

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

JOÃO VITOR FERRO SIMÕES

**DISPOSITIVO PARA MONITORAMENTO DE ÂNGULOS E MOVIMENTOS
REPETITIVOS EM ATIVIDADES FÍSICAS**

PONTA GROSSA

2022

JOÃO VITOR FERRO SIMÕES

**DISPOSITIVO PARA MONITORAMENTO DE ÂNGULOS E MOVIMENTOS
REPETITIVOS EM ATIVIDADES FÍSICAS**

Device for monitoring angles and repetitive movements in physical activities

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Elétrica do Curso de Bacharelado em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Sergio Luiz Stevan Jr.

PONTA GROSSA

2022



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Esta licença permite download e compartilhamento do trabalho desde que sejam atribuídos créditos ao(s) autor(es), sem a possibilidade de alterá-lo ou utilizá-lo para fins comerciais. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

JOÃO VITOR FERRO SIMÕES

**DISPOSITIVO PARA MONITORAMENTO DE ÂNGULOS E MOVIMENTOS
REPETITIVOS EM ATIVIDADES FÍSICAS**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Bacharel em Engenharia Elétrica
do Curso de Bacharelado em Engenharia
Elétrica da Universidade Tecnológica Federal do
Paraná.

Data de aprovação: 20/junho/2022

Sergio Luiz Stevan Jr.

Doutorado

Universidade Tecnológica Federal do Paraná - UTFPR/PG

Hugo Valadares Siqueira

Doutorado

Universidade Tecnológica Federal do Paraná - UTFPR/PG

Cristhiane Gonçalves

Doutorado

Universidade Tecnológica Federal do Paraná - UTFPR/PG

PONTA GROSSA

2022

Dedico este trabalho ao meu pai Carlos Roberto (*in memoriam*) e minha mãe Simone, por sempre me apoiarem em minhas decisões, sem a força deles, eu não estaria aqui hoje.

AGRADECIMENTOS

Gostaria de agradecer a todas as pessoas que, através desses anos de universidade, me ajudaram tanto.

Agradeço ao meu professor orientador Prof. Dr. Sergio Luiz Stevan Junior, por toda a ajuda e força durante este trabalho.

Aos meus amigos de universidade por todo o apoio durante o curso. Embora não tenhamos mais tanto contato como antes, continuam em meu coração.

À minha família e amigos, por, mesmo nos momentos mais difíceis, me apoiaram e me ajudaram.

The moral thing I should wish to say to them is very simple: I should say love is wise, hatred is foolish. In this world which is getting more and more closely interconnected, we have to learn to tolerate each other; we have to learn to put up with the fact that some people say things we don't like. We can only live together in that way and if we are to live together and not die together, we must learn a kind of charity and a kind of tolerance, which is absolutely vital to the continuation of human life on this planet (RUSSEL, Bertrand, 1959).

RESUMO

O presente trabalho tem por objetivo desenvolver um dispositivo para identificação de angulação, amplitude de movimento e que realize a contagem de repetições utilizando um sensor inercial para aquisição de dados brutos de aceleração e velocidade. Através da análise dos dados obtidos de movimentos repetitivos do antebraço, foi possível definir métodos para a criação de algoritmos de cálculo de amplitude angular e contagem de repetições. Para a identificação de ângulos absolutos, foi utilizado um suporte de alumínio e um transferidor, realizando o alinhamento em posições específicas e aferindo o erro entre o valor lido no transferidor e o obtido pelo dispositivo. A identificação de amplitude e tempo de movimentos simples, além da utilização do suporte, utilizou o dispositivo acoplado ao antebraço. Para contagem de repetições o dispositivo foi acoplado ao torso para testes de abdominal, nas costas para testes de flexão e também, da mesma forma que anteriormente no braço. Foram obtidos erro absoluto máximo de $2,4^\circ$ para identificação de ângulo e erro absoluto máximo de 8° para identificação de amplitude de movimentos simples. Para repetições foram encontrados erros de contagem de no máximo dois movimentos. Observou-se então que o dispositivo não aparenta possuir fontes de erros associado à indicação de direção da velocidade e com isso pode ser utilizado para a identificação de sentido e repetição do movimento. Além disso, verifica-se que o trabalho desenvolvido obtém sucesso na contagem de repetição de movimentos, podendo ser útil para a aplicação em testes físicos ou reabilitação de pacientes em recuperação.

Palavras-chave: esp32; mpu6050; sensor inercial; atividades físicas.

ABSTRACT

The present work aims to develop a device for identification of angulation, range of motion and that performs the counting of repetitions using an inertial sensor to acquire raw data of acceleration and velocity. Through the analysis of data obtained from repetitive movements of the forearm, it was possible to define methods for the creation of algorithms for calculating angular amplitude and counting repetitions. For the identification of absolute angles, an aluminum support and a protractor were used, performing the alignment in specific positions and checking the error between the value read on the protractor and that obtained by the device. The identification of amplitude and time of simple movements, in addition to the use of the support, used the device attached to the forearm. To count repetitions, the device was attached to the torso for abdominal tests, on the back for flexion tests and also, in the same way as before on the arm. A maximum absolute error of 2.4° was obtained for identification of angle and a maximum absolute error of 8° for identification of simple range of motion. For repetitions, counting errors of at most two movements were found. It was then observed that the device does not appear to have sources of errors associated with the speed direction indication and thus can be used to identify the direction and repetition of the movement. In addition, it appears that the work developed is successful in counting the repetition of movements, which may be useful for application in physical tests or rehabilitation of patients in recovery.

Keywords: esp32; mpu6050; inertial sensor; physical activities.

LISTA DE FIGURAS

Figura 1 – Divisão da aptidão total	17
Figura 2 – Teste de flexão de braço - posicionamento e músculos trabalhados	20
Figura 3 – Teste abdominal com os joelhos flexionados	22
Figura 4 – Teste abdominal com os joelhos estendidos	22
Figura 5 – Graus de liberdade para um corpo de prova	23
Figura 6 – Placas de GY-512, do módulo inercial MPU6050	26
Figura 7 – Placas de prototipagem rápida baseadas em microcontroladores de baixo custo, sendo: a) Arduino Mega2560; b) Arduino UNO; c) Arduino Nano; d) shield de comunicação Bluetooth (para as placas Arduinos sem comunicação sem fio nativa); e) ESP01 (ESP8266); f) ESP32; g) ESP32 (com módulo de bateria)	28
Figura 8 – Placa de desenvolvimento baseada em ESP32 com suporte para bateria. . .	29
Figura 9 – Funcionalidades do MIT App Inventor 2	30
Figura 10 – Conexões entre o ESP32 e o GY-521	33
Figura 11 – Caixa impressa em 3D para acomodação de todos os componentes	33
Figura 12 – Fluxograma geral das aplicações possíveis	34
Figura 13 – Eixos cartesianos aplicados ao conjunto ESP32+MPU6050	35
Figura 14 – Ângulo ϕ e α	36
Figura 15 – Funcionamento do algoritmo de captura de ponto máximo e mínimo de um movimento simples	37
Figura 16 – Exemplificação do cálculo de velocidade e amplitude pelo algoritmo	37
Figura 17 – Fluxograma que representa o algoritmo de identificação de movimento . . .	38
Figura 18 – Exemplificação de identificação de movimentos repetitivos	39
Figura 19 – Suporte de alumínio, transferidor e impressão 3D de encaixe utilizados para validação da captura de Ângulos e movimentos	39
Figura 20 – Movimento realizado para identificação de amplitude angular	40
Figura 21 – Fluxograma do algoritmo para detecção de movimento	43
Figura 22 – Funcionamento de limiar de ativação para detecção de movimento	44
Figura 23 – Exemplo de identificação de parada de movimento baseado na inclinação da reta entre pares de pontos	44

Figura 24 – Exemplo de identificação de inversão de movimento baseado na inclinação da reta entre pares de pontos consecutivos	45
Figura 25 – Fluxograma representativo da tomada de decisão do código para identificação de início de movimento	46
Figura 26 – Diagrama de blocos para captura e armazenamento de dados brutos	47
Figura 27 – Diagrama de blocos para captura e armazenamento de dados de movimento único	47
Figura 28 – Diagrama de blocos para captura de movimentos repetitivos com limitador de tempo	48
Figura 29 – Diagrama de blocos para captura de movimentos repetidos baseados em séries e repetições	48
Figura 30 – Esquemático de transmissão de intervalos fixos definidos pelo ESP32	49
Figura 31 – Telas do aplicativo no dispositivo android	50
Figura 32 – Requisição gerada automaticamente pelo <i>smartphone</i>	51
Figura 33 – Aquisição dos ângulos absolutos pelo suporte	52
Figura 34 – Posicionamento inicial do braço e fixação do suporte	53
Figura 35 – Exemplo de momento com dados deslocados de um movimento de flexão do braço, sendo a) os dados coletados; b) os dados corrigidos; e c1 a c3) a indicação do movimento	54
Figura 36 – Conjunto de três coletas de dados para movimentos lentos com escala de 2g	55
Figura 37 – Conjunto de três coletas de dados para movimentos rápidos com escala de 2g	56
Figura 38 – Dados de angulação de movimento rápido para a escala 2g sendo (a) dado bruto; e (b) com ajuste condicional (para adequação entre 0 e 180°)	57
Figura 39 – Conjunto de três coletas de dados para movimentos lentos, repetindo-se em (a), (b) e (c) com escala em 8g	58
Figura 40 – Conjunto de três coletas de dados para movimentos rápidos com escala em 8g	59
Figura 41 – Dados de angulação de movimento rápido para escala de 8g, sendo (a) dado bruto; e (b) com ajuste condicional (para adequação entre 0 e 180°)	60
Figura 42 – Conjunto de de três coletas de dados para movimentos lentos com escala em 16g	61
Figura 43 – Conjunto de de três coletas de dados para movimentos rápidos com escala em 16g	62

Figura 44 – Dados de angulação de movimento rápido para escala de 16g, sendo (a) dado bruto; e (b) com ajuste condicional (para adequação entre 0 e 180°) . . .	62
Figura 45 – Conjunto de três coletas de dados de velocidade para movimentos lentos . . .	63
Figura 46 – Conjunto de três coletas de dados de velocidade para movimentos Rápidos .	64
Figura 47 – Conjunto de dados de velocidade e angulação normalizados. Velocidade na cor laranja e angulação obtida pelo acelerômetro na cor azul.	65
Figura 48 – Conjunto de três coletas de dados de angulação obtidos através do giroscópio com calibração inicial feita pelo acelerômetro	66
Figura 49 – Fixação do sensor para os testes de abdominal (no peito) e indicação do movimento.	68
Figura 50 – Fixação do sensor para os testes de flexão (nas costas) e a indicação do movimento	69
Figura 51 – Tela inicial projetada para o aplicativo	77
Figura 52 – Blocos para a programação dos 5 botões da tela inicial	78
Figura 53 – Blocos para a programação do botão de escolha de dispositivo <i>bluetooth</i> . .	78
Figura 54 – Segunda tela projetada do aplicativo, com função de iniciar a transmissão de dados brutos	79
Figura 55 – Blocos para a programação do botão "Iniciar"	80
Figura 56 – Blocos para a programação do componente invisível "Clock"	80
Figura 57 – Blocos para a programação do botão de "Salvar"	81
Figura 58 – Terceira tela projetada do aplicativo, com função de receber os dados de amplitude angular e tempo de movimentos simples	81
Figura 59 – Blocos para a programação do componente "Clock" com função de aguardar calibração e armazenar variáveis	81
Figura 60 – Quarta tela projetada do aplicativo, com função de receber aviso de que foi realizada uma repetição de movimento, e contar como repetição.	82
Figura 61 – Blocos para a programação do componente "Clock" com função de aguardar calibração e realizar contagem de movimentos	82
Figura 62 – Quinta tela projetada do aplicativo, em que o usuário pode escolher o número de repetições e séries de movimentos.	83

LISTA DE TABELAS

Tabela 1 – Tabela do teste de flexão de braço - avaliação da resistência muscular para homens	20
Tabela 2 – Tabela do teste de flexão de braço - avaliação da resistência muscular para mulheres	21
Tabela 3 – Teste de Abdominal - avaliação da resistência muscular localizada para homens	22
Tabela 4 – Teste de Abdominal - avaliação da resistência muscular localizada para mulheres	23
Tabela 5 – Parâmetros técnicos de sensores inerciais comerciais	25
Tabela 6 – Mapa parcial de registradores do sensor MPU6050	26
Tabela 7 – Conexões entre a placa GY-521 e o módulo TTGO	32
Tabela 8 – Tabela de resultados para angulação estática detectada pelo acelerômetro .	52
Tabela 9 – Tabela de resultados para teste de amplitude utilizando-se do suporte	67
Tabela 10 – Tabela de resultados para teste de amplitude utilizando-se do suporte	67
Tabela 11 – Tabela de resultados para repetições de flexão, abdominal e flexão de braço	69

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Objetivos	14
1.1.1	Objetivo Geral	15
1.1.2	Objetivos Específicos	15
1.2	Justificativa	15
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Delimitações dos Exercícios e do uso de dispositivos de monitoramento	17
2.1.1	Teste de Aptidão Física	17
2.1.1.1	Teste de flexão de braço	19
2.1.1.2	Teste de Abdominal	21
2.2	Sensores Inerciais	22
2.2.1	Sensor MPU6050	26
2.3	Microcontroladores de baixo custo	27
2.3.1	Módulo ESP32	28
2.4	Interface remota com o usuário	29
2.4.1	MIT App Inventor	30
3	MATERIAIS E MÉTODOS	32
3.1	Materiais	32
3.2	Métodos	34
3.2.1	Análise de angulação e velocidade	35
3.2.2	Análise de amplitude angular e tempo de movimento	36
3.2.3	Análise de repetições	37
3.2.4	Metodologia experimental	39
3.2.4.1	Captura de Ângulos	40
3.2.4.2	Identificação de amplitude e tempo de movimento simples	40
3.2.4.3	Identificação de repetições	41
3.2.5	Metodologia de avaliação dos resultados	41
3.2.6	Procedimento de processamento de dados	42
3.2.7	Comunicação entre o dispositivo de captura e o dispositivo Android	48
4	RESULTADOS	50

4.1	Aplicativo android	50
4.2	Análise da identificação de ângulos pelo acelerômetro	51
4.3	Análise dos dados	52
4.3.1	Análise dos dados de angulação	52
4.3.2	Análise dos dados de Velocidade	63
4.3.3	Análise dos conjuntos de dados	64
4.4	Análise de captura de movimentos simples	65
4.5	Análise da identificação de movimentos	68
5	CONCLUSÃO	71
	REFERÊNCIAS	72
	APÊNDICE A DESENVOLVIMENTO DO APLICATIVO ANDROID	77
	APÊNDICE B CÓDIGO DESENVOLVIDO NO ESP32	85

1 INTRODUÇÃO

Segundo dados do IBGE (2020) 40,3% da população é classificada como insuficientemente ativos. Para atividades físicas no lazer, apenas 34,2% dos homens praticam o nível recomendado de atividade física. Esta porcentagem cai para 26,4% para mulheres.

Em um estudo realizado por Ritter e Rigo (2016) foram utilizados *smartphones* e dispositivos *wearables* para captura de dados decorridos do monitoramento de atividades físicas em pacientes, obtendo aceitação satisfatória por profissionais da nutrição.

Morouço *et al.* (2021) desenvolveram um dispositivo aplicado para a área de natação, com o auxílio de um sensor inercial para obter dados de aceleração e velocidade e assim gerar indicadores de desempenho levando em consideração aspectos essenciais de natação como elevação de tronco e idealização de rotação de corpo durante os movimentos.

Utilizando um conjunto de sensores inercias e processando os dados através de um microcontrolador, Amaro e Aquino (2019) criaram um sistema de captura de movimento humano integrado a um sistema operacional robótico, que estima a localização de membros do corpo humano.

Da mesma maneira, Ramadhani, Wisana e Nugraha (2021) projetaram um dispositivo de monitor de apneia para detectar sintomas de apneia do sono. A contribuição neste estudo foi de um sistema de monitoramento ou monitoramento remoto para que outras pessoas possam acompanhar a condição do paciente mesmo que não o estejam acompanhando.

De maneira muito semelhante ao dispositivo desenvolvido neste trabalho, Itou, Inoue e Washino (2019) produziram um sensor inercial que fica ligado ao tórax ou abdômen que retira o vetor da aceleração, velocidade angular e deriva o quatérnion da função *Digital Measurements Processor* no módulo do sensor inercial e detecta a respiração a partir de alterações de séries temporais.

Assim, fica visível que a utilização de sensores inerciais para captura de movimentos humanos possui aplicabilidade também nas áreas da saúde, principalmente quando alinhadas com um microcontrolador para correto processamento e envio dos dados.

Tendo estes dados em mente, foi proposta a criação de um dispositivo de baixo custo para contagem de repetições, sendo aplicado para testes de abdominal e flexão. O dispositivo também é capaz de enviar dados brutos de aceleração e velocidade por *bluetooth* para um aplicativo desenvolvido para android, encarregado de armazenar os dados recebidos. Também foi desenvolvido um algoritmo de detecção de amplitude angular, podendo auxiliar em pesquisas futuras.

1.1 Objetivos

A seguir serão apresentados todos os objetivos gerais e específicos do presente Trabalho de Conclusão de Curso.

1.1.1 Objetivo Geral

O objetivo geral deste trabalho é desenvolver um dispositivo portátil para a identificação de ângulo e repetibilidade de movimento que possa ser utilizado como auxílio ou monitoramento para testes de aptidão física.

1.1.2 Objetivos Específicos

São objetivos específicos:

- Desenvolver um dispositivo capaz de realizar cálculo de ângulo absoluto, velocidade e que possa identificar amplitude de movimentos simples, além de realizar contagem de repetições;
- desenvolver uma interface para comunicação com o dispositivo em um *smartphone* baseada em android;
- analisar possíveis aplicações do dispositivo;
- validar o uso do dispositivo;
- avaliar os resultados obtidos e avaliar outras possíveis aplicações do dispositivo.

1.2 Justificativa

Os testes físicos são importantes indicadores da condição física e da avaliação da saúde das pessoas. Estes testes podem ser utilizados para avaliar a saúde de um grupo em específico ou de um único indivíduo. Profissões como bombeiro, polícia militar ou carteiros, necessitam de um bom condicionamento físico para atuarem em suas áreas e, por isso, a seleção em concursos públicos destes profissionais aplica uma bateria de testes físicos.

Além disso, os testes físicos podem ser úteis quando trata-se da recuperação de pacientes em fase de reabilitação de movimentos, pois são capazes de avaliarem condições físicas como equilíbrio, força nos membros e resistência. Para isso, uma série de repetições é realizada e a avaliação da capacidade física é determinada através da comparação com parâmetros existentes de capacidade física. Normalmente, estes testes são realizados por duas pessoas, onde uma realiza e outra faz a contagem de tempo e repetição. Com isso, dificuldades são apresentadas na realização deste processo. Em primeiro lugar, vê-se a dificuldade da realização destes testes por uma única pessoa, o que pode alterar os resultados de contagem e tempo, pois um único indivíduo pode se atrapalhar no controle dos dois dados simultâneos. Além disso, a realização em um grupo grande, pode acarretar um longo período para aplicação dos testes.

Assim, o dispositivo desenvolvido transmite, através de comandos específicos pelo protocolo *bluetooth*: os dados obtidos de angulação absoluta (em graus) e de velocidade (em graus por segundo); os dados de amplitude angular de um movimento simples com início e fim estáticos. O dispositivo também identifica movimentos repetitivos, enviando um aviso à cada vez que uma repetição é enviada, possibilitando que o aplicativo do *smartphone* realize a contagem de sinais e conseqüentemente de movimentos.

Para os casos estudados, após a análise dos dados e desenvolvimento das funções, o dispositivo visa auxiliar na contagem de exercícios de repetição, destacando-se os testes de aptidão física, com exibição de contagem em um *smartphone* android.

Portanto, verifica-se a importância deste estudo para solucionar os problemas relatados e proporcionar uma opção de auxílio de realização de testes físicos e em aplicações diversas que utilizem repetitividade. Além dos testes de aptidão física de repetibilidade, o dispositivo têm a importância de, através dos outros menus programados no aplicativo, facilitar a captura dos dados brutos para futuras pesquisas relacionadas ao sensor inercial e, através da identificação de amplitude e velocidade de movimentos simples, auxiliar em futuras pesquisas relacionadas à reabilitação e engenharia biomédica.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são abordadas as possibilidades de aplicações do dispositivo desenvolvido, levando em consideração os testes de aptidão física e especificamente os testes de flexão e abdominal. Também são estudados sensores inerciais para análise do movimento e a utilização de microcontroladores para processamento e transmissão dos dados. Também é demonstrado exemplos de interface com o usuário e especificando-se em dispositivos móveis para android.

2.1 Delimitações dos Exercícios e do uso de dispositivos de monitoramento

O presente trabalho apresenta o desenvolvimento de um dispositivo que captura ângulos absolutos com base nas acelerações e velocidades dadas pelo sensor inercial MPU6050. A partir disso, realiza-se cálculos para determinar sentido, amplitude e tempo de um movimento.

Propõe-se então diversas aplicações que podem ser realizadas com o dispositivo, tais como: testes de aptidão física, contagem de repetições, determinação de ângulo específico, exercícios de recuperação e reabilitação e, aplicações em fisioterapia.

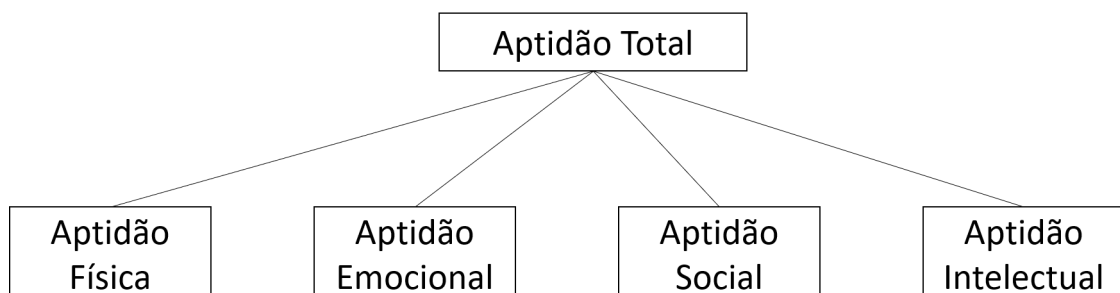
As subseções à seguir apresentam o ponto de vista de diversos autores, a respeito de 2 possíveis aplicações sugeridas: Teste de flexão e teste de abdominal.

2.1.1 Teste de Aptidão Física

O condicionamento físico do ser humano sempre foi um fator determinante para seu nível de saúde. Para Larson (1974), o ser humano é responsável pelo controle e direcionamento de sua saúde.

Para Hebbelinck (1984), a aptidão total de uma pessoa pode ser classificada em 4 itens: Aptidão física, Aptidão emocional, Aptidão social e a Aptidão intelectual, conforme a Figura 1.

Figura 1 – Divisão da aptidão total



Fonte: adaptado de (HEBBELINCK, 1984)

O termo aptidão física, pode ser conceituado segundo Caspersen, Powell e Christenson (1985), como a junção de habilidades adquiridas ou existentes em um indivíduo, relacionada com a capacidade de realizar atividades físicas.

Neste aspecto, a aptidão física pode envolver o aprimoramento da saúde, bem como o desenvolvimento de habilidades (força, flexibilidade, composição corporal e resistência aeróbica). Além disso, a aptidão física relaciona-se também com a agilidade, coordenação motora e velocidade (LARSON, 1974). Para BÖHME (1993) a aptidão física está relacionada diretamente à aptidão fisiológica de aspectos da saúde, como: força e resistência musculares, flexibilidade, resistência cardiorrespiratória e composição corporal.

Dessa forma, evidencia-se a importância da aptidão física para o ser humano e, os benefícios que seu desenvolvimento oferece.

Nesta perspectiva, criaram-se os testes de aptidão física (TAF). Os TAF possuem como principal objetivo, avaliar a condição física e de saúde de um indivíduo. Estes testes são muito utilizados para a avaliação de saúde de um determinado grupo, em concursos públicos para profissionais que necessitam de um bom condicionamento físico durante o exercício de suas profissões (bombeiros, policiais, etc) e para avaliação de desempenho esportivo (THOMAZ, 2003).

Existem diversos tipos de TAF, cada um direcionado para uma área específica de desenvolvimento humano. O autor Fleishman (1964) e a *American Alliance for Health, Physical Education and Recreation - AAHPERD* classificaram os TAF por baterias de avaliação da aptidão física.

Para Fleishman (1964), a aptidão física pode ser avaliada através de um conjunto de testes, sendo eles:

- Teste de Rotação do Tronco - Flexibilidade Extensiva
- Teste de Flexão e Rotação do Tronco - Flexibilidade Dinâmica
- Corrida de ida e volta - Força Explosiva
- Lançamento da Bola de Softbol - Força Explosiva
- Teste de preensão Manual - Força Estática
- Flexão e Extensão de Braços na Barra Fixa - Força Dinâmica
- Teste de Elevação das Pernas - Força de Tronco
- Teste de Salto sobre o Cabo - Coordenação Grossa Corporal
- Teste de Equilíbrio - Equilíbrio Grosso Corporal
- Corrida de 600 jardas (548 metros) - Resistência Cardiovascular

Para AAHPERD (1980), tem-se duas classificações: Testes e medidas e componentes de Aptidão física. Os testes de medidas e componentes podem ser citados por:

- Corrida de 9 minutos;
- Dobras Cutâneas Tricipital e Subescapular;
- Abdominal Modificado;
- Teste de Flexibilidade.

Ainda para AAHPERD (1980) os testes de componentes de Aptidão física podem ser classificados em:

- Resistência Cardiorrespiratória;
- Composição Corporal;
- Resistência/Força Muscular;
- Flexibilidade.

2.1.1.1 Teste de flexão de braço

O teste de flexão de braço possui como principal objetivo, a avaliação da força de resistência dos membros superiores (bíceps e grande pectoral) e extensão de antebraços (tríceps). Este exercício consiste na realização de flexões de braço, com o apoio ao solo (FILHO; NOVARES, 2013).

O teste de flexão de braço é de grande alteração positiva para a densidade mineral óssea, o que pode levar a diminuição de doenças cardiometabólicas (ORTEGA *et al.*, 2009). Além disso, este teste pode ser utilizado para reabilitação, uma vez que apresenta resultados significantes quando utilizados por profissionais de áreas da saúde (BARBOSA, 2015).

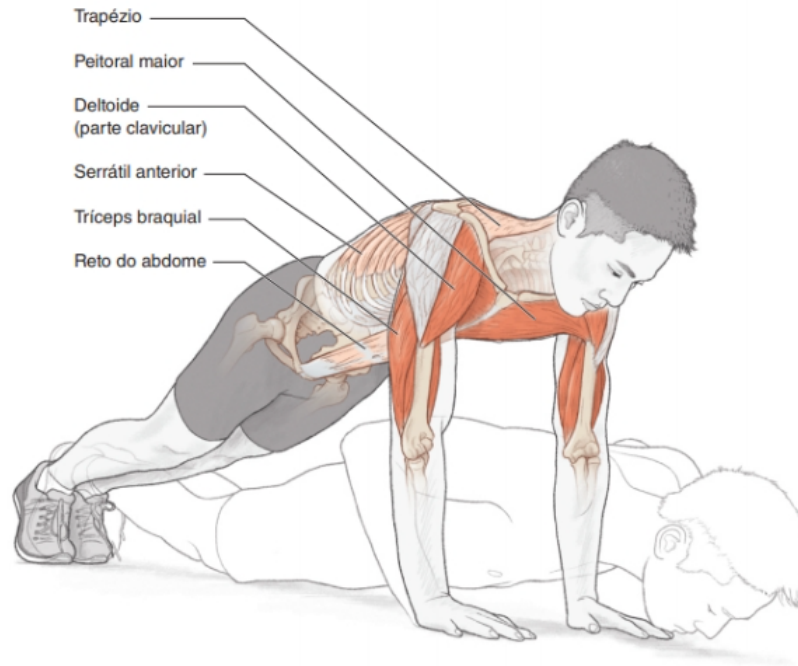
Para a realização do teste, existem algumas orientações. O teste deve ser realizado em um período de 1 minuto. O indivíduo deve estar posicionado em decúbito ventral, com o corpo apoiado nos pés e mão. As mãos devem estar apoiadas ao solo na linha dos ombros, com os dedos voltados para frente. Os cotovelos devem iniciar-se estendidos.

Após o posicionamento, deve-se realizar a máxima quantidade de flexões que conseguir, podendo descansar durante o tempo se necessário. Deve-se realizar flexões e extensões dos cotovelos no plano transversal até o tórax tocar o chão. Mulheres podem realizar o teste com o apoio dos joelhos ao solo. (POLLOCK; WILMORE, 1993).

A Figura 2 exemplifica a posição ideal para a realização do teste de flexão de braço e os músculos trabalhados durante a prática.

São consideradas flexões corretas, as que no momento de descida o cotovelo resultar em um ângulo de 90.º e no momento de descida, os braços estiverem totalmente esticados.

Figura 2 – Teste de flexão de braço - posicionamento e músculos trabalhados



Fonte: (HANSEN; KENNELLY, 1993)

Estes valores são comparados em tabelas disponíveis por autores, para a identificação do nível da capacidade física.

Os autores Pollock e Wilmore (1993), descrevem os resultados do teste de flexão de braço para homem e mulher, conforme idade e o número de repetições por minuto, como pode ser observado nas tabelas 1 e 2.

Tabela 1 – Tabela do teste de flexão de braço - avaliação da resistência muscular para homens

Idade	Excelente	Acima da Média	Média	Abaixo da Média	Fraco
15 - 19	+39	29 a 38	23 a 28	18 a 22	-17
20 - 29	+36	29 a 35	22 a 28	17 a 21	-16
30 - 39	+30	22 a 29	17 a 21	12 a 16	-11
40 - 49	+22	17 a 21	13 a 16	10 a 12	-09
50 a 59	+21	13 a 20	10 a 12	07 a 09	-06
60 - 69	+18	11 a 17	08 a 10	05 a 07	-04

Fonte: Adaptado de Pollock e Wilmore (1993)

Portanto, devido a importância do uso do teste de flexão de braço para a saúde, determinação de capacidade física e, avaliação de força, o dispositivo desenvolvido pode ser utilizado como auxílio à execução de tais testes.

Tabela 2 – Tabela do teste de flexão de braço - avaliação da resistência muscular para mulheres

Idade	Excelente	Acima da Média	Média	Abaixo da Média	Fraco
15 - 19	+33	25 a 32	18 a 24	12 a 17	-11
20 - 29	+30	21 a 29	15 a 20	10 a 14	-09
30 - 39	+27	20 a 26	13 a 19	08 a 12	-07
40 - 49	+24	15 a 23	11 a 14	05 a 10	-04
50 a 59	+21	11 a 22	07 a 10	02 a 06	-01
60 a 69	+17	12 a 16	05 a 11	02 a 04	-01

Fonte: Adaptado de Pollock e Wilmore (1993)

2.1.1.2 Teste de Abdominal

O teste abdominal trata-se de um TAF com principal objetivo de avaliação de força de resistência dos músculos abdominais. Estes exercícios, são de grande importância na composição dos exercícios físicos (SCHOENELL; AUTHERS, 1993).

O músculo abdominal possui grande influência no equilíbrio postural e no alinhamento de todo o corpo. A região abdominal fortalecida, pode prevenir diversas complicações posturais, tais como: lesões na coluna ou hérnias abdominais (LIMA, 2013).

O músculo abdominal deve ser fortalecido e possuir grandes resistências, visto que, este músculo é indispensável para a segurança dos órgãos internos. Além disso, é o grande responsável por manter o equilíbrio do corpo e controle da postura (DELFINO; HUMBER, 2013).

Para a realização deste teste, deve-se posicionar os joelhos em um ângulo de 90.º. O indivíduo deve deitar em posição de decúbito dorsal, com os pés apoiados sobre o solo, unidos e a uma distância de 30 a 45 cm do quadril. As mãos devem estar atrás da cabeça, com os dedos juntos. No momento da flexão abdominal, os cotovelos devem chegar até os joelhos (SCHOENELL; AUTHERS, 1993).

Para melhores resultados do teste, o indivíduo avaliado pode contar com a ajuda de uma pessoa, para segurar e firme seus pés sobre o solo. Assim, a força abdominal poderá resultar em um resultado mais preciso. Além disso, segundo, o teste pode ser realizado também com os joelhos flexionados (ABREU *et al.*, 2007).

A Figura 3, apresenta a posição adequada para a realização do teste proposto de abdominais com joelhos flexionados e a Figura 4 apresenta a posição adequada para realização do teste de abdominal com joelhos estendidos.

Para a avaliação do teste apresentado, alguns autores realizaram a avaliação do mesmo. Para que a referência seja a mesma utilizada na análise de resultados do teste de flexão de braços, utilizaremos os autores Pollock e Wilmore (1993), conforme as tabelas 3 e 4.

Figura 3 – Teste abdominal com os joelhos flexionados



Fonte: (ABREU *et al.*, 2007)

Figura 4 – Teste abdominal com os joelhos estendidos



Fonte: (ABREU *et al.*, 2007)

Tabela 3 – Teste de Abdominal - avaliação da resistência muscular localizada para homens

Idade	Excelente	Acima da Média	Média	Abaixo da Média	Fraco
15 - 19	+48	42 a 47	38 a 41	33 a 37	-32
20 - 29	+43	37 a 42	33 a 36	29 a 32	-28
30 - 39	+36	31 a 35	27 a 30	22 a 26	-21
40 - 49	+31	26 a 30	22 a 25	17 a 21	-16
50 - 59	+26	22 a 25	18 a 21	13 a 17	-12
60 - 69	+23	17 a 22	12 a 16	07 a 11	-06

Fonte: Adaptado de Pollock e Wilmore (1993)

Considerando a metodologia deste teste, pode-se propor uma sugestão para a aplicação do dispositivo, uma vez que seus requisitos são todos realizados pelo sistema. Além disso, o teste de abdominal apresenta-se muito relevante para a sociedade e com grandes benefícios para a saúde do ser humano.

2.2 Sensores Inerciais

Sensores inerciais são dispositivos micro-eletromecânicos (MEMS - *Micro Electro Mechanical Systems*) que medem velocidade e aceleração, angular ou linear através de transdutores mecânicos, que realizam o sensoriamento (TORRES, 2014).

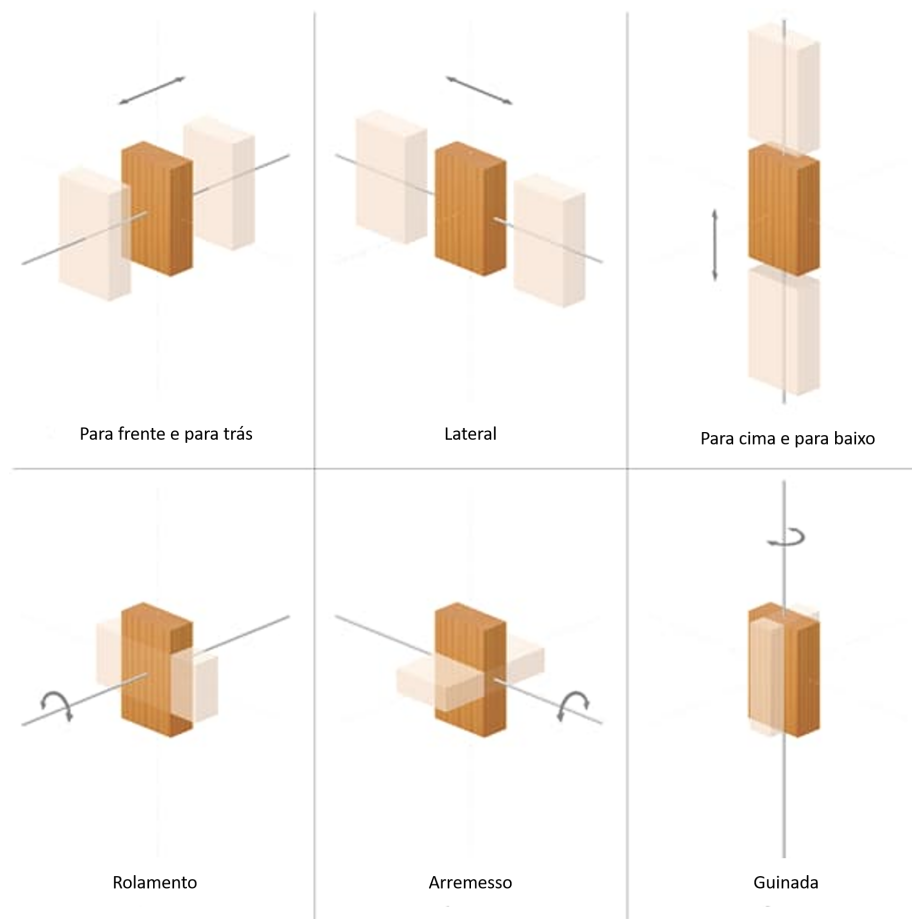
Tabela 4 – Teste de Abdominal - avaliação da resistência muscular localizada para mulheres

Idade	Excelente	Acima da Média	Média	Abaixo da Média	Fraco
15 - 19	+42	36 a 41	32 a 35	27 a 31	-26
20 - 29	+36	31 a 35	25 a 30	21 a 24	-20
30 - 39	+29	24 a 28	20 a 23	15 a 19	-06
40 - 49	+25	20 a 24	15 a 19	07 a 14	-06
50 - 59	+19	12 a 18	05 a 11	03 a 04	-02
60 - 69	+16	12 a 15	04 a 11	02 a 03	-01

Fonte: Adaptado de Pollock e Wilmore (1993)

Para decidir então, qual tipo de sensor inercial usar, é preciso atentar-se aos graus de liberdade disponíveis em cada um. Os graus de liberdade são componentes de movimentação que definem configuração ou estado. Para um corpo rígido no espaço tanto a translação quanto a rotação possuem três componentes (GOLATA, 2021). A Figura 5 ilustra os seis graus de liberdade e seus movimentos.

Figura 5 – Graus de liberdade para um corpo de prova



Fonte: Adaptado de (GOLATA, 2021)

Giroscópios, por medirem orientação nos eixos, possuem três graus de liberdade, acelerômetros, por medirem posição nos eixos, possuem mais três graus de liberdade. Já o mag-

netômetro pode medir o rumo direcional nos eixos, possuindo assim, mais três graus de liberdade(GOLATA, 2021).

O nível de acurácia e confiabilidade de sensores inerciais sugerem que eles podem ser utilizados para analisar movimentos repetitivos específicos em diversos contextos (CUESTA-VARGAS ALEJANDRO GALÁN-MERCANT, 2010). Todos os acelerômetros e giroscópios são passíveis de erro, limitando-se sua acurácia na medição de aceleração e velocidade angular. Estes erros podem ser sistemáticos ou aleatórios. Os erros sistemáticos incluem falhas inerentes ao processo de fabricação do tipo de sensor e dos componentes utilizados nele, já erros aleatórios provém de fontes de ruído, interferências ou instabilidade (BHATTACHARYYA S.; DE-PRETTERE, 2018)

A Tabela 5 apresenta diferentes modelos disponíveis em sensores inerciais de 16 bits de resolução, com características semelhantes, levando em consideração os graus de liberdade, a faixa de operação e sensibilidade do acelerômetro e giroscópio.

Tabela 5 – Parâmetros técnicos de sensores inerciais comerciais

Sensor	Graus de liberdade	Faixa de operação do acelerômetro (g)	Sensibilidade do acelerômetro(LSB/g)	Faixa de operação do giroscópio (°/s)	Sensibilidade do giroscópio(LSB/(°/s))	Preço médio (R\$) *
MPU6050		±2	16.384	±250	131	20,00
		±4	8.192	±500	65,5	
		±8	4.096	±1.000	32,8	
		±16	2.048	±2.000	16,4	
MPU9250		±2	16.384	±250	131	85,00
		±4	8.192	±500	65,5	
		±8	4.096	±1.000	32,8	
		±16	2.048	±2.000	16,4	
ICM20649		±4	8.192	±500	65,5	260,00
		±8	4.096	±1.000	32,8	
		±16	2.048	±2.000	16,4	
		±30	1.024	±4000	8,2	
BMI160		±2	16.384	±125	262,4	23,00
		±4	8.192	±250	131,2	
		±8	4.096	±500	65,6	
		±16	2.048	±1.000	32,8	
				±2.000	16,4	

Fonte: (INVENSENSE, 2012) (INVENSENSE, 2016) (INVENSENSE, 2021) (BOSCH, 2018)

* valores referentes a consulta em Julho de 2022

2.2.1 Sensor MPU6050

O módulo inercial MPU6050 conta com giroscópio e acelerômetro de três eixos cada, com conversores analógico-digitais de 16 bits integrados, podendo realizar a captura de todos os sensores de giroscópio e acelerômetro de forma simultânea, sem a necessidade de multiplexador. Possui tensão de operação entre 2,375V e 3,46V e corrente de operação máxima de 3,9mA. Ele utiliza comunicação I2C, agindo sempre como escravo e possuindo endereço fixo (INVENSENSE, 2012). O mapa de registradores do sensor MPU6050 para o acelerômetro, giroscópio e sensor de temperatura pode ser encontrado na Tabela 6.

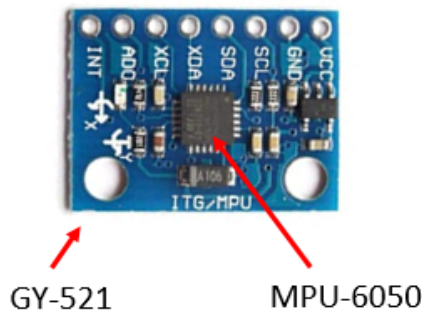
Tabela 6 – Mapa parcial de registradores do sensor MPU6050

Endereço	Nome do Registrador	Bits 0 até 7
3B	ACCEL_XOUT_H	ACCEL_XOUT[15:8]
3C	ACCEL_XOUT_L	ACCEL_XOUT[7:0]
3D	ACCEL_XOUT_H	ACCEL_YOUT[15:8]
3E	ACCEL_XOUT_L	ACCEL_YOUT[7:0]
3F	ACCEL_XOUT_H	ACCEL_ZOUT[15:8]
40	ACCEL_XOUT_L	ACCEL_ZOUT[7:0]
41	TEMP_XOUT_H	TEMP_OUT[15:8]
42	TEMP_XOUT_L	TEMP_OUT[7:0]
43	GYRO_XOUT_H	GYRO_XOUT[15:8]
44	GYRO_XOUT_L	GYRO_XOUT[7:0]
45	GYRO_XOUT_H	GYRO_YOUT[15:8]
46	GYRO_XOUT_L	GYRO_YOUT[7:0]
47	GYRO_XOUT_H	GYRO_ZOUT[15:8]
48	GYRO_XOUT_L	GYRO_ZOUT[7:0]

Fonte: Adaptado de InvenSense (2013)

O sensor MPU6050 possui a versão comercial em uma placa para prototipagem rápida denominada de GY-521, ilustrada na Figura 6.

Figura 6 – Placas de GY-521, do módulo inercial MPU6050



Fonte: adaptado de (EASYTRONICS, 2022)

Ela possui 8 pinos de interação, os quais estão indicados e descritos:

- Vcc - Pino de alimentação de 3 até 5 V;
- GND - Pino terra (referência);
- SCL - Pino de clock de saída para o mestre (I2C);
- SDA - Pino de dados de saída para o mestre (I2C);
- XCA - Pino de clock de saída para dispositivo auxiliar;
- XCL - Pino de dados de saída para dispositivo auxiliar;
- AD0 - Pino de definição do endereço, podendo ser 0x68 ou 0x69;
- INT - Interrupção para a saída.

Este sensor foi escolhido para este projeto devido ao seu custo-benefício apresentado para análise de erros simples, sendo o inercial mais barato da lista e que conta com resolução e sensibilidade parecida com os demais, tornando-o passível para o uso nas aplicações propostas.

2.3 Microcontroladores de baixo custo

Para realizar a aquisição e pré-processamento dos dados, levando em consideração o protocolo de comunicação I2C praticado pelo sensor inercial MPU6050, se faz necessário a utilização de um microcontrolador para aquisição, pré-processamento e processamento dos dados.

Os microcontroladores da família ATmega possuem memórias flash de 4 KBytes até 32 KBytes com EEPROM (memória estática) entre 256 Bytes e 1 KBytes e memória RAM (*Rapid Access Memory*) entre 512 Bytes e 2KBytes (MICROCHIP, 2020).

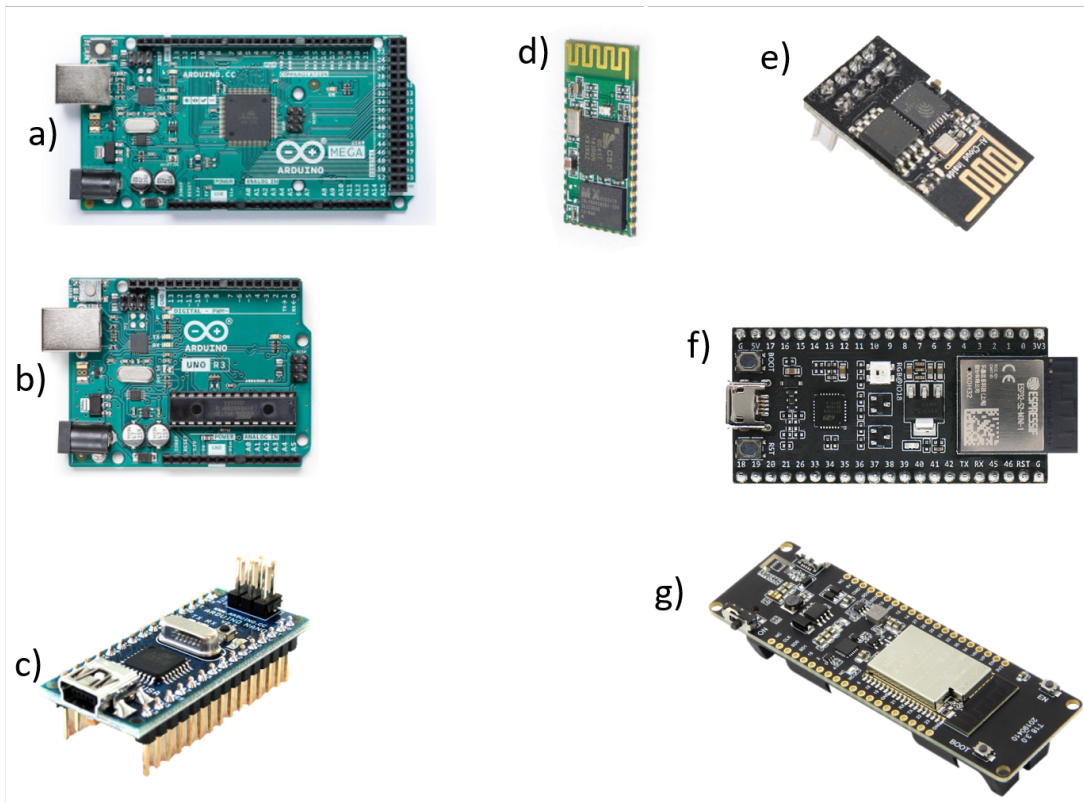
Baseado nos microcontroladores AVR ATmega, em 2005 na Itália, com o objetivo de obter um circuito de prototipagem de interface simples e de baixo custo, foi desenvolvida a interface Arduino (SILVA, 2019).

Outra opção de microcontrolador, o ESP8266, desenvolvido em 2014 pela Expressif Systems possui memória flash mínima de 512kBytes, memora RAM máxima utilizável de 50kBytes e possui como característica diferencial a utilização de protocolo de comunicação Wi-Fi com até 72,2Mbps de fluxo de dados (EXPRESSIF, 2018) .

Diversas opções de placas de prototipagem rápida podem ser utilizadas para este fim, como ilustrado na Figura 7. Algumas destas são muito disseminadas como a família Arduino (Mega2560, Uno e Nano, ilustrados na Figura 7 a, b e c, respectivamente), entretanto, não possuem comunicação sem fio nativa disponível. Para disporem de protocolos *bluetooth*, precisam de placas adicionais, como o HC-05, ilustrado na Figura 7 d). Já, outras famílias como a ESP, possuem *wi-fi* e *bluetooth* nativas (ESP01, ESP32, ESP32 com módulo de bateria - ilustrados

na Fig. 7 e), f) e g), respectivamente), o que lhes confere uma maior versabilidade aos projetos onde são aplicados.

Figura 7 – Placas de prototipagem rápida baseadas em microcontroladores de baixo custo, sendo: a) Arduino Mega2560; b) Arduino UNO; c) Arduino Nano; d) shield de comunicação Bluetooth (para as placas Arduinos sem comunicação sem fio nativa); e) ESP01 (ESP8266); f) ESP32; g) ESP32 (com módulo de bateria)



Fonte: adaptado de (ARDUINO, 2022c), (ARDUINO, 2022a), (ARDUINO, 2022b), (ITEAD, 2010), (AITHINKER, 2015), (ESPRESSIF, 2022), (LILYGO, 2019)

2.3.1 Módulo ESP32

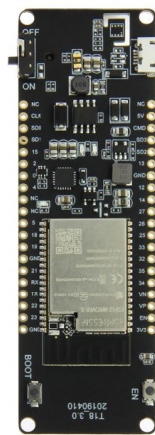
Como evolução do ESP8266 a Expressif Systems desenvolveu o ESP32, microcontrolador com memória flash de 4Mb, 520KBytes de RAM e 448KBytes de ROM, possui como característica diferencial a utilização de protocolo WiFi e Bluetooth, com processador de dois núcleos, capazes de operar individualmente ou em conjunto (EXPRESSIF, 2022).

O ESP32 pode ser programado através da própria interface Arduino em linguagem C ou em placas mais novas através de outras extensões em linguagem Micropython, atraindo as vantagens da programação orientada à objeto (MAIER; SHARP; VAGAPOV, 2017). O ESP32 é uma opção também para aplicações de IoT devido à suas características de construção,

propriedades e preço, superando seu predecessor ESP8266 devido, majoritariamente ao seu dois núcleos (BABIUCH; FOLTÝNEK; SMUTNÝ, 2019).

Como a principal vantagem do ESP32 é a comunicação sem fio, a fabricante TTGO criou uma placa de desenvolvimento baseada em ESP32 com suporte para bateria de lítio e limitador de carga. A placa desenvolvida pela TTGO encontra-se na Figura 8 sendo o suporte para a bateria acoplado ao lado oposto da placa.

Figura 8 – Placa de desenvolvimento baseada em ESP32 com suporte para bateria.



Fonte:(LILYGO, 2019)

Um dispositivo foi desenvolvido por Zhang e Cai (2021) com o objetivo da criação de um robô que coleta dados de inércia e dados do motor por meio do codificador Hall baseado em ESP32, utilizando o inercial MPU9250. Assim como Liu *et al.* (2021) realizaram um projeto de sistema de rastreamento automático de satélite de baixa órbita com base em ESP32.

Em outro estudo, Setiawan, Purnamasari *et al.* (2019) utilizaram o microcontrolador ESP32 para detectar a segurança de alojamentos de terra arumsari em Cirebon Distrito quando deixado por seus habitantes. Wen *et al.* (2022) desenvolveram um modo de desbloqueio de reconhecimento facial, para coletar, armazenar imagens e enviar para o computador por meio do barramento de dados com base em ESP32.

2.4 Interface remota com o usuário

Existem diferentes tipos de *interface* para interação ao usuário. São elas as responsáveis por transformar todo o sistema um dispositivo usual para quem o acessa. Saber se um *software* é adequado para o usuário, implica em uma série de testes e desenvolvimento das tarefas para que ele é utilizado (PRATES; BARBOSA, 2003).

Através da utilização de um microcontrolador, transmissor *bluetooth* e da interface android Fajrin *et al.* (2021) desenvolveram um aplicativo para monitoramento de batimentos cardíacos com alcance de até 13 metros.

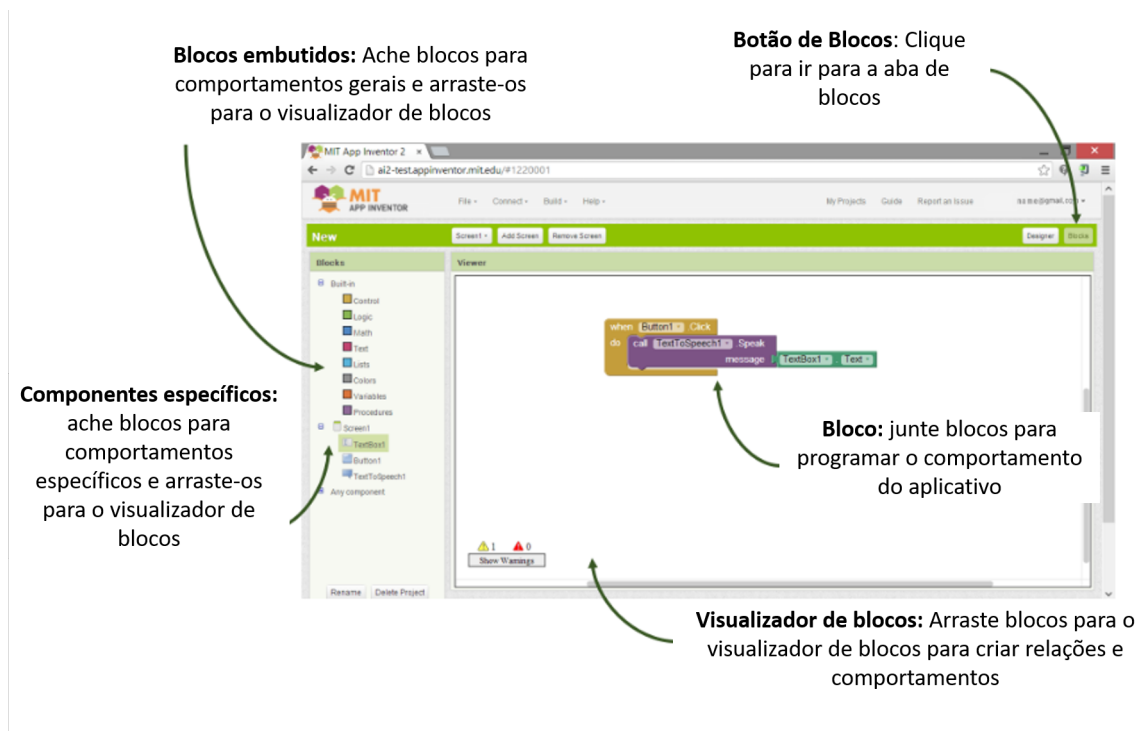
O Android é o sistema operacional móvel líder mundial e pertencente a Google. Essa plataforma para *smartphones*, baseia-se no sistema operacional Linux, que possui ferramentas para a criação de aplicativos pelos seus diversos componentes, com uma grande disponibilidade de *interface* gráfica e bibliotecas (CATAPAN, 2009). O sistema operacional android foi escolhido para o desenvolvimento deste trabalho. Dessa forma, a *interface* de interação escolhida torna-se fácil e de barato acesso é disponibilizada ao usuário.

Dessa forma, a aplicação Android trata-se de uma ferramenta eficaz e com grande possibilidade de apresentação de dados. Por isso, o dispositivo possui o envio de dados em tempo real por *bluetooth* a um aplicativo Android desenvolvido para essa finalidade. O ambiente de desenvolvimento do aplicativo da *interface* ao usuário, foi o MIT App Inventor.

2.4.1 MIT App Inventor

O MIT App inventor 2 é uma plataforma de desenvolvimento de aplicativos em forma de programação de blocos para sistemas android e IOS. Foi desenvolvido pelo MIT-CSAIL(*Massachusetts Institute of Technology - Computer Science and Artificial Inteligente Laboratory*) e possui 6 milhões de usuários registrados(INVENTOR, 2022). A figura 9 mostra o funcionamento da programação de blocos do MIT App inventor.

Figura 9 – Funcionalidades do MIT App Inventor 2



Fonte: (INVENTOR, 2022)

Esta IDE é um tipo de dispositivo de programação focada na contextualização em blocos que possibilita que qualquer pessoa, até mesmo um iniciante, comece a programar e desenvolver aplicativos funcionais para dispositivos Android. Esta ferramenta consiste em um tipo de *interface* fácil e ágil para o desenvolvimento de projetos (PRADO, 2017).

Segundo Cordeiro (2017) o App Inventor compõe de uma *interface* simples e de maneira fácil de usar. Este *software* não possui linhas de programação conforme o que vê-se atualmente e possibilita que os usuários, mesmo que não entendam de uma linguagem aprofundada de programação, produzam e publiquem seus aplicativos. Devido à função *drag and drop*, traduzido em português como: arrastar e soltar, a sua linguagem acontece de maneira muito intuitiva e fácil.

Uma das principais vantagens do uso do *App inventor*, é a facilidade em que seu desenvolvimento é oferecido. Sua programação acontece de maneira tão simples, que um dispositivo complexo pode ser desenvolvido rápido e de forma eficiente, visto que não exige programações de alta complexidade.

Portanto, escolheu-se este ambiente de desenvolvimento para a criação da *interface* do dispositivo. As informações completas do seu desenvolvimento serão expostas ao decorrer deste trabalho.

3 MATERIAIS E MÉTODOS

Neste capítulo são apresentados os materiais utilizados, as metodologias de análise, experimental e de validação dos resultados obtidos.

3.1 Materiais

Para o desenvolvimento do sistema, foram utilizados os seguintes materiais:

- Placa de desenvolvimento TTGO baseada em ESP32;
- Placa de sensor inercial Gy-521, baseada em MPU6050;
- *Smartphone* com sistema operacional android.

Realizou-se a comunicação entre as placas através das portas 21 e 22 do ESP32, configurando-as para comunicação I2C. A tabela 7 apresenta a conexão dos pinos do ESP32 com os pinos da placa GY-521.

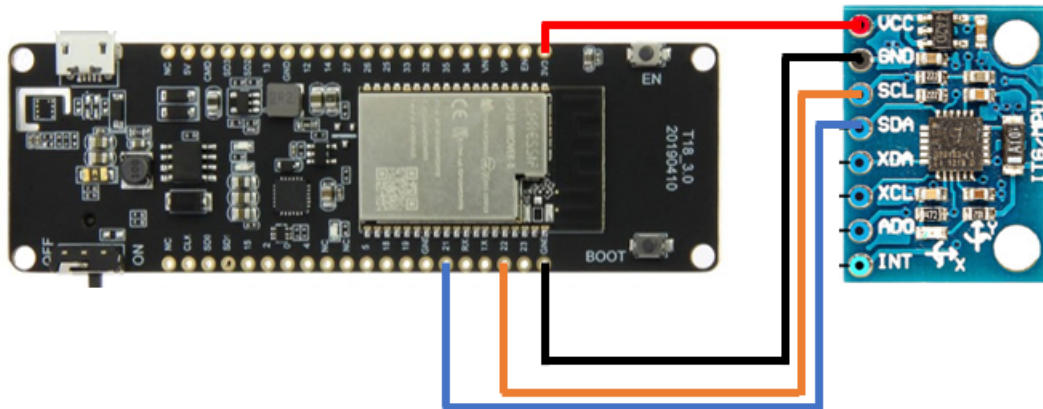
Tabela 7 – Conexões entre a placa GY-521 e o módulo TTGO

Pino ESP32	Pino GY-521
3V3	VCC
GND	DNG
21	SDA
22	SCL

Fonte: O autor

Na Figura 10 observa-se as conexões entre o ESP32 e a placa Gy-521, sendo esta soldada diretamente no microcontrolador através de uma placa perfurada. A soldagem auxilia a diminuição de ruídos provenientes da movimentação e diminui interferências que poderiam ser causadas pela movimentação de fios.

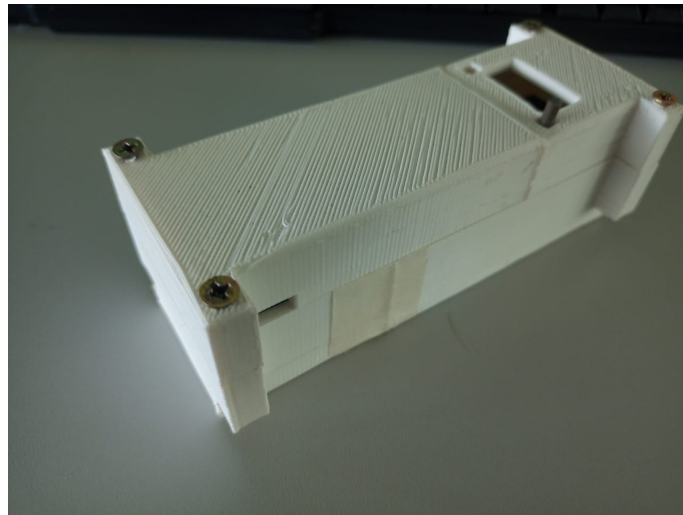
Figura 10 – Conexões entre o ESP32 e o GY-521



Fonte: O autor.

O ESP32 e o MPU6050, já soldados, foram acomodados em uma caixa desenvolvida em uma impressora 3D. Para sua confecção foi atribuído o material PLA *Polylactic Acid*, possuindo as dimensões de 35mm de largura por 40mm de altura por 120mm de comprimento. A caixa possui paredes de 3mm de espessura. Para o fechamento com parafusos foram feitos encaixes de 6mm de largura por 6mm de comprimento nos quatro cantos da caixa. Na Figura 11 ilustra-se a caixa impressa em 3D, que acomoda o circuito com o ESP32 e o sensor MPU6050.

Figura 11 – Caixa impressa em 3D para acomodação de todos os componentes



Fonte: O autor.

3.2 Métodos

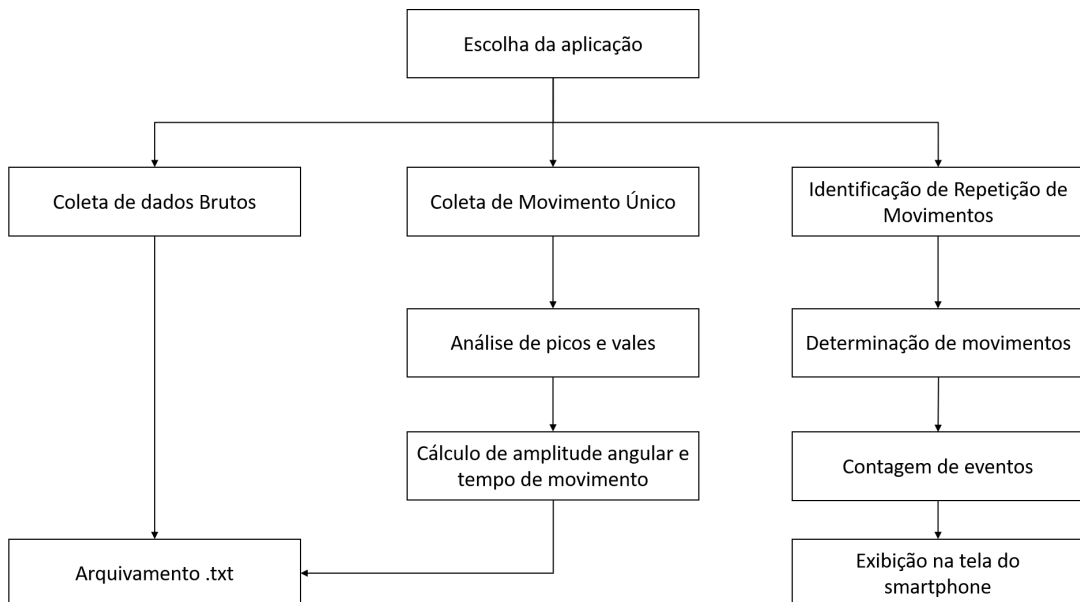
A placa GY-521 foi utilizada para a captura dos dados referentes à aceleração e velocidade angular. A placa baseada em ESP32 foi utilizada para realizar o pré-processamento, processamento e envio dos dados para o *smartphone*, que ficou encarregado de realizar o recebimento dos dados e a *interface* com o usuário.

Assim, através da combinação de sensor inercial, microcontrolador e *smartphone*, pretende-se utilizar o dispositivo para realizar a análise e identificação de angulação e velocidade, tanto de forma *online* quanto *offline*. Também se propõe a identificação de repetições de maneira *online*.

Para o recebimento e armazenamento dos dados brutos, processados e para a contagem de movimentos, utilizou-se um aplicativo android, criado a partir da *interface* de desenvolvimento MIT App Inventor 2. Para captura de valores brutos dos sensores, captura de ângulo e velocidade de movimento, os dados ficam salvos na memória interna do celular em formato .txt. Com isso, se faz necessária a permissão do aplicativo para acesso ao armazenamento do dispositivo Android.

Desta maneira, a Figura 12 representa as aplicações possíveis, selecionadas pelo usuário no menu do aplicativo.

Figura 12 – Fluxograma geral das aplicações possíveis



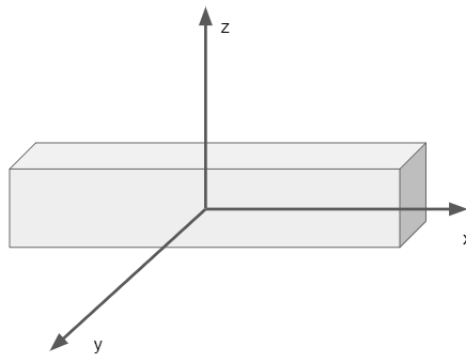
Fonte: O autor.

Os registradores utilizados do MPU6050 foram os registradores 3B até 48. Assim, foram obtidos 14 registradores que representam os valores de aceleração nos eixos X, Y e Z, registrador de temperatura (não utilizado) e três registradores de giroscópio. Dessa forma, através

da comunicação I2C, conseguiram-se os valores absolutos de aceleração e velocidade nestes eixos.

Na Figura 13, é possível observar os eixos cartesianos do conjunto ESP32 e MPU6050. Devido ao formato do conjunto e visando o conforto, o eixo X foi colocado de forma longitudinal ao membro a ser estudado, assim, as funções desenvolvidas levaram em consideração variações de velocidade e ângulo com relação ao eixo rotacional Y.

Figura 13 – Eixos cartesianos aplicados ao conjunto ESP32+MPU6050



Fonte: O autor.

3.2.1 Análise de angulação e velocidade

Para a análise de angulação e velocidade, foram feitas capturas dos dados brutos do acelerômetro e giroscópio. Foi verificado através de testes preliminares que a amostragem adequada se encontra em torno de 10 milissegundos (100Hz). Os ângulos ϕ e α , observados na Figura 14 são obtidos através das equações 1 e 2 respectivamente. A sensibilidade do acelerômetro foi calibrada para 2g, 8g e 16g.

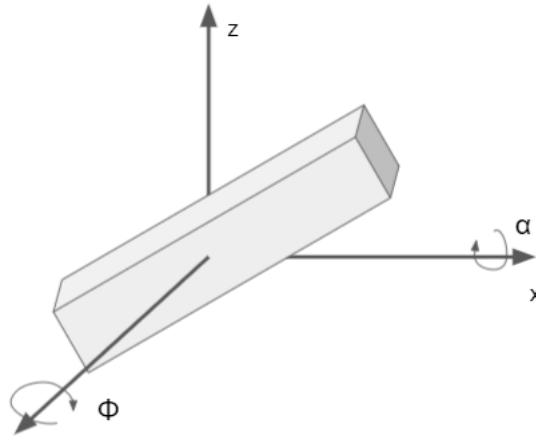
$$\phi = \arctan\left(\frac{A_x}{\sqrt{A_y^2 + A_z^2}}\right) \quad (1)$$

$$\alpha = \arctan\left(\frac{A_y}{\sqrt{A_x^2 + A_z^2}}\right) \quad (2)$$

Sendo:

- A_x a aceleração no eixo X

Figura 14 – Ângulo ϕ e α



Fonte: O autor.

- A_y a aceleração no eixo Y
- A_z a aceleração no eixo Z

Para o giroscópio, os dados de velocidade foram obtidos em radianos por segundo e convertidos para graus por segundo através da equação 3.

$$1 \text{ Radiano} = \frac{180.}{\pi} \quad (3)$$

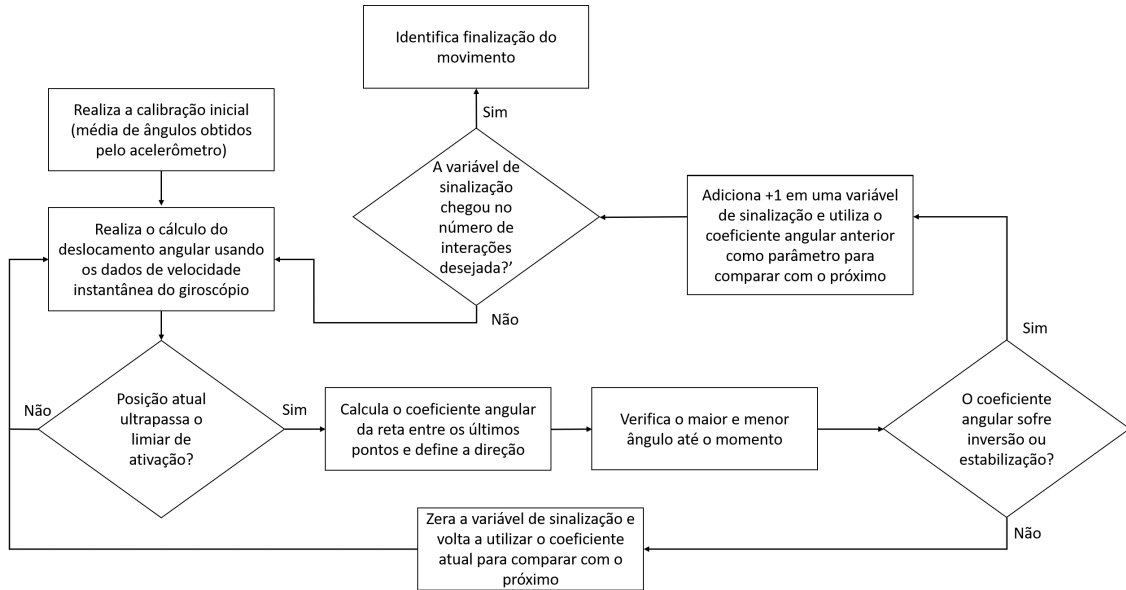
Os dados brutos foram analisados de forma *offline*, ou seja, foram salvos em formato ".txt" e processados posteriormente baseado nas características apresentadas.

3.2.2 Análise de amplitude angular e tempo de movimento

Através dos dados obtidos na etapa anterior foi desenvolvido um algoritmo de detecção de movimento, que, como pode ser observado pelo fluxograma da Figura 15, realiza, através da detecção de estabilidade do coeficiente de inclinação da reta, os cálculos de amplitude e tempo.

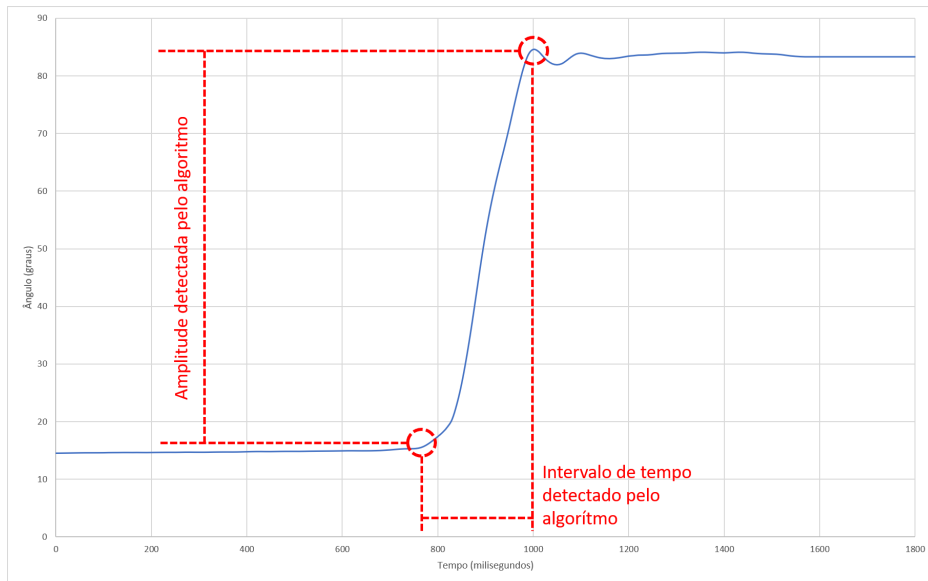
Porém, tanto o tempo do movimento quanto a amplitude levam em consideração o maior e menor ponto em graus, como pode ser visto na Figura 16, em que encontram-se circulos em vermelho a maior e menor posição em graus.

Figura 15 – Funcionamento do algoritmo de captura de ponto máximo e mínimo de um movimento simples



Fonte: O autor.

Figura 16 – Exemplificação do cálculo de velocidade e amplitude pelo algoritmo



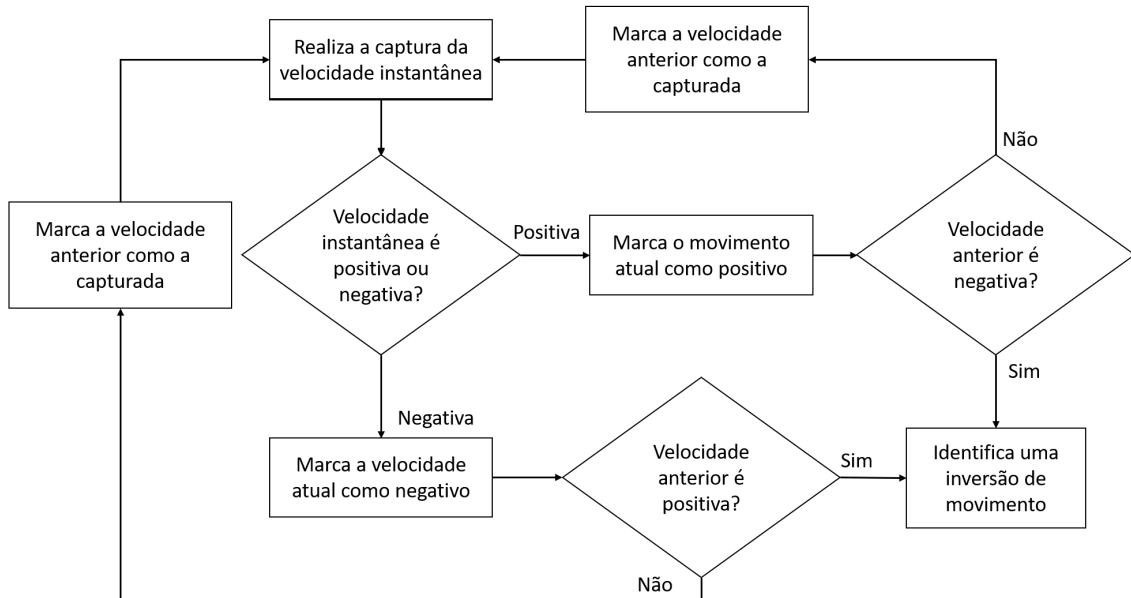
Fonte: O autor.

3.2.3 Análise de repetições

Levando em consideração os dados obtidos na etapa de análise de angulação e velocidade, foi desenvolvido um algoritmo de detecção de repetição.

O algoritmo proposto detecta inversão ou parada de movimento, levando em consideração a velocidade angular, posição inicial e direção inicial do movimento. O seu funcionamento pode ser descrito pela Figura 17.

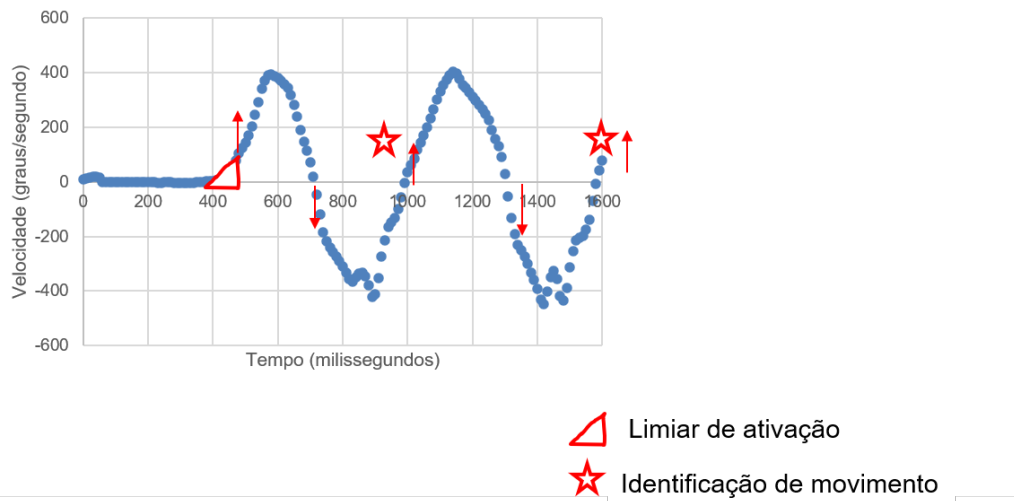
Figura 17 – Fluxograma que representa o algoritmo de identificação de movimento



Fonte: O autor.

Através do desenvolvimento do algoritmo, foi estabelecido que a cada duas inversões de movimento, à partir do início, seria considerado uma repetição (ida e volta). Assim, a Figura 18 mostra o funcionamento do algoritmo em um gráfico de velocidade por tempo, sendo as setas vermelhas indicativas da mudança de direção. O limiar de ativação neste caso é a integração da velocidade em função do tempo ou seja, utiliza-se o deslocamento em graus para identificar o início do movimento.

Figura 18 – Exemplificação de identificação de movimentos repetitivos

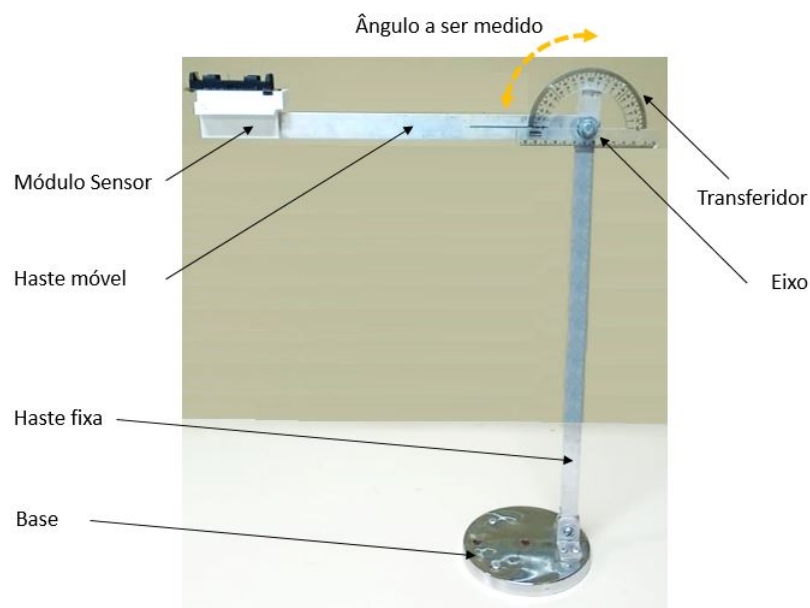


Fonte: O autor.

3.2.4 Metodologia experimental

Para a validação experimental foi utilizado o suporte de alumínio, ilustrado na Figura 19 para realizar a validação da captura de ângulos. Também na primeira etapa foram realizados testes de movimentação com o dispositivo acoplado ao antebraço, torso e costas.

Figura 19 – Suporte de alumínio, transferidor e impressão 3D de encaixe utilizados para validação da captura de Ângulos e movimentos



Fonte: O autor.

3.2.4.1 Captura de Ângulos

Para validar os ângulos absolutos do acelerômetro, o suporte de alumínio e o transferidor foram alinhados para que o ângulo 0° identificado pelo MPU6050 fosse o mesmo ângulo 0° indicado no transferidor.

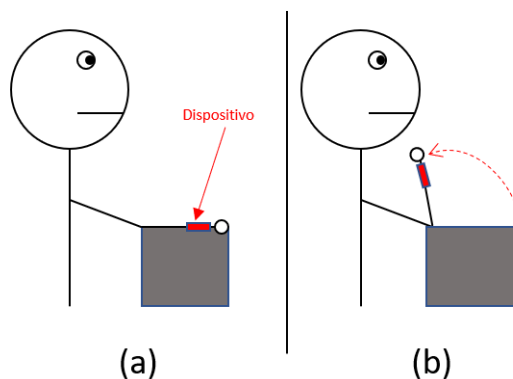
Foi feito então o posicionamento em diferentes ângulos da haste de alumínio com o ESP32 acoplado na ponta, assim, a cada 5° movidos, tendo-se como referência o transferidor, foram anotados os Ângulos obtidos pelo acelerômetro.

3.2.4.2 Identificação de amplitude e tempo de movimento simples

A identificação da captura de movimentos foi feita em um primeiro momento pelo suporte de alumínio, utilizando ângulos de 20° , 40° , 60° , 80° e 100° orientados por um transferidor. Como os movimentos foram realizados manualmente, estima-se um erro de $\pm 1^\circ$ para cada posição. Todos os movimentos foram repetidos três vezes cada.

Em um segundo momento a captura da angulação foi feita com o dispositivo acoplado ao braço do autor, realizando os movimentos mostrados na Figura 20, em que (a) representa a posição inicial e (b) a posição final, determinada pela amplitude máxima de contração possível do antebraço.

Figura 20 – Movimento realizado para identificação de amplitude angular



Fonte: O autor.

Como referência de angulação, utilizou-se uma captura de dados brutos, fazendo então a média dos ângulos máximos e a média de ângulos mínimos. Após a definição de referência de angulação mínima e máxima, foram feitos dez capturas com a amplitude total do braço, cinco com movimentos rápidos e cinco com movimentos lentos, utilizando o algoritmo identificador de movimentos para definir amplitude e velocidade.

Para a aquisição da velocidade dada pelo giroscópio não foi possível realizar a validação, pois seria necessário usar outros meios (como gravação de imagem com alta taxa de quadros

por segundo) para comparação, tendo-se que assumir um método como mais ou menos preciso. Supõe-se que a correta obtenção da amplitude angular de movimento pode ser associada à correta obtenção da velocidade, já que a variação angular está diretamente associada à velocidade instantânea obtida pelo giroscópio integrada pelo tempo decorrido entre as aquisições de dados.

3.2.4.3 Identificação de repetições

Através do terceiro menu do aplicativo, para a contagem de movimentos, o dispositivo foi acoplado nas costas do autor, realizando-se flexões por 15 segundos; Após, o dispositivo foi acoplado no torso do autor, sendo feito abdominais por 15 segundos e contando-os; Foram então feitos movimentos de flexão do braço, por 15 segundos. Para todas as etapas os movimentos foram repetidos cinco vezes.

3.2.5 Metodologia de avaliação dos resultados

Após a coleta dos dados de angulação obtidos do acelerômetro, os dados foram, caso a caso, inseridos em gráficos de dispersão, sendo o eixo x o tempo, em milissegundos e o eixo y o ângulo obtido, em graus.

Para a avaliação dos resultados obtidos pela captura de ângulos absolutos, utilizou-se a média de 700 pontos, equivalente a 7000 milissegundos de coleta entre os pontos de movimentação do suporte, descartando os pontos de movimentação ou estabilização. Estes dados foram inseridos em uma tabela comparativa que considera como posição real a indicada visualmente pelo transferidor. A tabela também conta com o erro absoluto para cada medida e o erro percentual.

Considerando os dados de angulação para movimentos do acelerômetro, os conjuntos de dados foram comparados entre movimentos lentos e rápidos, foram analisados conforme a distribuição dos pontos para as escalas de 2g, 8g e também 16g, levantando em cada um deles as fontes prováveis de erro.

Após a coleta dos dados de velocidade obtidos pelo giroscópio, novamente foram feitos gráficos de dispersão, com velocidade no eixo y (em graus por segundo) e tempo no eixo x (em milissegundos).

Os dados foram então analisados tanto para os casos de movimentos rápidos quanto para o caso de movimentos lentos, comparando o gráfico dos dados obtidos com o acelerômetro. Também foi realizada a análise do conjunto de dados do acelerômetro e giroscópio, ambos normalizados para movimentos lentos, assim, comparando a velocidade e ângulo no mesmo gráfico.

Utilizando o algoritmo para detecção de movimentos simples, com o suporte de alumínio, os dados de amplitude pretendida e amplitude obtida foram colocadas em uma tabela comparativa, fazendo-se a análise de erro absoluto e desvio padrão.

Para a detecção de movimento simples do braço, foi feita uma tabela que associa o tipo de movimento (lento ou rápido) com a amplitude obtida. Foi calculado a média destes movimentos separado por tipo e analisado conforme dados brutos obtidos do mesmo tipo de movimento.

Na comparação dos resultados em testes de aptidão física, os valores de referência das tabelas 1 e 3 foram divididos por 4, adequando os parâmetros aos 15 segundos realizados.

3.2.6 Procedimento de processamento de dados

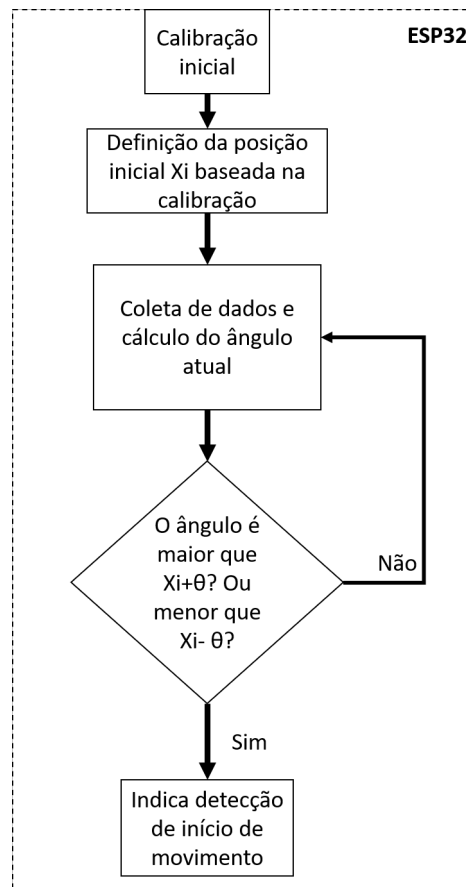
Foi necessário estabelecer conexões I2C de leitura e gravação através da biblioteca "Wire.h", utilizando-se do endereço específico da placa Gy-521.

Após estabelecida a conexão I2C, foi necessário realizar a leitura dos registradores do sensor MPU6050 pela função é feita através da função "Wire1.read()".

Para conseguir obter automaticamente o início do movimento, foi utilizada a técnica de limiar de ativação, então, para melhor detecção, foi definido um valor angular θ de 10° para mais e para menos. Assim, a partir de uma calibração de 1 segundo com amostragem de 100Hz em que o usuário necessita ficar parado na posição inicial, gerando uma média inicial de pontos denominada X_i .

O fluxograma ilustrado na Figura 21 apresenta o funcionamento do algoritmo de detecção de movimento que, para os experimentos realizados, foi de θ de 10° .

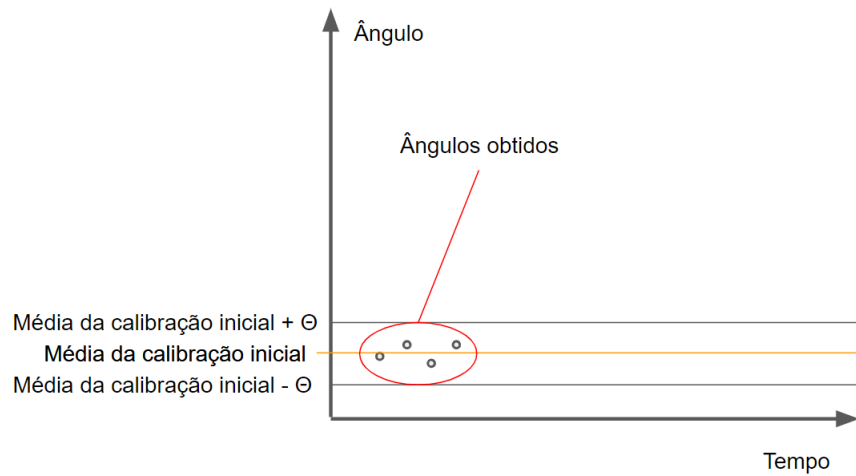
Figura 21 – Fluxograma do algoritmo para detecção de movimento



Fonte: O autor.

A figura 22 mostra como funciona o limiar de ativação para a média de pontos realizadas durante a calibração. No exemplo dado, nenhum dos pontos seria detectado como movimento, pois estão dentro do limiar de ativação.

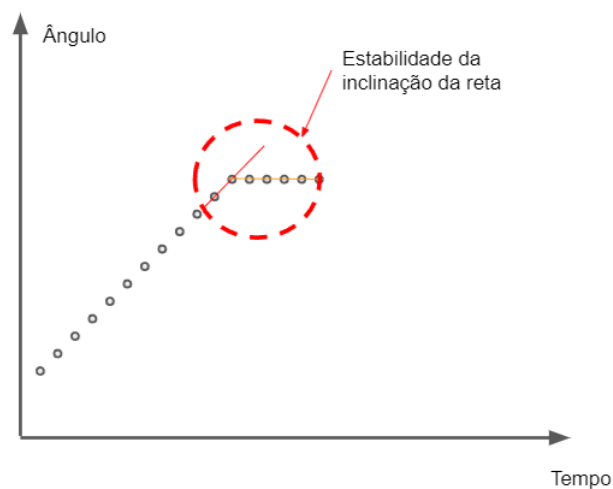
Figura 22 – Funcionamento de limiar de ativação para detecção de movimento



Fonte: O autor.

A detecção da amplitude do movimento foi realizada através da criação de um identificador de parada ou inversão de movimento. O algoritmo identificador de parada detecta se, após o início do movimento, ocorreu a parada do movimento. Isso acontece através do cálculo do coeficiente angular da reta referente aos ângulos com variação no tempo. Dessa forma, ao identificar o ponto que ocorre a estabilidade da inclinação da reta, o algoritmo supõe uma parada. Este comportamento pode ser observado como exemplo na Figura 23, que ilustra a parada do movimento baseando-se na equação da reta.

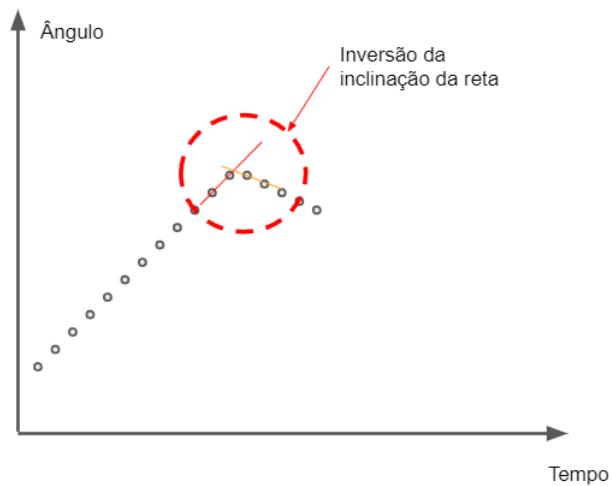
Figura 23 – Exemplo de identificação de parada de movimento baseado na inclinação da reta entre pares de pontos



Fonte: O autor.

O identificador de inversão de movimento detecta se a equação da reta referente aos últimos pares de pontos sofreu uma mudança de positiva para negativa ou negativa para positiva. A Figura 24 mostra como foi executada a identificação de inversão de movimento.

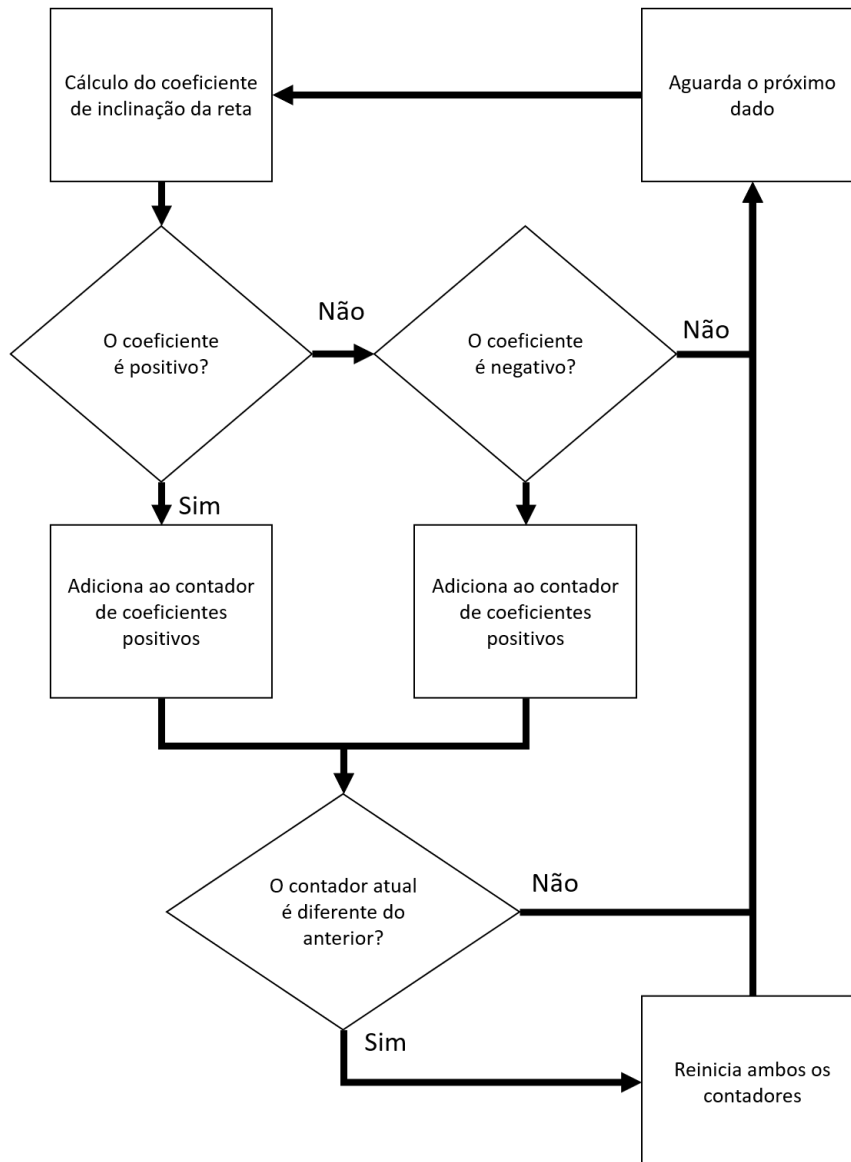
Figura 24 – Exemplo de identificação de inversão de movimento baseado na inclinação da reta entre pares de pontos consecutivos



Fonte: O autor.

O funcionamento das decisões tomadas pelo código para realizar a detecção de movimentos pode ser observado pelo fluxograma da Figura 25.

Figura 25 – Fluxograma representativo da tomada de decisão do código para identificação de início de movimento

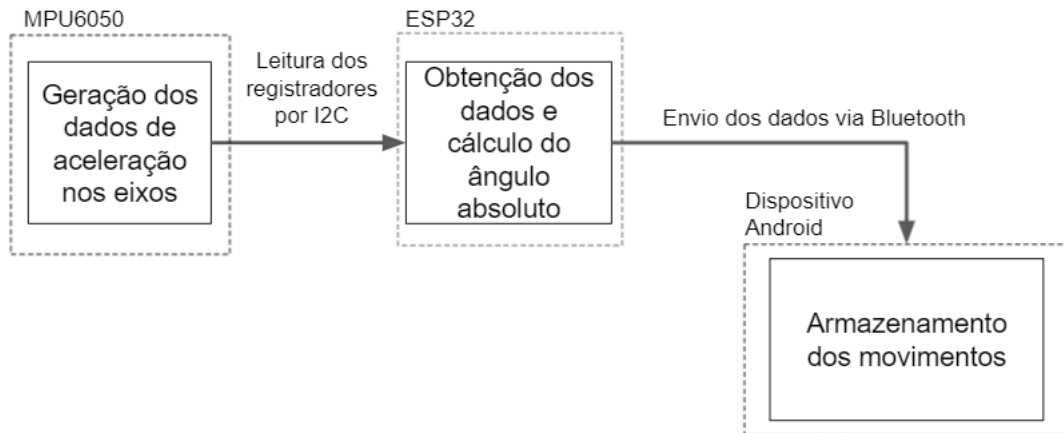


Fonte: O autor.

A captura da amplitude utiliza a diferença entre o menor e maior ângulo absoluto atingido. O cálculo do tempo de realização do movimento é realizado através da diferença entre o tempo de inversão ou estabilidade dos ângulos com o tempo de início do movimento. Esta diferença de tempo é realizada através da função "millis()", que fornece o tempo em milissegundos.

O funcionamento do sistema pode ser considerado em quatro diagrama de bloco. Cada um representa uma função que é escolhida pelo usuário no início do exercício, o diagrama de blocos representado pela Figura 26 mostra o funcionamento para a captura de dados brutos, ou seja, o dispositivo Android é encarregado apenas de receber os dados brutos de angulação do vetores espaciais x e y e armazená-los em um documento ".txt", no armazenamento interno do dispositivo.

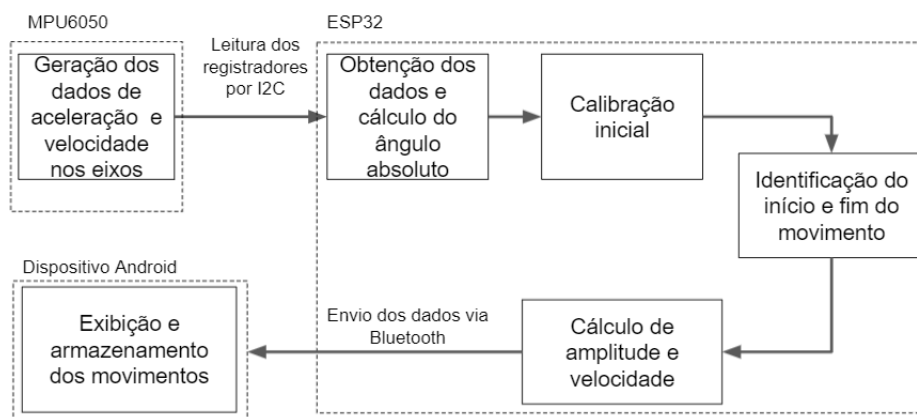
Figura 26 – Diagrama de blocos para captura e armazenamento de dados brutos



Fonte: O autor.

O segundo diagrama de blocos, representado pela Figura 27 mostra o funcionamento da função de captura de movimento simples, em que os dados que representam a amplitude e velocidade do movimento são armazenados um a um, com calibrações iniciais individuais. Desta forma, se torna possível realizar, em um mesmo conjunto de movimentos, diferentes identificações individuais de amplitude e velocidade, armazenando em quantos arquivos ".txt" forem necessários.

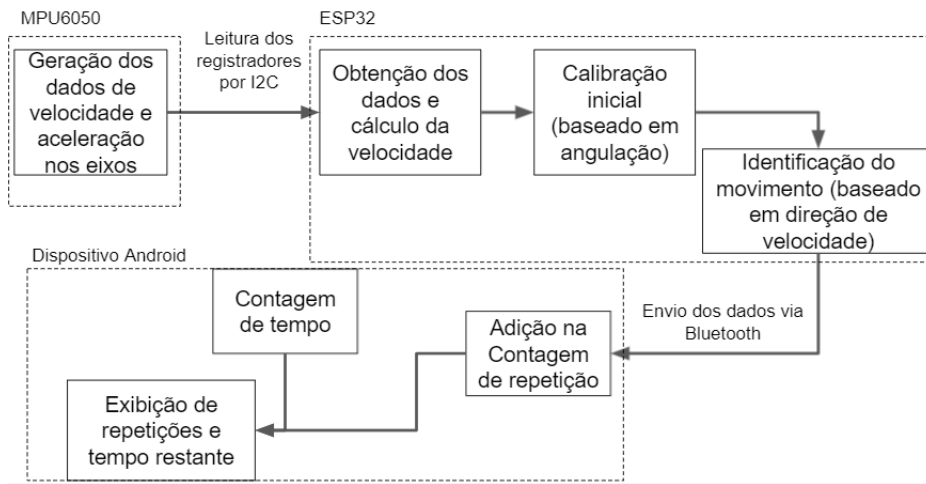
Figura 27 – Diagrama de blocos para captura e armazenamento de dados de movimento único



Fonte: O autor.

O terceiro diagrama de blocos, observado na Figura 28 demonstra a função de captura de movimentos repetitivos com limite de tempo. Para esta função, o ESP32 é responsável pelo processamento de dados brutos de velocidade e identificação do movimento, enquanto o dispositivo android é responsável pela contagem dos movimentos e do tempo de execução do exercício.

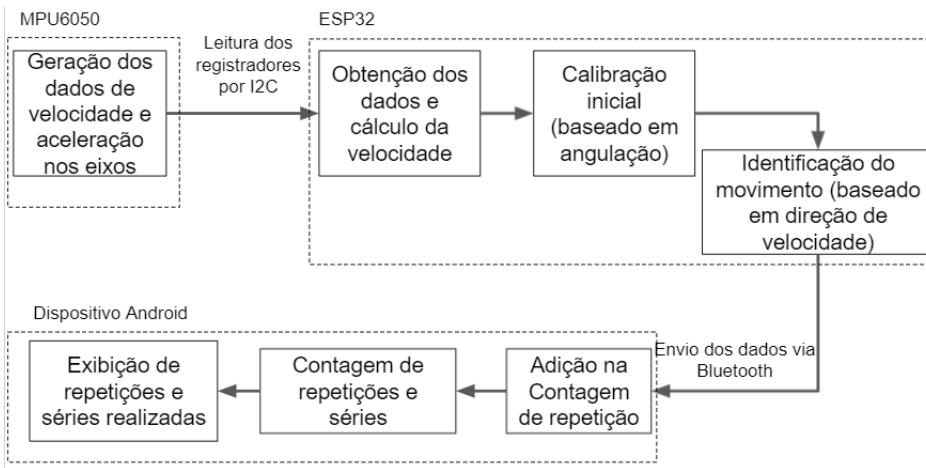
Figura 28 – Diagrama de blocos para captura de movimentos repetitivos com limitador de tempo



Fonte: O autor.

O quarto e último diagrama de blocos, demonstrado pela Figura 29 mostra a função em que é feita a captura de movimentos repetitivos baseados em séries e repetições, sem limite de tempo.

Figura 29 – Diagrama de blocos para captura de movimentos repetidos baseados em séries e repetições



Fonte: O autor.

A criação e programação do Aplicativo android é descrita no Apêndice A deste documento.

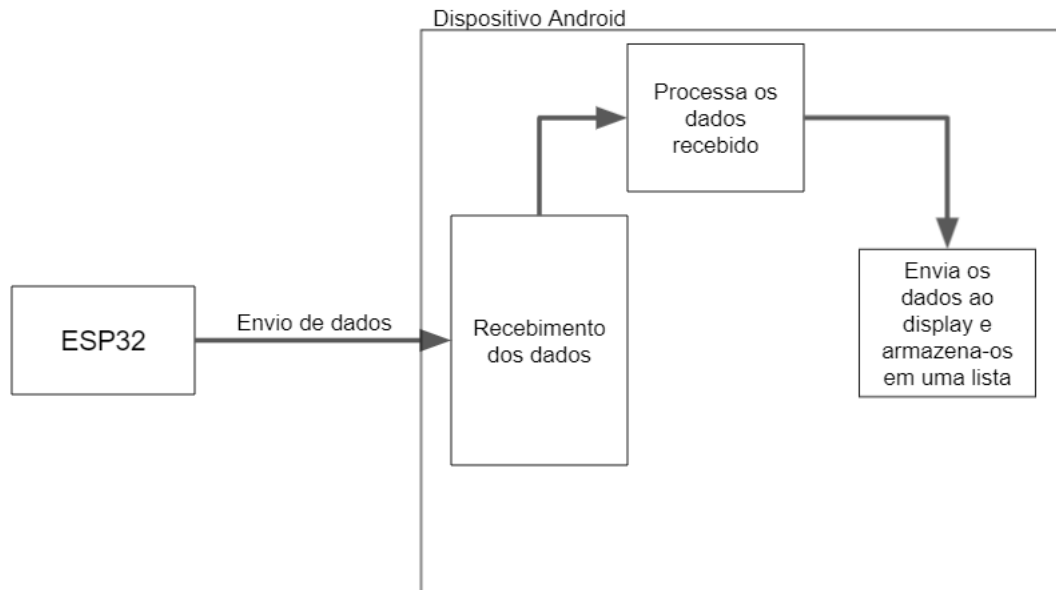
3.2.7 Comunicação entre o dispositivo de captura e o dispositivo Android

Visando a praticidade e facilidade para o usuário, a transmissão dos dados foi feita através de *bluetooth*. A comunicação foi realizada através de dados seriais pelo componente *Blueto-*

othClient do App Inventor e a biblioteca *BluetoothSerial.h* para ESP32. O tempo de amostragem para os dados e seu subsequente envio foi definido pelo *timer* do ESP32, projetado para ser executado a cada 10ms.

A Figura 30 mostra o diagrama de ações onde o *timer* do ESP32 é responsável pelo envio.

Figura 30 – Esquemático de transmissão de intervalos fixos definidos pelo ESP32



Fonte: O autor.

Identifica-se assim que ao usar o timer do ESP32 têm-se a possibilidade de obter intervalos de tempo mais precisos, porém, como o sistema definido pelo App Inventor não possui interrupções para identificar dados por bluetooth, pode-se ocorrer acúmulo de dados no *buffer* da porta Serial *bluetooth*.

Como os dados são transmitidos via porta serial, se faz necessário diminuir ao máximo o tamanho dos dados, assim, as variáveis para definição de início e parada foram do tipo *char*, possuindo apenas 1 byte de comprimento; os dados de ângulo, tanto para amplitude quanto para os dados brutos utilizaram o tipo *short int* (inteiros) que possui 2 bytes de comprimento; os dados de tempo foram enviados como *float*, com 4 bytes de comprimento.

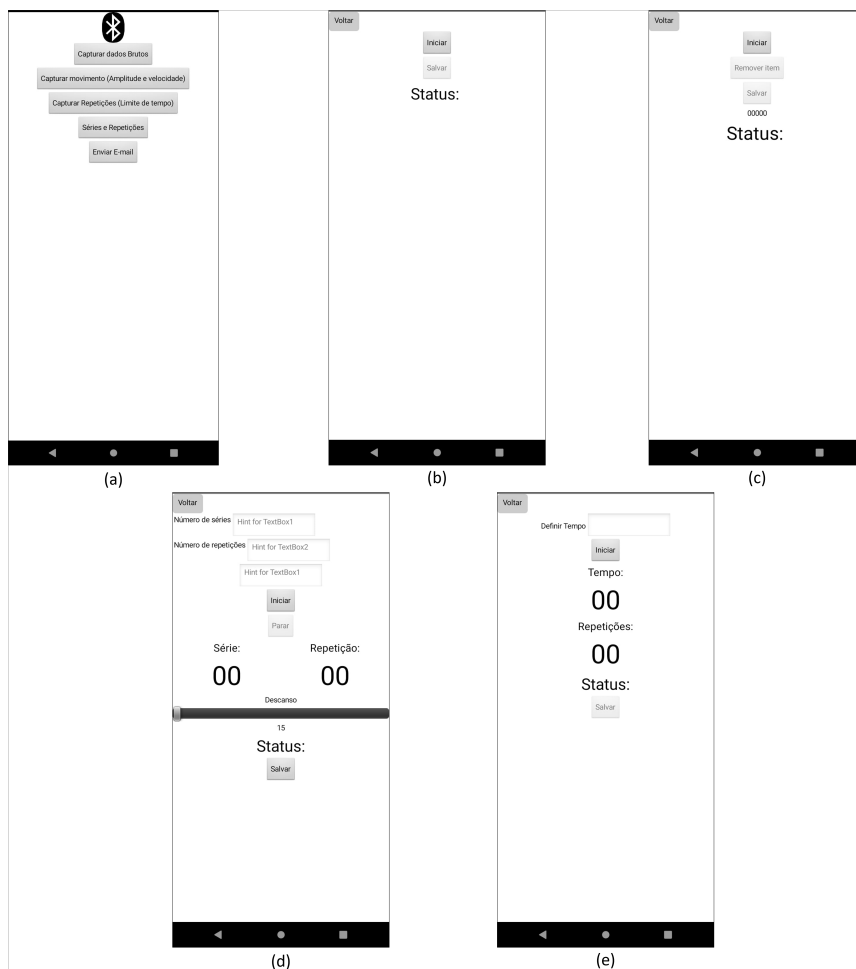
4 RESULTADOS

Neste capítulo serão apresentados os resultados para a análise dos dados, para o aplicativo android, para os algoritmos de identificação de amplitude de movimento e também para algoritmos de detecção de repetição.

4.1 Aplicativo android

Como a *interface* de desenvolvimento *MIT App Inventor 2* fornece um componente visual para organização dos itens, não ocorreram grandes diferenças visuais entre o desenvolvido e o obtido. Nas figuras 31 (a) até (e) é possível observar respectivamente, as telas inicial, tela de captura de dados brutos, tela de captura de amplitude de movimento, tela de identificação e contagem de movimentos baseado em tempo e, por último a tela de identificação e contagem de movimentos baseado em séries e repetições.

Figura 31 – Telas do aplicativo no dispositivo android

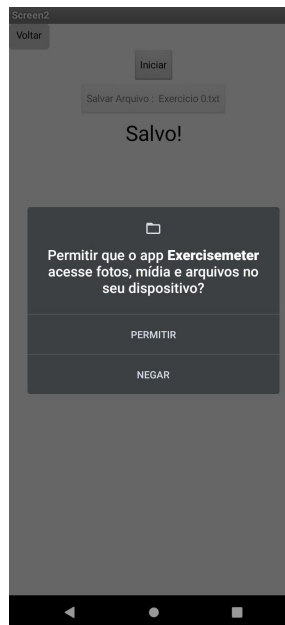


Fonte: O autor.

O aplicativo finalizado possui o tamanho de 10,4 Mb e, utiliza-se do armazenamento interno aberto do aparelho para armazenar os dados em .txt.

A Figura 32 apresenta o pedido gerado automaticamente ao clicar em qualquer botão "salvar". Esta etapa não foi configurada, pois o próprio sistema do *smartphone* que faz a requisição da permissão. Caso a requisição de permissão seja negada, se faz necessário autorizar nas configurações do aplicativo para o acesso ao armazenamento, do contrário, não é possível salvar as atividades realizadas.

Figura 32 – Requisição gerada automaticamente pelo *smartphone*

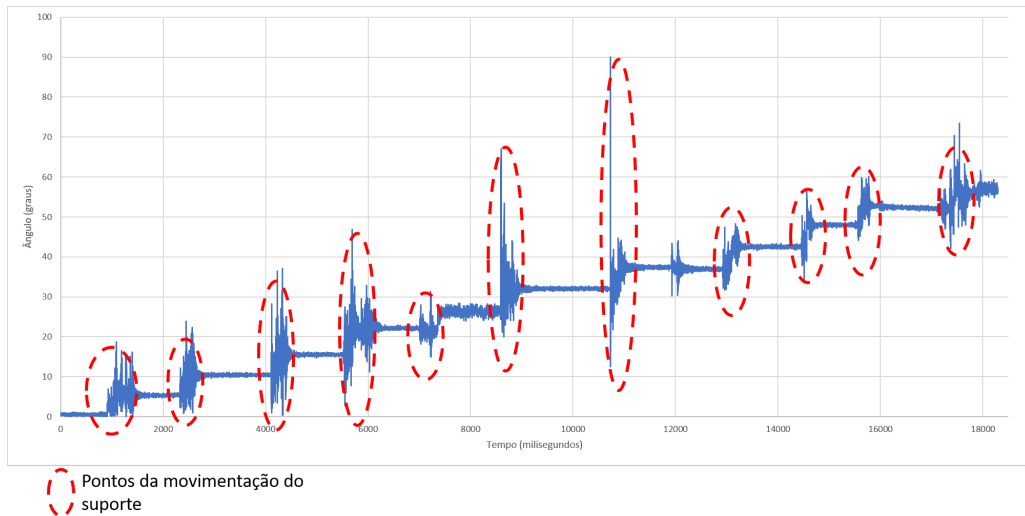


Fonte: O autor.

4.2 Análise da identificação de ângulos pelo acelerômetro

A identificação de ângulos pelo acelerômetro realizada por meio do suporte de alumínio, foi utilizada com a captura de dados brutos pelo primeiro menu do aplicativo. Após a estabilização do suporte, que inclui a movimentação até o ângulo desejado e aperto do parafuso para melhor estabilidade, utilizou-se a média de 700 pontos, equivalente a 7000 milissegundos de coleta entre as movimentações do suporte. Na Figura 33 são ilustrados os pontos obtidos e, circulado em vermelho estão os dados provenientes dos procedimentos de estabilização descritos anteriormente, os quais não foram considerados no cálculo da média de pontos. Assim, conforme a Tabela 8, um erro percentual médio de 4,26% e um erro absoluto médio de 1,29°.

Figura 33 – Aquisição dos ângulos absolutos pelo suporte



Fonte: O autor.

Tabela 8 – Tabela de resultados para angulação estática detectada pelo acelerômetro

Posição (°)	Valor médio obtido (°)	Erro Absoluto	Erro Percentual (%)
0	0,56	0,56	-
5	5,35	0,35	6,96
10	10,45	0,45	4,5
15	14,98	0,02	0,12
20	21,48	1,48	7,4
25	25,59	0,59	2,34
30	31,50	1,5	4,99
35	36,86	1,86	5,32
40	41,99	1,99	4,97
45	47,43	2,43	5,40
50	51,60	1,6	3,20
55	56,35	1,35	2,45
60	62,46	2,46	4,1

Fonte: O autor.

4.3 Análise dos dados

Para todos os casos de análise os dados brutos foram inseridos em um gráfico de dispersão de pontos, com a relação de angulação por tempo.

4.3.1 Análise dos dados de angulação

Para realizar uma rápida análise dos dados coletados, realizou-se o movimento ilustrado na Figura 35 (C1 a C3). O braço do autor foi posicionado conforme a posição indicada em (c1), com o sensor preso ao pulso. Posteriormente, descreveu-se uma angulação em torno de 150°

(c2) e, retornou-se o braço à posição inicial conforme indicado em (c3). A Figura 34 apresenta a fixação do sensor no braço e a posição inicial do movimento.

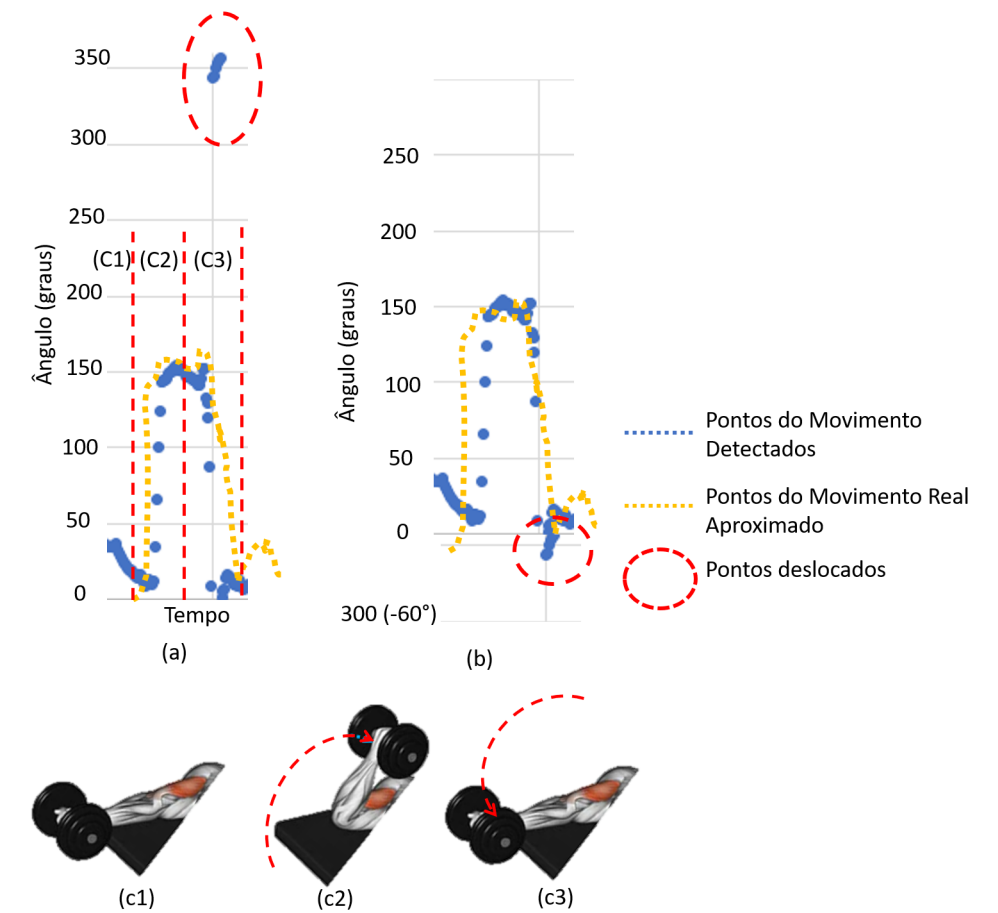
Figura 34 – Posicionamento inicial do braço e fixação do suporte



Fonte: O autor.

Verificou-se que o modo como o sensor interpreta os ângulos é limitado a translações de 360° . Assim, ao realizar o curso de uma angulação maior, o sensor transcreve os ângulo a partir do 0° (ex: 362° são ilustrados como 2°) e, de forma semelhante, ângulos abaixo de 0° são descritos abaixo de 360° (ex: -4° são ilustrados como 356°), como ilustrado na Figura 35 (a), a qual parte dos dados coletados no retorno do movimento, (de 150° para 0°) passam da origem e aparecem como se tivessem valores perto de 350 - 360° . Na Figura 35 (b), ilustram-se os dados realocados para estimar a posição angular real.

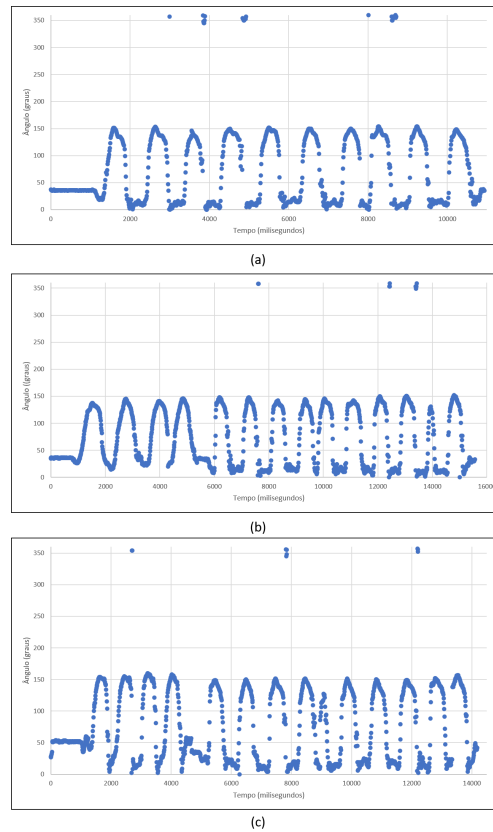
Figura 35 – Exemplo de momento com dados deslocados de um movimento de flexão do braço, sendo a) os dados coletados; b) os dados corrigidos; e c1 a c3) a indicação do movimento



Fonte: O autor.

Os dados do acelerômetro, captados com escala de 2g apresentaram acúmulo de pontos, tanto para movimentos lentos quanto para movimentos rápidos. Para movimentos lentos, como pode ser observado na Figura 36 foram detectados 5 acúmulos de pontos deslocados em (a) e, 3 acúmulos de pontos deslocados tanto (b) quanto para (C).

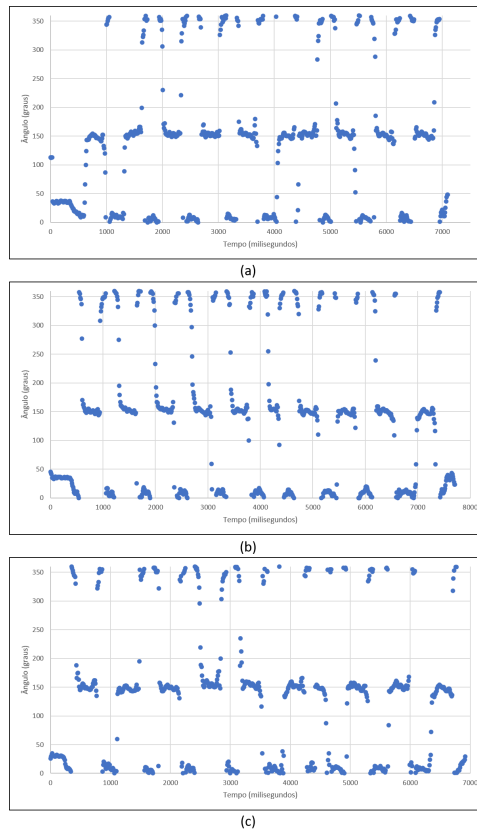
Figura 36 – Conjunto de três coletas de dados para movimentos lentos com escala de 2g



Fonte: O autor.

Para os movimentos rápidos foram detectados diversas regiões com deslocamento e acúmulo de pontos, tanto para o primeiro conjunto de movimentos realizados (a) quanto para os segundo e terceiro conjuntos de movimentos (b) e (c), observados na Figura 37.

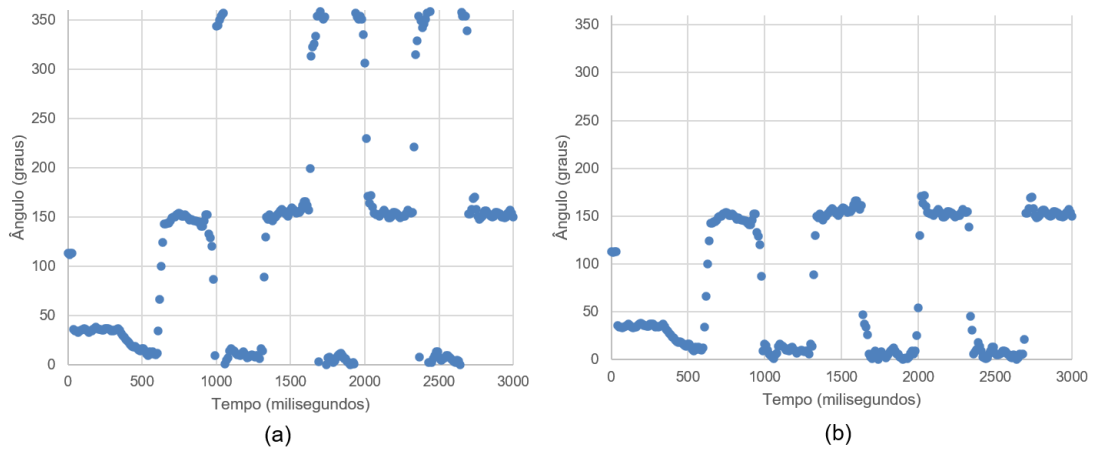
Figura 37 – Conjunto de três coletas de dados para movimentos rápidos com escala de 2g



Fonte: O autor.

Para melhor análise, observou-se o posicionamento dos pontos e tendo conhecimento da amplitude máxima do movimento, foi realizado um condicional baseado em limiar de ativação, limitando a amplitude do movimento em até 180° e, realizando a readequação dos dados que passarem deste valor, subtraindo o valor do ponto de 360° e rebatendo os dados identificados erroneamente no 3° e 4° quadrante para o 1° e 2° . A Figura 38 (b) apresenta a readequação dos pontos de movimentos do gráfico (a) utilizando o condicional.

Figura 38 – Dados de angulação de movimento rápido para a escala 2g sendo (a) dado bruto; e (b) com ajuste condicional (para adequação entre 0 e 180°)

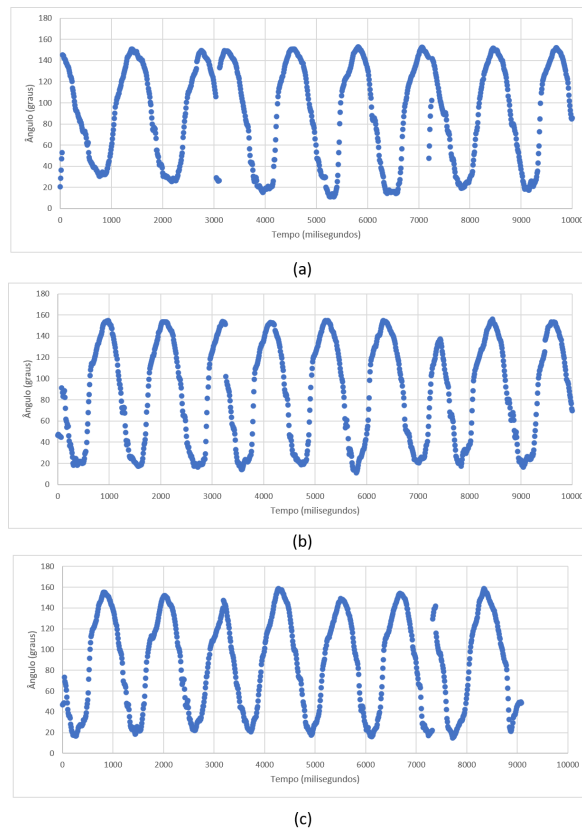


Fonte: O autor.

Supôs-se que os erros poderiam estar associados à escala de 2g, pois há um grande acúmulo de pontos e possuem poucos pontos médios entre o ângulo máximo e mínimo. Foram então refeitos os testes com escala de 8g.

Assim, fez-se a captura dos mesmos conjuntos de movimentos com escala de 8g, o qual pode ser observado no conjunto de gráficos da Figura 39, não exibiu para nenhum dos três conjunto de dados (a), (b) ou (c) deslocamentos de dados fora do intervalo 0° até 180°.

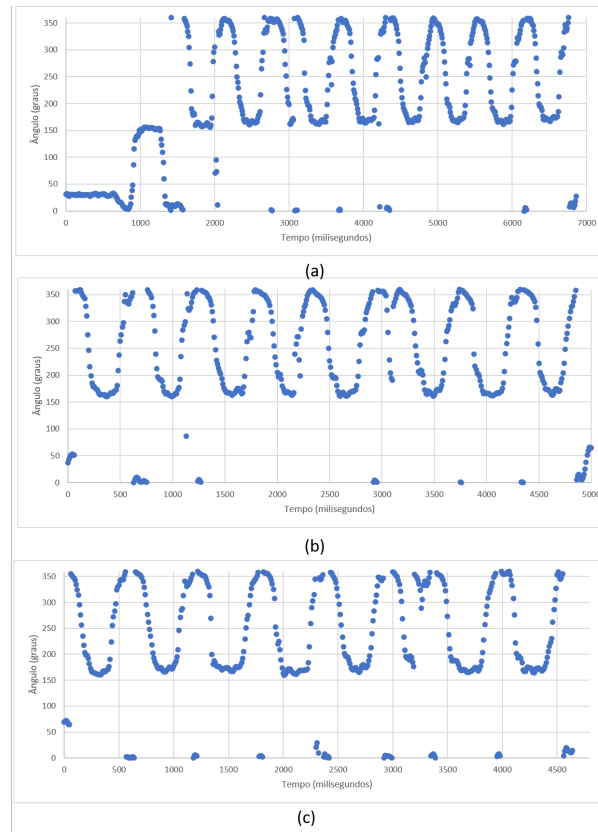
Figura 39 – Conjunto de três coletas de dados para movimentos lentos, repetindo-se em (a), (b) e (c) com escala em 8g



Fonte: O autor.

Para os conjuntos de movimentos rápidos, observou-se um deslocamento nos dados, mantendo-se entre 150° e 360° . Na Figura 40 item (a), a alternância entre movimentos rápidos e lentos gera dois tipos de escala para o mesmo movimento, já para (b) e (c), mantém-se o padrão de deslocamento, já que não ocorreu alternância na velocidade.

Figura 40 – Conjunto de três coletas de dados para movimentos rápidos com escala em 8g

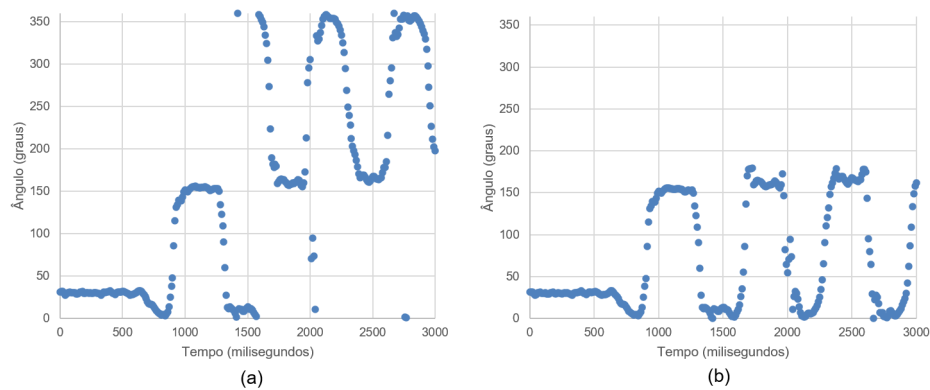


Fonte: O autor.

Observando os erros obtidos e modificando os dados do gráfico para pontos, identificou-se um possível espelhamento nos eixos, pois, como o acelerômetro trabalha em uma faixa circular de 1° até 360° , constatou-se que os dados para movimentos rápidos estavam deslocados de forma padronizada.

Aplicou-se então a mesma técnica de condicional utilizada para a escala de 2g, assim, a Figura 41 (b) apresenta os dados da mesma Figura, item (a), rebatidos para o intervalo de 0° até 180° .

Figura 41 – Dados de angulação de movimento rápido para escala de 8g, sendo (a) dado bruto; e (b) com ajuste condicional (para adequação entre 0 e 180°)



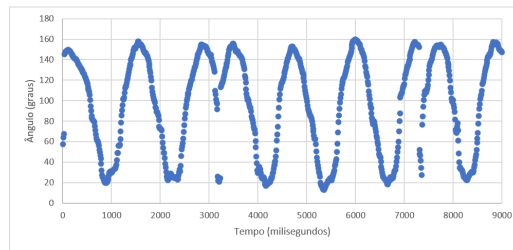
Fonte: O autor.

Embora os pontos tenham sido readequados para o 1° e 2° quadrantes (de 0° até 180°), ainda ocorre um acúmulo de pontos nas inversões de movimento com pontos intermediários que indicam o percurso do movimento angular, diferente da escala 2g.

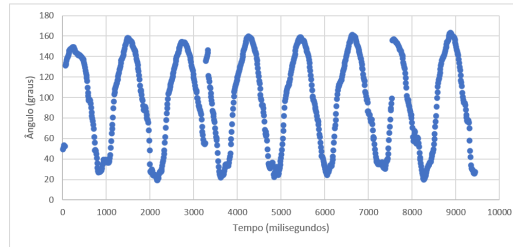
Mudando a escala do acelerômetro para 16g, como pode ser visto na Figura 42, não foram detectados deslocamentos de dados para movimentos lentos em nenhum dos conjuntos de repetições (a), (b) ou (c).

Nos movimentos rápidos, o deslocamento dos dados segue o mesmo padrão conforme a escala 8g, como pode ser observado na Figura 43, nos conjuntos de repetições (a), (b) e (c), que representam os movimentos rápidos para escala de 16g.

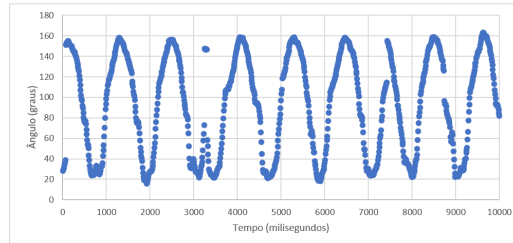
Figura 42 – Conjunto de de três coletas de dados para movimentos lentos com escala em 16g



(a)



(b)

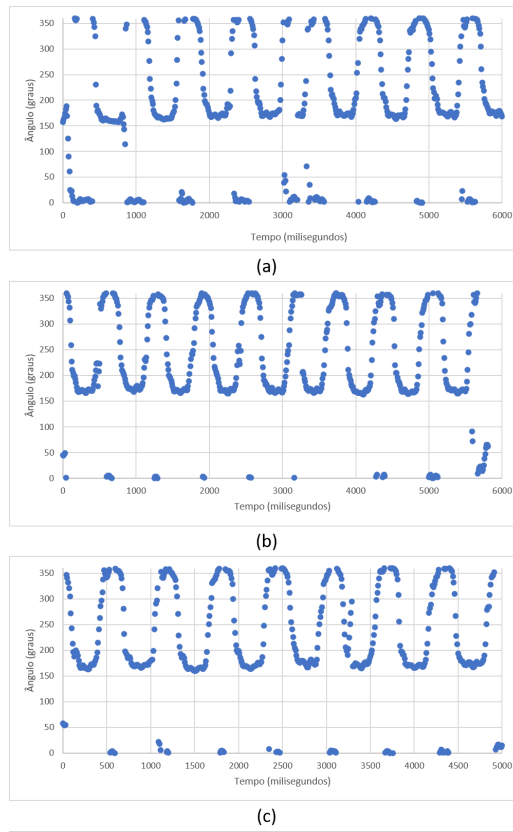


(c)

Fonte: O autor.

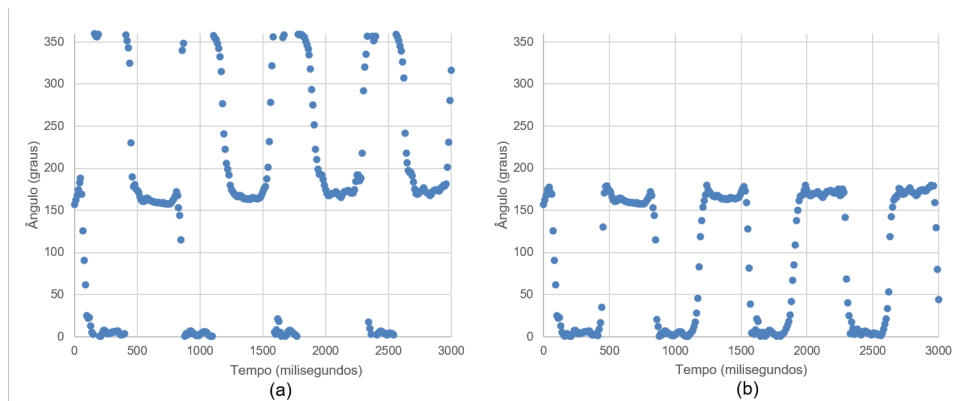
Foi então aplicado o mesmo procedimento de condicionamento de dados para a Figura 44 item (a) entre 0° e 180° , gerando o gráfico da Figura 44 item (b). Observa-se novamente acúmulo de pontos nas inversões de movimento ainda com pontos intermediários.

Figura 43 – Conjunto de de três coletas de dados para movimentos rápidos com escala em 16g



Fonte: O autor.

Figura 44 – Dados de angulação de movimento rápido para escala de 16g, sendo (a) dado bruto; e (b) com ajuste condicional (para adequação entre 0 e 180°)



Fonte: O autor.

Por meio desta análise dos dados, foi possível observar a necessidade de outro método para contar movimentos de forma precisa, pois, mesmo com o objetivo de identificar apenas inversão de movimento, o deslocamentos dos pontos gerados pela variação de velocidade faz

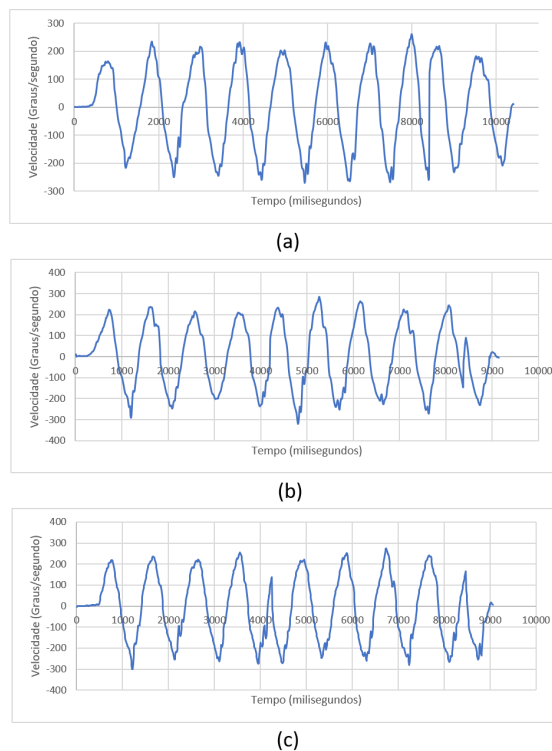
com que se torne inviável a utilização dos dados brutos do acelerômetro para identificação do movimento.

Já com a utilização condicional proposto, teoricamente é possível realizar a identificação de movimento por meio de limiares de ativação, porém, isso restringiria os movimentos ao primeiro e segundo quadrante, entre 0° e 180° . O filtro condicional poderia ser utilizado para identificação de amplitude angular em uma série de movimentos repetitivos, necessitando-se apenas identificar a diferença angular entre o maior e menor acúmulo de pontos.

4.3.2 Análise dos dados de Velocidade

Para a análise dos dados brutos do giroscópio, que são dados em radianos por segundos, convertendo-os para graus e, utilizando-se o mesmo conjunto de movimentos, foi identificado conforme a Figura 45 que os movimentos considerados lentos, tanto para as repetições (a), (b) e (c), não possuem pontos deslocados, tornando-se possível observar a velocidade positiva, a parada e a velocidade negativa.

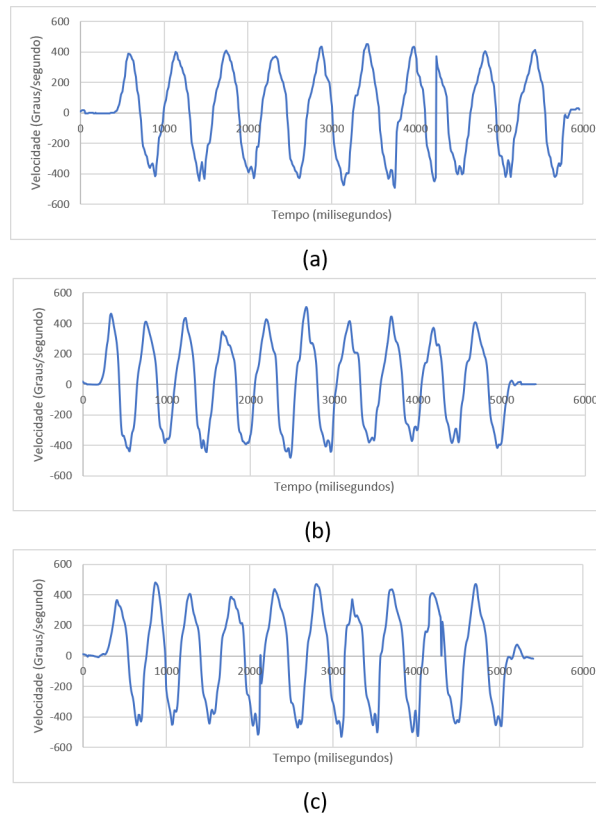
Figura 45 – Conjunto de três coletas de dados de velocidade para movimentos lentos



Fonte: O autor.

Diferente do que ocorre com o acelerômetro, para movimentos rápidos também não foram encontrados deslocamentos de pontos obtidos pelo sensor, como pode ser observado na Figura 46, ilustrando três conjuntos de dados de três repetições (a), (b) e (c).

Figura 46 – Conjunto de três coletas de dados de velocidade para movimentos Rápidos



Fonte: O autor.

4.3.3 Análise dos conjuntos de dados

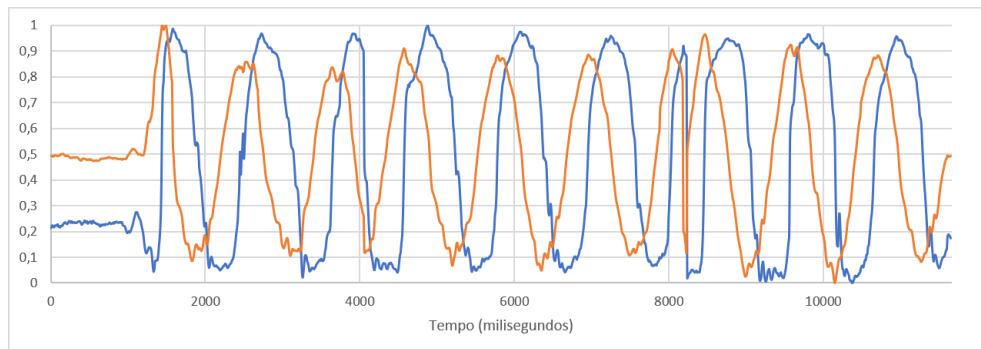
Realizando então a captura dos dados de acelerômetro e giroscópio de forma simultânea para movimentos lentos, normalizando os dados e inserindo em um gráfico, é possível observar na Figura 47 com a velocidade em laranja e angulação obtida pelo acelerômetro em azul, que o momento de angulação máximo e mínimo coincidem com a inversão do sentido da velocidade, indicando que o sistema está funcionando de forma correta.

Assim, levou-se em conta a necessidade de um método que pudesse indicar angulação tanto para movimentos lentos quanto rápidos. Como a velocidade do giroscópio é angular, ou seja, nos fornece dados em radianos por segundo, seria necessário, para saber o deslocamento entre leituras, apenas saber o tempo decorrido.

Assim, por meio da função "millis()" foi calculado o tempo exato entre execuções sendo utilizado para prever o deslocamento em graus.

Tendo o deslocamento entre dois pontos agora conhecido, necessita-se ter um referencial inicial do acelerômetro. Desta forma, foi definido que, antes de realizar os exercícios, seria essencial um período de calibração de 1 segundo, com o sensor parado na posição inicial. Por meio dos dados coletados no período de tempo proposto, é feita uma média, que é definida

Figura 47 – Conjunto de dados de velocidade e angulação normalizados. Velocidade na cor laranja e angulação obtida pelo acelerômetro na cor azul.



Fonte: O autor.

como o ponto inicial do movimento. Este método também auxiliou na definição do limiar de ativação de 10° .

4.4 Análise de captura de movimentos simples

Por meio da análise feita anteriormente, foi possível desenvolver um algoritmo de captura de movimentos simples, que leva em consideração a média de pontos definida antes de cada movimento como ângulo inicial (calibração) e o deslocamento em graus através do cálculo do deslocamento em graus, utilizando a velocidade angular obtida pelo giroscópio.

Os dados de angulação para movimentos simples com calibração inicial, obtidos por meio da combinação de acelerômetro e giroscópio, podem ser vistos na Figura 48, com aproximadamente 70° de amplitude angular para (a), iniciando-se em 15° e com pico em 85° ; aproximadamente 40° para (b), com início em 25° e maior ponto em 65° ; e, para (c), atingindo aproximadamente 100° , iniciando em 20° e tendo o máximo em 120° . Nos três casos é possível observar uma crescente em angulação sem nenhum tipo de momento de instabilidade para os três movimentos rápidos realizados.

Figura 48 – Conjunto de três coletas de dados de angulação obtidos através do giroscópio com calibração inicial feita pelo acelerômetro



(a)



(b)



(c)

Fonte: O autor.

A análise da captura de amplitude realizada no suporte de alumínio pode ser vista na tabela 9, em que observa-se o erro médio absoluto máximo de 8°.

Tabela 9 – Tabela de resultados para teste de amplitude utilizando-se do suporte

Amplitude pretendida (°)	Amplitude obtida (°)	Erro absoluto(°)	Desvio Padrão(°)
20	23	3	
20	28	8	2,05
20	25	5	
40	36	4	
40	43	3	2,86
40	40	0	
60	64	4	
60	60	0	1,63
60	62	2	
80	79	1	
80	82	2	1,25
80	80	0	
100	101	1	
100	102	2	2,62
100	96	4	

Fonte: O autor.

A análise da captura de amplitude de movimento do braço utilizou-se dos dados brutos processados de forma *offline* para obter os valores de amplitude, resultando em uma amplitude máxima média, entre 4 amostras de dados brutos, de 115,0°. Foram então feitas capturas com movimentos lentos e rápidos, obtendo os resultados observados na tabela 10 chegando em uma amplitude média de movimentos lentos de 115,6°, e para movimentos rápidos, uma amplitude média de 122,6°.

Tabela 10 – Tabela de resultados para teste de amplitude utilizando-se do suporte

Tipo de movimento (°)	Amplitude obtida (°)
Lento	111
Lento	117
Lento	118
Lento	115
Lento	117
Rápido	122
Rápido	124
Rápido	118
Rápido	125
Rápido	124

Fonte: O autor.

Neste caso não é possível propor erro médio e absoluto, visto que cada movimento seria único, com posição inicial e final apenas aproximadas. Pode-se apenas comparar qualitativamente as amplitudes obtidas pelo algoritmo de identificação de movimento da Tabela 9 com a amplitude angular de movimento de 115° média dos dados brutos, identificando que as amplitudes obtidas pelo cálculo do aparelho possuem uma variação máxima de 10° com relação à obtida através do pós processamento dos dados brutos.

4.5 Análise da identificação de movimentos

O algoritmo de identificação de movimentos foi desenvolvido baseado nos resultados obtidos dos dados brutos. Assim, como visto anteriormente, para uma melhor identificação de movimentos, se faz necessário utilizar o giroscópio, devido à ausência de momentos de instabilidade. O acelerômetro ainda é utilizado como identificação de ângulo inicial, estabelecendo um limiar de ativação.

O algoritmo identifica inversões de movimento da velocidade, sendo aplicado somente à partir do momento que ocorre a ultrapassagem do limiar de ativação, seja de forma positiva ou negativa.

Para os testes de 15 segundos, utilizando o menu Capturar repetições (Limite de tempo), com os sensores posicionados conforme Figura 49 para abdominais e Figura 50 para flexões, foram encontrados os resultados descritos na tabela 11, que mostra o número de repetições feitas e contabilizadas pelo dispositivo, identificando através da divisão por 4 dos requisitos estabelecidos pelas Tabelas 1 e 3, a avaliação de resistência muscular, tanto para os movimentos contados sem o aplicativo quanto para movimentos contados com o aplicativo.

Figura 49 – Fixação do sensor para os testes de abdominal (no peito) e indicação do movimento.



Fonte: O autor.

Figura 50 – Fixação do sensor para os testes de flexão (nas costas) e a indicação do movimento



Fonte: O autor.

Tabela 11 – Tabela de resultados para repetições de flexão, abdominal e flexão de braço

Movimento	Número de repetições realizadas	Repetições obtidas pelo dispositivo	Número de erros do exercício
Flexão	9 - Excelente	11 - Excelente	3
	11 - Excelente	12 - Excelente	
	8 - Acima da média	8 - Acima da média	
	10 - Excelente	10 - Excelente	
	9 - Excelente	9 - Excelente	
Abdominal	6 - Fraco	6 - Fraco	1
	8 - Abaixo da média	9 - Média	
	9 - Média	9 - Média	
	11 - Excelente	11 - Excelente	
	10 - Acima da média	10 - Acima da média	
Flexão do braço	14	14	1
	20	21	
	21	21	
	26	26	
	32	32	

Fonte: O autor.

É possível observar que para os três tipos de movimento, em 15 segundos, ocorreram erros absolutos de no máximo dois movimentos.

Neste caso, de acordo com a tabela 1 e 3, para nenhuma das contagens propostas foi identificada diferença em relação à avaliação de resistência muscular, porém, é possível que em diferentes casos ocorra um erro de contagem que desloque a avaliação de resistência muscular para condicionamentos melhores do que o real. Assim, ainda se faz necessário a contagem em paralelo.

O último menu do aplicativo não foi utilizado, servindo apenas de exemplo de aplicação, porém, ele utiliza o mesmo algoritmo do menu de contagem por tempo, tendo a diferença ape-

nas em que, no momento do recebimento do aviso de movimento, as repetições são subtraídas de um total informado pelo usuário em vez de adicionadas. Assim, muda-se somente a programação do dispositivo que recebe o aviso de movimento, mantendo o algoritmo de detecção inalterado.

Das fontes causadoras de erro para a contagem de movimentos, pode-se especular o movimento peitoral da respiração (para abdominal e flexão), movimentos parciais ou incompletos, entre outras fontes fisiológicas ou aleatórias de erro.

5 CONCLUSÃO

Foi desenvolvido um dispositivo que identifica ângulo e repetibilidade de movimento através do sensor inercial MPU6050. Criou-se também um aplicativo para dispositivos android responsável por receber, exibir e armazenar os dados enviados pelo microcontrolador ESP32.

Através da análise dos dados brutos de acelerômetro do sensor inercial, foi possível identificar fontes de erro associadas à movimentos rápidos. Feito isto, assumiu-se a aplicação de contagem de movimento para testes de aptidão física.

Pelos resultados obtidos dos testes preliminares utilizando dados não processados, para a correta predição de posição angular em movimentos, se faz necessário o uso do giroscópio, com referencial inicial dado pelo acelerômetro.

Já para os dados brutos analisados do giroscópio, foi possível identificar que ele não possui fontes de erro associada à velocidade, portanto, além de ser utilizado para a predição de ângulo junto do acelerômetro, ele também se torna capaz de realizar predição de direção de movimento. Assim, observou-se que, através de um algoritmo para a correta classificação da direção indicada pelos dados do giroscópio, se torna possível realizar a contagem de movimentos repetitivos.

Pelos resultados obtidos na identificação e contagem de movimentos, é possível observar que existem, nos dados brutos do giroscópio, fontes de ruído. As fontes de ruído podem gerar contagens errôneas no algoritmo proposto, sendo possível que o desenvolvimento de algum tipo de filtro especificamente para o giroscópio, amenizasse o problema.

Como trabalhos futuros, têm-se o objetivo de desenvolver outros tipos de filtros para evitar identificações errôneas de movimento e realizar a validação dos dados capturados em grupos maiores de pessoas.

Também têm-se como objetivo realizar um estudo sobre a possibilidade de implementação de uma rede neural artificial para a classificação de diferentes tipos de movimento baseando-se nas características específicas de cada um.

REFERÊNCIAS

- AAHPERD. Health related physical fitness test manual. 1980.
- ABREU, A. V. *et al.* Avaliação clínico radiográfica da mobilidade da lordose lombar. 2007. Disponível em: <http://rbo.org.br/detalhes/1152/pt-BR/avaliacao-clinico-radiografica-da-mobilidade-da-lordose-lombarf>.
- AITHINKER. **ESP01 WiFi Module**. 2015. Disponível em: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1179098/ETC2/ESP-01.html>.
- AMARO, A. C. A.; AQUINO, S. S. Sistema modular de medição do movimento humano baseado em ROS e sensores inerciais. 2019.
- ARDUINO. **Arduino Mega 2560 - Product Reference Manual**. 2022. Disponível em: <https://docs.arduino.cc/static/5188f3f82fa6482eb8e117e568010aef/A000067-datasheet.pdf>.
- ARDUINO. **Arduino Nano - User Manual**. 2022. Disponível em: <https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf>.
- ARDUINO. **Arduino Uno R3 - Product Reference Manual**. 2022. Disponível em: <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>.
- BABIUCH, M.; FOLTÝNEK, P.; SMUTNÝ, P. Using the esp32 microcontroller for data processing. In: **2019 20th International Carpathian Control Conference (ICCC)**. [S.l.: s.n.], 2019. p. 1–6.
- BARBOSA, J. A. Análise cinética, cinemática e eletromiográfica da flexão de braços: um estudo exploratório. 2015. Disponível em: https://iepapp.unimep.br/biblioteca_digital/pdfs/docs/28052015_135627_josealexandrebarbosa_ok.pdf.
- BHATTACHARYYA S.; DEPRETTERE, E. L. R. T. J. **Handbook of Signal Processing Systems**. [S.l.]: Springer Cham, 2018.
- BÖHME, M. T. S. Aptidão física: aspectos teóricos. **Revista Paulista de Educação Física**, v. 7, n. 2, p. 52–65, 1993.
- BOSCH. Bmi160 datasheet. 2018. Disponível em: <https://www.mouser.com/datasheet/2/783/BST-BMI160-DS000-1509569.pdf>.
- CASPERSEN; POWELL; CHRISTENSON. Physical activity, exercise, and physical fitness: definitions and distinctions for health-related research. **Public Health**, p. 126 – 311, 1985.
- CATAPAN, D. R. Desenvolvimento de aplicativo para android sdk. 2009. Disponível em: <https://cepein.femanet.com.br/BDigital/arqTccs/0511150155.pdf>.
- CORDEIRO, F. **App Inventor: Guia de Criação de Apps**. 2017. Disponível em: <https://www.androidpro.com.br/blog/desenvolvimento-android/app-inventor/>.
- CUESTA-VARGAS ALEJANDRO GALÁN-MERCANT, J. M. W. A. I. The use of inertial sensors system for human motion analysis. **Physical Therapy Reviews**, 2010. Disponível em: <https://doi.org/10.1179/1743288X11Y.0000000006>.
- DELFINO, T.; HUMBER, P. M. Níveis de força/resistência abdominal e índice de massa corporal em alunos de 12 a 15 anos do município de Armazém, sc. **EFDesportes**, 2013. Disponível em: <http://www.efdesportes.com/efd178/niveis-de-forca-resistencia-abdominal-de-12-a-15.htm>.

- EASYTRONICS. **MPU6050 - Acelerômetro e Giroscópio de 3 Eixos**. 2022. Disponível em: <https://www.easytronics.com.br/mpu6050-acelerometro-e-giroscopio-de-3-eixos>.
- ESPRESSIF. **Development Boards**. 2022. Disponível em: <https://www.espressif.com/en/products/devkits>.
- EXPRESSIF. Esp8266ex datasheetversion 6.0. 2018. Disponível em: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1148030/ESPRESSIF/ESP8266EX.html>.
- EXPRESSIF. **ESP32 Series Datasheet**. 2022. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.
- FAJRIN, H. R. *et al.* Telemedicine-equipped android interface-based heart rate monitoring. **Indones. J. Electr. Eng. Comput. Sci**, v. 21, n. 3, p. 1332–1340, 2021.
- FILHO, J. F.; NOVAES, J. S. Eficácia do teste de flexão e extensão de braços, corrigido pelo índice de massa corporal, na determinação da resistência muscular localizada absoluta em mulheres do exército brasileiro. 2013.
- FLEISHMAN, E. Structure and measurement of physical fitness. 1964.
- GOLATA, P. **Set Yourself Straight on Inertial Measurement Units (IMUs)**. 2021. Disponível em: https://www.mouser.com/blog/set-yourself-straight-imus?utm_source=embarcados&utm_medium=display&utm_campaign=imus&utm_content=imus.
- HANSEN, D.; KENNELLY, S. Anatomia da pliometria: guia ilustrado da potência muscular em movimentos esportivos de salto, corrida, arremesso, flexão e agachamento. **Editora Manoele**, 1993. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788520458853/>.
- HEBBELINCK. The concept of health related to physical fitness. **International Journal of Physical**, p. 09 – 29, 1984.
- IBGE. **PNS 2019: Cai o consumo de tabaco, mas aumenta o de bebida alcoólica**. 2020. Disponível em: <https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/29471-pns-2019-cai-o-consumo-de-tabaco-mas-aumenta-o-de-bebida-alcoolica>.
- INVENSENSE. Mpu-6000 and mpu-6050 product specification: Revision 3.3. 2012. Disponível em: <https://pdf1.alldatasheet.com/datasheet-pdf/view/517744/ETC1/MPU-6050.html>.
- INVENSENSE. **MPU-6000 and MPU-6050 Register Map and Descriptions Revision 4.2**. 2013. Disponível em: <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Register-Map1.pdf>.
- INVENSENSE. Mpu-9250 product specification: Revision 1.1. 2016. Disponível em: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1132036/TDK/PS-MPU-9250A-01.html>.
- INVENSENSE. Icm-20649 datasheet. 2021. Disponível em: <https://3cfeqx1hf82y3xcoull08ihx-wpengine.netdna-ssl.com/wp-content/uploads/2021/07/DS-000192-ICM-20649-v1.1.pdf>.
- INVENTOR, M. A. **About Us**. 2022. Disponível em: <https://appinventor.mit.edu/about-us>.
- ITEAD. **HC-05 Datasheet**. 2010. Disponível em: https://components101.com/sites/default/files/component_datasheet/HC-05%20Datasheet.pdf.

- ITOU, S.; INOUE, S.; WASHINO, K. Respiration detection using inertial measurement unit. In: **2019 International Symposium on Electrical and Electronics Engineering (ISEE)**. [S.l.: s.n.], 2019. p. 119–123.
- LARSON, L. A. **Fitness, health, and work capacity: International standards for assessment**. [S.l.: s.n.], 1974.
- LILYGO. **LILYGO® TTGO T-Energy ESP32 8MByte PSRAM WiFi amp; Bluetooth Module 18650 Battery ESP32-WROVER-B Development Board**. 2019. Disponível em: http://www.lilygo.cn/prod_view.aspx?TypeId=50033&Id=1170&FId=t3:50033:3.
- LIMA, R. F. Análise do nível de resistência muscular localizada em escolares do ensino fundamental de 9 a 11 anos. 2013. Disponível em: <https://repositorio.uniceub.br/jspui/bitstream/235/5855/1/21015383.pdf>.
- LIU, K. *et al.* Automatic tracking system of leo satellite based on sgp4. In: **ACM Turing Award Celebration Conference-China (ACM TURC 2021)**. [S.l.: s.n.], 2021. p. 192–199.
- MAIER, A.; SHARP, A.; VAGAPOV, Y. Comparative analysis and practical implementation of the esp32 microcontroller module for the internet of things. In: **2017 Internet Technologies and Applications (ITA)**. [S.l.: s.n.], 2017. p. 143–148.
- MICROCHIP. mega avr® data sheet. 2020. Disponível em: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1148030/ESPRESSIF/ESP8266EX.html>.
- MOROUÇO, P. *et al.* Unidade de medida inercial para controlo do treino de nadadores. **Motricidade**, Edições Desafio Singular, v. 17, p. 26–27, 2021.
- ORTEGA *et al.* Physical fitness levels among european adolescents: the helena study. 2009.
- POLLOCK, M. L.; WILMORE, J. H. Exercícios na saúde e na doença avaliação e prescrição para prevenção. **Médica e Científica**, 1993. Disponível em: <https://www.scienceopen.com/document?vid=348b448b-a262-437e-ba88-0f332ad5f3fa>.
- PRADO, S. **Introdução ao funcionamento interno do Android**. 2017. Disponível em: <https://sergioprado.org/introducao-ao-funcionamento-interno-do-android/>.
- PRATES, R. O.; BARBOSA, S. D. J. Avaliação de interfaces de usuário—conceitos e métodos. In: SN. **Jornada de Atualização em Informática do Congresso da Sociedade Brasileira de Computação, Capítulo**. [S.l.], 2003. v. 6, p. 28.
- RAMADHANI, U. A.; WISANA, I. D. G. H.; NUGRAHA, P. C. Smartphone based respiratory signal monitoring and apnea detection via bluetooth communication. **Jurnal Teknokes**, v. 14, n. 2, p. 49–55, 2021.
- RITTER, E. W.; RIGO, S. J. Fitdata: Um sistema para monitoramento de atividade física baseado em dispositivos móveis. In: SBC. **Anais do XII Simpósio Brasileiro de Sistemas de Informação**. [S.l.], 2016. p. 550–557.
- SCHOENELL, M. C. W.; AUTHERS. Correlação e reprodutibilidade de testes abdominais em mulheres jovens. **Rev. Bras. Ciênc. Esporte**, v. 35, n. 3, 1993. Disponível em: <https://www.scielo.br/j/rbce/a/bz5RXM8qxJTR9WpXgys8vck/?format=pdf&lang=pt>.
- SETIAWAN, A.; PURNAMASARI, A. I. *et al.* Pengembangan smart home dengan microcontrollers esp32 dan mc-38 door magnetic switch sensor berbasis internet of things (iot) untuk meningkatkan deteksi dini keamanan perumahan. **Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)**, v. 3, n. 3, p. 451–457, 2019.

SILVA, W. M. A. M. M. C. R. O. da. Visão geral sobre microcontroladores e prototipagem com arduino. **REVISTA TECNOLOGIAS EM PROJEÇÃO**, v. 10, n. 1, 2019. Disponível em: <http://revista.faculdadeprojecao.edu.br/index.php/Projecao4/article/view/1357/1062>.

THOMAZ, J. J. Tabela de aptidão física: proposta de provas, índices e critérios justos que reflitam a realidade do policial militar de alagoas. 2003.

TORRES, H. **Sensores Inerciais – Parte 1**. 2014. Disponível em: <https://www.embarcados.com.br/sensores-inerciais-parte-1/>.

WEN, J. *et al.* Face recognition system design based on esp32. In: IEEE. **2022 International Seminar on Computer Science and Engineering Technology (SCSET)**. [S.l.], 2022. p. 114–116.

ZHANG, Y.; CAI, W. Design of self-balancing robot based on esp32. In: IEEE. **2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology (ICCASIT)**. [S.l.], 2021. p. 1039–1043.

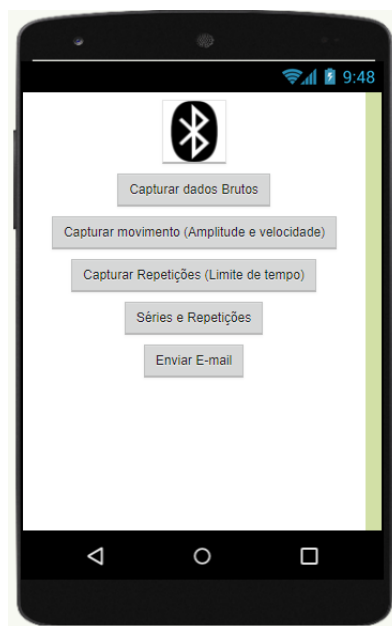
APÊNDICE A – Desenvolvimento do aplicativo Android

O aplicativo android foi desenvolvido através da plataforma *online* MIT APP Inventor 2, que utiliza a programação de blocos para criar um aplicativo android de forma simplificada.

Como primeiro passo, foi necessário desenvolver a parte gráfica das telas do aplicativo, sendo cada uma delas feitas através da aba "Designer". Para a programação foi necessário usar a tela "Blocks", também para cada uma das cinco telas.

A primeira tela funciona como um menu inicial, podendo ser visto na figura 51, descrevendo as quatro funções desenvolvidas e com a possibilidade de enviar um e-mail pré escrito direcionado para o desenvolvedor do aplicativo.

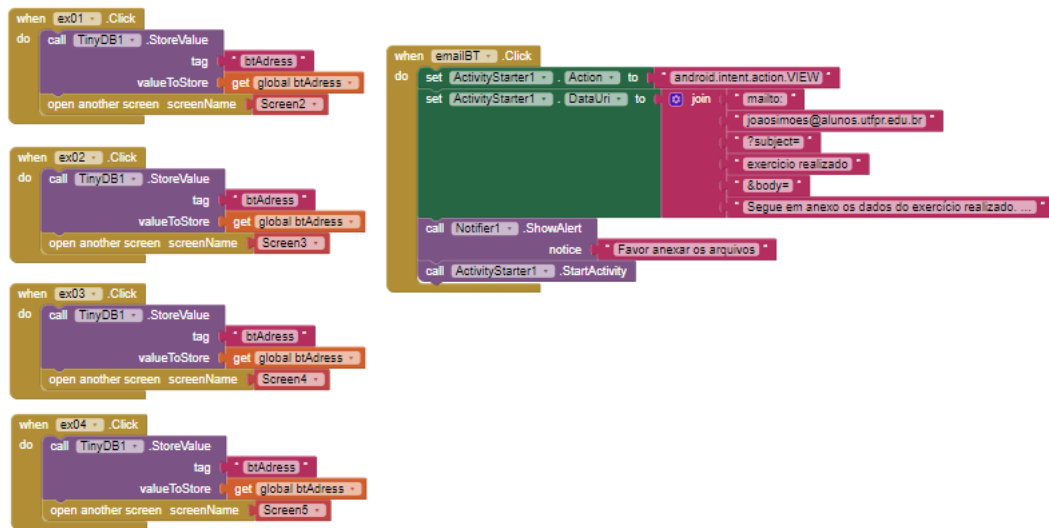
Figura 51 – Tela inicial projetada para o aplicativo



Fonte: O autor.

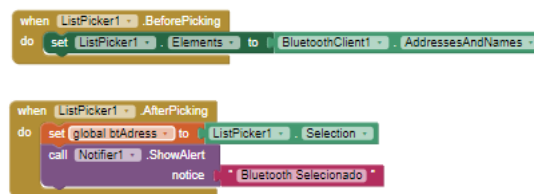
Para a primeira tela, foi necessário configurar cada um dos 5 botões, com a função de abrir uma nova tela e armazenar o nome do dispositivo *bluetooth*, e o botão (*ListPicker*) para procurar o dispositivo *bluetooth* e armazenar seu nome em uma variável. A programação em blocos dos botões pode ser encontrado na Figura ?? e a programação para obtenção do endereço específico *bluetooth* é visto na Figura ??.

Figura 52 – Blocos para a programação dos 5 botões da tela inicial



Fonte: O autor.

Figura 53 – Blocos para a programação do botão de escolha de dispositivo *bluetooth*



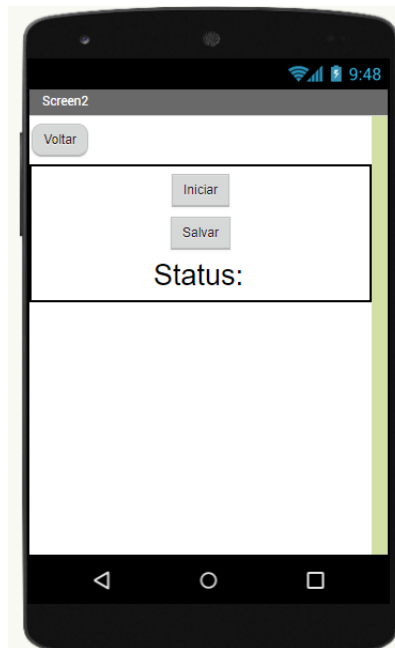
Fonte: O autor.

A segunda tela tem a função de receber e armazenar os dados brutos de aceleração e velocidade dos eixos X e Y, como visto na Figura 54 ela não demonstra dados e apenas possui um botão para Iniciar/Parar e outro para salvar os dados em .txt. O Texto "Status" muda para mostrar ao usuário se o dispositivo encontra-se em modo de captura ou não.

Como consequência da simplicidade da tela, a programação em blocos funciona de forma que o botão de Iniciar também possui a função de parar e de realizar a conexão bluetooth, através do dado armazenado na tela anterior. A Figura ?? mostra a configuração do botão "Iniciar". O componente invisível "Clock" é responsável por armazenar os dados em uma lista, sendo a programação deste componente mostrada pela Figura ?? e a Figura ?? mostra a configuração do botão "Salvar". A função envia a letra "k" como indício de início de coleta para o ESP32. Para parar a coleta, todas as telas enviam a letra "p".

Para a terceira tela, realizada para capturar amplitude e velocidade de movimentos simples, foi realizada uma interface simples, vista na Figura 58 com a possibilidade de armazenar amplitude e velocidade de diversos movimentos no mesmo arquivo .txt, assim, fêz-se apenas necessária a adição de um botão para dar a possibilidade do usuário remover a última captura

Figura 54 – Segunda tela projetada do aplicativo, com função de iniciar a transmissão de dados brutos



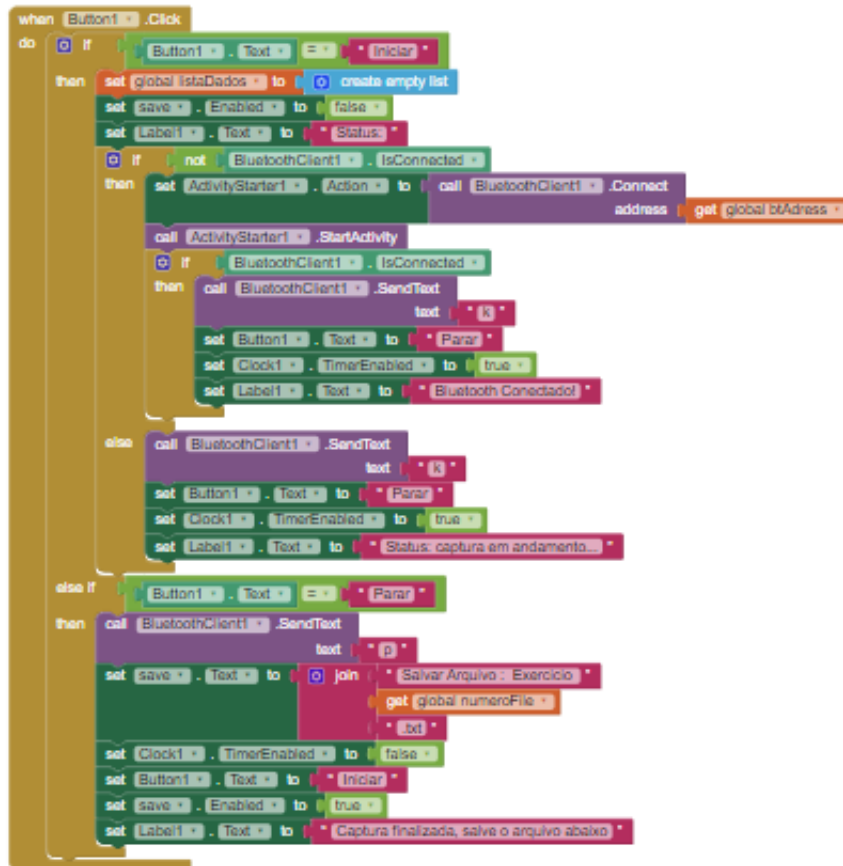
Fonte: O autor.

simples realizada. O botão de Iniciar conta com a mesma lógica do da tela anterior, porém, a letra enviada por bluetooth simbolizando a função é "m".

A amplitude angular, depende da calibração inicial, ou seja, da definição de um ângulo inicial, definido pela média de 100 pontos com uma amostragem de 100Hz, sendo assim de 1 segundo. O indício de finalização da calibração é o envio da letra "c" do ESP32 para o dispositivo android, como pode ser visto no componente "clock" da Figura 59.

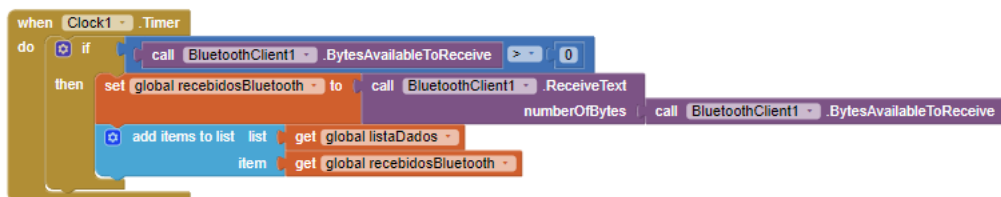
A quarta tela do aplicativo, demonstrada na Figura 60 possibilita que o usuário defina um limite de tempo para a realização de movimentos repetitivos, a tela também conta com aviso sonoro de início e parada para facilitar a utilização.

Figura 55 – Blocos para a programação do botão "Iniciar"



Fonte: O autor.

Figura 56 – Blocos para a programação do componente invisível "Clock"

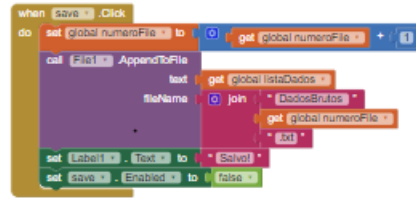


Fonte: O autor.

O limite de tempo possui um componente oculto "clock" específico, com um segundo de tempo de ativação, que é ativado à partir do início do exercício, subtraindo-se uma unidade por unidade do número inserido pelo usuário.

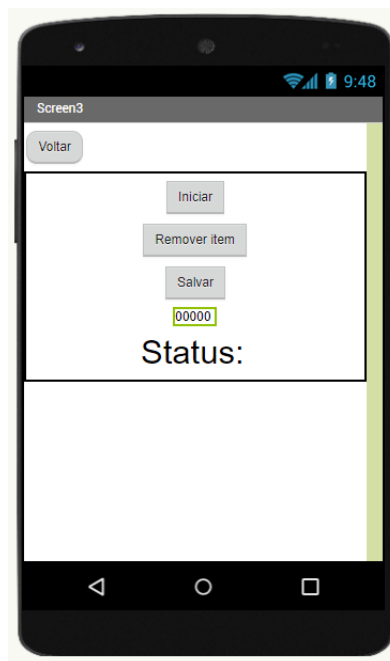
Para esta tela, foram usadas as mesmas lógicas para o componente "clock" da anterior, só que com envio automático da letra "p" após o fim do tempo do exercício e em vez de armazenar as variáveis o componente adiciona uma unidade ao item de contagem de repetições e

Figura 57 – Blocos para a programação do botão de "Salvar"



Fonte: O autor.

Figura 58 – Terceira tela projetada do aplicativo, com função de receber os dados de amplitude angular e tempo de movimentos simples



Fonte: O autor.

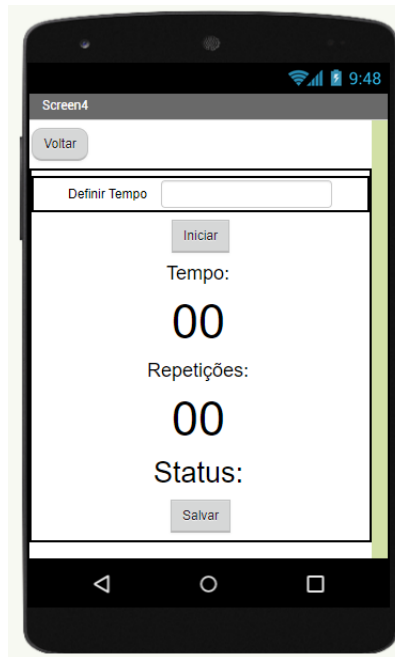
Figura 59 – Blocos para a programação do componente "Clock" com função de aguardar calibração e armazenar variáveis

Fonte: O autor.

envia a letra "j" como identificador de funções para o ESP32. A programação do componente oculto "clock" pode ser observada na Figura 61.

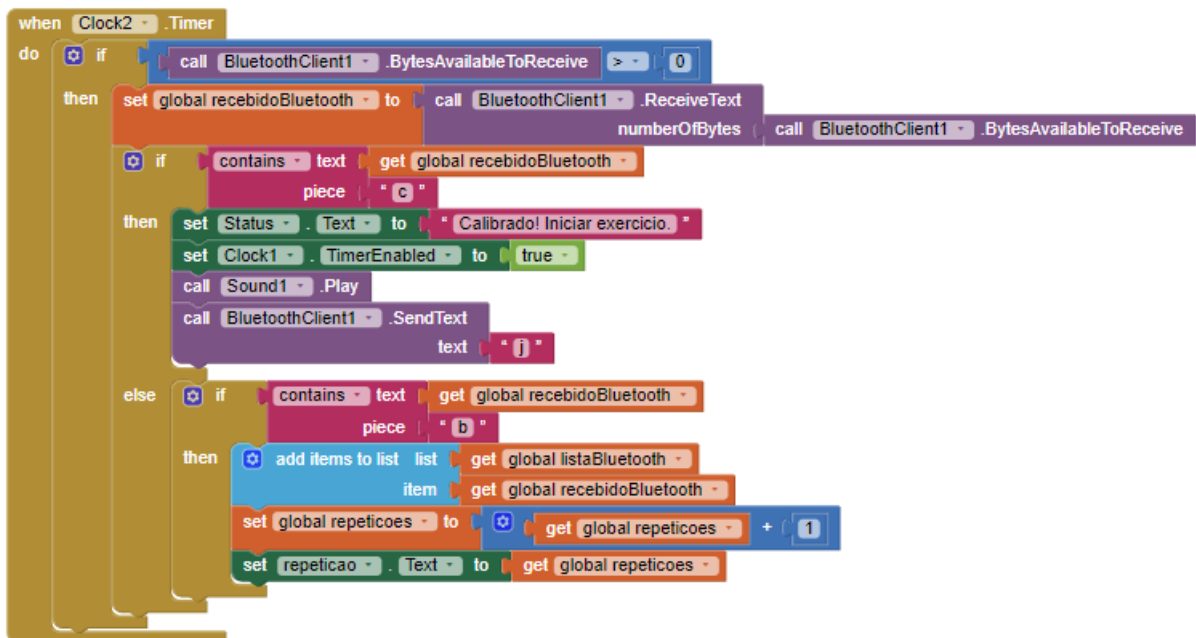
A quinta e última tela proporciona uma interface que permite que o usuário selecione a quantidade de movimentos e séries de movimentos necessárias, também sendo possível configurar intervalos de tempo entre séries (chamados de descanso). A tela também utiliza sinais sonoros para parada e início de movimento, além de demonstrar visualmente quantas repetições e séries foram cumpridas, como pode ser visto na Figura 62.

Figura 60 – Quarta tela projetada do aplicativo, com função de receber aviso de que foi realizada uma repetição de movimento, e contar como repetição.



Fonte: O autor.

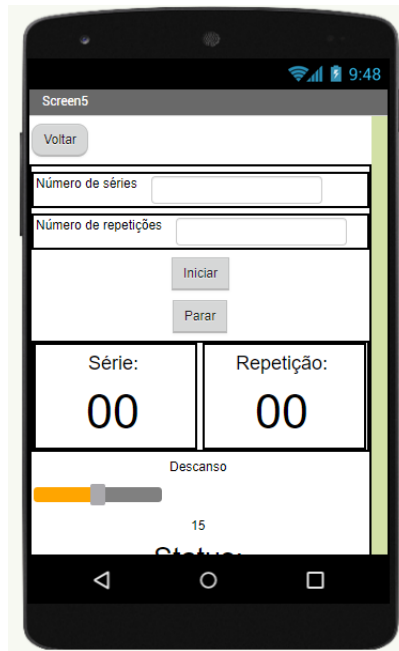
Figura 61 – Blocos para a programação do componente "Clock" com função de aguardar calibração e realizar contagem de movimentos



Fonte: O autor.

A última tela utiliza a lógica inversa da quarta tela para a questão de séries. É atribuído uma variável para número de séries e repetições, e a cada movimento, é subtraído do número de

Figura 62 – Quinta tela projetada do aplicativo, em que o usuário pode escolher o número de repetições e séries de movimentos.



Fonte: O autor.

repetições pedidas, quanto obtida as repetições, é subtraído uma unidade do valor das séries e um componente oculto "Clock" realiza a contagem de segundos de descanso, reenviando então a letra de início de movimento após o fim do descanso.

APÊNDICE B – Código desenvolvido no ESP32

```

1
2 #include <Wire.h>                // Only needed for Arduino 1.6.5 and
   earlier
3 #include "SSD1306Wire.h"        // legacy: #include "SSD1306.h"
4
5
6 #include "BluetoothSerial.h"
7
8 #if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
9 #error Bluetooth is not enabled! Please run `make menuconfig` to and
   enable it
10 #endif
11
12 BluetoothSerial SerialBT;
13
14 const int MPU_ADDR = 0x68; // I2C address of the MPU-6050
15 int16_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ; //int16_t
16 double Row, Pitch, Yaw;
17 int Row2, Pitch2, Yaw2;
18 int16_t AcX2, AcY2, AcZ2;
19 int16_t calibX, calibY;
20 float contagem;
21 char leituraSerialTeste;
22 char var;
23 float x;
24 float y;
25 float Gx;
26 float Gy;
27 int xAnterior, yAnterior, GxAnterior, GyAnterior;
28 int yAnt;
29 int interacao;
30 double coefY, coefYAntigo, coefX, coefXAntigo;
31 int contadorPositivoY, contadorNegativoY, contadorPositivoX,
   contadorNegativoX, contadorNeutroX, contadorNeutroY;

```

```
32 int contadorVelocidadeY;
33 int amplitudeAngularY, amplitudeAngularX;
34 int limiarDeAtivacao;
35 boolean movimentoY, movimentoX;
36 int maiorAnguloY, menorAnguloY;
37 int maiorAnguloX, menorAnguloX;
38 float tempoDoMovimentoY, tempoDoMovimentoX;
39 unsigned long t1Y, t2Y, t1X, t2X;
40 bool ex1, ex2, ex3, ex4;
41 int mediaPontosY, mediaGyBruto;
42 int mediaPontosX, mediaGxBruto;
43 bool movimentoPositivoX, movimentoNegativoX, movimentoPositivoY,
    movimentoNegativoY;
44 unsigned long tempo1, tempo2, maiorTempo;
45 float tempoCorrido;
46 float tempX, tempY, tempSomaX, tempSomaY;
47 float anguloGyroX, anguloGyroY;
48 bool primeiraInteracao;
49 String envioY;
50 int idaEVolta;
51 bool movimentoTemporarioY;
52 int movimentoTemporarioDirecao;
53 int count;
54
55
56 void setup() {
57     Serial.begin(115200);
58     Serial.println();
59     Serial.println();
60     contagem = 0;
61
62
63     //definições genéricas para menor e maior ngulo
64     menorAnguloY = 99999;
```

```
65 maiorAnguloY = -99999;
66 maiorTempo = -999999;
67 menorAnguloX = 99999;
68 maiorAnguloX = -99999;
69
70
71 limiarDeAtivacao = 10;
72
73
74 primeiraInteracao = ex1 = ex2 = ex3 = false;
75
76 // Initialising the UI will init the display too.
77
78
79 //initializing the MPU6050
80 Wire1.begin(21, 22, 100000); // sda, scl, clock speed
81 Wire1.beginTransmission(MPU_ADDR);
82 Wire1.write(0x6B); // PWR_MGMT_1 register
83 Wire1.write(0X00); // set to zero (wakes up the MPU 6050 )
84 Wire1.endTransmission(true);
85
86 //CONFIGURA O DO GYRO
87 Wire1.beginTransmission(0x68); //begin, Send the slave
    adress (in this case 68)
88 Wire1.write(0x1B); //We want to write to the
    GYRO_CONFIG register (1B hex)
89 Wire1.write(0x10); //Set the register bits as
    00010000 (1000dps full scale)
90 Wire1.endTransmission(true); //End the transmission
    with the gyro
91
92 //CONFIGURA O DO ACC
93 Wire1.beginTransmission(MPU_ADDR); //Start communication
    with the address found during search.
```



```
94 Wire1.write(0x1C); //We want to write to the
    ACCEL_CONFIG register
95 Wire1.write(0x10); //Set the register bits as
    00010000 (+/- 8g full scale range)
96 Wire1.endTransmission(true);
97
98 tempo2 = millis();
99
100 //iniciando Bluetooth
101 SerialBT.begin("ESP32test"); //Bluetooth device name
102 Serial.println("Setup complete");
103
104 }
105
106
107
108
109 void recebeBluetooth() {
110
111     if (SerialBT.available()) {
112         var = SerialBT.read();
113         SerialBT.flush();
114         if (var == 'k') {
115             funcaoDadosBrutos();
116         }
117         if (var == 'm') {
118             funcaoDadosMovimentoSimples();
119         }
120         if (var == 'j') {
121             funcaoDadosMovimentoPorTempo();
122         }
123         if (var == 'p') {
124             if (ex1 == true) {
125                 ex1 = false;
```

```
126     }
127     if (ex2 == true) {
128         ex2 = false;
129     }
130     if (ex3 == true) {
131         ex4 = false;
132     }
133 }
134 }
135 }
136
137 void identificaMovimento() {
138
139     //a identificação do movimento não muda, mantém-se a mudança do
140     //threshold, porque os ngulos com função no Gyro estão sendo
141     //calculados antes, eles podem ser usados ou não
142
143     if (ex2 == true) {
144         y = anguloGyroY;
145         x = anguloGyroX;
146         xAnterior = GxAnterior;
147         yAnterior = GyAnterior;
148     }
149
150     if (ex3 == true) {
151         y = anguloGyroY;
152         x = anguloGyroX;
153         xAnterior = GxAnterior;
154         yAnterior = GyAnterior;
155     }
156
157     if (ex4 == true) {
158         y = anguloGyroY;
159         x = anguloGyroX;
160         xAnterior = GxAnterior;
161         yAnterior = GyAnterior;
162     }
163 }
```

```
158
159 if (movimentoY == false) {
160     if (y > (mediaPontosY + limiarDeAtivacao)) {
161         movimentoY = true;
162         movimentoPositivoY = true;
163         t1Y = millis();
164         Serial.println("movimento positivo");
165     }
166     if (y < (mediaPontosY - limiarDeAtivacao)) {
167         movimentoY = true;
168         movimentoNegativoY = true;
169         t1Y = millis(); //t1 é facil, é onde inicia
170         Serial.println("Primeiro movimento negativo");
171     }
172 }
173 }
174
175
176 if (movimentoY == true) {
177     if (y > maiorAnguloY) {
178         maiorAnguloY = y;
179         t2Y = millis();
180     }
181     if (y < menorAnguloY) {
182         menorAnguloY = y;
183         t2Y = millis();
184     }
185 }
186
187 if (movimentoY == true) {
188     pegaCoeficienteY();
189 }
190
191 }
```

```
192
193 void pegaCoeficienteY() {
194
195     coefY = (y - yAnterior) / (interacao - (interacao - 1)); //a interação neste caso é 1 unidade, se mudar para 10 unidades muda-se o número da formula
196
197     if(ex2 == true){
198         if (coefY > 0.1) {
199             contadorPositivoY++;
200             contadorNegativoY = 0;
201             contadorNeutroY = 0;
202         }
203         else if (coefY < -0.1) {
204             contadorNegativoY++;
205             contadorPositivoY = 0;
206             contadorNeutroY = 0;
207         }
208         else{
209             contadorNegativoY = 0;
210             contadorPositivoY = 0;
211             contadorNeutroY++;
212         }
213     }
214
215     if(ex3 == true || ex4 == true){
216         if (Gy > 0) {
217             contadorVelocidadeY++;
218         }
219         else if (Gy < 0) {
220             contadorVelocidadeY--;
221         }
222     }
223
```

```
224 if(contadorVelocidadeY == 10){ //se movimento positivo
225     contadorVelocidadeY = 0;
226     if(movimentoTemporarioY == false){//se for a primeira interação
227         movimentoTemporarioDirecao = 1;
228         movimentoTemporarioY = true;
229         Serial.println("Identificou movimento positivo");
230     }
231     else{
232         if(movimentoTemporarioDirecao == -1){ //se o movimento era
           negativo (-1) e se tornar positivo (contadorVelocidadeY == 4) então
           o virou para o positivo
233             Serial.println("Identificou virada para positivo");
234             movimentoTemporarioDirecao = 1;
235             count++;
236             if(count == 2){
237                 Serial.println("MOVIMENTO!");
238                 SerialBT.println("b");
239                 count = 0;
240             }
241             // movimentoTemporarioY = false;
242         }
243     }
244 }
245
246 else if(contadorVelocidadeY == -10){ //se movimento negativo
247     contadorVelocidadeY = 0;
248     if(movimentoTemporarioY == false){
249         Serial.println("Identificou movimento negativo");
250         movimentoTemporarioDirecao = -1;
251         movimentoTemporarioY = true;
252     }
253     else{
254         if(movimentoTemporarioDirecao == 1){
255             count++;
```

```

256     movimentoTemporarioDirecao = -1;
257     Serial.println("Identificou virada para negativo");
258     if(count == 2){
259         Serial.println("MOVIMENTO!");
260         SerialBT.println("b");
261         count = 0;
262     }
263     // movimentoTemporarioY = false;
264     }
265 }
266 }
267
268
269
270 if (contadorPositivoY == 4 && movimentoPositivoY == false) { //
    espera ter 4 interações para o mesmo lado para garantir que está
    acontecendo inversão de movimento (pode mudar o número de interaçõ
    es)
271     //movimentoY = false;
272     movimentoPositivoY = true;
273     contadorPositivoY = 0;
274     contadorNegativoY = 0;
275     contadorNeutroX = 0;
276     if(ex3 == true || ex4 == true){
277         idaEVolta++;
278         if(idaEVolta == 2){
279             idaEVolta = 0;
280             registraMovimentoY();
281             enviaMovimentoY();
282         }
283     }
284     else{
285         registraMovimentoY();
286         enviaMovimentoY();

```

```
287     }
288 }
289
290 if (contadorNegativoY == 4 && movimentoNegativoY == false) { //
    espera ter 4 interações para o mesmo lado para garantir que está
    acontecendo inversão de movimento (pode mudar o número de interações)
291     //movimentoY = false;
292     movimentoNegativoY = true;
293     contadorPositivoY = 0;
294     contadorNegativoY = 0;
295     contadorNeutroX = 0;
296     if(ex3 == true || ex4 == true){
297         idaEVolta++;
298         if(idaEVolta == 2){
299             idaEVolta = 0;
300             registraMovimentoY();
301             enviaMovimentoY();
302         }
303     }
304     else{
305         registraMovimentoY();
306         enviaMovimentoY();
307     }
308 }
309 if (contadorNeutroY == 4) { //espera ter 4 interações para o mesmo
    lado para garantir que está acontecendo inversão de movimento (
    pode mudar o número de interações)
310     movimentoY = false;
311     contadorPositivoY = 0; //adicionar aqui esse caralho de
    threshold para identificação de parada ou inversao?
312     contadorNegativoY = 0;
313     contadorNeutroX = 0;
314     Serial.println("Foi ida");
```

```
315     if(ex3 == true || ex4 == true){
316         idaEVolta++;
317
318         if(idaEVolta == 2){
319             Serial.println("Foi ida e volta");
320             idaEVolta = 0;
321             registraMovimentoY();
322             enviaMovimentoY();
323         }
324     }
325     else{
326         registraMovimentoY();
327         enviaMovimentoY();
328     }
329 }
330 }
331
332 void enviaMovimentoY() {
333     SerialBT.println("AY:" + String(amplitudeAngularY) + "/TY:" +
334         String(tempoDoMovimentoY));
335     Serial.println("Enviou?");
336     ex2 = false;
337 }
338 void registraMovimentoY() {
339
340     //pega o maior e o menor ngulo atingido e faz a diferença
341     amplitudeAngularY = (maiorAnguloY - menorAnguloY)+limiarDeAtivacao;
342     //utiliza a velocidade do movimento, utilizando o tempo em micros
343     //desde a primeira movimentação até a inversão do movimento
344     tempoDoMovimentoY = (t2Y - t1Y) / 1000.00;
345
346     //reset das definições genéricas para menor e maior ngulo
```



```
347 menorAnguloY = 99999;
348 maiorAnguloY = -99999;
349 }
350
351 void getAngles() {
352     yAnterior = y;
353     xAnterior = x;
354     GyAnterior = anguloGyroY;
355     GxAnterior = anguloGyroX;
356     interacao++;
357
358     x = (atan2(-AcY, -AcZ) + PI) * (180 / 3.14);
359     y = (atan2(-AcX, -AcZ) + PI) * (180 / 3.14);
360
361     if (primeiraInteracao == true) {
362         anguloGyroX = mediaPontosX + ((Gy * tempoCorrido)/1000) -
            mediaGxBruto;
363         anguloGyroY = mediaPontosY + ((Gy * tempoCorrido)/1000) -
            mediaGyBruto;
364         primeiraInteracao = false;
365     }
366     else if (primeiraInteracao == false && (ex2 == true || ex3 == true)
367         ) {
368         anguloGyroX = anguloGyroX + ((Gy * tempoCorrido)/1000) -
            mediaGxBruto;
369         anguloGyroY = anguloGyroY + ((Gy * tempoCorrido)/1000) -
            mediaGyBruto;
370     }
371
372     Serial.println(y);
373
374     if (ex1 == true) {
375
```

```

376     SerialBT.print(x); SerialBT.print(",");
377     SerialBT.print(Gx); SerialBT.print(",");
378     SerialBT.print(y); SerialBT.print(",");
379     SerialBT.println(Gy);
380 }
381
382 if (ex2 == true || ex3 == true) {
383     identificaMovimento();
384 }
385
386
387 }
388
389 void MPU6050 () {
390     tempo1 = tempo2;
391     tempo2 = millis();
392     tempoCorrido = (tempo2 - tempo1);
393     Wire1.beginTransaction(MPU_ADDR); //Faz a conexão com o endereço
        específico do sensor MPU6050
394     Wire1.write(0x3B); // Começa a leitura pelo registrador 0x3B (
        ACCEL_XOUT_H)
395     Wire1.endTransmission(false);
396     Wire1.requestFrom(MPU_ADDR, 14, true); // Faz a requisição das
        informações de 6 registradores
397     AcX = (Wire1.read() << 8 | Wire1.read()); // 0x3B (ACCEL_XOUT_H) &
        0x3C (ACCEL_XOUT_L)
398     AcY = (Wire1.read() << 8 | Wire1.read()); // 0x3D (ACCEL_YOUT_H) &
        0x3E (ACCEL_YOUT_L)
399     AcZ = (Wire1.read() << 8 | Wire1.read()); // 0x3F (ACCEL_ZOUT_H) &
        0x40 (ACCEL_ZOUT_L)
400     Tmp = Wire1.read() << 8 | Wire1.read(); // 0x41 (TEMP_OUT_H) & 0
        x42 (TEMP_OUT_L)
401     GyX = Wire1.read() << 8 | Wire1.read(); // 0x43 (GYRO_XOUT_H) & 0
        x44 (GYRO_XOUT_L)

```

```
402 GyY = Wire1.read() << 8 | Wire1.read(); // 0x45 (GYRO_YOUT_H) & 0
    x46 (GYRO_YOUT_L)
403 GyZ = Wire1.read() << 8 | Wire1.read(); // 0x47 (GYRO_ZOUT_H) & 0
    x48 (GYRO_ZOUT_L)
404 Gx = ((GyX * -1) / 32.8);
405 Gy = ((GyY * -1) / 32.8);
406 getAngles();
407 }
408
409 void enviaBluetooth() {
410     MPU6050();
411 }
412
413 void funcaoDadosBrutos() { //primeira função do aplicativo, referente
    ao envio de dados brutos
414     ex1 = false;
415     ex3 = false;
416     ex1 = true;
417     while (ex1 == true) {
418         var = SerialBT.read();
419         SerialBT.flush();
420         MPU6050();
421         if (var == 'p') {
422             ex1 = false;
423         }
424         delay(10);
425     }
426 }
427
428 void funcaoDadosMovimentoSimples() { //segunda função do aplicativo,
    referente ao calculo de amplitude angular e tempo
429     if (ex2 == false) {
430         ex1 = false;
431         ex3 = false;
```

```
432     mediaPontosX = 0;
433     mediaPontosY = 0;
434     mediaGxBruto = 0;
435     mediaGyBruto = 0;
436     Serial.println("Ta no loop");
437     for (int i = 0 ; i < 100 ; i++) { //pega 100 pontos para fazer a
média
438         MPU6050();
439         mediaPontosX += x;
440         mediaPontosY += y;
441         mediaGxBruto += Gx;
442         mediaGyBruto += Gy;
443         delay(10);
444     }
445     mediaGxBruto = mediaGxBruto / 100;
446     mediaGyBruto = mediaGyBruto / 100;
447     mediaPontosX = mediaPontosX / 100; //possui agora um ponto de
calibração, envia "c" para o celular identificar que a calibração
acabou
448     mediaPontosY = mediaPontosY / 100;
449     SerialBT.println("c");
450 }
451 primeiraInteracao = true;
452 ex2 = true;
453
454 while (ex2 == true) {
455     var = SerialBT.read();
456     SerialBT.flush();
457     if (var == 'p') {
458         ex2 = false;
459     }
460     MPU6050();
461     delay(10);
462 }
```

```
463 Serial.println("fugiu do loop");
464 anguloGyroX = 0;
465 anguloGyroY = 0;
466 primeiraInteracao = false;
467 ex2 = false;
468 }
469
470 void funcaoDadosMovimentoPorTempo() { //terceira e quarta funções do
    aplicativo, referente a identificação de movimento
471
472     if (ex3 == false) {
473         ex1 = false;
474         ex2 = false;
475         mediaPontosX = 0;
476         mediaPontosY = 0;
477         mediaGxBruto = 0;
478         mediaGyBruto = 0;
479         for (int i = 0 ; i < 100 ; i++) {
480             MPU6050();
481             mediaPontosX += x;
482             mediaPontosY += y;
483             mediaGxBruto += Gx;
484             mediaGyBruto += Gy;
485             delay(10);
486         }
487         mediaGxBruto = mediaGxBruto / 100;
488         mediaGyBruto = mediaGyBruto / 100;
489         mediaPontosX = mediaPontosX / 100;
490         mediaPontosY = mediaPontosY / 100;
491         SerialBT.println("c"); //envia c para avisar que a calibração
        acabou
492     }
493     primeiraInteracao = true;
494     ex3 = true;
```

```
495 while(ex3 == true){
496     var = SerialBT.read();
497     SerialBT.flush();
498     if (var == 'p') {
499         ex3 = false;
500         movimentoY = false;
501     }
502     MPU6050();
503     delay(10);
504 }
505 anguloGyroX = 0;
506 anguloGyroY = 0;
507 primeiraInteracao = false;
508 Serial.println("fugiu do loop");
509 ex3 = false;
510 }
511
512 void loop() {
513     recebeBluetooth();
514 }
```