlue* é possível escanear e conectar dispositivos *bluetooth* (*BluetoothDevice*) próximos do *smartphone* (DEMARCO, 2017). Uma vez que um dispositivo está conectado, é possível realizar a descoberta dos serviços disponíveis (*BluetoothService*), assim como suas características (*BluetoothCharacteristic*) e descritores (*BluetoothDescriptor*).

2.4.2 *Mailer*

Mailer é uma biblioteca de fácil utilização para compor e enviar e-mails em Dart. A aplicação dessa biblioteca possibilita a configuração e o envio de e-mail através de um servidor SMTP sem intervenção do usuário, a configuração é realizada através de uma conta Gmail, sendo necessário a criação de uma senha de aplicativo (SELLGREN, 2013).

2.4.3 *Telephony*

Telephony é um *plugin* Flutter que permite realizar o envio de SMS diretamente da aplicação, sem intervenção do usuário, nenhuma configuração prévia é necessária, utilizando recursos do próprio *smartphone* para realizar o envio de forma automática, requerendo apenas a permissão para fazer o envio do SMS (MULAY, 2020). Além disso, possibilita a execução de funcionalidades como o envio de mensagens SMS, iniciar chamadas de voz, observar a chegada de novos SMS's e obter diversos parâmetros de conexão.

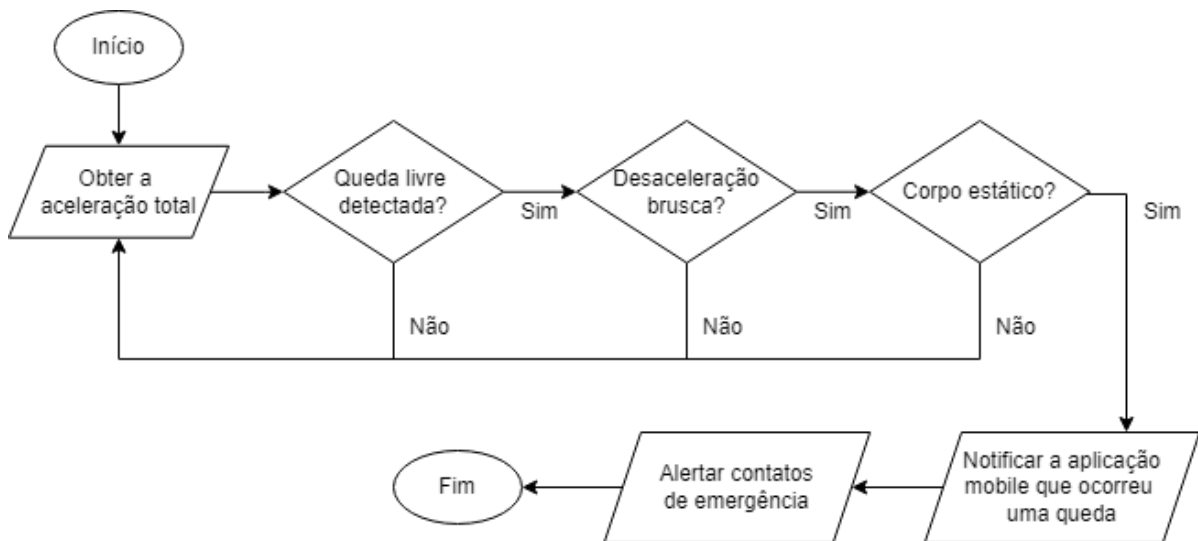
3 DESENVOLVIMENTO

Esse capítulo trata sobre a proposta de solução e todos os componentes utilizados para que fosse possível desenvolver o trabalho proposto.

3.1 Proposta de solução

Esse trabalho tem como objetivo criar um protótipo de baixo custo capaz de detectar quedas através dos dados obtidos por um sensor do tipo acelerômetro de 3-eixos, e uma aplicação para *smartphones* que comunique com o dispositivo de detecção de quedas via *Bluetooth*, e realize o alerta de uma queda para os contatos de emergência, tal processo é definido pelo fluxograma da Figura 2.

Figura 2 - Fluxograma do projeto



Fonte: Elaborado pelo autor.

O método aplicado para construir o software executado no microcontrolador é a soma vetorial dos três eixos do acelerômetro. Por causa dos movimentos e orientações serem imprevisíveis, monitorar os dados de cada eixo do acelerômetro individualmente podem gerar resultados instáveis.

O cálculo da aceleração total é mostrado na equação [2], onde A_X, A_Y, A_Z denotam a aceleração dos eixos X, Y e Z, respectivamente, e A_{TOTAL} é a aceleração total. Em um caso estático $A_{TOTAL} = 1g$.

$$A_{TOTAL} = \sqrt{A_X^2 + A_Y^2 + A_Z^2} \quad [2]$$

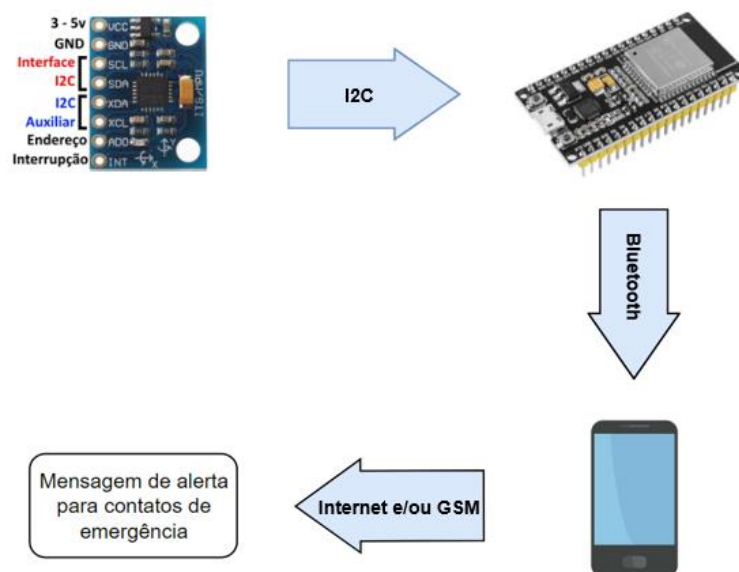
Com o protótipo proposto uma queda pode ser detectada independente da direção que a pessoa cair. Através da análise empírica dos dados de aceleração, foi possível determinar os seguintes parâmetros utilizados:

- $A_{TOTAL} < 0.4g$ por um tempo $t > 0.05s$ (Queda livre detectada);
- $A_{TOTAL} > 5g$ por mais de 3 vezes recorrentes após a queda livre (Desaceleração brusca detectada);
- $0.9g < \overline{A_{TOTAL}} < 1.5g$ por 0.5 segundos (Corpo estático, sem movimento).

É importante que os parâmetros utilizados não sejam muito altos nem muito baixos, em relação aos parâmetros definidos acima. Parâmetros muito altos irão resultar em falso negativos, onde uma queda real ocorreu, mas não foi reconhecida. Já parâmetros muito baixos vão indicar falsos positivos, ou seja, um alarme falso de uma queda que não ocorreu. Uma queda detectada com sucesso é chamada de verdadeiro positivo.

O diagrama de blocos da Figura 3 descreve quais dispositivos de hardware e meios de comunicação foram utilizados nesse projeto, visando o baixo custo de implementação do sistema de detecção de quedas.

Figura 3 - Diagrama de blocos do projeto



Fonte: Elaborado pelo autor.

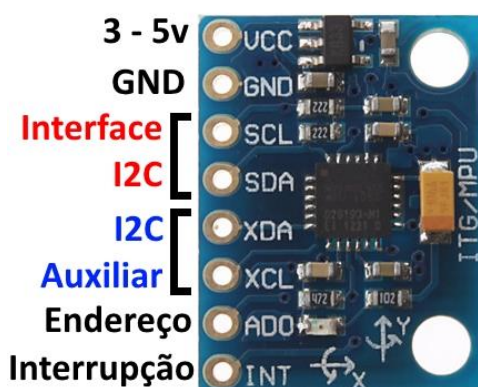
3.2 Componentes de Hardware

Esta seção apresenta os componentes de hardware utilizados no projeto para construir o protótipo e realizar a comunicação com o dispositivo *mobile*.

3.2.1 MPU-6050

O MPU-6050 é um chip MEMS acelerômetro e giroscópio (Figura 4), onde MEMS significa *Micro-Electro-Mechanical System* (Sistema Micro-Elétrico-Mecânico), capaz de medir a aceleração e rotação nos três eixos coordenados (X, Y, Z). O módulo também possui um sensor de temperatura capaz de medir entre -40°C a $+85^{\circ}\text{C}$ (TDK INVENSENSE, 2013).

Figura 4 - MPU-6050



Fonte: THOMSEN, 2014.

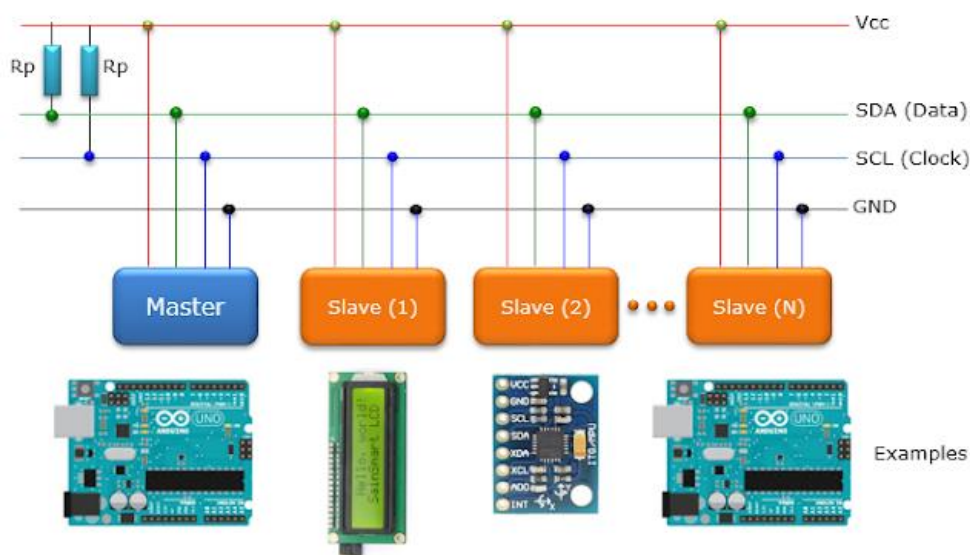
A alimentação é feita através dos pinos VCC e GND respectivamente, onde o VCC pode receber entre 3V e 5V. No MPU-6050, o pino SCL é responsável pelo *clock*, já o pino SDA é responsável por transmitir e receber os dados, ele utiliza a interface I2C, que requer apenas dois canais (fios). Nos pinos XDA e XCL pode-se ligar outros dispositivos com a interface I2C, como um magnetômetro, por exemplo. O pino ADO quando alimentado altera o endereço padrão (0x68) do MPU-6050 para o endereço alternativo (0x69), essa mudança permite que dois módulos do MPU-6050 sejam utilizados em um mesmo circuito.

3.2.2 Interface I2C

O *Inter-Integrated Circuit*, ou Circuito Inter-Integrado, (I2C, ou I²C) é um tipo de interface serial utilizada para comunicação com diversos dispositivos. Foi criado pela Philips, tem como vantagem a simplicidade e o baixo custo.

Há dois canais de comunicação: (a) o canal de dados seriais, chamado de *Serial Data* (SDA); e (b) o canal de sincronização, chamado *Serial Clock* (SCL). Estes dois canais são bidirecionais, ou seja, os dispositivos dos dois lados podem utilizar os canais. Para haver comunicação, o arranjo dos dispositivos tem que ser do tipo “Mestre” e “Escravo” (*Master and Slave*). O mestre é responsável por coordenar a comunicação, ou seja, gerar *clock* e iniciar a comunicação. Enquanto o escravo apenas recebe os dados e responde quando é solicitado. Cada escravo possui um endereço individual de 7 bits, que é utilizado para um mestre conseguir comunicar de forma individual com eles. A Figura 5 apresenta um exemplo de comunicação via I2C.

Figura 5 – Exemplo de comunicação I2C



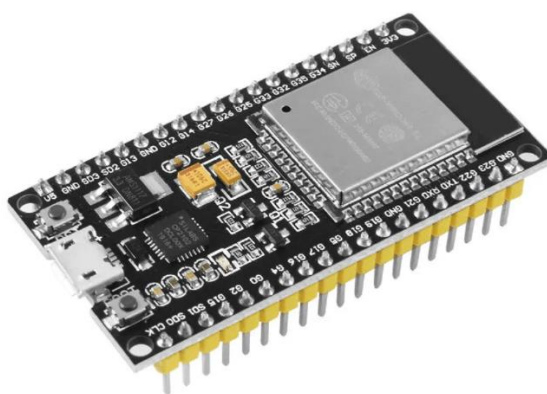
Fonte: GIRALDO, 2020.

A desvantagem do I2C é a baixa velocidade de comunicação, visto que uma única linha recebe e transmite os dados (comunicação *Half-duplex*). Ou seja, não é possível fazer os dois simultaneamente e o tempo gasto é dobrado. O valor médio de velocidade é de 100 kbit/s (GUIMARÃES, 2022).

3.2.3 ESP32

O ESP32 é um componente de baixo custo e alta performance, desenvolvido pela *Espressif Systems*, que se caracteriza por ser um *System on a Chip* (SOC) contendo um microcontrolador, Wi-Fi e *Bluetooth Low Energy* (BLE) integrados. O ESP32, Figura 6, foi projetado para aplicações moveis, dispositivos vestíveis e Internet das Coisas (IoT). Com um design robusto, ele é capaz de operar em ambientes extremos, como indústrias por exemplo, com uma margem de operação de temperatura de -40°C a $+125^{\circ}\text{C}$ (ESPRESSIF SYSTEMS, 2021).

Figura 6 – ESP-32



Fonte: CACPNRJ, 2021.

Neste trabalho, o ESP32 foi utilizado para comunicar com o sensor MPU-6050, obter os dados de aceleração e realizar o tratamento dos dados. O algoritmo de detecção de quedas foi desenvolvido, utilizando a linguagem de programação C++, voltado para essa plataforma.

3.3 Bluetooth Low Energy

O *Bluetooth Low Energy* (BLE) é um tipo de *bluetooth* que consome cerca de 10% de energia em relação ao *Bluetooth* clássico. O BLE foi desenvolvido para transmissão de dados, porém com o foco em maximizar a economia de bateria, para garantir um longo tempo de transmissão de informações (ARAUJO; VASCONCELLOS, 2012).

Trata-se de um protocolo que surgiu na versão 4.0 do *Bluetooth* clássico. Uma das características de dispositivos que utilizam o BLE, é que na maior parte do tempo, ele atua em modo *sleep*, sendo ativado apenas para realizar conexões que duram milésimos de segundo,

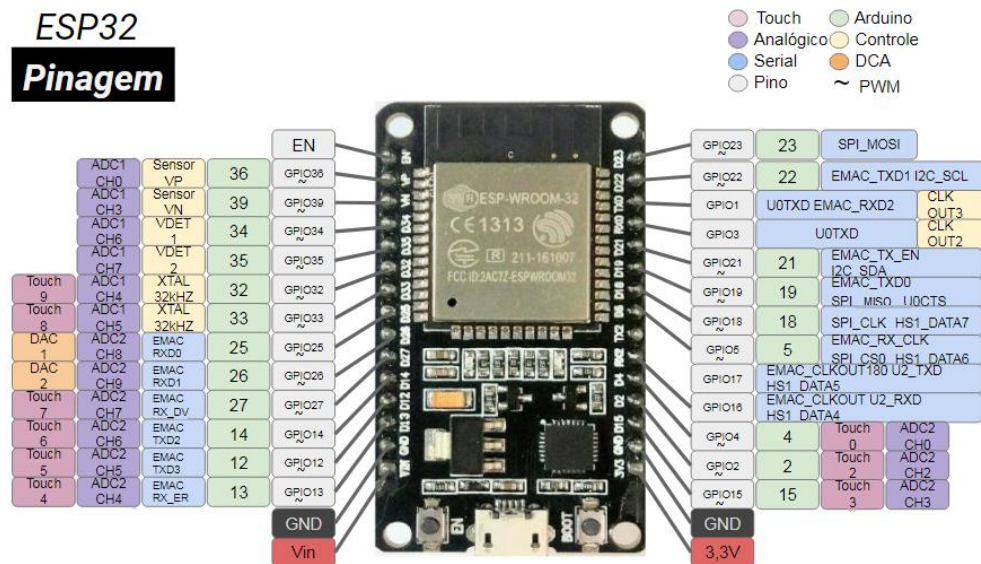
para transmitir uma pequena quantidade de dados, se tornando ideal para aplicações que notificam de forma esporádica.

Devido a tais características, esse protocolo foi escolhido para realizar a comunicação entre o ESP32 e a aplicação móvel, sendo que ao detectar uma queda, o evento será notificado através do protocolo BLE.

3.4 Construção do protótipo

A construção do protótipo utiliza os dispositivos descritos nas seções 3.2.1 e 3.2.3. Para realizar a comunicação entre os dispositivos foi utilizado a interface de comunicação I2C, descrita na seção 3.2.2 para integrar todos os dispositivos foi utilizado uma Protoboard de 830 pinos e conectores jumpers. A Figura 7 apresenta a pinagem do microcontrolador ESP32, sendo que a conexão de dados com o sensor acelerômetro foi feita utilizando as portas GPIO21 e GPIO22.

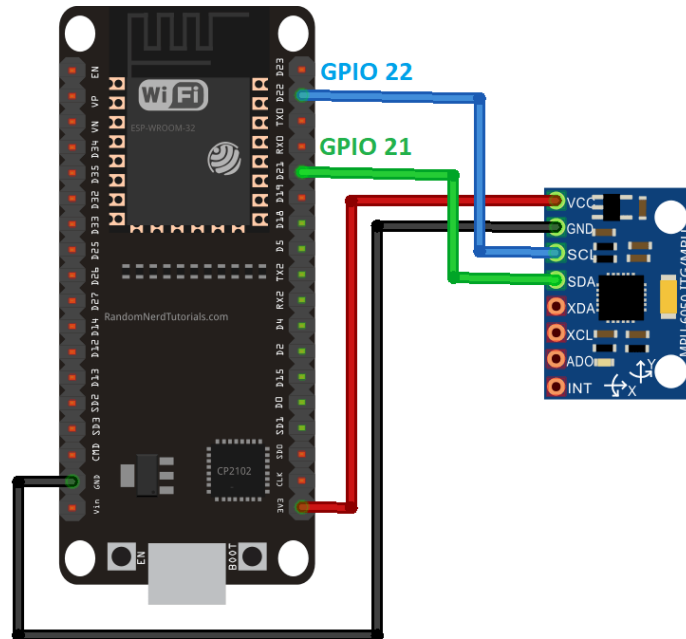
Figura 7 – Pinagem ESP-32



Autor: BLOG DA CURTO, 2018.

O sensor acelerômetro foi alimentado pelas conexões do próprio microcontrolador, que fornece uma tensão de 3.3V combinado com o pino GND, já o microcontrolador é alimentado pela porta Micro USB, sendo possível também realizar a alimentação pelo pino VIN. Na Figura 8 é ilustrado um esquema semelhante a implementação real do protótipo.

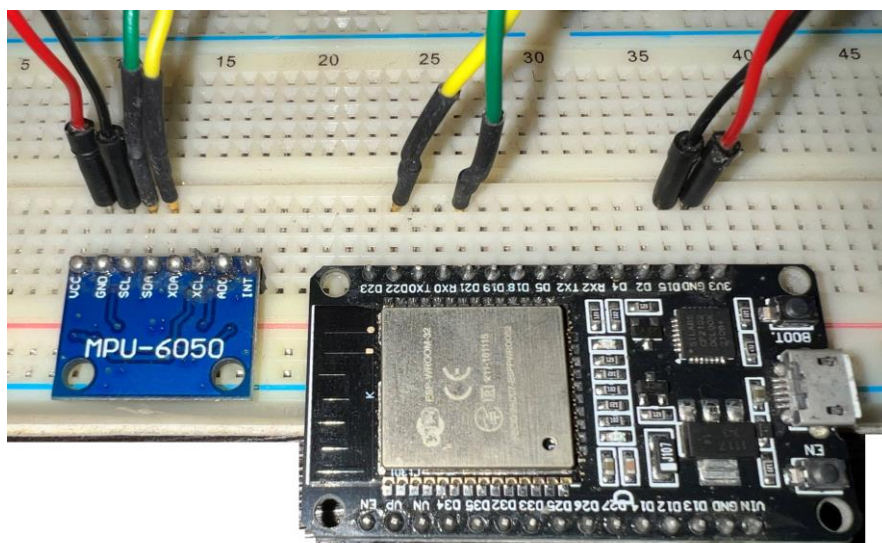
Figura 8 – Esquema de comunicação ESP-32 – MPU-6050



Fonte: RANDOM NERD TUTORIALS, 2021.

A Figura 9 apresenta o protótipo construído utilizando uma protoboard de 830 pinos, juntamente com jumpers de diversas cores e os componentes principais, descritos nas seções 3.2.1 e 3.2.3.

Figura 9 – Protótipo



Fonte: Elaborado pelo autor.

3.5 Firmware de testes

Para realizar os testes foi desenvolvido um *firmware* para gerar dados para análise e visualização, o código do *firmware* de testes foi escrito utilizando a linguagem C++, em conjunto com as bibliotecas I2Cdev, Wire e MPU6050 para realizar a comunicação com o sensor acelerômetro. O código fonte está disponível no Apêndice A – Firmware de Testes.

Para realizar a leitura dos dados foi utilizado o software PuTTY, um terminal de simulação *open source*, desenvolvido para agir como um cliente de conexões através de protocolos TCP, Telnet, serial, dentre outros.

Após a coleta dos dados brutos, para tratar e gerar gráficos para a visualização e análise dos resultados, foi utilizado a linguagem de programação Python juntamente com as bibliotecas Pandas, Seaborn e Matplotlib. O código para gerar os gráficos está disponível no Apêndice B – Código Python para gerar os gráficos.

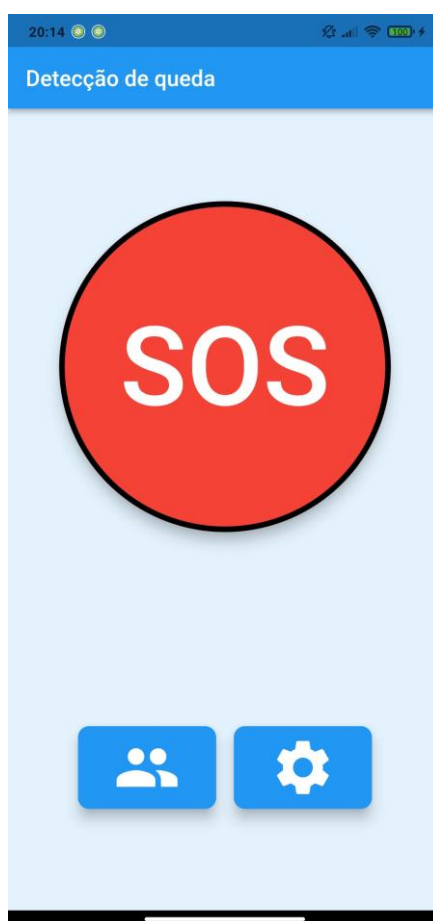
4 APLICAÇÃO MÓVEL

Este capítulo apresenta a criação de uma aplicação móvel com o objetivo de realizar a comunicação com o protótipo de detecção de quedas via *Bluetooth* e o alerta da queda para contatos de emergência, utilizando o *Framework* Flutter. O repositório com o código da aplicação móvel disponível em: https://github.com/Wellington-gc/tcc_app.

4.1 Tela inicial

A aplicação foi construída de forma a ter uma interface simples e intuitiva para que seja acessível aos usuários. A tela inicial, ilustrada na Figura 10, possui um botão de pânico de fácil acesso, que invoca a tela de alerta de queda, um botão para a tela de contatos e um botão para a tela de configurações.

Figura 10 – Tela inicial

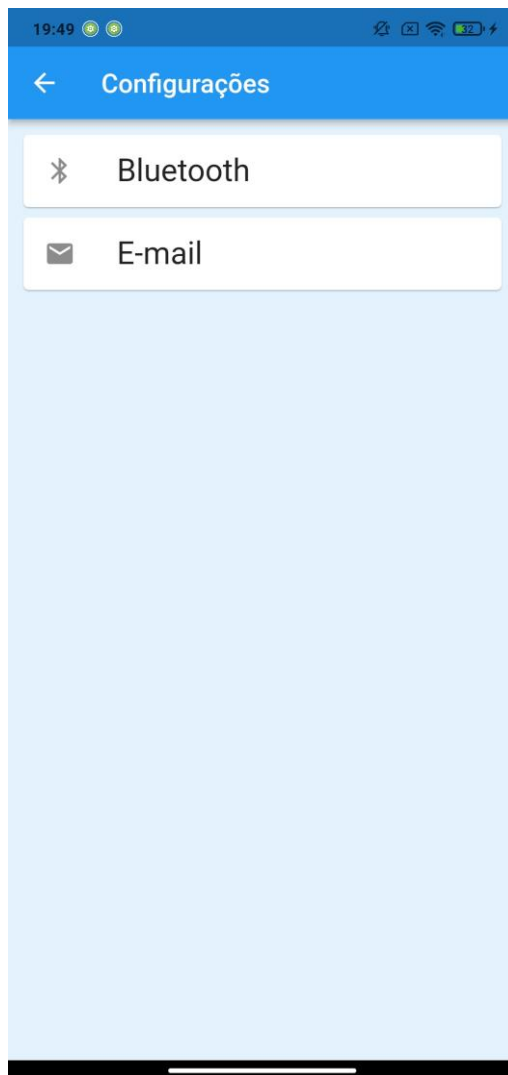


Fonte: Elaborado pelo autor.

4.2 Tela de Configurações

A tela de Configurações, ilustrada na Figura 11, consiste em um menu para obter acesso a configurações específicas da aplicação, como *Bluetooth* e E-mail.

Figura 11 – Menu de configurações

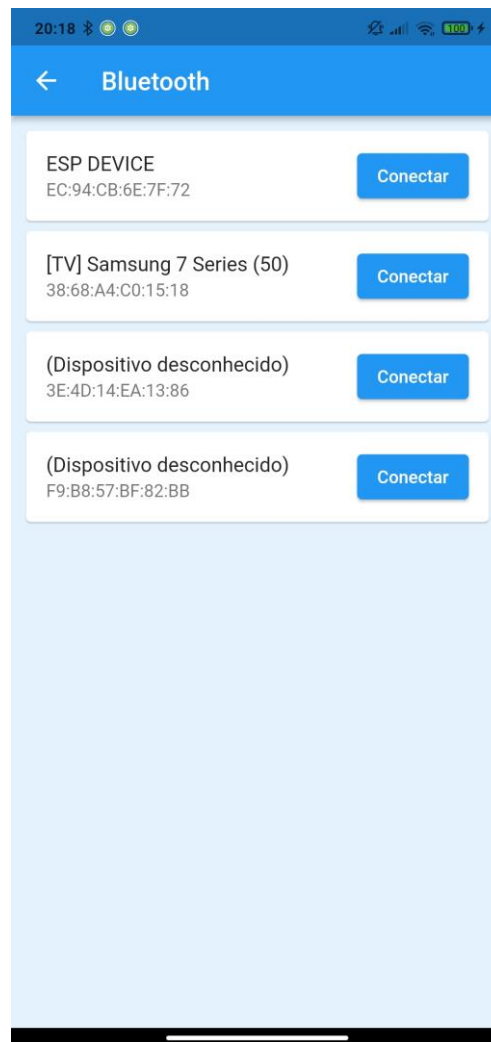


Fonte: Elaborado pelo autor.

4.2.1 Configuração do *Bluetooth*

A tela de Configuração do *Bluetooth*, ilustrada na Figura 12, consiste em uma lista de dispositivos *Bluetooth* próximos disponíveis para conexão. Ao conectar com o dispositivo “*ESP DEVICE*”, é feita a descoberta dos Serviços, e para cada serviço é feita a descoberta das características, a comunicação é realizada a partir de uma característica do tipo “*Notify*”, configurado na ESP32. Ao conectar no dispositivo de detecção de quedas, a aplicação aguarda uma notificação do dispositivo que uma queda ocorreu. Para realizar a comunicação *Bluetooth* foi utilizada a biblioteca FlutterBlue descrita na seção 2.4.1.

Figura 12 – Configurações *Bluetooth*



Fonte: Elaborado pelo autor.

4.2.2 Configuração de E-mail

A tela de configuração de E-mail, ilustrada na Figura 13, é utilizada para configurar o servidor SMTP utilizado para enviar o alerta de queda, o e-mail é enviado a partir da conta Gmail do usuário, a senha utilizada é uma senha de aplicativo configurada na seção de segurança da conta Google do usuário. Com isso, é possível realizar o envio de e-mail sem nenhuma ação do usuário.

Figura 13 – Configurações de E-mail



22:09

E-mail

Nome Completo

E-mail

Senha

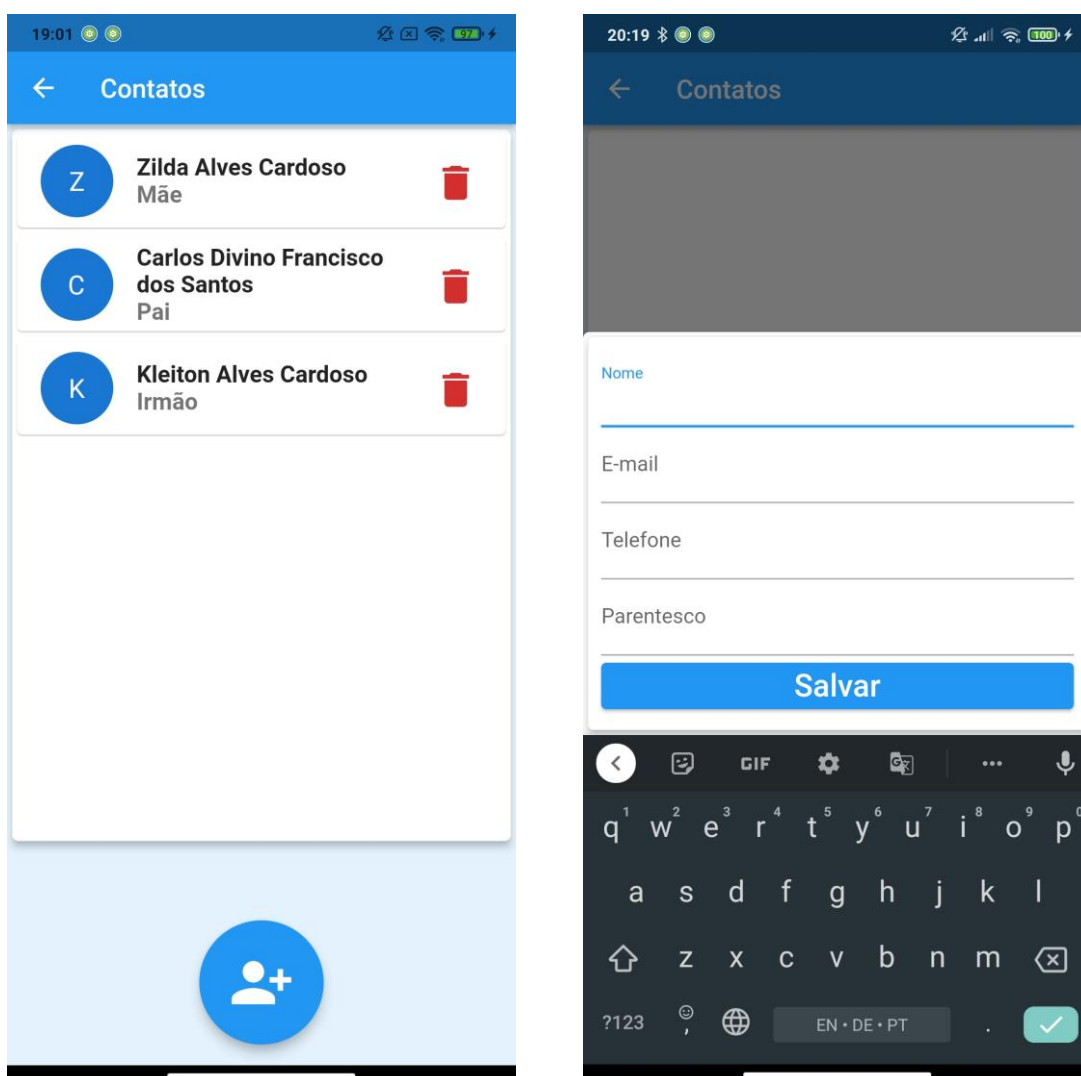
Salvar

Fonte: Elaborado pelo autor.

4.3 Tela de Contatos

O usuário pode cadastrar uma lista de contatos que será utilizada para realizar o envio do alerta de queda, para isso foi criada uma tela com a visualização de todos os contatos cadastrados e um formulário para cadastrar novos contatos, ilustrados na Figura 14. Para armazenar os contatos foi utilizado o Hive, um banco de dados do tipo chave-valor, com armazenamento local NoSQL.

Figura 14 – Tela de contatos



Fonte: Elaborado pelo autor.

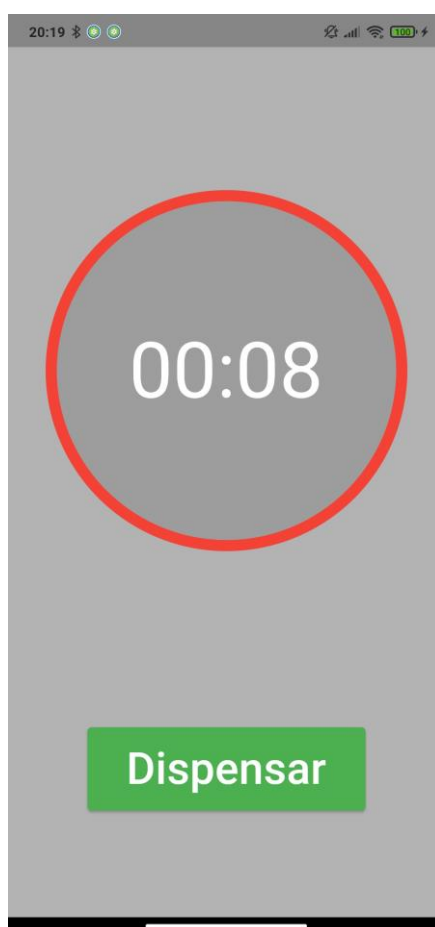
4.4 Tela de Alerta

A tela de alerta, ilustrada na Figura 15, pode ser invocada de duas maneiras diferentes: (a) ao pressionar o botão SOS na tela inicial; ou (b) com uma notificação de tela inteira, semelhante ao funcionamento de um despertador. Quando a tela é instanciada um timer é iniciado, o valor padrão é de 10 segundos.

Quando chega ao final do timer, é disparado as mensagens de alerta, via E-mail e SMS para todos os contatos cadastrados pelo usuário, por fim é iniciada uma ligação para o primeiro contato da lista de contatos. Caso o usuário tenha apertado o botão por engano, ou o dispositivo tenha alertado uma falsa queda, o usuário pode apertar o botão “Dispensar”, para abortar o envio do alerta.

Para realizar o envio de E-mail foi utilizado a biblioteca Mailer, e para realizar o envio de SMS e a ligação foi utilizada a biblioteca Telephony.

Figura 15 – Tela de alerta



Fonte: Elaborado pelo autor.

4.5 Mensagem de alerta

A mensagem de alerta é enviada através de e-mail e SMS para a lista de contatos cadastrados, o envio da mensagem de alerta é um passo importante da proposta de solução, por isso está sendo utilizado mais de um meio de comunicação para realizar o envio da mensagem para mais de um contato.

4.5.1 E-mail

O envio de E-mail foi realizado através de um servidor SMTP, disponibilizado através da biblioteca Mailer descrita na seção 2.4.2. O corpo do E-mail é composto pela mensagem, nome do usuário, data e hora do evento e instruções, ilustrado na Figura 16.

Figura 16 – E-mail de alerta



Fonte: Elaborado pelo autor.

4.5.2 SMS

O envio de SMS foi implementado utilizando a biblioteca Telephony descrita na seção 2.4.3. Devido à limitação de 160 caracteres do padrão de mensagem SMS, o corpo da mensagem foi reduzido, contendo a mensagem de queda, nome do usuário, data do evento e instruções, ilustrada na Figura 17.

Figura 17 – SMS de alerta



Fonte: Elaborado pelo autor.

5 TESTES E RESULTADOS

Este capítulo descreve como foram realizados os testes do protótipo, apresentando e explicando gráficos gerados durante os testes, e como os dados obtidos foram utilizados para gerar o algoritmo de detecção de quedas citado na seção 3.1.

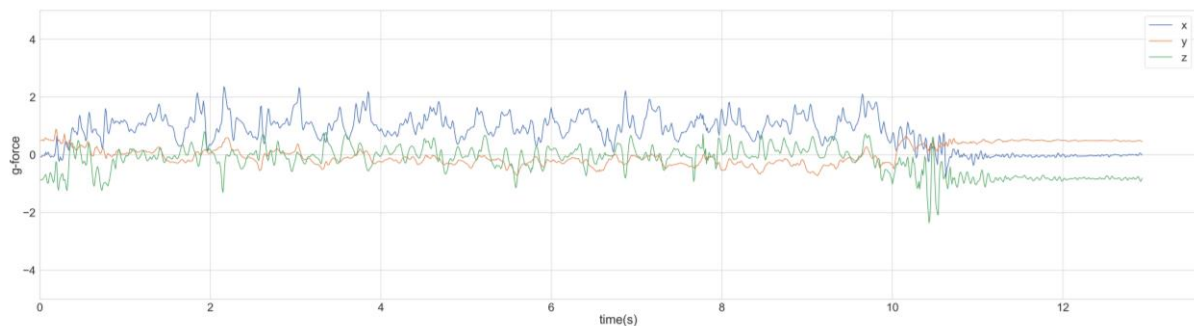
5.1 Testes do protótipo

Essa seção apresenta a análise dos testes das leituras de aceleração de diferentes atividades do usuário realizadas pelo protótipo, a análise dos dados presentes nos gráficos desse capítulo serviu de base para criar o algoritmo de detecção de quedas, de tal maneira que atividades rotineiras não fosse interpretada como um falso positivo de queda.

5.1.1 *Andando*

Quando o usuário andou ou caminhou, uma leitura bem característica e estável, ilustrada na Figura 18, sem grandes variações, uma atividade comum como esse será filtrada, sem riscos de haver um falso positivo na detecção de queda. Os testes de caminhada foram feitos em um corredor sem alterações de elevação ou obstáculos, num percurso de aproximadamente 10 metros por 10 segundos, ao final podemos visualizar uma estabilização, que significa que o usuário está parado.

Figura 18 – Gráfico andando

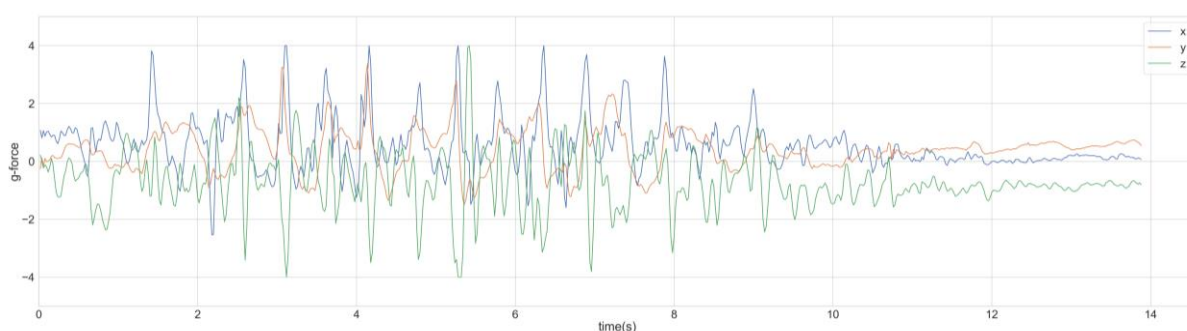


Fonte: Elaborado pelo autor.

5.1.2 Correndo

Quando o usuário estiver correndo, uma leitura com picos acentuados e repetitivos ilustrada na Figura 19 foi observada, assim como quando o usuário está caminhando. Os testes de corrida foram realizados em um corredor com uma superfície lisa e sem alterações de elevações, num percurso de aproximadamente 10 metros por 8 segundos, e a cada 0.5 segundos em média, foi possível visualizar o impacto de cada passo nas leituras de aceleração.

Figura 19 – Gráfico correndo

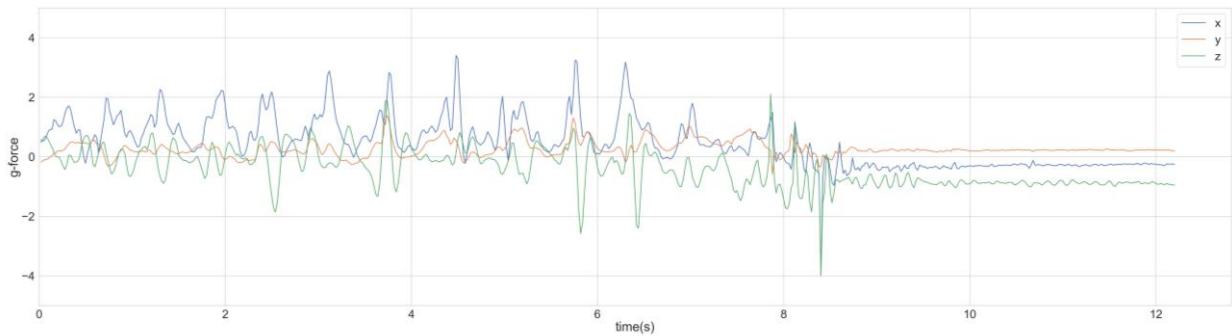


Fonte: Elaborado pelo autor.

5.1.3 Subindo e descendo escadas

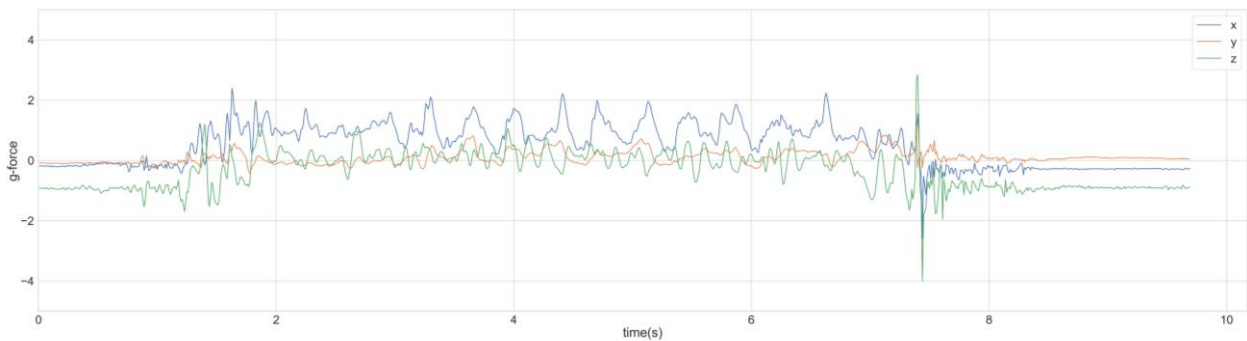
Uma outra atividade comum que deve ser analisada e mitigada, é o trânsito do usuário por escadas ilustrado na Figura 20 e Figura 21, pois em certos casos, as leituras obtidas em escadas podem ser semelhantes à de uma queda, causando um falso alerta de queda. Os testes foram feitos em uma escadaria com cerca de 10 degraus por aproximadamente 8 segundos, sendo estes mais evidentes quando está subindo as escadas.

Figura 20 – Gráfico subindo escadas



Fonte: Elaborado pelo autor.

Figura 21 – Gráfico descendo escadas



Fonte: Elaborado pelo autor.

5.1.4 Queda Frontal

Nessa seção serão mostrados os resultados de eventuais quedas. Estes testes foram realizados com o protótipo ligado ao corpo do usuário, que simulou quedas em um colchão. O usuário voluntário para simular a queda possui 170 cm de altura. Primeiramente uma queda de frente foi simulada, com o dispositivo no pulso do usuário, e o pulso na altura próxima dos ombros, fazendo uma trajetória simulando uma queda real, onde os braços tendem a se projetar em frente a face, com o objetivo de proteger a cabeça, ilustrada na Figura 22.

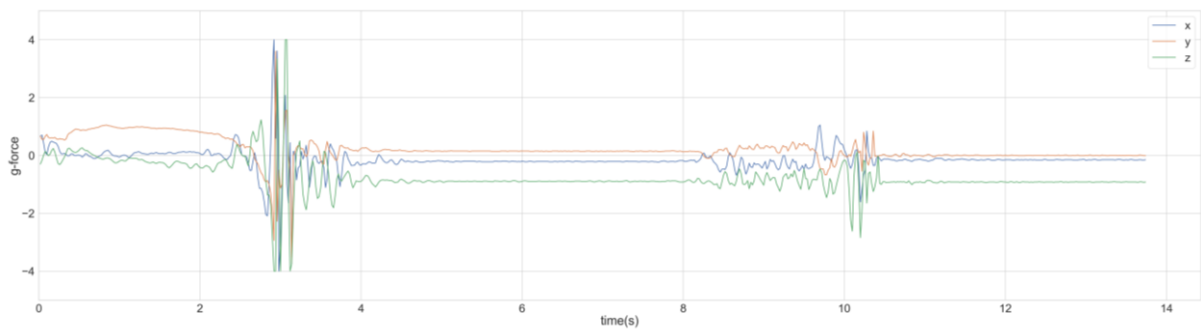
Figura 22 – Ilustração de queda frontal



Fonte: Adaptado de FOROUGHJI (2009).

Na Figura 23 temos a visualização do gráfico de uma queda frontal com as características descritas na seção 3.1, onde a aceleração tende a zero, seguida da desaceleração brusca com picos de leituras, por fim o usuário estático sem se movimentar por alguns segundos.

Figura 23 – Gráfico de queda frontal



Fonte: Elaborado pelo autor.

5.1.5 Queda de costas

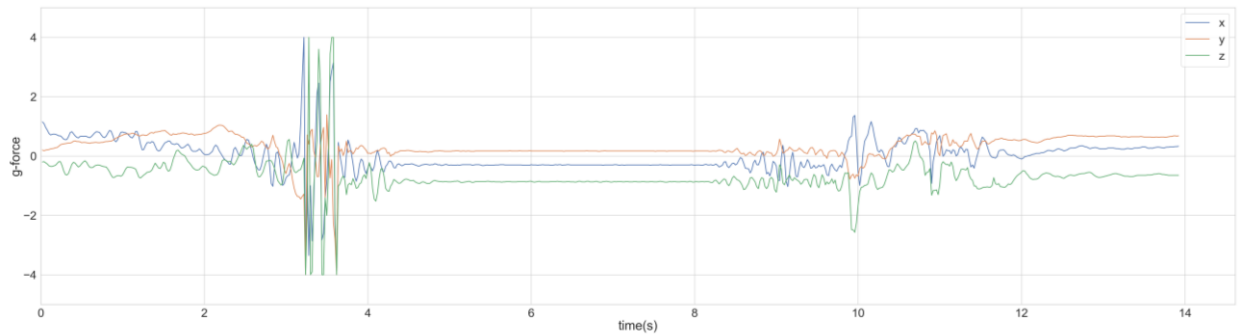
A queda de costas, ilustrada na Figura 24, gera leituras semelhantes a queda frontal, porém com mais picos de desaceleração, demonstrados no gráfico da Figura 25.

Figura 24 – Ilustração de queda de costas



Fonte: Adaptado de FOROUGHJI (2009).

Figura 25 – Gráfico de queda de costas

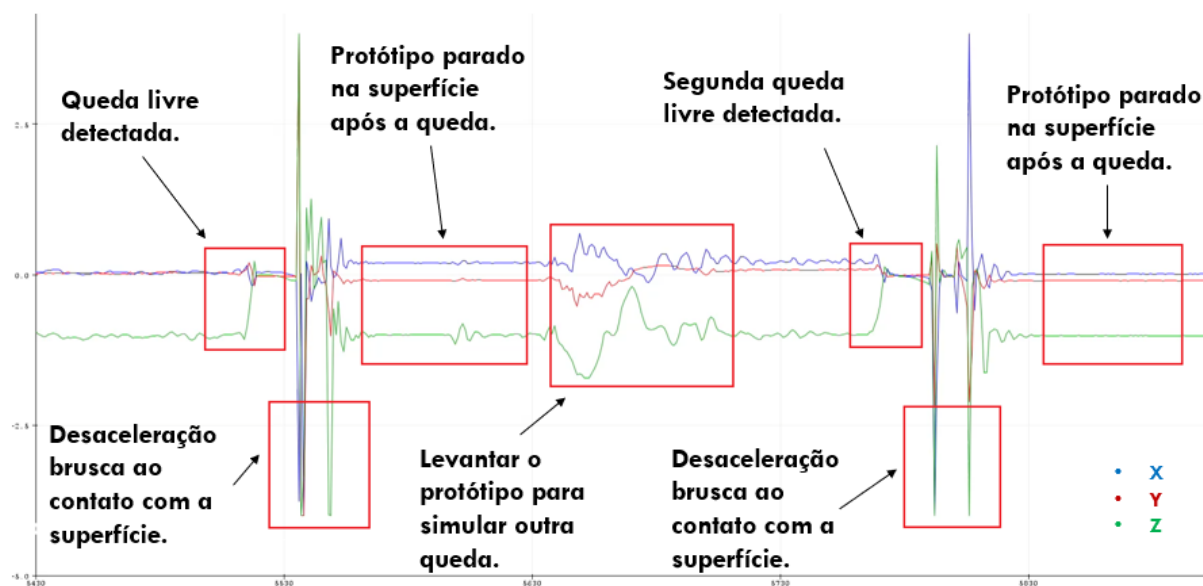


Fonte: Elaborado pelo autor.

5.1.6 Visualização da queda do protótipo

A queda do protótipo, ilustrada na figura 26, possibilita a visualização detalhada de como uma queda ocorre, com os seguintes passos: (a) queda livre, (b) desaceleração brusca, (c) protótipo sem grandes variações nas leituras. Foi feita a simulação de duas quedas em sequência, sendo possível visualizar entre as quedas, quando o protótipo foi movimentado para realizar a segunda queda.

Figura 26 - Visualização da queda do protótipo



Fonte: Elaborado pelo autor.

5.2 Implementação do firmware

Para implementar o firmware de detecção e alerta de quedas, foi utilizada uma implementação da biblioteca FreeRTOS, com essa implementação foi possível criar uma tarefa (*task*) para realizar a leitura dos dados do sensor acelerômetro e notificar o *smartphone* via a interface de comunicação BLE.

Sempre que o dispositivo de detecção de quedas for ligado, será iniciado a descoberta para outros dispositivos *Bluetooth* próximos, habilitando o dispositivo para ser pareado e conectado pelo *smartphone*, o nome padrão escolhido para o dispositivo de detecção foi “ESP DEVICE”. Uma característica *Bluetooth* foi criada, com um ID único. A implementação do firmware está disponível em: <https://github.com/Wellington-gc/Fall-Detection-Firmware-ESP32>.

6 CONSIDERAÇÕES FINAIS

Este trabalho apresentou o desenvolvimento e a documentação de uma solução de baixo custo composta por: (a) hardware, para a detecção de quedas, e (b) aplicativo para dispositivos móveis, utilizado para o envio de mensagens de alerta. Para o desenvolvimento do protótipo foi utilizado componentes de hardware de baixo custo para obter e enviar dados. Estes componentes serviram para realizar as leituras de aceleração do usuário que foram utilizadas para monitorar possíveis quedas.

A utilização do *Framework* Flutter possibilitou a implementação de um aplicativo móvel multiplataforma para realizar a comunicação com o protótipo, sendo o aplicativo responsável por realizar o envio de mensagens de emergência para os contatos cadastrados e armazenados no dispositivo do usuário.

O objetivo deste trabalho foi propor uma solução de baixo custo para identificação e alerta de quedas que fosse capaz de agilizar a prestação de socorro de pessoas necessitadas. Para a detecção de quedas que o protótipo identificou durante os testes, foi realizado o envio de mensagens de alerta via e-mail e SMS, em que a mensagem de alerta contém: nome do usuário, descrição, data e hora do incidente.

Por fim, esse apresentou resultado satisfatório ao detectar a ocorrência de quedas, possibilitando a notificação aos contatos de emergência cadastrados. O monitoramento, juntamente com o alerta de quedas, é importante e precisa ser tratado com urgência, para garantir o atendimento rápido à vítima, aumentando as chances de sobrevivência do usuário e minimizando o risco de sequelas.

6.1 Sugestões de trabalhos futuros

- Enviar a geolocalização do *smartphone* do usuário ao alertar uma queda;
- Melhorar o protótipo para ser um dispositivo vestível pelo usuário;
- Melhorar a forma de realizar detecção de quedas com a aplicação de inteligência artificial;
- Testar o aplicativo desenvolvido, utilizando o *Framework* Flutter, no sistema operacional iOS;
- Realizar o envio de mensagens via redes sociais, como Whatsapp ou Telegram.

REFERÊNCIAS

A. H. Nasution and S. Emmanuel, "**Intelligent Video Surveillance for Monitoring Elderly in Home Environments**," 2007 IEEE 9th Workshop on Multimedia Signal Processing, 2007, pp. 203-206, doi: 10.1109/MMSP.2007.4412853.

APPLE BRASIL. **Watch**. 6 jun. 2022. Disponível em: <https://www.apple.com/br/watch/>. Acesso em: 7 jun. 2022.

ARAUJO, André; VASCONCELLOS, Pedro. **BLE | bluetooth low energy**. 21 fev. 2012. Disponível em: https://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2012_2/bluetooth/ble.htm. Acesso em: 17 mar. 2022.

CACPNRJ. **Instalando a placa ESP32 no Arduino IDE (instruções para Mac OS X e Linux)** - Cap Sistema. 4 jul. 2021. Disponível em: <https://capsistema.com.br/index.php/2021/07/04/instalando-a-placa-esp32-no-arduino-ide-instrucoes-para-mac-os-x-e-linux/>. Acesso em: 2 abr. 2022.

DEMARCO, Paul. **flutter_blue | Flutter Package**. 1 set. 2017. Disponível em: https://pub.dev/packages/flutter_blue. Acesso em: 20 jun. 2022.

ELENA, Dalia; RODRIGUES, Danilo. **Idosos no contexto da pandemia da COVID-19 no Brasil**: efeitos nas condições de saúde, renda e trabalho. 31 mar. 2021. Disponível em: <https://www.scielo.br/j/csp/a/gXG5RYBXmdhc8ZtvKjt7kzc/?lang=pt>. Acesso em: 23 set. 2021.

ESPRESSIF SYSTEMS. **ESP32 wi-fi & bluetooth MCU I espressif systems**. 3 abr. 2021. Disponível em: <https://www.espressif.com/en/products/socs/esp32>. Acesso em: 11 nov. 2021.

Foroughi, Homa & Aski, Baharak & Pourreza, Hamid. (2009). **Intelligent video surveillance for monitoring fall detection of elderly in home environments**. 219 - 224. 10.1109/ICCITECHN.2008.4803020.

GIRALDO, Sergio. **Bus de comunicación I2C con arduino**. 6 jan. 2020. Disponível em: <https://controlautomaticoeducacion.com/arduino/bus-comunicacion-i2c/>. Acesso em: 1 jun. 2022.

GOMES, Renata. **Sistema domiciliar de cuidados em saúde**: mecanismo de detecção de quedas de pessoas. 2019. 20 p. Dissertação de mestrado — Instituto Federal de São Paulo (IFSP), São Paulo, 2019. Disponível em: https://hto.ifsp.edu.br/portal/images/thumbnails/images/IFSP/Cursos/Coord_ADS/Arquivos/TCCs/2018/TCC_RenataGomesdaSilva_HT1520792.pdf. Acesso em: 9 nov. 2021.

GUIMARÃES, Fábio. **I2C - Mundo Projetado**. 22 ago. 2022. Disponível em: <https://mundoprojetado.com.br/i2c/>. Acesso em: 20 jun. 2022.

HAMMARSTEDT, Ola. **Fall detection bracelet with an accelerometer and cellular connectivity**. 2019. 63 p. Dissertação de mestrado — Uppsala University, Uppsala, 2019.

M. Alwan et al., "A Smart and Passive Floor-Vibration Based Fall Detector for Elderly," 2006 2nd International Conference on Information & Communication Technologies, 2006, pp. 1003-1007, doi: 10.1109/ICTTA.2006.1684511.

M. Popescu, Y. Li, M. Skubic and M. Rantz, "An acoustic fall detector system that uses sound height information to reduce the false alarm rate," 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2008, pp. 4628-4631, doi: 10.1109/IEMBS.2008.4650244.

M. Yu, A. Rhuma, S. M. Naqvi, L. Wang and J. Chambers, "A Posture Recognition-Based Fall Detection System for Monitoring an Elderly Person in a Smart Home Environment," in IEEE Transactions on Information Technology in Biomedicine, vol. 16, no. 6, pp. 1274-1286, Nov. 2012, doi: 10.1109/TITB.2012.2214786.

MULAY, Shounak. **telephony | Flutter Package**. 3 jul. 2020. Disponível em: <https://pub.dev/packages/telephony>. Acesso em: 20 jun. 2022.

PEETERS, G.; SCHOOR, Natasja; LIPS, Paul. **Fall risk: the clinical relevance of falls and how to integrate fall risk with fracture risk**. 14 dez. 2009. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S1521694209000977?via=ihub#!>. Acesso em: 29 set. 2021.

RANDOM NERD TUTORIALS. **ESP32 MPU-6050 accelerometer and gyroscope (arduino) | random nerd tutorials**. 12 jan. 2021. Disponível em: <https://randomnerdtutorials.com/esp32-mpu-6050-accelerometer-gyroscope-arduino/>. Acesso em: 1 jun. 2022.

SELLGREN, Kai. **mailer | Dart Package**. 10 set. 2013. Disponível em: <https://pub.dev/packages/mailer>. Acesso em: 20 jun. 2022.

SOUZA, Carinne; BOSCO, Natalia; FIGUEIREDO, Odail. **Estudo divulga perfil dos idosos brasileiros**. 12 fev. 2021. Disponível em: <https://www.correiobraziliense.com.br/brasil/2021/02/4906292-estudo-divulga-perfil-dos-idosos-brasileiros.html>. Acesso em: 15 set. 2021.

TDK INVENSENSE. **Mpu-6050 | tdk**. 19 ago. 2013. Disponível em: <https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/>. Acesso em: 10 dez. 2021.

THOMAS, Gaël. **What is flutter and why you should learn it in 2020**. 12 dez. 2019. Disponível em: <https://www.freecodecamp.org/news/what-is-flutter-and-why-you-should-learn-it-in-2020/>. Acesso em: 14 abr. 2022.

THOMSEN, Adilson. Tutorial: **Acelerômetro MPU6050 com Arduino - FilipeFlop**. 30 set. 2014. Disponível em: <https://www.filipeflop.com/blog/tutorial-acelerometro-mpu6050-arduino/>. Acesso em: 2 mar. 2022.

TORQUATO, Lukas; MONTEIRO, André; ROCHA, Felipe. **Fall monitor**: detector de quedas para dispositivos móveis para cuidado a pessoas com mobilidade debilitada. 3 set. 2017. Disponível em: <https://www.sbrt.org.br/sbrt2017/anais/1570361999.pdf>. Acesso em: 3 nov. 2022.

TUMELERO, Naína. **Pesquisa aplicada**: guia com exemplos e características. 18 set. 2019. Disponível em: <https://blog.mettzer.com/pesquisa-aplicada/>. Acesso em: 16 set. 2021.

TUMELERO, Naína. **Pesquisa experimental**: o que é e como fazer em 5 passos. 30 set. 2019. Disponível em: <https://blog.mettzer.com/pesquisa-experimental/>. Acesso em: 17 jun. 2022.

APÊNDICE A – FIRMWARE DE TESTES

```
#include "I2Cdev.h"
#include "Wire.h"
#include "MPU6050.h"

#define MPU 0x68 // Address I2C to communicate with the MPU
MPU6050 mpu(MPU); // Start the communication with the MPU 6050

void setup(){
  Wire.begin();
  Serial.begin(115200);

  mpu.setSleepEnabled(false);
  mpu.setClockSource(0);
  mpu.setTempSensorEnabled(false);
  mpu.setStandbyYGyroEnabled(true);
  mpu.setStandbyYGyroEnabled(true);
  mpu.setStandbyZGyroEnabled(true);
  mpu.setFullScaleAccelRange(1);
}

void loop(){
  x = mpu.getAccelerationX() / 8192.00;
  y = mpu.getAccelerationY() / 8192.00;
  z = mpu.getAccelerationZ() / 8192.00;

  Serial.print(x, 2);
  Serial.print(",");
  Serial.print(y, 2);
  Serial.print(",");
  Serial.println(z, 2);
  delay(10);
}
```

APÊNDICE B – CÓDIGO PYTHON PARA GERAR OS GRÁFICOS

```
import os
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

files = os.listdir("data")

for file in files:
    filename = file.split('.')[0]

    df = pd.read_csv('data/'+file)
    df.index = df['time']
    df = df.drop(['time'], axis=1)

    sns.set(rc={'figure.figsize':(80,20)}, context='poster', font_scale = 2, style='whitegrid')
    ax = df.plot()
    ax.set_xlim(xmin = 0)
    ax.set_ylim(ymin = -5, ymax = 5)
    ax.set_ylabel('g-force')
    ax.set_xlabel('time(s)')
    ax.set_xticklabels((df.index-1)*2)
    plt.savefig(filename + '.png')
```

APÊNDICE C – TERMO DE AUTORIZAÇÃO DE PUBLICAÇÃO DE PRODUÇÃO ACADÊMICA



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
GABINETE DO REITOR

Av. Universitária, 1069 • Setor Universitário
Caixa Postal 86 • CEP 74605-010
Goiânia • Goiás • Brasil
Fone: (62) 3946.1000
www.pucgoias.edu.br • reitoria@pucgoias.edu.br

RESOLUÇÃO n° 038/2020 – CEPE

ANEXO I

APÊNDICE ao TCC

Termo de autorização de publicação de produção acadêmica

O(A) estudante Wellington Alves dos Santos
do Curso de Engenharia de Computação, matrícula 20173003302819,
telefone: 62 9 93673299 e-mail wellington_gc@outlook.com na qualidade de titular dos
direitos autorais, em consonância com a Lei n° 9.610/98 (Lei dos Direitos do autor),
autoriza a Pontifícia Universidade Católica de Goiás (PUC Goiás) a disponibilizar o
Trabalho de Conclusão de Curso intitulado
Solução para detecção e alerta de queda com hardware de
baixo custo, gratuitamente, sem ressarcimento dos direitos autorais, por 5
(cinco) anos, conforme permissões do documento, em meio eletrônico, na rede mundial
de computadores, no formato especificado (Texto (PDF); Imagem (GIF ou JPEG); Som
(WAVE, MPEG, AIFF, SND); Vídeo (MPEG, MWV, AVI, QT); outros, específicos da
área; para fins de leitura e/ou impressão pela internet, a título de divulgação da
produção científica gerada nos cursos de graduação da PUC Goiás.

Goiânia, 25 de junho de 2022.

Assinatura do(s) autor(es): Wellington Alves dos Santos

Nome completo do autor: Wellington Alves dos Santos

Assinatura do professor-orientador: Ludmilla R. P. dos Santos

Nome completo do professor-orientador: Ludmilla Reis Pinheiro dos Santos