



**TRABALHO DE GRADUAÇÃO**

**Estudo de Choque e Vibração em Raquetes de  
Tênis**

**Davilh Guimarães Nakamura de Lira  
Fernando Queiroz Campelo**

**Brasília, 20 de fevereiro de 2019**

**UNIVERSIDADE DE BRASÍLIA**

**FACULDADE DE TECNOLOGIA**

UNIVERSIDADE DE BRASÍLIA  
Faculdade de Tecnologia

## TRABALHO DE GRADUAÇÃO

# Estudo de Choque e Vibração em Raquetes de Tênis

Davilh Guimarães Nakamura de Lira  
Fernando Queiroz Campelo

Relatório submetido como requisito parcial para obtenção  
do grau de Engenheiro de Redes de Comunicações

### Banca Examinadora

Prof. Claudia Jacy Barenco Abbas, UnB/ ENE  
(Orientador)

---

Prof. Ricardo Zelenovsky, UnB/ ENE

---

Prof. Georges Daniel Amvame Nze, UnB/ ENE

---

## FICHA CATALOGRÁFICA

LIRA, DAVIH GUIMARÃES NAKAMURA. CAMPELO, FERNANDO QUEIROZ  
ESTUDO DE CHOQUE E VIBRAÇÃO EM RAQUETES DE TÊNIS, Distrito Federal,  
2019.

xii, 112 p., 210 x 297 mm (ENE/FT/UnB, Engenheiro, Engenharia de Redes e  
Comunicação, 2019).

Projeto Final de Graduação – Universidade de Brasília – Faculdade de Tecnologia.  
Departamento de Engenharia Elétrica

1. Raquete de Tênis

2. Vibração

3. Ondas de Choque

4. Impacto

I. ENE/FT/UnB

II. Título (Série)

## REFERÊNCIA BIBLIOGRÁFICA

LIRA, D. G. N.; CAMPELO, F. Q. (2019). *ESTUDO DE CHOQUE E VIBRAÇÃO EM RAQUETES DE TÊNIS*. Projeto de Graduação. Departamento de Engenharia Elétrica. Universidade de Brasília, Brasília, DF, 119 p.

## CESSÃO DE DIREITOS

AUTORES: Davih Guimarães Nakamura Lira e Fernando Queiroz Campelo

TÍTULO: ESTUDO DE CHOQUE E VIBRAÇÃO EM RAQUETES DE TÊNIS

GRAU: Engenheiros de Redes e Comunicação

É concedida a Universidade de Brasília permissão para reproduzir cópias desde Projeto Final de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. Os autores reservam outros direitos de publicação e nenhuma parte deste Projeto Final de Graduação pode ser reproduzido sem a autorização por escrito dos autores.

---

Davih Guimarães Nakamura Lira  
Depto. De Engenharia Elétrica (ENE) – FT  
Universidade de Brasília (UnB)  
Campus Darcy Ribeiro  
CEP: 70919-970 – Brasília – DF - Brasil

---

Fernando Queiroz Campelo  
Depto. De Engenharia Elétrica (ENE) – FT  
Universidade de Brasília (UnB)  
Campus Darcy Ribeiro  
CEP: 70919-970 – Brasília – DF - Brasil

### **Dedicatória(s)**

*À minha família, por sua capacidade de acreditar em mim e investir em mim. Mãe, seu cuidado e dedicação foi que deram, em alguns momentos, a esperança para seguir. Pai, sua presença significou segurança e certeza de que não estou sozinho nessa caminhada. Nathália, seu amor e carinho me ajudou a nunca desistir e nunca duvidar de mim.*

*Daviv Guimarães Nakamura de Lira*

*Á Deus, à minha família, à minha namorada Ana Carolina e aos meus amigos, que fazem de mim uma pessoa mais corajosa.*

*Fernando Queiroz Campelo*

## **Agradecimentos**

*Primeiramente a Deus, que sempre me iluminou e guiou meu caminho e pela força, coragem e fé que me ajudou a nunca desistir. Tudo é graças a Ele.*

*À minha família, Papai, Mamãe e Enzo, pelo amor, incentivo e apoio incondicional, que me conduziu as melhores decisões e nunca deixou de acreditar em mim. Aos meus avós e meus tios, vocês foram sempre meu ponto de conforto a quem podia contar nos melhores e nos piores momentos. Obrigado por me amarem incondicionalmente.*

*À minha namorada Nathália, pelo amor, carinho e atenção, que sempre segurou minha mão e caminhou comigo me apoiando em tudo e nunca deixando que me sentisse sozinho e desamparado. Me lembrou quem eu era no momento mais difícil e me mostrou o quão capaz eu sou de conseguir aquilo que eu quero. Eu amo você um tanto, mas um tanto. Que você esteja do meu lado em todas outras conquistas. Obrigado por ser você comigo.*

*Aos meus amigos, da UnB, do Maristão, da Nossa Senhora de Guadalupe, e aqueles que fiz durante todo o meu caminho, vocês são um bem de valor incalculável na minha vida, especialmente aqueles que dedicaram seu tempo livre para nos auxiliar neste projeto, sempre estando disponível para as incontáveis solicitações e desesperos feitos por nós neste projeto, como Antônio, com seu amor por ajudar e um carinho muito enorme por nós, Eduardo, com seu caminho e conhecimentos que nos guiaram muito bem, e Thales, com seu carinho durante todos estes anos de faculdade e total apoio com o que eu necessitava, nunca me senti sozinho, e a todos os outros que estão desde o começo desta jornada acadêmica e minha vida e que vão continuar presentes em minha vida com certeza, eu amo vocês.*

*À nossa orientadora Cláudia, pelo empenho dedicado à elaboração deste trabalho e pela paciência que teve durante todo o período, sem você isso não seria possível.*

*À minha dupla Fernando, pela amizade, entendimento, apoio, brincadeiras, que tornou este projeto mais fácil, mais tranquilo com sua alegria, inteligência e carinho. Agradeço a Deus por ter te conhecido e você passar esses anos de faculdade comigo, com inúmeras experiências, aulas e provas.*

*À todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado, de coração.*

*Daviv Guimarães Nakamura de Lira*

*Primeiramente agradeço a Deus por me abençoar e me guiar nas minhas escolhas com a dádiva de teu amor.*

*À minha família, Adenilda, Francisvaldo e Gabi, pelo amor e pelo apoio prestados ao longo de toda minha vida, sempre incentivando a realizar meus sonhos e fornecendo meios para que sejam concretizados.*

*À minha namorada, Ana Carolina, que desperta o que há de melhor em mim, e que foi fundamental para a realização deste - e de outros - projetos, você é a minha luz.*

*À minha orientadora e professora Claudia Barenco, que sempre se mostrou muito prestativa, acessível e forneceu todo amparo durante esse trabalho de 1 ano. Pela sua sabedoria e sensibilidade de nos guiar, elogiar e criticar quando necessário. Meu muito obrigado.*

*Aos meus amigos da faculdade André Andrade, Gustavo Madruga, Pedro Burached, Matheus Paulino, e, em especial ao Thales Von Sperling, que se colocou à disposição, nas suas disputadíssimas horas vagas, para contribuir com seus conhecimentos de programação e eletrônica em boa parte do processo, ao Antônio Bigonha, que possui o coração enorme e uma empatia inigualável, a minha dupla de projeto final e várias matérias Davih Nakamura, que possui uma sintonia incrível neste projeto e na nossa amizade, e sua capacidade de lidar, dividir e solucionar nossos desafios. Ao Eduardo Oliveira, que passou sua experiência e as diretrizes para o sucesso deste trabalho. Obrigado a todos envolvidos no projeto, diretamente e indiretamente.*

*Aos meus amigos da vida, que sempre estiveram ao meu lado e me motivando a atingir meus sonhos e objetivos, em especial Christopher Queiroz, que transborda motivação e energias positivas. De coração aberto, meu obrigado a todos.*

*Fernando Queiroz Campelo*

---

## RESUMO

O ato de jogar tênis expõe o corpo humano a ondas de choque e a forças de vibração causadas pelo impacto da bola de tênis com a raquete. Por meio da raquete, essas ondas de choque se propagam para o sistema musculoesquelético, afetando assim a saúde e a eficiência dos jogadores de tênis dentro de quadra. De forma objetiva, este estudo apresenta a comparação dos valores de vibração e do impacto de uma raquete com uma bola de tênis, com o acréscimo de massa a mesma com o intuito de comprovar se as tecnologias que encontram-se no mercado, como a Kinetic Q, realmente trazem benefícios com relação a redução da vibração e do impacto. Este estudo foi realizado por meio de testes em condições controladas utilizando sensores piezoelétricos para detecção de choque e vibrações e transmissões de dados via Bluetooth para um aplicativo no celular e uma posterior análise através da transformada de Fourier. Os valores coletados para análise do choque e da vibração foram: amplitude da frequência de vibração, potência de impacto da bola e duração da vibração. Neste estudo, conseguiu-se uma ótima redução do choque, 62,79%, e da duração da vibração, 47,22%, com estudos realizados com força denominada neste estudo como forte.

---

## ABSTRACT

The act of playing tennis exposes the human body to shock waves and the vibrational forces caused by the impact of the tennis ball with the racket. Through the racket, these shock waves propagate to the musculoskeletal system, thus affecting the health and efficiency of indoor tennis players. In an objective way, this study presents the comparison of the values of vibration and the impact of a racket with a tennis ball, with the addition of mass to the same one in order to verify if the technologies that are in the market, such as Kinetic Q, really bring benefits in terms of vibration and impact reduction. This study was carried out through tests under controlled conditions using piezoelectric sensors for shock and vibration detection and data transmission via Bluetooth to a mobile application and a later analysis through the Fourier transform. The values collected for shock and vibration analysis were: amplitude of vibration frequency, ball impact power and duration of vibration. In this study, a great reduction of shock, 62.79%, and vibration duration, 47.22%, was achieved with studies performed with strength denominated in this study as strong.

# SUMÁRIO

<b>1. INTRODUÇÃO</b> .....	14
<b>2. FUNDAMENTAÇÃO TEÓRICA</b> .....	16
2.1 BLUETOOTH LOW ENERGY.....	16
2.1.1 Introdução .....	16
2.1.2 Arquitetura .....	17
2.1.3 Aplicações .....	19
2.1.4 Bluetooth 4.1 .....	20
2.1.5 Bluetooth 4.2.....	20
2.2 TENSÃO MECÂNICA NA RAQUETE TRANSMITIDA PARA O BRAÇO .....	22
2.3 FATORES QUE AFETAM A VIBRAÇÃO DAS RAQUETES DE TÊNIS.....	25
2.4 TRANSFORMADA DE FOURIER.....	30
2.5 PÊNDULO FÍSICO .....	31
<b>3. EQUIPAMENTOS E SOFTWARES UTILIZADOS</b> .....	33
3.1 ARDUÍNO IDE 1.8.5 .....	33
3.2 ACELERÔMETRO.....	34
3.2.1 Introdução .....	34
3.2.2 Como funciona o acelerômetro .....	35
3.2.3 Acelerômetro utilizado no projeto .....	35
3.3 REDBEAR BLE NANO V2.....	36
3.4 BATERIA-MOEDA E SUPORTE.....	38
3.5 ESP32 .....	40
3.5.1 Conversor de Analógico para Digital.....	41
3.5.2 Características do ESP32 .....	42
3.6 BLYNK.....	44
3.7 SENSORES MINISENSE 100.....	46
3.7.1 Introdução .....	46
3.7.2 Funcionamento.....	47
3.7.3 Sensibilidade fora do eixo.....	49
3.7.4 Aplicações .....	49
3.8 MÓDULO BUZZER 5V PASSIVO .....	50
<b>4. EXPERIMENTOS REALIZADOS</b> .....	52
4.1 METODOLOGIA.....	52
4.2 TESTES REALIZADOS.....	53
4.2.1 Teste de comunicação RedBear BLE nano e IOS.....	53



4.2.2	Teste do Funcionamento do sensor de vibração com o envio para o celular.....	55
4.2.3	Passos para acoplar na raquete e testar o choque.....	60
4.2.4	Substituição do micro controlador e teste de precisão do choque .....	61
4.2.5	Dispositivo acoplado na raquete e primeiros resultados .....	63
4.2.6	Aperfeiçoamento do experimento .....	66
4.2.7	Repetição do experimento adicionando 3 massas de 3g .....	73
4.2.8	Estudo da vibração com os valores da batida forte .....	79
5.	<b>CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS.....</b>	<b>88</b>
6.	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>90</b>
	APÊNDICES .....	91

# LISTA DE FIGURAS

Figura 1 - Arquitetura BLE [1].....	18
Figura 2 - Pontos Fundamentais na raquete [9].....	23
Figura 3 - Uma raquete de tênis modelada como um feixe uniforme de comprimento L com uma massa adicional $m_A$ localizada a uma distância d da ponta. A bola impacta a uma distância B da ponta [18] .....	25
Figura 4 - Cálculos para um feixe uniforme de massa, com suporte livre, de 0,3 kg com resistência à flexão $EI = 300 \text{ Nm}^2$ quando a bola de tênis atinge $B = 60 \text{ mm}$ , a $10 \text{ m / s}$ . Os resultados mostram o deslocamento de y e a velocidade, $dy = dt$ , da ponta do feixe durante e após o impacto [18] .....	28
Figura 5 - Resultados calculados para feixes uniformes livremente suportados de massa variável e rigidez à flexão mostrando (a) a frequência de vibração fundamental do feixe (eixo direito), a amplitude de vibração da ponta final e (b) o coeficiente aparente de restituição, eA, quando bola de tênis é incidente normalmente a $10 \text{ m / s}$ para impactar o feixe em $B = 60 \text{ mm}$ [18] .....	28
Figura 6 - Resultados calculados para uma viga uniforme de 250 g mostrando os efeitos da adição de uma massa de 50 g na ponta ou ponta final ou ponto nodal na extremidade da ponta. A bola é incidente normalmente a $10 \text{ m / s}$ para impactar o feixe em (a) $B = 6 \text{ cm}$ ou (b) $B = 27 \text{ cm}$ [18].....	29
Figura 7 - Os resultados calculados para a viga de 250 g, $EI = 350 \text{ Nm}^2$ na Fig. 3 mostram o efeito da variação da rigidez do plano da corda, $k_s$ para três valores diferentes da rigidez da esfera, $k_1$ . A bola é incidente normalmente a $10 \text{ m / s}$ para impactar o feixe em $B = 60 \text{ mm}$ [18] .....	30
Figura 8 - Pêndulo Simples [24].....	32
Figura 9 - Funcionamento do Arduino IDE.....	34
Figura 10 - Pinos X, Y, Z já descritos.[17].....	36
Figura 11 - Pinos do RedBear BLE Nano 2 (Referencia de um projeto cujo link esta no arquivo referencia) [17].....	37
Figura 12 - Placas BLE Nano 2 e DAPlink [17] .....	38
Figura 13 - Bateria-moeda CR2032 [26] .....	39
Figura 14 - Suporte Lilypad [27].....	39
Figura 15 - Circuito do suporte LilyPad [27].....	40
Figura 16 - Pinagem do ESP32 [28].....	41
Figura 17 - Chip ESP32 [28].....	42
Figura 18 - Características do Conversor ADC [28] .....	42
Figura 19 - Pinos do ESP32 [28] .....	43
Figura 20 - Diagrama de blocos de componentes do ESP32 [28].....	44
Figura 21 - Funcionamento do Blynk [20] .....	45
Figura 22 - Interface do Aplicativo .....	46
Figura 23 - dimensões, em mm, dos sensores utilizados [21].....	47
Figura 24 - gráficos da linearidade e da resposta na frequência [21] .....	48
Figura 25 - Características elétricas do sensor [21] .....	48
Figura 26 - Resposta do sensor nos eixos [21].....	49
Figura 27 – Buzzer [23].....	51
Figura 28 - Circuito do módulo Buzzer [23] .....	51
Figura 29 - Trecho do código “SimpleChat” .....	54
Figura 30 - Ferramenta “Serial Monitor” mostrando os dados enviados e recebidos	54

Figura 31 - Interface do dispositivo móvel .....	55
Figura 32 - Declaração das variáveis.....	56
Figura 33 - Montagem do Circuito utilizando o RedBear e o Sensor MiniSense 100	57
Figura 34 - Resultado com movimentos feitos no sensor .....	58
Figura 35 - Criação dos objetos.....	59
Figura 36 - Função Atualiza Sensor.....	59
Figura 37 - Definição do Tempo do Ticker.....	59
Figura 38 - Mini Protoboard utilizada [29] .....	60
Figura 39 - Localização do Sensor na Raquete [30] .....	60
Figura 40 - Circuito montado com o ESP32 .....	62
Figura 41 - Valores da Voltagem impressos em formato de tabela no aplicativo Blynk .....	62
Figura 42 - Circuito com o ESP32 e o buzzer .....	63
Figura 43 - Raquete com o sensor encaixado na rede .....	65
Figura 44 - Valores sendo impressos na tabela com o sensor na rede .....	65
Figura 45 - Valores apresentados no app Blynk .....	66
Figura 46 - Demarcação da posição ideal da raquete .....	68
Figura 47 - Protótipo do experimento .....	68
Figura 48 - Posição do sensor .....	69
Figura 49 - Peso da bola de tênis.....	69
Figura 50 - 4 choques arremessados com a força fraca .....	70
Figura 51 - 5 choques arremessados com a força forte .....	71
Figura 52 - Valores para força fraca no aplicativo Blynk .....	72
Figura 53 - Posicionamento das fitas de 3g.....	75
Figura 54 - Ambiente semelhante ao do teste da seção 4.2.6 .....	76
Figura 55 - 5 batidas fracas com o acréscimo de peso de 9g .....	77
Figura 56 - Captura de valores dos choques.....	77
Figura 57 - 5 batidas fortes com o acréscimo de peso de 9g.....	78
Figura 58 - Gráfico em Volts da batida forte sem massa (Segundos x Volts) .....	80
Figura 59 - Gráfico de Fourier da batida forte sem massa (Hz x Amplitude) .....	80
Figura 60 - Gráfico da batida forte com massa (Segundos x Volts) .....	81
Figura 61 - Gráfico de Fourier da batida forte com massa (Hz x Amplitude) .....	81

## LISTA DE TABELAS

Tabela 1 - Dados Resultantes com a batida fraca e forte .....	72
Tabela 2 - Dados resultantes com a batida fraca e forte e o acrescimo de 9g.....	78
Tabela 3 - Valores calculados após a análise com a transformada de Fourier .....	81
Tabela 4 - Valores das amplitudes calculados pela Transformada de Fourier .....	82
Tabela 5 - Valores calculados após a análise da transformada de Fourier e priorização da frequência harmônica .....	85
Tabela 6 - Valores das amplitudes calculados pela Transformada de Fourier sem as frequências secundárias .....	85

# LISTA DE SÍMBOLOS

## Símbolos Latinos

$A$	Área	[m <sup>2</sup> ]
$C_p$	Calor específico a pressão constante	[kJ/kg.K]
$h$	Entalpia específica	[kJ/kg]
$\dot{m}$	vazão mássica	[kg/s]
$T$	Temperatura	[°C]
$U$	Coeficiente global de transferência de calor	[W/m <sup>2</sup> .K]

## Símbolos Gregos

$\alpha$	Difusividade térmica	[m <sup>2</sup> /s]
$\Delta$	Varição entre duas grandezas similares	
$\rho$	Densidade	[m <sup>3</sup> /kg]

## Grupos Adimensionais

$Nu$	Número de Nusselt
$Re$	Número de Reynolds

## Subscritos

$amb$	ambiente
$ext$	externo
$in$	entrada
$ex$	saída

## Sobrescritos

$\dot{\phantom{x}}$	Varição temporal
$\bar{\phantom{x}}$	Valor médio

## Siglas

ABNT	Associação Brasileira de Normas Técnicas
------	--

# 1. INTRODUÇÃO

*Este capítulo apresenta as principais motivações e estudos relacionados ao impacto e vibração nas raquetes de tênis durante o choque com a bola. Também são descritas as premissas do projeto e as tecnologias utilizadas no mesmo.*

Este trabalho teve como motivação o trabalho desenvolvido por Isabelle Rogowski [8]. Neste estudo, nove tenistas realizaram uma série de movimentos de *forehand*, que é uma técnica de golpe direta realizada no tênis movimentando a raquete com a palma da mão direcionada para a frente, usando uma raquete leve e altamente vibrante. Dois acelerômetros foram fixados na estrutura da raquete e no pulso do jogador. A análise dos sinais de vibração nos domínios de tempo e frequência não mostrou efeito de interação entre as condições de velocidade e de raquete, tanto no comportamento de vibração da raquete quanto na transmissão de choque. Um aumento na velocidade de batida aumentou a quantidade de tempo de vibração na raquete e no pulso, mas alterou fracamente sua vibração. Em comparação com uma raquete altamente vibrante, uma raquete levemente vibrante amortecia mais o impacto, e exibia uma quantidade menor de amplitude na frequência harmônica do sinal de vibração na raquete e no pulso. Estas descobertas indicaram que a velocidade de jogo deve ser controlada quando se investiga as vibrações devido à raquete em condições reais de jogo. Da mesma forma, uma redução de vibração percebida pelo tenista estaria ligada à diminuição da amplitude do sinal de vibração da raquete, que pode concentrar a energia do sinal nas baixas frequências e assim tender a estabilidade.

O objetivo geral deste estudo é comparar os valores de vibração e do impacto de uma raquete com uma bola de tênis, com o acréscimo de massa a mesma com o intuito de comprovar se as tecnologias que encontram-se no mercado, como a Kinetic Q, realmente trazem benefícios com relação a redução da vibração e do impacto.

Em nossos estudos, serão feitos testes com uma raquete e com mudanças na massa desta, utilizando pequenas fitas de massas, para verificar o choque de uma bola de tênis com a raquete e conseguir mensurar esse efeito em volts. Com isso, nosso estudo será diferente do mencionado em [8]. Após a análise em volts, os gráficos obtidos a partir desta serão utilizados para um estudo acerca das vibrações

posteriores ao choque, relacionando-as com as frequências, por meio da transformada de Fourier.

Os objetivos específicos para este estudo são: Determinar o microcontrolador adequado; Escolher um sensor e sua posição na raquete; Estudar as melhores posições para adição das massas na raquete; Definir metodologia a ser realizada no projeto; Criar um ambiente mais controlado possível; Desenvolver um aplicativo para recepção dos dados; Realizar os testes; Analisar o impacto da bola de tênis na raquete com o método escolhido

Serão utilizados sensores piezoelétricos, transmissores Bluetooth e um microcontrolador para enviar os dados para o celular e será utilizado o próprio Bluetooth do celular para receber estes dados. Os transmissores Bluetooth utilizaram a tecnologia do Bluetooth orientado à conexão com o dispositivo móvel para poder enviar esses dados de maneira frequente durante um jogo ou durante o tempo escolhido para realização dos testes.

## 2. FUNDAMENTAÇÃO TEÓRICA

*Essa seção do trabalho apresenta alguns conceitos e fundamentos teóricos que são necessários para que torne mais simples a compreensão do contexto em que o projeto se aplica e as etapas que foram seguidas até a conclusão do experimento. Primeiramente é explicado a principal idéia do experimento, a comunicação via Bluetooth, e, posteriormente, são explicados quais equipamentos e softwares foram utilizados: Redbear, Arduino IDE, sniffers, sensores, acelerômetro, dispositivo iOS e também os algoritmos aplicados ao estudo.*

### 2.1 BLUETOOTH LOW ENERGY

#### 2.1.1 Introdução

Bluetooth Low Energy (BLE) é uma tecnologia sem fio com baixo consumo de energia projetada para ser incorporada em vários dispositivos. É considerado como *power friendly*, pois permite que os dispositivos consumam menos energia durante a execução em um longo período de tempo.

Outra característica interessante do BLE é que esta tecnologia já vem incorporada em dispositivos tablets e smartphones. Essa capacidade permite que a solução deste projeto possa ser usada em bilhões de dispositivos que já estão no mercado.

Seu sistema é projetado para transmitir pequenos pacotes de dados, com menos energia e custo, que fazem com que o dispositivo dure durante anos utilizando apenas baterias minúsculas [13], como a bateria-moeda CR2032.

O BLE é a base de *smart devices*, como relógios, brinquedos, mouses e muitos outros equipamentos. O BLE possui muitos recursos como: “ultra-low peak, ultra-low cost e um tamanho pequeno para acessórios e dispositivos de interface humana”. O custo de implementação é muito baixo e possui um sistema seguro compatível com vários fornecedores.



Em 2012 o IETF 6LoWPAN working group (WG) reconheceu a importância do BLE para a *Internet of things*, onde, eles estabeleceram uma especificação que aprimora a transmissão de pacotes IPV6 sobre BLE. [1]

### 2.1.2 Arquitetura

A arquitetura do BLE, como mostrado na Figura 1, consiste em três blocos básicos: Controlador, Host e Aplicativos. Cada bloco tem uma função e um protocolo específico:

Primeiro bloco: bloco de controlador que contém camada física, camada de enlace e modo de teste direto.

O segundo bloco é a bloco Host que é executado em um processador de aplicativos e funciona no nível superior e consiste em vários protocolos que são: “Link lógico e protocolo de adaptação (L2CAP)”, “protocolo de atributo (ATT)”, “protocolo gerenciador de segurança (SMP)”, “atributo genérico perfil (GATT)” e “perfil de acesso genérico (GAP)”. Existe uma “interface de controlador de host (HCI)” que atua como uma interface e como um ponto de comunicação entre o Controlador e a bloco de Host.

O terceiro bloco é o bloco de aplicações.

Agora, será esclarecido cada camada e funções deste protocolo de maneira breve:

1. Camada Física: BLE trabalha na banda 2.4 GHz ISM, onde possui 40 canais de radiofrequência com espaçamento de canal de 2 MHz, este recurso favorece o surgimento de chips de rádio baratos e simples. A BLE RF possui dois tipos de canais: canais de publicidade e canais de dados.
2. Camada de enlace: transporta os pacotes físicos na banda de rádio, e gerencia o tempo para envio e recebimento de pacotes.
3. L2CAP: é responsável por multiplexar e segmentar os pacotes de dados para as camadas de nível superior ATT, SMP e sinalização da camada de enlace, bem como coletar os pacotes do controlador antes que eles transmitam para os níveis mais altos. L2CAP tem 23 bytes como tamanho de *payload* onde os pacotes de dados do protocolo da camada superior não podem exceder esse tamanho.

4. ATT: é responsável por fornecer uma comunicação para um pequeno pacote de dados sobre o canal L2CAP. O protocolo de atributos pode ser descoberto, lido ou escrito por outro dispositivo porque possui valores ou atributos que são armazenados e gerenciados pelo GATT. No entanto, o cliente pode enviar um pedido para o servidor para conhecer os atributos do servidor, por outro lado, o servidor pode enviar notificações ou indicações para o cliente.
5. GATT: é usado para determinar a estrutura ATT e a transferência de pacotes que contém um valor e propriedades (características) entre dispositivos.
6. Segurança: O BLE fornece um ambiente seguro para troca de dados entre dispositivos, onde os serviços suportados podem ser trocados em "modo de segurança LE 1 ou modo de segurança LE 2". Esses dois modos operam o propósito de segurança na camada de enlace e na camada ATT. A camada de enlace utiliza o "*Cipher Block Chaining*" para auxiliar o processo de criptografia e autenticação. Além disso, o recurso de privacidade de suporte BLE permite que o dispositivo envie e receba usando um endereço particular.
7. GAP e perfis de aplicativos: o GAP está localizado na camada superior da pilha BLE que promete a conexão do servidor com clientes usando os serviços, onde ele gerencia e define as funções dos dispositivos de descoberta e é responsável por determinar os modos e métodos para realizar uma conexão segura. O GAP com a melhoria do controlador desempenha quatro funções: "emissora, observador, periférico e central". [1]

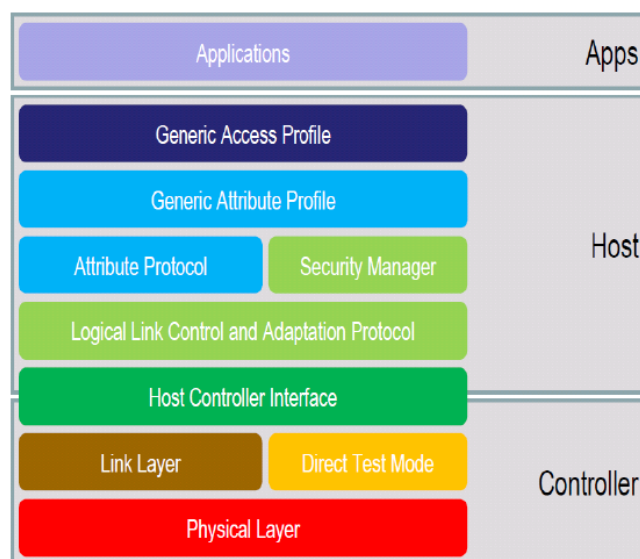


Figura 1 - Arquitetura BLE [1]

### 2.1.3 Aplicações

O Bluetooth Low Energy é projetado para funcionar em diversas aplicações em dispositivos mais modernos, especialmente para dispositivos que buscam baixo consumo de energia, onde foram os maiores problemas enfrentados pelo Bluetooth clássico. Seu baixo custo e uso simples facilita sua produção. Além disso, eles podem usar o serviço por anos [12] usando, por exemplo, bateria-moeda CR2032 que facilitam a manutenção. Os próximos parágrafos mostram alguns usos do BLE:

- A. Smart Energy: Hoje em dia muitos smartphones são capazes de se conectar uns aos outros através da implementação de Bluetooth v4.0, onde a Apple foi a primeira no mercado que instalou Bluetooth v4.0 em seus dispositivos, em seguida, outros como Android, RIM Blackberry, Symbian e Windows Phone 7. Além disso, uma das aplicações interessantes é o sistema de chamadas sem uso de mão usado na maioria dos novos carros e fones de ouvido. O BLE é encontrado em sensores domésticos com perfil de segurança, como o alarme de proximidade que pode ser usado para crianças, em que este dispara um alarme para os pais quando as crianças se moverem de alguma forma.
- B. Saúde e bem-estar: O crescimento do uso da tecnologia BLE, permite ao usuário verificar e usar modelos médicos para monitorar a frequência cardíaca e a pressão arterial em casa apenas conectando-se a seus smartphones. Existem vários dispositivos médicos disponíveis no mercado, como monitores de glicose no sangue, oxímetros de pulso, monitores de frequência cardíaca, inaladores de asma e estetoscópios.
- C. Esporte e fitness: O BLE é usado no campo de esporte e *fitness* onde, a pessoa pode usar seu smartphone que possui o BLE como no Android que tem uma aplicação de saúde que conta horas de caminhada do usuário e auxiliam no exercício informando sua perda de peso, organizando um cronograma para seu exercício e monitora também sua alimentação utilizando a medição de calorias. Alguns materiais esportivos também vêm com o chip BLE como, por exemplo, alguns modelos de calçados esportivos, o monitor de frequência cardíaca e os relógios esportivos que mostram os sinais vitais de um atleta durante o exercício e transmitem seus passos ao celular para compartilhar sua localização e mapear com os outros. Também pode ser utilizado em esportes como o tênis que é o foco deste estudo. [1]

#### **2.1.4 Bluetooth 4.1**

A especificação Bluetooth 4.1 foi lançada em 2013. O Bluetooth 4.1 incorpora uma mudança fundamental no que diz respeito ao suporte de rede de malha BLE; um dispositivo, independentemente de sua função de camada de enlace, pode executar várias instâncias da camada de enlace simultaneamente, sem limitação. Assim, um escravo pode ser conectado simultaneamente a mais de um mestre. Além disso, um dispositivo pode atuar como escravo em determinados intervalos e como mestre em outros, mantendo comunicações paralelas com seus vizinhos. Isso abre as portas para a criação de topologias de rede estendidas além da topologia de estrela, como a topologia de malha. No entanto, a arquitetura e os mecanismos para a formação e operação de uma rede de malha BLE não estão definidos na especificação Bluetooth 4.1. [3]

#### **2.1.5 Bluetooth 4.2**

O Bluetooth 4.2, publicado em 2014, incorpora melhorias principalmente em três áreas; conectividade com a internet, segurança aprimorada e maior rendimento. Essas atualizações destinam-se a aumentar as possibilidades do BLE como uma tecnologia para o IoT. No entanto, o Bluetooth 4.2 não oferece funcionalidade adicional para suportar redes em malha BLE. [3]

##### **2.1.1.1 Conectividade com a Internet**

Embora seja compatível com o Bluetooth 4.1, o Perfil de Suporte ao Protocolo da Internet (IPSP) é lançado com o Bluetooth 4.2. O IPSP fornece suporte a uma central de Bluetooth habilitada para IPv6 e a um periférico para descobrir um ao outro e estabelecer uma conexão de camada de enlace. Os Perfis de Atributos Genéricos (GATT) do Bluetooth LE ajudam a descobrir se o IPSP é suportado e o Modo de Controle de Fluxo Baseado em Crédito Bluetooth LE L2CAP é usado para trocar dados. No GATT, o Serviço de Suporte IP (IPSS) é usado para determinar o suporte

para a função Nó do IPSP. Embora houvesse esforços anteriores para conectar dispositivos Bluetooth LE com a Internet [5] [4] como um esboço da IETF, onde se discute a maneira padronizada de transmitir pacotes IPv6 através de Bluetooth LE. Eles explicam o uso de técnicas 6LoWPAN e a conexão entre 6LoWPAN e Bluetooth LE. Eles adaptam o mecanismo comprimido do cabeçalho 6LoWPAN através do Bluetooth LE.

#### 2.1.1.2 Melhoria na Segurança

Para melhorar a eficiência energética, o Bluetooth LE 4.0 inicialmente tem uma segurança mais fraca em comparação com o Bluetooth padrão. Ao contrário das versões 4.1 e anteriores do Bluetooth, o Bluetooth 4.2 vem com uma forte segurança que traz o nível de segurança Bluetooth LE igual ao Bluetooth padrão. Para o gerenciamento de chaves, o Bluetooth 4.2 é equipado com a Criptografia de Curva Elíptica (ECC) assimétrica em escala industrial, com curvas elípticas recomendadas pelo FIPS (Federal Information Processing Standard). Ele também usa a criptografia AES-CCM (Advanced Encryption Standard - Counter with CBC-MAC) aprovada do FIPS para criptografia de mensagens. Atualmente, os recursos de segurança do Bluetooth 4.2 fornecem segurança entre dois dispositivos vizinhos. Essa segurança de camada de enlace pode proteger os links sem fio em contraespionagem passiva e, em alguns casos, ataques MITM (*Man-in-the-Middle*), dependendo da disponibilidade de um monitor / teclado para o dispositivo. [4]

#### 2.1.1.3 Melhoria na Privacidade

Desde o Bluetooth 4.0, o Bluetooth LE suporta a mudança frequente do endereço do dispositivo Bluetooth (chamado de endereço privado) para limitar a capacidade de rastrear dispositivos Bluetooth LE por um período de tempo. Um dispositivo Bluetooth LE que deseje estabelecer uma conexão com outro dispositivo deve ser capaz de resolver endereços privados que são gerados usando a chave de identificação de resolução (IRK) compartilhada durante o processo de conexão. Além da resolução de endereço privado no host (com suporte no Bluetooth 4.1), o Bluetooth 4.2 adiciona resolução de endereço particular no controlador. Além disso, o Bluetooth 4.2 suporta uma lista branca de endereços privados no Controlador. Isso significa que

o Controlador na camada de enlace pode resolver e gerar endereços privados sem envolver o Host e pode aceitar ou rejeitar a solicitação recebida consultando a lista de permissões. Isso reduz significativamente a frequência de ativação do host (ou seja, chips Bluetooth LE) e, portanto, consome menos energia. Quando o Host está envolvido, por exemplo, quando o Controlador não pode armazenar todos os IRKs, a filtragem do dispositivo deve ser desativada, o que resulta em maior consumo de energia. [4]

#### 2.1.1.4 Melhoria no Tamanho do Pacote

Um dos recursos proeminentes do Bluetooth 4.2 é o aumento do tamanho do pacote Bluetooth LE, quase em 10 vezes em comparação com o Bluetooth 4.1 LE. Além disso, a taxa de transferência de dados no Bluetooth 4.2 LE é aumentada em até 2,5 vezes. Essa capacidade aumentada facilita a comunicação eficiente e abre novas possibilidades de uso do Bluetooth LE. Por exemplo, devido ao aumento do tamanho do pacote, a comunicação IP tornou-se eficiente em dispositivos Bluetooth LE. O uso adicional de maior capacidade é: atualizações de firmware rápidas e frequentes, capacidade de executar protocolos de segurança e comunicação na Internet e *upload* rápido de registros de dados de sensores para o back-end de computação (um computador, smartphone ou nuvem). [4]

## 2.2 TENSÃO MECÂNICA NA RAQUETE TRANSMITIDA PARA O BRAÇO

Na referência [9] explica-se como a tensão gerada pela raquete de tênis é transmitida para o braço. O artigo diz que jogar tênis expõe o corpo humano a impactos repetitivos da bola com as cordas, causando repetidas ondas de choque, descendo pela raquete e propagando-se através do sistema musculoesquelético. Após a bola ter deixado as cordas, o braço do jogador começa a experimentar não apenas as tensões mecânicas resultantes do choque da bola com a raquete, mas também os esforços resultantes das vibrações da raquete. Tais vibrações induzem desconforto ao jogador. Embora não exista evidência clínica publicada até o momento, supõe-se que a transmissão repetitiva de choques no antebraço afete os membros superiores dos tenistas e contribua para a ocorrência de lesões por uso repetitivo.

Para limitar esses riscos, as empresas de equipamentos oferecem projetos inovadores, materiais compostos e apertos ou dispositivos de amortecimento destinados a reduzir as vibrações da raquete. Enquanto os amortecedores de cordas, que são pequenas peças de borracha colocadas nas cordas para reduzir sua vibração, são responsáveis por reduzir o desconforto na mão e no braço, estudos anteriores concluíram que eles não têm nenhum outro efeito nas sensações do jogador além de reduzir o som do impacto [8]. Um amortecedor de núcleo de várias camadas inserido na corda da raquete como para reduzir a rigidez da raquete, e por sua vez, limita o tempo de amortecimento e amplitudes de vibrações. Os benefícios reivindicados são principalmente baseados em resultados obtidos por simulação em testes laboratoriais. Como uma raquete fixa pode se comportar de maneira diferente de uma raquete na mão do jogador, a investigação das vibrações da raquete e da transmissão de choque sob condições reais de jogo seria de interesse para trazer conhecimento adicional sobre a relação jogador-raquete.

Seja qual for o desenho e os componentes do material da raquete, duas áreas notáveis na região das cordas fornecem benefícios particulares para o jogador. Primeiro, a área de mínima vibração após o impacto, na Figura 2 é chamado de vibration node, se uma bola impactar neste ponto, a força transmitida para a mão é suficientemente pequena para que o jogador quase não perceba que o impacto ocorreu. Um impacto no nó cancela a frequência fundamental do quadro da raquete. Em segundo lugar, o centro da percussão é a área de choques mínimos para o braço (ver Figura 2).

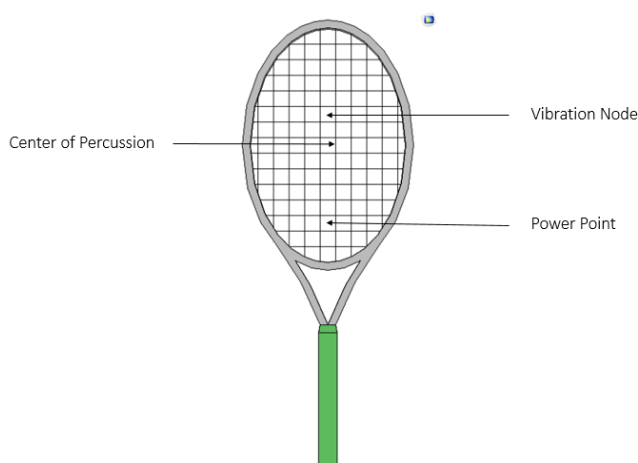


Figura 2 - Pontos Fundamentais na raquete [9]

Os jogadores de tênis, no entanto, nem sempre acertam as bolas de tênis nestas duas áreas ótimas durante os movimentos de *forehand* reais. A falta de habilidades de tênis para jogadores iniciantes ou uma melhor exploração das propriedades da raquete para acertar um tiro poderoso para jogadores experientes são as principais razões para impactos descentralizados, que resultam em aumento de vibrações de raquete e aumento transferência de choque para o braço do jogador. A maior parte da energia de vibrações detectada na raquete, subsequentemente, ao impacto com a bola, está relacionada a frequências abaixo de 3000 Hz, com a maior proporção na direção fora do plano nas frequências de 80–200 Hz para a estrutura da raquete. A maioria das frequências acima de 500 Hz é atribuída ao *stringbed*. Curiosamente, uma análise de vibração anterior indicou que há energia nas frequências de dados no plano associadas a modos de vibração que são predominantemente fora do plano. Isso sugere que as formas de modo da raquete sugerem modelo tridimensional ao invés do modelo de bidimensional. Além disso, o acoplamento mecânico da mão com a raquete reduz a frequência do modo fundamental em 10% e o tempo de amortecimento da vibração de 175 a 780 ms para 20 a 30 ms. Os picos de algumas frequências observadas durante a análise modal são eliminados quando se pratica um verdadeiro *forehand*.

Condições de jogo como força de agarramento da raquete e velocidades de bola e raquete afetam a carga na mão e podem, portanto, influenciar também as vibrações da raquete e a transmissão de choque. Estas descobertas anteriores destacam a necessidade de se criar um modelo tridimensional de vibração na raquete de tênis em condições reais de jogo, a fim de contribuir para uma melhor compreensão da transmissão do choque para o antebraço do jogador.

O objetivo do estudo feito era, portanto, investigar os efeitos de duas raquetes diferentes e velocidades de *forehand* no comportamento de vibração da raquete e transmissão de choque para o pulso do jogador em verdadeiros *forehand*. Hipotetizou-se que as raquetes de tênis seriam discriminadas com base nos descritores de tempo e frequência do comportamento de vibração e que quanto mais rápido o impulso frontal, maior a vibração da raquete e a transmissão de choque. [8]

Este trabalho tem o objetivo, também, de melhorar o descrito em [8] com o uso de um sensor mais preciso de detecção de choque, transmissões com maior capacidade, mais confiável e de maneira mais amigável, com o desenvolvimento de



um aplicativo para celular utilizando a tecnologia bluetooth e com a solução de rede de dados implantada, possa se tornar um produto de mercado.

## 2.3 FATORES QUE AFETAM A VIBRAÇÃO DAS RAQUETES DE TÊNIS

Medições e cálculos são apresentados sobre a vibração do cabo de uma raquete de tênis (ver Figura 4, Figura 5, Figura 6, Figura 7). É mostrado que a amplitude da vibração depende principalmente da rigidez da raquete, mas também depende da massa da raquete, da distribuição da massa, da frequência de vibração, da localização do impacto, da duração do impacto, da rigidez da bola e da rigidez do plano da corda. A massa adicionada na ponta de uma raquete, como sugere a maioria dos fabricantes dessas pequenas massas, geralmente reduz as vibrações do cabo quando a bola impacta perto da ponta, mas aumenta as vibrações do cabo para impactos na região da garganta.

A massa adicionada à alça da raquete (ver Figura 3) geralmente reduz as vibrações da alça, independentemente do ponto de impacto. A massa adicionada a um determinado ponto de vibração tem efeito na frequência de vibração, apenas um pequeno efeito na amplitude e pode ser útil adicionada a raquetes muito leves e rígidas para aumentar o poder da raquete sem afetar a baixa amplitude de vibração destas raquetes. A amplitude do modo de vibração fundamental diminui à medida que a duração do impacto aumenta, diminuindo para zero em todos os pontos de impacto quando a duração do impacto é 1,5 vezes maior que o período de vibração fundamental, como apresenta a seção 4.2 deste trabalho.

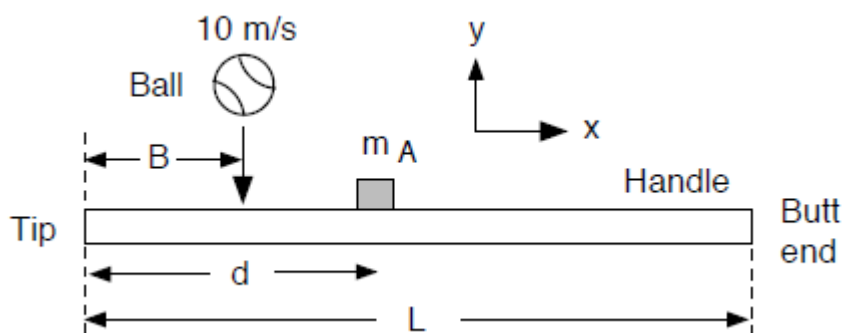


Figura 3 - Uma raquete de tênis modelada como um feixe uniforme de comprimento  $L$  com uma massa adicional  $m_A$  localizada a uma distância  $d$  da ponta. A bola impacta a uma distância  $B$  da ponta [18]

Uma característica desejável de uma raquete de tênis é que ela tenha uma sensação agradável nas mãos de um jogador. Para esse fim, os fabricantes adotaram muitos métodos diferentes para reduzir as vibrações transmitidas à mão e ao braço quando um jogador golpeia uma bola. Um desses métodos é endurecer o quadro aumentando sua largura na direção de dobra. Raquetes deste tipo são conhecidas como raquetes de corpo largo. O aumento da rigidez significa que a raquete é mais difícil de dobrar, com o resultado que a frequência de vibração é aumentada e a amplitude de vibração é reduzida. Os métodos disponíveis para um jogador incluem o uso de cordas em baixa tensão e / ou uma corda de baixa rigidez, e usando um punho macio cobrindo o braço da raquete. Uma rigidez do plano de corda baixa atua reduzindo a força de impacto, diminuindo assim a força de flexão e a amplitude de vibração.

A maioria dos outros métodos empregados pelos fabricantes para reduzir a vibração envolve o uso de materiais passivos de absorção de energia dentro da raquete para amortecer as vibrações. O amortecimento ativo também tem sido usado, através do qual, elementos piezoelétricos são usados para detectar e dobrar a raquete de tal forma que as vibrações sejam parcialmente canceladas. Amortecedores de cordas são usados por muitos jogadores para reduzir as vibrações das cordas, mas são leves demais para ter qualquer efeito significativo nas vibrações da estrutura. Seu objetivo principal é reduzir o som do impacto, apesar do fato de que os jogadores afirmam que uma raquete "se sente" melhor quando um amortecedor de corda é usado. A mão humana também atua como um amortecedor de vibrações muito eficaz [19]

Muitos dos fatores que contribuem para lidar com vibrações determinam a amplitude de vibração de uma raquete de tênis. Uma é que o modo de vibração fundamental não é excitado se a bola impactar no nó de vibração fundamental próximo ao meio das cordas. Em outros pontos de impacto, o modo fundamental é excitado com amplitude crescente à medida que o ponto de impacto se afasta mais do ponto de nó, como explicitado na seção 2.2.

Outro efeito está relacionado ao fato de que a duração do impacto com uma bola de tênis é tipicamente de cerca de 5 ms. Como resultado, os modos com uma frequência de vibração superior a aproximadamente 200 Hz, são excitados apenas de uma forma baixa. Para excitar modos de vibração de alta frequência com amplitude relativamente grande, é necessário reduzir a duração do impacto para cerca de 1 ms

ou menos, por exemplo, batendo numa armação de raquete com um martelo de impacto. Conseqüentemente, os modos de vibração fundamental e todos os de alta frequência de uma raquete de tênis são relativamente fracos em condições normais de jogo se a frequência de vibração fundamental for de aproximadamente 200 Hz ou mais. A frequência de vibração fundamental de uma raquete de tênis moderna é tipicamente entre cerca de 130 Hz e cerca de 160 Hz, dependendo da massa e rigidez do quadro [18]. No entanto, existem algumas raquetes muito rígidas de corpo largo que têm uma frequência de vibração fundamental em torno de 200 Hz ou mais. Tais raquetes podem ser descritas como tendo um sweet point relativamente grande, significando que a amplitude de vibração é pequena para impactos sobre a maior parte do plano da corda. Um exemplo do último tipo de raquete é a raquete Prince Viper, que se tornou disponível no início dos anos 2000. Sendo muito rígida e com uma massa de apenas 260 g, tem uma frequência de vibração fundamental de cerca de 230 Hz, dependendo da corda e da massa do punho.

A força de impacto em uma raquete quando é atingida por uma bola, pode ser representada por uma forma de onda meio senoidal. Se a duração do impacto for  $s$ , o espectro de frequência dessa forma de onda em particular terá amplitude máxima na frequência zero e diminuirá na amplitude zero em uma frequência de  $1: 5 = s$ . Conseqüentemente, o modo fundamental não é excitado se a frequência do modo fundamental for  $1/5 = s$ . Por exemplo, se a duração do impacto for estendida para 7 ms usando cordas suaves ou uma bola macia, e se a frequência de vibração fundamental da raquete for 214 Hz, o modo fundamental não será excitado, independentemente do ponto de impacto. Mesmo que a frequência de vibração seja de apenas 200 Hz, ou se a duração do impacto for de 6,5 ms em vez de 7,0 ms, a amplitude da vibração ainda será muito pequena. De fato, a duração do impacto poderia exceder 7,0 ms se jogadores casuais usassem bolas antigas e cordas antigas, batessem na bola suavemente e impactassem a bola na garganta, em vez da região da ponta da raquete. É somente quando os jogadores acertam a bola firmemente usando cordas duras e bolas novas e duras que a duração do impacto é tipicamente de cerca de 5 ms. Para um impacto de alta velocidade, a duração do impacto é tipicamente de cerca de 4 ms.

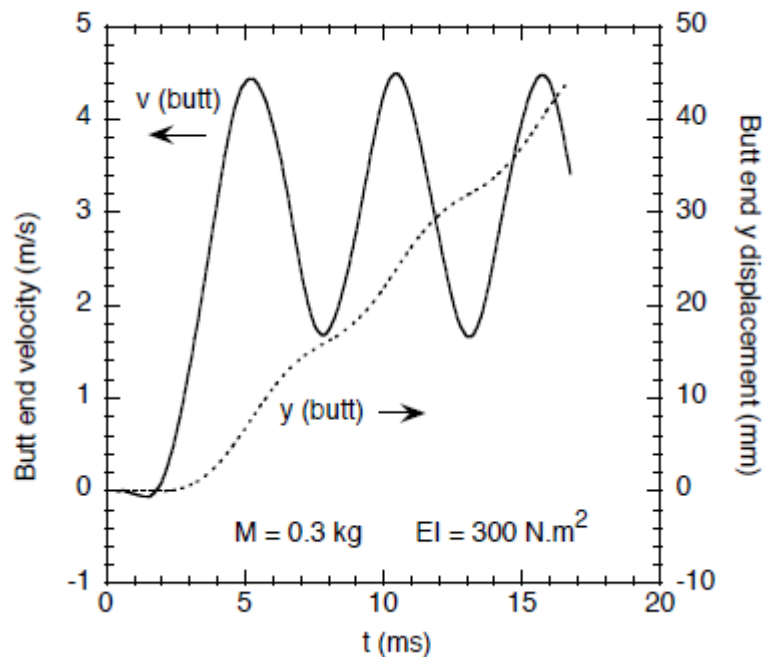


Figura 4 - Cálculos para um feixe uniforme de massa, com suporte livre, de 0,3 kg com resistência à flexão  $EI = 300 \text{ Nm}^2$  quando a bola de tênis atinge  $B = 60 \text{ mm}$ , a  $10 \text{ m/s}$ . Os resultados mostram o deslocamento de  $y$  e a velocidade,  $dy = dt$ , da ponta do feixe durante e após o impacto [18]

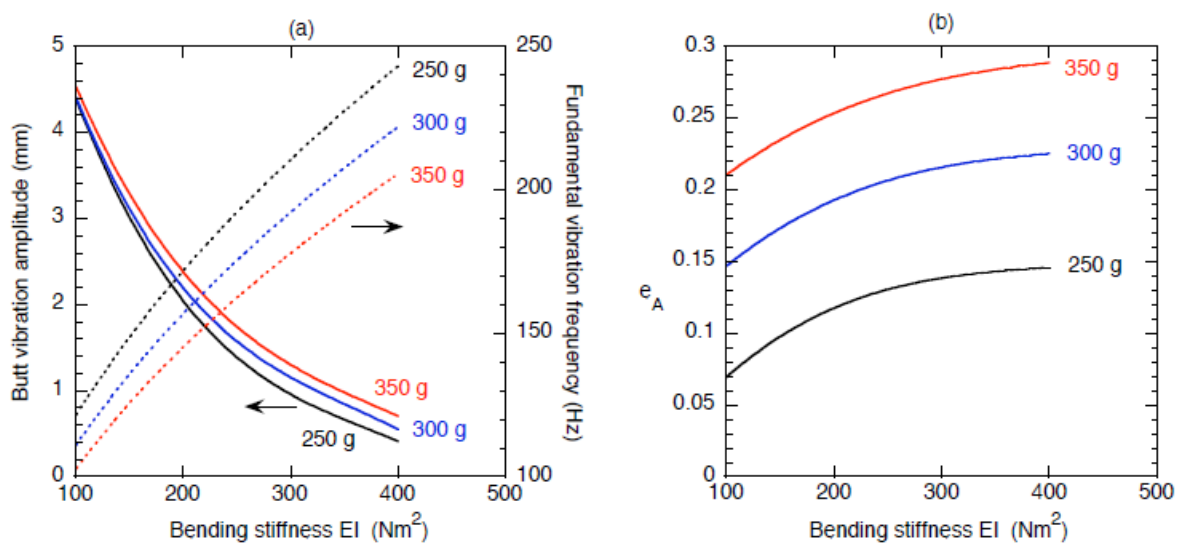


Figura 5 - Resultados calculados para feixes uniformes livremente suportados de massa variável e rigidez à flexão mostrando (a) a frequência de vibração fundamental do feixe (eixo direito), a amplitude de vibração da ponta final e (b) o coeficiente aparente de restituição,  $e_A$ , quando bola de tênis é incidente normalmente a  $10 \text{ m/s}$  para impactar o feixe em  $B = 60 \text{ mm}$  [18]

As raquetes de corpo largo tendem a ser relativamente leves, uma vez que são projetadas principalmente para jogadores casuais e não profissionais. A massa adicionada a tal raquete tenderá a diminuir a frequência de vibração e pode, portanto,

aumentar a amplitude de vibração. No entanto, o resultado final dependerá de onde a massa adicional está localizada. Em teoria, não deve haver nenhum efeito sobre a frequência de vibração ou a amplitude de vibração se a massa extra for adicionada a qualquer um ou a todos os três nós fundamentais de vibração no quadro (ver Figura 2). Por outro lado, o efeito máximo pode ser esperado se a massa for adicionada em um anti-nó, como a ponta ou extremidade do cabo ou o meio da raquete. As vantagens potenciais de adicionar ou concentrar a massa em um ponto nodal em uma raquete não foram previamente investigadas, mesmo por fabricantes de raquetes.

Em nossos estudos, foi medida a influência de pesos adicionados na raquete e foi feita a comparação com o estudo mostrado nos testes em [17].

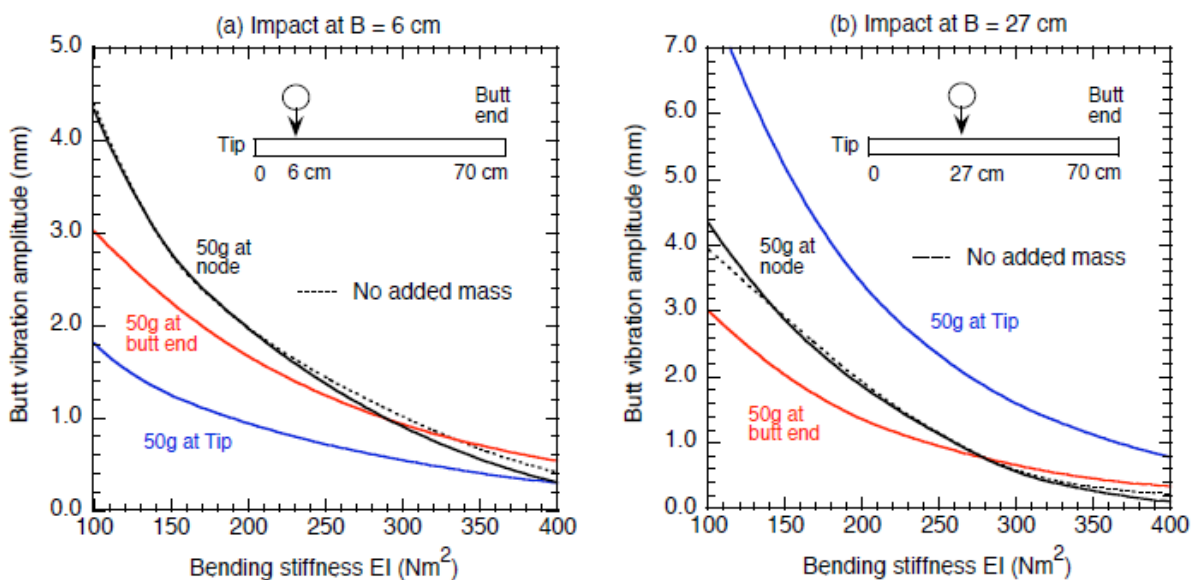


Figura 6 - Resultados calculados para uma viga uniforme de 250 g mostrando os efeitos da adição de uma massa de 50 g na ponta ou ponta final ou ponto nodal na extremidade da ponta. A bola é incidente normalmente a 10 m / s para impactar o feixe em (a)  $B = 6$  cm ou (b)  $B = 27$  cm [18]

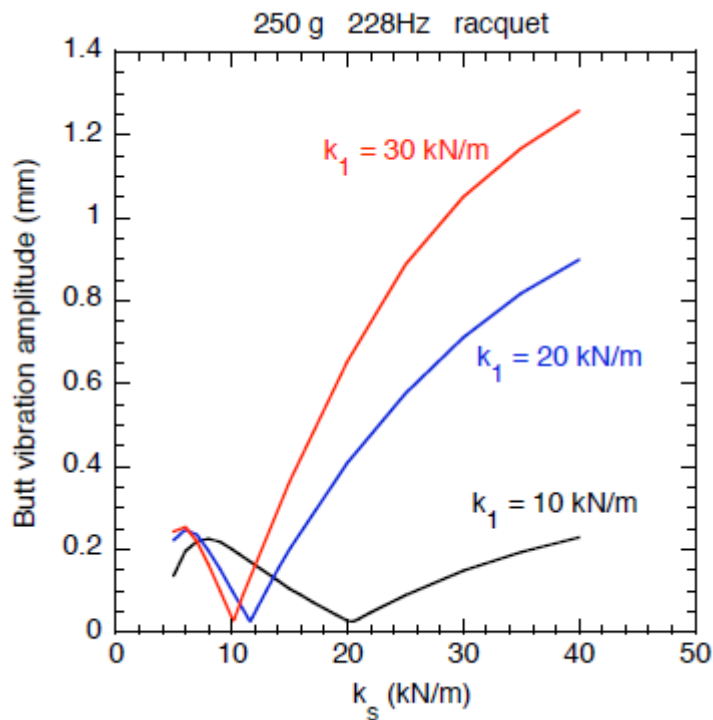


Figura 7 - Os resultados calculados para a viga de 250 g,  $EI = 350 \text{ Nm}^2$  na Fig. 3 mostram o efeito da variação da rigidez do plano da corda,  $k_s$  para três valores diferentes da rigidez da esfera,  $k_1$ . A bola é incidente normalmente a 10 m / s para impactar o feixe em  $B = 60 \text{ mm}$  [18]

## 2.4 TRANSFORMADA DE FOURIER

Para conseguirmos quantificar a frequência, amplitude e duração da vibração transmitida da raquete para o braço, precisamos do valor em Hertz. O sensor minisense100, que será descrito adiante, apresenta os valores captados em Volts, com isso, precisa-se da transformada de Fourier para transformar Volts no domínio do tempo para Volts no domínio da frequência.

A Transformada Rápida de Fourier (FFT) é um algoritmo computacional otimizado para implementar a Transformada Discreta de Fourier em uma matriz de amostras  $2^N$ . Ela permite determinar a frequência de um sinal discreto, representar o sinal no domínio da frequência. Este algoritmo tem uma complexidade de  $O(N \cdot \log_2(N))$ . Na verdade, a complexidade do algoritmo é um pouco maior porque os dados precisam ser preparados por uma operação chamada reversão de bits. Esta

seção de reversão de bit é apresentada no livro “Numerical Recipes In C” [9] como uma complexidade  $O(2N)$ . Com uma pequena alteração feita no código referenciado em [2], ele fica em  $O(N)$ . Isso representa algo como uma melhora de 8% no desempenho.

A FFT é calculada em duas partes. A primeira transforma a matriz de dados original em uma matriz de ordem reversa de bits aplicando o método de reversão de bits. Isso torna os cálculos matemáticos da segunda parte "muito mais fáceis". A segunda parte processa a FFT em operações  $N * \log_2(N)$  (aplicação do algoritmo de Danielson-Lanzcos. [15]).

Começando com uma matriz de dados complexos. Essa matriz poderia ser, por exemplo, nesse caso, uma matriz de floats em que o dado [even\_index] é a parte real e os dados [odd\_index] é a parte complexa. (Isso pode ser adaptado para uma matriz de dados reais, apenas preenchendo os valores complexos com 0s, ou usar a matriz real FFT implementada no livro “Numerical Recipes In C” [9].) O tamanho da matriz deve estar em uma ordem  $N^2$  (2, 4, 8, 16, 32, 64, etc ...). Caso a amostra não corresponda a esse tamanho, basta colocá-la em uma matriz com o próximo tamanho  $2^N$  e preencher os espaços restantes com 0s.

Apenas uma consideração pequena e não muito significativa: o código original usa matrizes de dados considerando que o início das informações está no índice 1 -> data [1], e data [0] é ignorado. O código usado modifica isso para começar de 0 -> data [0]. [2]

O código utilizado neste estudo está mostrado no Apêndice 3.

## 2.5 PÊNDULO FÍSICO

Pêndulos são de uso comum, e, no projeto em questão, foi necessário seu uso para que seja possível quantificar e obter as forças de soltura da bola de tênis mais próximas possíveis nos testes realizados.

Alguns pêndulos têm usos cruciais, como nos relógios; alguns são para diversão, como o balanço de uma criança; e alguns estão ali, como o chumbador em uma linha de pesca. Para pequenos deslocamentos, um pêndulo é um simples

oscilador harmônico. Um pêndulo simples é definido para ter um objeto que tenha uma massa pequena, também conhecida como pêndulo, suspensa por um fio ou corda leve. O deslocamento linear do equilíbrio é  $s$ , o comprimento do arco. Também são mostradas as forças no trenó, que resultam em uma força resultante de  $F = -mg \sin \theta$  em direção à posição de equilíbrio - isto é, uma força restauradora.

Na Figura 8, uma barra horizontal é apresentada de forma a fixar, em um ponto, o fio do pêndulo. Uma linha pontilhada perpendicular a barra, representando o equilíbrio do pêndulo, é desenhada para baixo. Uma corda de comprimento  $L$  é amarrada à barra no ponto de equilíbrio, e, na sua outra extremidade, é presa uma massa  $m$ , que neste estudo era uma bola de tênis, que está a uma distância  $s$  do ponto de equilíbrio. A corda está em uma abertura de um ângulo de teta com o equilíbrio na barra. Uma seta vermelha mostrando a força de tração da corda é mostrada ao longo da linha da corda em direção à barra. Uma seta em direção ao equilíbrio mostra sua força restauradora como seno teta. Uma seta perpendicular em direção ao chão representa a força Peso como  $W$ , sendo igual a  $mg$ , e esta seta está em um ângulo teta com direção descendente de corda [24]

O pêndulo foi utilizado em nossos testes como forma de controlar o ambiente, de forma que a força de arremesso da bola de tênis seja sempre a mesma, e, assim, conseguir um resultado cada vez mais exato e confiável. Em um pêndulo sempre com mesma massa e com o mesmo comprimento da corda, a única influência é a altura em que a bola é solta, com isso, os resultados são mais confiáveis e controlados.

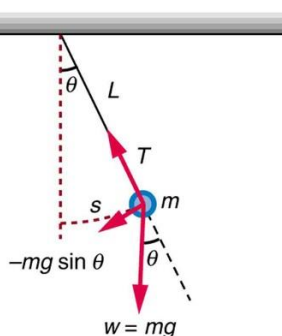


Figura 8 - Pêndulo Simples [24]



### 3. EQUIPAMENTOS E SOFTWARES UTILIZADOS

*Essa subseção apresenta os diversos componentes e dispositivos que são imprescindíveis para a execução do projeto. Cada componente será descrito e apresentado de modo que o leitor compreenda, com certo grau de profundidade, o que cada ferramenta realiza e sua importância para que os objetivos do experimento sejam atingidos com êxito. A compreensão desta subseção é de extrema importância para o entendimento do próximo capítulo, onde serão discutidos a função de cada componente ao longo do projeto.*

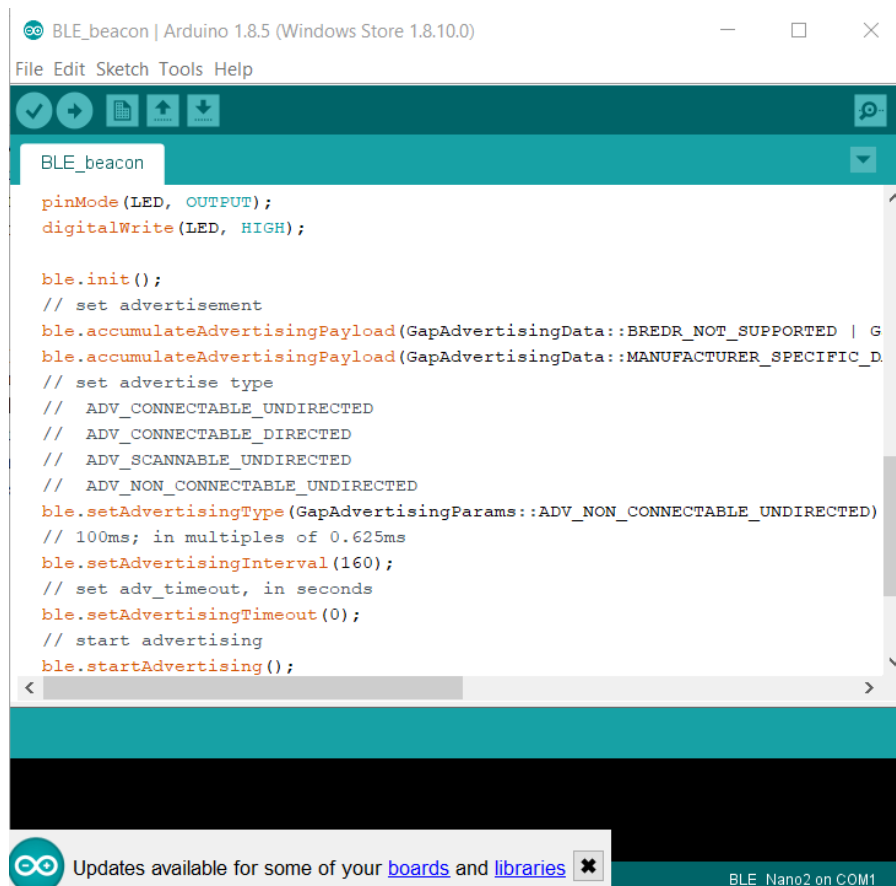
#### 3.1 ARDUÍNO IDE 1.8.5

O software utilizado para a realização dos códigos utilizados no Redbear foi a plataforma Arduino IDE (*Integrated Development Environment*). Esse ambiente de desenvolvimento apresenta diversas vantagens para o projeto, por ser um software de código aberto e com alta compatibilidade com diversos dispositivos e sistemas operacionais existentes. Devido a este fator, existe uma ampla biblioteca de códigos e exemplos na internet.

O Arduino IDE possui uma interface intuitiva e de fácil utilização, como mostra a Figura 9. Nela, é possível observar algumas das funções apresentadas pelo software, como verificação do programa, gravação, novo programa e monitoramento serial. Toda interação e upload dos códigos são feitos por meio da interface USB, ou seja, a IDE interage com o Redbear por meio desta porta, e, além disso, o ambiente de desenvolvimento permite a escolha da porta em que se deseja fazer o upload do código. É possível programar para diversas placas ao mesmo tempo.

A interface bluetooth, do Redbear, será programada por meio deste software e o *upload* do código ocorrerá via porta USB. O fabricante do Redbear fornece uma biblioteca de funções disponíveis para a placa, então, é imprescindível o download dessa biblioteca e a integração dessa biblioteca no ambiente de desenvolvimento. Dentre os recursos nela disponíveis, existe uma ampla lista de códigos exemplos que ficaram disponíveis na aba “tools” após a importação da mesma. A linguagem de programação utilizada foi C/C++. Os códigos resultantes do projeto foram

modificações realizadas a partir de códigos exemplos da seção BLE da biblioteca importada, facilitando o desenvolvimento dessa etapa do trabalho.



```
BLE_beacon | Arduino 1.8.5 (Windows Store 1.8.10.0)
File Edit Sketch Tools Help
BLE_beacon
pinMode(LED, OUTPUT);
digitalWrite(LED, HIGH);

ble.init();
// set advertisement
ble.accumulateAdvertisingPayload(GapAdvertisingData::BREDR_NOT_SUPPORTED | G
ble.accumulateAdvertisingPayload(GapAdvertisingData::MANUFACTURER_SPECIFIC_D
// set advertise type
// ADV_CONNECTABLE_UNDIRECTED
// ADV_CONNECTABLE_DIRECTED
// ADV_SCANNABLE_UNDIRECTED
// ADV_NON_CONNECTABLE_UNDIRECTED
ble.setAdvertisingType(GapAdvertisingParams::ADV_NON_CONNECTABLE_UNDIRECTED)
// 100ms; in multiples of 0.625ms
ble.setAdvertisingInterval(160);
// set adv_timeout, in seconds
ble.setAdvertisingTimeout(0);
// start advertising
ble.startAdvertising();
```

Figura 9 - Funcionamento do Arduino IDE

## 3.2 ACELERÔMETRO

### 3.2.1 Introdução

O acelerômetro é um sensor que mede a aceleração dinâmica de um dispositivo físico na forma de tensão. Esses sensores versáteis seriam utilizados no projeto em questão para realizar medições das acelerações nos eixos x, y, z a que a raquete fica submetida durante um jogo de tênis, entretanto, nas fases posteriores do projeto, um novo sensor foi determinado mais adequado para a realização dos estudos acerca do choque da bola na raquete: o minisens100, que será detalhado nas próximas seções.

### 3.2.2 Como funciona o acelerômetro

A maioria dos acelerômetros utiliza o efeito piezoelétrico, que geram uma corrente elétrica certos tipos de metais sofrem uma pressão mecânica, então é gerada uma carga elétrica que é proporcional à força aplicada, portanto ela será proporcional à aceleração sofrida. Após gerada esta carga elétrica, é necessário que o acelerômetro tenha um amplificador externo ou um amplificador de carga em linha. Também são imprescindíveis ajustes na impedância de saída de acordo com os dispositivos de medição e procurar minimizar a susceptibilidade a fontes de ruídos externas e *crosstalk*.

### 3.2.3 Acelerômetro utilizado no projeto

O acelerômetro escolhido para utilização deste projeto foi o ADXL337, da fabricante SparkFun [17] (ver Figura 10). Pode-se destacar os principais motivos para a escolha do mesmo como sendo: baixo custo, tamanho e a informação da aceleração nos três eixos x, y e z.

O ADXL337 é um sensor completo que possui a capacidade de medir a aceleração nos três principais eixos: x, y e z, possui pequenas dimensões, é fino e de baixa potência. As tensões analógicas de saída são proporcionais à aceleração, e, estão em uma faixa de  $\pm 3g$ . Este acelerômetro consegue medir a aceleração estática da gravidade e também a aceleração dinâmica resultante de um movimento, choque ou vibração. Para que o ADXL337 funcione de forma correta, é necessário que a alimentação esteja em 3.3V e o GND conectado no Terra. A partir disso, as linhas X, Y, Z devem ser conectadas aos pinos do Redbear, e então a troca de dados estará fisicamente preparada. As larguras de banda também podem ser definidas pelo usuário de acordo com a aplicação desejada, entre os intervalos de 0,5 Hz a 1600Hz para os eixos x e y, e de 0,5Hz a 550Hz para o eixo z.



Figura 10 - Pinos X, Y, Z já descritos.[17]

### 3.3 REDBEAR BLE NANO V2

Após buscas no mercado a procura de um microcontrolador para utilizar na aplicação em questão, o RedBear BLE nano v2 foi a placa de desenvolvimento que atendeu aos pré-requisitos definidos no escopo do projeto: módulo bluetooth, baixo custo, compatibilidade com outros dispositivos, fácil acesso para programação da placa e baixo consumo de energia.

O BLE nano v2 da RedBearlab é uma das menores placas de desenvolvimento Bluetooth 4.2 / 5.0 Low Energy (BLE) existente no mercado atual [14]. No núcleo do módulo BLE nano v2 existe o processador Nordic nRF52832, um SoC (System on chip) ARM Cortex-M4F e BLE capaz de funcionar com um clock de 64MHz com consumo de energia baixíssimo. Outra vantagem desta placa de desenvolvimento é a alta compatibilidade com diversos dispositivos de diferentes sistemas operacionais, realizando uma comunicação sem fio, como por exemplo com dispositivos que executam iOS 7/8, Android 4.3 ou superior e Windows Phone 8.1.

Como mostra a Figura 11, pode-se observar a pinagem do Redbear BLE nano v2, que possui as seguintes características:

- 11 pinos de entrada/saída Digital
- 1 UART (Universal Asynchronous Receiver-Transmitter) com controle de fluxo em hardware com 4 pinos de entrada/saída
- 2 pinos I<sup>2</sup>C (Inter-Integrated Circuit)
- 12 pinos PWM (Pulse Width Modulation)

- 8 pinos SPI (Serial Peripheral Interface)

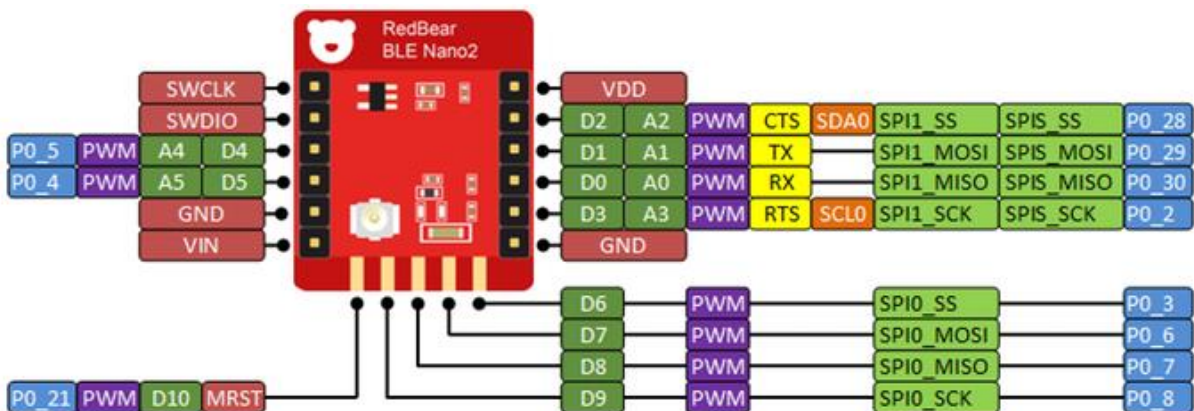


Figura 11 - Pinos do RedBear BLE Nano 2 (Referencia de um projeto cujo link esta no arquivo referencia) [17]

Para que o RedBear BLE Nano 2 realize as funções citadas anteriormente, a alimentação da placa de desenvolvimento deve ser efetuada de forma correta. Ele opera entre tensões de 1.8V a 3.6V, o que o torna capaz de trabalhar em conjunto com uma ampla variedade de componentes eletrônicos. Então, para gerar essa alimentação necessária para o bom funcionamento da placa, a ReadBearLab fornece em seu kit para desenvolvimento outra placa chamada DAPLink, que funciona como um dongle USB, ou seja, ela recebe 5V da porta USB e, por meio do LDO onboard, regula esta tensão para 3.3V, e enfim, pode operar como alimentação do BLE Nano 2. Além da função de regulador de tensão, a DAPLink também realiza a compilação dos códigos e envia para a placa BLE Nano 2 por interface USB. A Figura 12 mostra o kit completo de desenvolvimento. Segundo o User Guide do DapLink, este possui as seguintes características:

1. ARM 32bit Cortex-M3
2. FLASH DE 128KB, RAM DE 20KB
3. Suporte VIN 3v3 ~ 5v input
4. Saída 3v3 do VDD
5. MSC - memória flash
6. CDC - porta virtual com para emulação de log, rastreo e terminal
7. Canal de depuração compatível com HID - CMSIS-DAP

Aparelhos com suporte:

1. BLE Nano (nRF51822)
2. BLE Nano 2 (nRF52832)

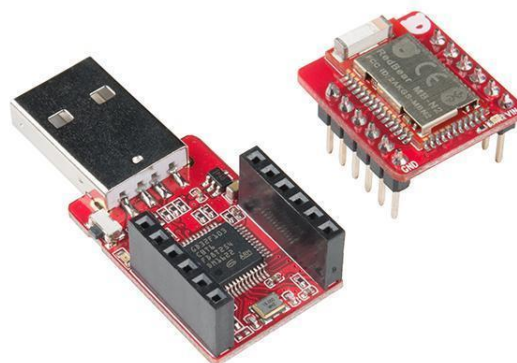


Figura 12 - Placas BLE Nano 2 e DAPLink [17]

### 3.4 BATERIA-MOEDA E SUPORTE

Baterias-moedas, também conhecidas popularmente como baterias-botão, é uma célula que fornece uma pequena quantidade de energia enquanto o sistema não está sob uma alimentação. Elas possuem dimensões pequenas, o que facilita o uso em pequenos sistemas, como relógios de pulso, calculadoras, relógios eletrônicos, brinquedos e outros. O que as diferencia de uma bateria usual é o fato de possuírem apenas uma célula, pois, em uma bateria normal, existem pelo menos duas ou mais células conectadas entre si.

A bateria escolhida para o projeto em questão foi a CR2032, ver Figura 13, que é uma célula de botão de lítio que gera 3V, corrente de 250mAh, e possui 20 mm de dimensão, que é ideal para alimentar o ReadBear BLE Nano 2. Entretanto, para que a bateria fosse utilizada neste projeto, também foi necessário o uso de um suporte para colocar as baterias em questão.

O suporte de bateria-moeda escolhido foi o da LilyPad (ver Figura 14), pois possui um pequeno interruptor em linha com o circuito, o que é uma vantagem pois torna possível ligar ou desligar o circuito quando conveniente, economizando a energia das baterias. Essa placa possui 20 mm de dimensão, assim como uma CR2032, e oferece quatro pontos de conexão, sendo dois negativos e dois positivos. O suporte foi projetado para que a bateria não saia acidentalmente ou ligue/desligue o sistema acidentalmente. Seu circuito interno é apresentado na Figura 15. [27]



Figura 13 - Bateria-moeda CR2032 [26]

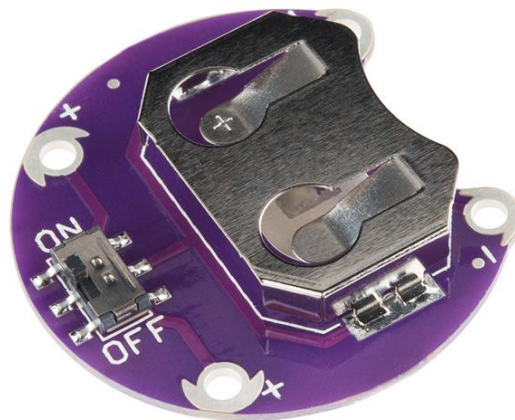


Figura 14 - Suporte Lilypad [27]

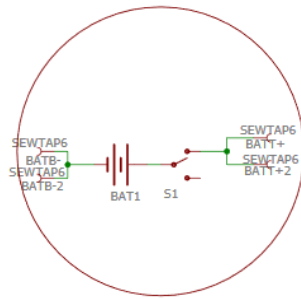


Figura 15 - Circuito do suporte LilyPad [27]

### 3.5 ESP32

O ESP32 é um único chip combo Wi-Fi e Bluetooth de 2,4 GHz projetado com a tecnologia de 40 nm de baixa potência da TSMC (Taiwan Semiconductor Manufacturing Company, Limited) (ver Figura 17). Ele é projetado para obter a melhor potência e desempenho de RF, mostrando robustez, versatilidade e confiabilidade em uma ampla variedade de aplicações e cenários de energia.

O ESP32 é projetado para aplicativos móveis, eletrônicos vestíveis e Internet das Coisas. Ele apresenta todas as características de ponta dos chips de baixa potência, incluindo relógio com granulação fina, múltiplos modos de potência e escalonamento de potência dinâmico. Por exemplo, em um cenário de aplicação de Hub de sensor de baixa potência, o ESP32 é despertado periodicamente e somente quando uma condição especificada é detectada. Ciclo de baixa demanda é usado para minimizar a quantidade de energia que o chip gasta. A saída do amplificador de potência também é ajustável, contribuindo para um balanceamento ideal entre a faixa de comunicação, taxa de dados e consumo de energia.

O ESP32 é uma solução altamente integrada para aplicativos IoT, Wi-Fi e Bluetooth, com cerca de 20 componentes externos. O ESP32 integra um comutador de antena, balun de RF, amplificador de potência, amplificador de recepção de baixo ruído, filtros e módulos de gerenciamento de energia. Como tal, toda a solução ocupa uma área mínima de placa de circuito impresso (PCB). O ESP32 utiliza o CMOS para rádio e banda-base totalmente integrados em um único chip, além de integrar circuitos de calibração avançados que permitem que a solução remova as imperfeições do



circuito externo ou ajuste as alterações nas condições externas. Como tal, a produção em massa de soluções ESP32 não requer equipamento de teste Wi-Fi caro e especializado.

### 3.5.1 Conversor de Analógico para Digital

O ESP32 integra ADCs SAR de 12 bits e compartilhado em 18 canais (pinos habilitados para analógico) (ver Figura 16). O ULPcoprocessor no ESP32 também é projetado para medir tensão, enquanto opera no modo de suspensão, o que permite baixo consumo de energia. A CPU pode ser acordada por uma configuração de limite e/ou por outros gatilhos. Com configurações apropriadas, os ADCs podem ser configurados para medir a tensão em até 18 pinos. As características do conversor são apresentadas na Figura 18.

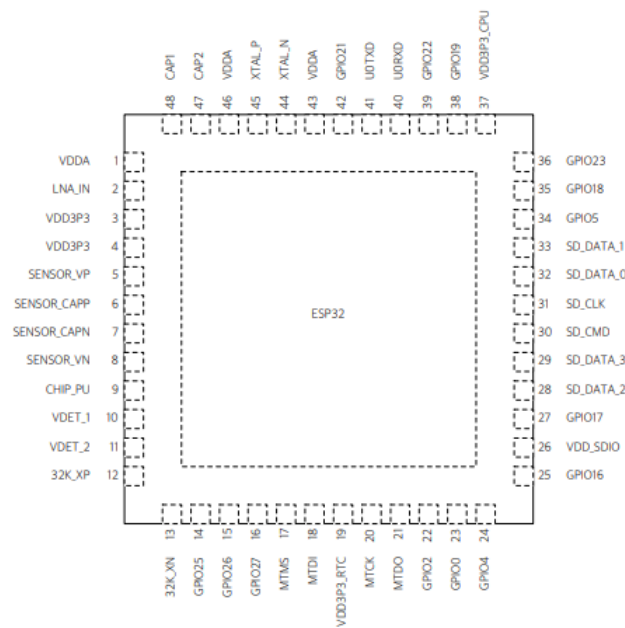


Figura 16 - Pinagem do ESP32 [28]

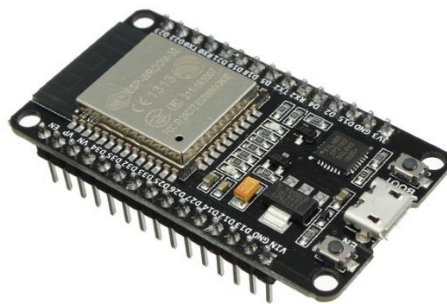


Figura 17 - Chip ESP32 [28]

Parameter	Description	Min	Max	Unit
DNL (Differential nonlinearity)	RTC controller; ADC connected to an external 100 nF capacitor;	-7	7	LSB
INL (Integral nonlinearity)	DC signal input; ambient temperature at 25 °C; Wi-Fi&BT off	-12	12	LSB
Sampling rate	RTC controller	-	200	kpsps
	DIG controller	-	2	Mpsps

Figura 18 - Características do Conversor ADC [28]

### 3.5.2 Características do ESP32

Para o projeto descrito, foi utilizado o microcontrolador ESP32 devido às suas características versáteis e sua alta compatibilidade com os componentes utilizados no trabalho. As principais características do microcontrolador estão apresentadas a seguir:

- Processador principal: LX6 32-bit Dual-core, operando 2-240 MHz
- Processador secundário: ULP (Ultra Low Power coprocessor) 8MHz e consome 150uA
- FLASH: 4MB, suportando até 16MB de memória externa
- RAM: 520kB
- GPIO: 34, com 3.3V e 12mA
- ADC: 18, com resolução de 12-bit
- DAC: 2, com resolução 8-bit
- WiFi: 2,4 GHz, 802.11 b/g/n
- Bluetooth: Bluetooth Low Energy v4.2(BLE)

- Acelerador via hardware para encriptações, hash e afins.
- 4 Timers de 64-bit
- 4 Watchdogs
- 10 Sensores de Touch Capacitivo
- 1 Sensor de temperatura interno
- 1 Sensor de efeito Hall

Tais características demonstram a qualidade e a superioridade do ESP32 em relação ao RedBear, e, devido à incompatibilidade deste com o software Blynk para iOS, optou-se por seguir o trabalho utilizando o ESP32, que possui uma versatilidade maior.

Observa-se a presença do processador dual-core, ou seja, possui dois núcleos de processamento, e, também possui aproximadamente 500kBytes de memória SRAM, permitindo a execução de programas mais complexos. O coprocessador, o processador secundário, ULP, delega tarefas simples para execução enquanto os processadores principais estão desligados. Portanto, ele funciona como um modo “sleep”. Outro fator importante para o desenvolvimento do projeto é a presença do módulo bluetooth 4.2, que é o BLE, que permite a conectividade com uma cobertura ampla de dispositivos móveis, utilizando um modo de transmissão de dados sem fio de baixo consumo. A Figura 16 descreve os pinos e a Figura 20, o diagrama de blocos de componentes do ESP32. [28]

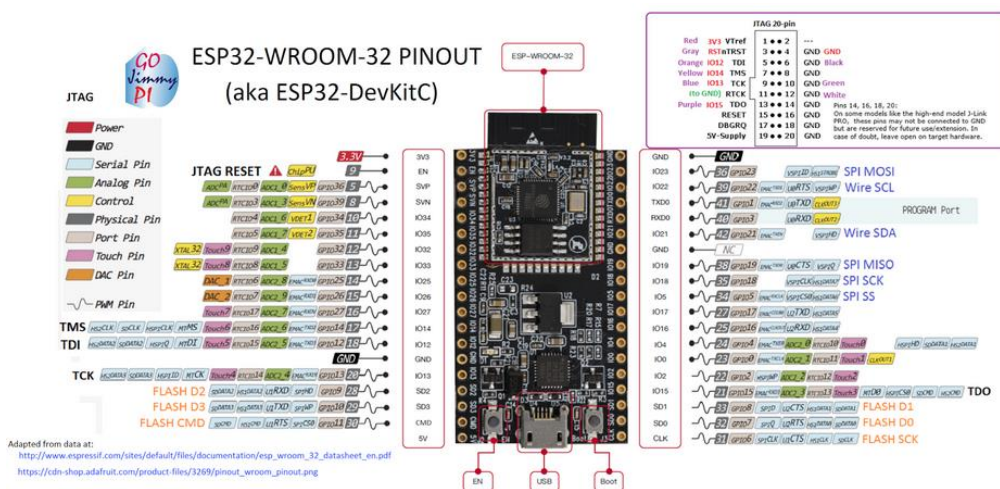


Figura 19 - Pinos do ESP32 [28]

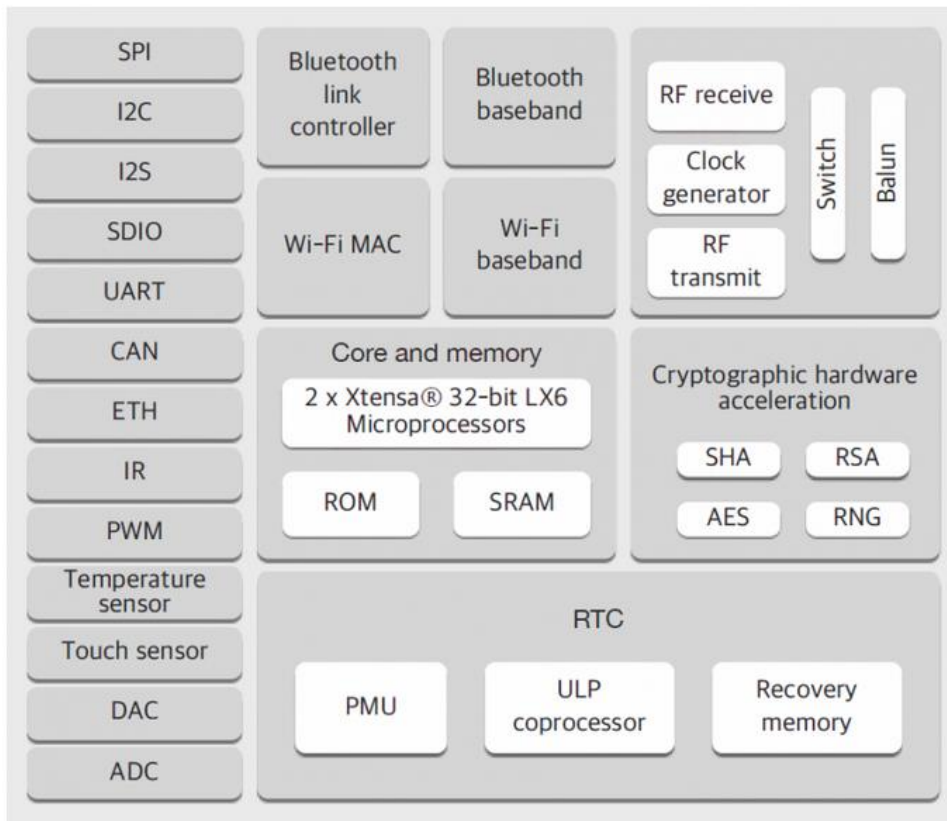


Figura 20 - Diagrama de blocos de componentes do ESP32 [28]

### 3.6 BLYNK

Blynk é um recurso disponível para aplicativos iOS e Android que pode ser integrado à sistemas embarcados como Arduino, RedBear, ESP32 Raspberry Pi e outros. É um painel digital com diversas funções que são inseridas por meio de widgets, entre essas funções, pode-se destacar a integração de mapeamento GPS, LED's, timers, conectividade bluetooth, notificações via twitter, notificação celular e e-mail [20]

Esta aplicação é bastante versátil (ver Figura 21), permitindo a integração e o desenvolvimentos em diferentes sistemas embarcados e o uso de diversas tecnologias de comunicação, como WIFI, bluetooth, ethernet e outros. Com o uso de tecnologias com acesso à inter net, é possível acessar e utilizar as funções fornecidas pelos desenvolvedores da plataforma, por meio do Blynk Server. É uma ferramenta muito útil para desenvolvimento de projetos IoT (Internet das Coisas), como recebimento de dados de sensores, armazenamento de dados, e visualização de

dados para monitoramento. No projeto em questão, utilizou-se a integração do Blynk ao ESP32 por meio da tecnologia bluetooth 4.2, por isso, algumas das funções disponíveis via internet não estavam disponíveis para o projeto, a interface do aplicativo pode ser vista na Figura 22. Pode-se explicitar três componentes na plataforma, são eles:

Blynk App - permite criar interfaces para os projetos usando vários widgets (ver Figura 22).

Blynk Server - responsável por todas as comunicações entre o smartphone e o hardware. É possível usar a Blynk Cloud ou executar um servidor Blynk privado localmente.

Bibliotecas Blynk - para todas as plataformas de hardware populares - permitem a comunicação com o servidor e processam todos os comandos de entrada e saída.

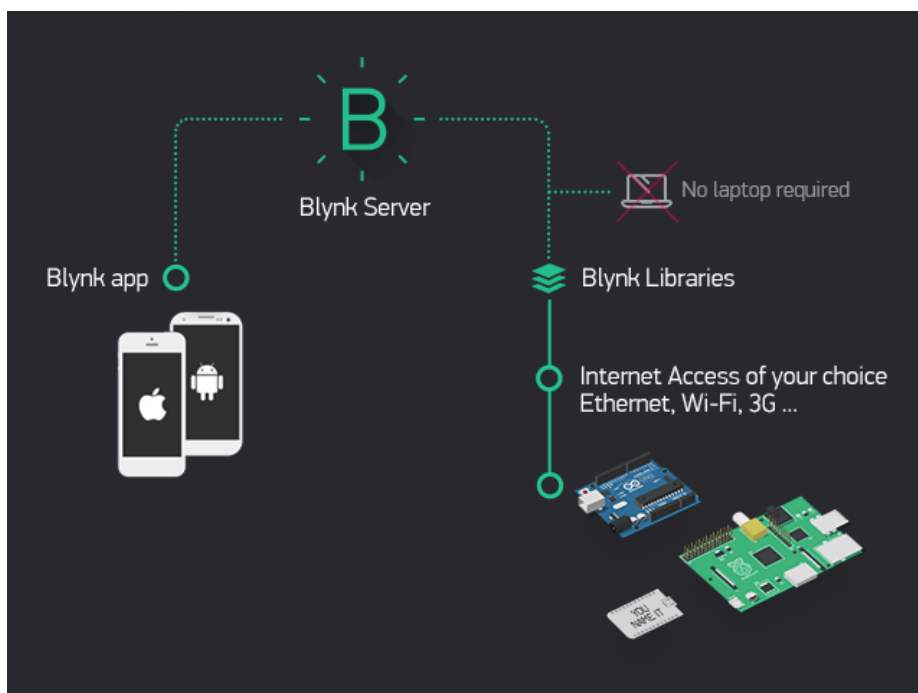


Figura 21 - Funcionamento do Blynk [20]

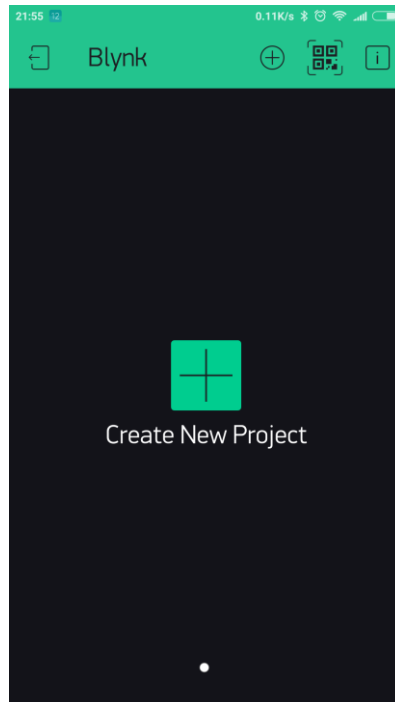


Figura 22 - Interface do Aplicativo

### 3.7 SENSORES MINISENSE 100

Essa seção do trabalho apresenta um estudo detalhado acerca dos sensores utilizados nas raquetes de tênis. A priori seria utilizado um acelerômetro que captaria as acelerações nos eixos x,y e z, como mostrado na seção 4.2, e, entretanto, ao longo do projeto, foram encontrados sensores mais adequados para a medição de vibração durante o choque da bola de tênis com as cordas da raquete, o que diminui o erro de precisão, e logo substituíram o acelerômetro. Esta seção será exclusivamente destinada a detalhar as características e o uso desses sensores.

#### 3.7.1 Introdução

O sensor utilizado, o sensor Minisense 100, é um sensor piezelétrico. Esse tipo de sensor é denominado dessa forma, pois, são dispositivos que medem a pressão ou a tensão utilizando a piezeletricidade, que é um fenômeno onde certos materiais podem gerar uma corrente elétrica quando deformado, ou seja, a piezoeletricidade é a capacidade que alguns cristais possuem de gerarem corrente elétrica em resposta a uma pressão mecânica exercida sobre o mesmo [21]. Esse tipo de material, que converte uma forma de energia em outra, são muito utilizados em aplicações sensoriais.

O sensor Minisense 100, da Measurement Specialties, é um sensor de vibração de baixo custo carregado por uma massa. Ele possui uma alta sensibilidade para detectar vibrações em baixas frequências, como por exemplo a vibração da raquete ao receber o choque da bola de tênis. Pequenas correntes e uma alta voltagem são geradas quando o sensor se move para frente e para trás ou para baixo e para cima. É necessário o uso de um resistor simples para baixar a tensão para níveis ADC. Além dessas características, esse sensor foi escolhido devido à alta capacidade de absorver choques e sua flexibilidade, o que o torna ideal para a detecção de vibrações com uma precisão alta. Seus pinos são facilmente soldados e seu tamanho é ideal para a funcionalidade desejada, que é analisar o impacto da bola de tênis na raquete. A Figura 23 apresenta suas dimensões.

Para o projeto da raquete foi utilizado um sensor igual a este descrito no parágrafo anterior.

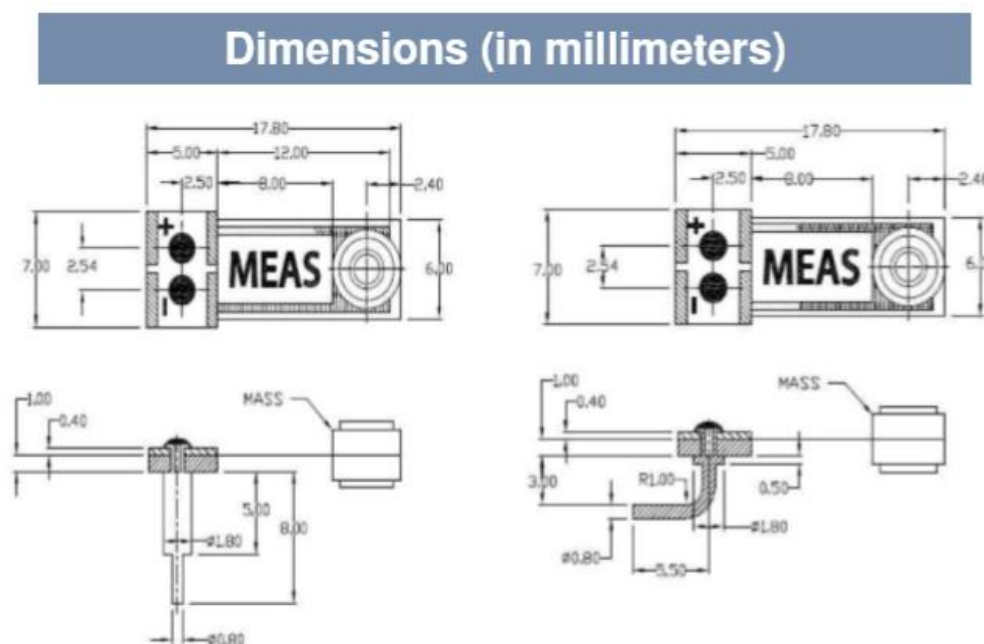


Figura 23 - dimensões, em mm, dos sensores utilizados [21]

### 3.7.2 Funcionamento

O MiniSense 100 atua como um acelerômetro de feixe cantilever. Quando montado na horizontal, aparece uma inclinação no eixo vertical devido à inércia da massa na ponta do feixe. Essa tensão causada cria uma resposta piezoelétrica, que é detectada como uma carga ou saída de tensão nos eletrodos do sensor. Portanto, este sensor pode ser usado na detecção de vibrações, impactos contínuos ou

impulsivos. Para frequências de excitação abaixo da frequência de ressonância, que é 75Hz [22], o sensor produz uma saída linear controlada pela sensibilidade da linha no gráfico da Figura 24. A sensibilidade na frequência ressonante é significativamente maior. Impactos nos componentes de alta frequência irão excitar a frequência de ressonância, como é possível observar nos gráficos da Figura 24.

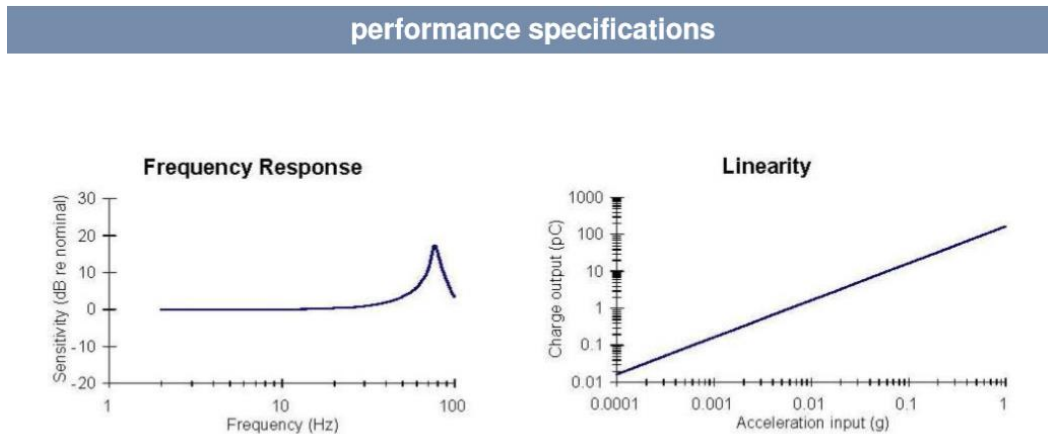


Figura 24 - gráficos da linearidade e da resposta na frequência [21]

Por uma perspectiva elétrica, o sensor MiniSense 100 comporta-se como um capacitor “ativo”, ou seja, ele pode ser descrito como uma fonte de tensão, onde a tensão seria proporcional à aceleração aplicada nele, em série com a capacitância do dispositivo. Qualquer entrada externa ou carga conectada ao circuito formará um filtro passa-alta, com uma frequência de supressão calculada por meio da fórmula  $f = 1 / (2\pi RC)$ . A impedância do sensor é de aproximadamente 650 Mohms a 1Hz (ver Figura 25).

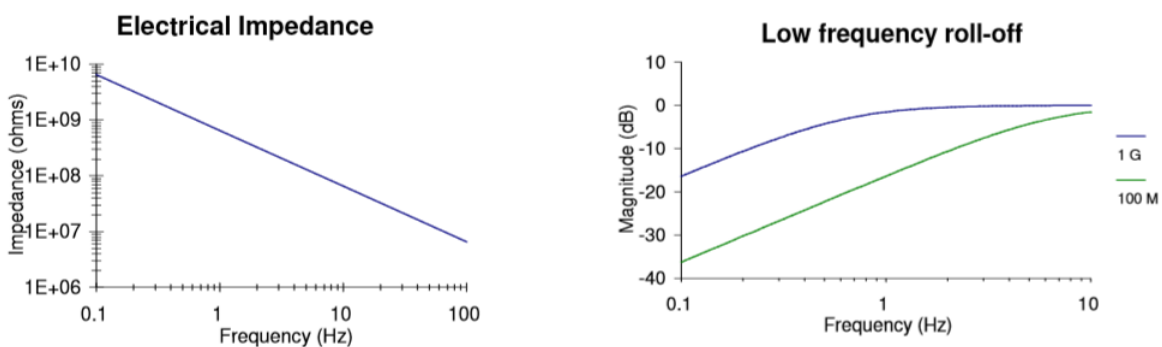


Figura 25 - Características elétricas do sensor [21]



### 3.7.3 Sensibilidade fora do eixo

A sensibilidade do Minisense 100 segue uma lei do cosseno, quando girada horizontalmente em torno de seu eixo, ou verticalmente ao redor de seu ponto médio. A uma rotação de 90 graus em ambos os planos, tanto a sensibilidade da linha de base quanto a sensibilidade à ressonância estão no mínimo. Em teoria, a sensibilidade deve ser zero nessa condição. É provável que alguma sensibilidade em torno da frequência de ressonância ainda seja observada - mas isso pode ser imprevisível e provavelmente será de pelo menos -16 dB com referência à resposta no eixo. Observe que a sensibilidade a 30 graus de rotação é de -1,25 dB (87% da resposta no eixo), a 60 graus, cai para -6 dB (50%).

Os gráficos da Figura 26 mostram a mudança na sensibilidade observada para:

- 1) Rotação sobre o eixo principal do elemento sensor, ou
- 2) Rotação sobre o ponto médio do elemento sensor.

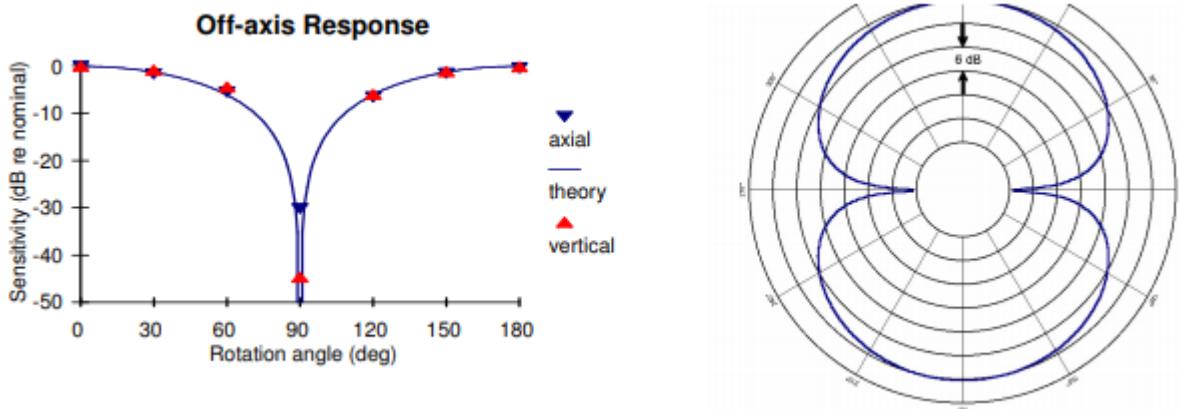


Figura 26 - Resposta do sensor nos eixos [21]

### 3.7.4 Aplicações

Por meio das especificações do fabricante, Measurement Specialties, foi possível determinar as funcionalidades do sensor Minisense 100 que foi utilizado no projeto. Dentre as aplicações apresentadas no manual técnico dos sensores, foram separadas

duas funções que são interessantes para o estudo de sensores nas raquetes de tênis e mais algumas funções que são amplamente utilizadas:

1. Pontuação esportiva: sensores piezoelétricos podem ser usados para medir o tempo de impacto, precisão e a força. Tais parâmetros são desejáveis em diversas aplicações de pontuação esportiva. O grande desafio nessa aplicação é a robustez do alvo sem a introdução de um retorno repentino no design. No projeto descrito, foi analisada a tensão do choque entre a bola de tênis e as cordas da raquete.
2. Acelerômetro: Os acelerômetros piezoelétricos são amplamente utilizados para medições de aceleração, choque e vibração de propósito geral. Este tipo de dispositivo utiliza o efeito piezoelétrico, sendo formado, basicamente, por uma massa (denominada massa sísmica) que é posta em contato direto com o componente piezoelétrico. Quando um movimento acelerado é aplicado ao acelerômetro, o cristal piezoelétrico experimenta uma força, fazendo com que nele se desenvolva uma carga elétrica proporcional à aceleração. Este sinal elétrico é, a seguir, correlacionado à aceleração.
3. Microfones: Uma aplicação comum de sensores piezoelétricos é em microfones, quando a força exercida por ondas de som é convertida em sinais elétricos e gravado ou amplificado.
4. Automóveis: Para os carros os sensores podem ser utilizados de várias maneiras como sensor de movimento do veículo e dispositivos antirroubo.
5. Monitoramento de Sinais Vitais: Os batimentos cardíacos também são monitorados por sensores piezoelétricos.

### 3.8 MÓDULO BUZZER 5V PASSIVO

O módulo buzzer 5V passivo é um componente importante para projetos que necessitam de funções sonoras. No projeto em questão, sua função é auxiliar na detecção de falsos positivos, onde, o buzzer está integrado com o microcontrolador e com o sensor minisense 100 descrito na seção 3.8. O módulo possui 3 pinos, são eles: VCC, GND e I/O, onde sua tensão de operação é de 5V. Suas dimensões são de 30 x 13 x 10 mm. Ele pode ser conectado à portas digitais do microcontrolador e

também em pinos PWM, que irá produzir variações sonoras de acordo com o código implementado [23]. O módulo está representado na Figura 27 e o circuito interno dos pinos na Figura 28.

Características do Buzzer:

- Corrente:  $\leq 42\text{mA}$ ;
- Som de saída:  $\geq 85\text{DB}$ ;
- Frequência de Ressonância:  $2300 \pm 300\text{HZ}$ ;
- Temperatura de Operação:  $-20^{\circ}\text{C} \sim +45^{\circ}\text{C}$ ;
- Temperatura de armazenamento:  $-20^{\circ}\text{C} \sim +60^{\circ}\text{C}$ ;
- Sinalizador piezoelétrico de 12mm.

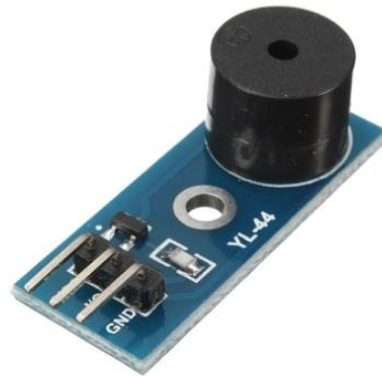


Figura 27 – Buzzer [23]

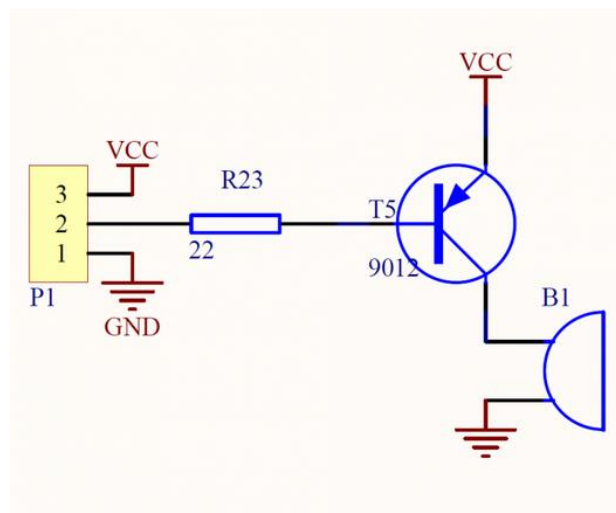


Figura 28 - Circuito do módulo Buzzer [23]

## 4. EXPERIMENTOS REALIZADOS

*Essa seção do trabalho apresenta as etapas que foram seguidas para a realização do projeto em questão. Os conceitos apresentados no capítulo 2 são imprescindíveis para a compreensão de cada parte deste projeto. Essa seção tem como objetivo descrever as etapas realizadas, os objetivos esperados, códigos utilizados, testes realizados, resultados obtidos e análise dos mesmos. Nesta seção será detalhado cada parte do processo do estudo até atingir a conclusão do estudo comparativo.*

Nos primeiros experimentos, foi utilizado o acelerômetro ADXL337. Durante os testes, foi encontrado o Minisense 100, em que fazia mais sentido para o estudo, devido a que necessita-se também uma análise da frequência, amplitude e duração da vibração. E durante todo o estudo, foi utilizado o Minisense 100, que apresentou resultados consistentes.

### 4.1 METODOLOGIA

No capítulo 1 deste documento, a Introdução, foi especificada a motivação do projeto a ser realizado e os objetivos deste estudo. Tendo em vista os objetivos estabelecidos no cenário descrito, iniciou-se uma série de etapas, definidas metodicamente, para a realização do projeto. Para isso, foram decididos os seguintes estágios de desenvolvimento do trabalho:

1. Compreensão da funcionalidade de cada componente, pois é de suma importância para o entendimento da utilização de cada recurso de forma eficiente no estudo desenvolvido.
2. Verificação da condição de uso de cada dispositivo.
3. Estabelecer a comunicação entre os dispositivos ESP32 e um aparelho móvel iOS Iphone 8 plus e, conseqüentemente, a troca de dados entre eles.

4. Conseguir receber e interpretar os dados do sensor utilizado, para o nosso estudo, é o sensor minisense 100.
5. Transmitir os dados recebidos do sensor.
6. Interpretação dos resultados.
7. Anotação dos resultados e conclusões.

## 4.2 TESTES REALIZADOS

### 4.2.1 Teste de comunicação RedBear BLE nano e IOS

Este teste consiste em estabelecer uma comunicação via bluetooth entre o RedBear BLE nano e um dispositivo móvel, que no caso é um ios. Para isso, foi utilizado a interface de programação *Arduino IDE 1.8.5*. Ao ter acesso à biblioteca do nRF5x BLE, o código base apropriado para este teste foi o “SimpleChat” da própria biblioteca, como mostra a Figura 29. Na interface do dispositivo móvel, o aplicativo utilizado para observar a troca de dados foi o *RedBear BLE Controller*.

Após as devidas modificações no código mencionado acima, o mesmo foi compilado e instalado no RedBear BLE nano, e então iniciou-se o processo de monitoramento de troca de dados entre os dispositivos. O Arduino IDE possui uma ferramenta chamada “Serial Monitor”, que o usuário seleciona a porta e este recurso permite o monitoramento, o envio e o recebimento de dados.

#### **Resultados:**

Na interface do dispositivo móvel, por meio do aplicativo do RedBear, foram escaneados os dispositivos ao seu alcance pela interface bluetooth e então escolhido o RedBear BLE nano. A comunicação entre ambos os dispositivos foi efetuada e logo em seguida a troca de dados, como é possível observar pelas *Figuras 29,30 e 31*. Pôde-se concluir que os objetivos do teste foram realizados com sucesso.

```
SimpleChat2
static uint8_t rx_state=0;

// The uuid of service and characteristics
static const uint8_t service1_uuid[] = {0x71, 0x3D, 0, 0, 0x50, 0x3E, 0x4C, 0x75, 0xBA, 0x94, 0x31, 0x48};
static const uint8_t service1_tx_uuid[] = {0x71, 0x3D, 0, 3, 0x50, 0x3E, 0x4C, 0x75, 0xBA, 0x94, 0x31, 0x48};
static const uint8_t service1_rx_uuid[] = {0x71, 0x3D, 0, 2, 0x50, 0x3E, 0x4C, 0x75, 0xBA, 0x94, 0x31, 0x48};
static const uint8_t uart_base_uuid_rev[] = {0x1E, 0x94, 0x8D, 0xF1, 0x48, 0x31, 0x94, 0xBA, 0x75, 0x4C, 0x3E};

uint8_t tx_value[TXRX_BUF_LEN] = {0,};
uint8_t rx_value[TXRX_BUF_LEN] = {0,};

// Initialize value of chars
GattCharacteristic characteristic1(service1_tx_uuid, tx_value, 1, TXRX_BUF_LEN, GattCharacteristic::BLE_GATT_CHARACTERISTIC_READ);
GattCharacteristic characteristic2(service1_rx_uuid, rx_value, 1, TXRX_BUF_LEN, GattCharacteristic::BLE_GATT_CHARACTERISTIC_WRITE);
GattCharacteristic *uartChars[] = {&characteristic1, &characteristic2};
GattService uartService(service1_uuid, uartChars, sizeof(uartChars) / sizeof(GattCharacteristic *));

void disconnectionCallback(const Gap::DisconnectionCallbackParams_t *params) {
    Serial.println("Disconnected!");
    Serial.println("Restarting the advertising process");
    ble.startAdvertising();
}

void gattServerWriteCallBack(const GattWriteCallbackParams *Handler) {
    uint8_t buf[TXRX_BUF_LEN];
    uint8_t index;
    uint16_t bytesRead = TXRX_BUF_LEN;

    Serial.println("onDataWritten : ");
    if (Handler->handle == characteristic1.getValueAttribute().getHandle()) {
        ble.readCharacteristicValue(characteristic1.getValueAttribute().getHandle(), buf, &bytesRead);
        Serial.print("bytesRead: ");
        Serial.println(bytesRead, HEX);
        for(index=0; index<bytesRead; index++) {

```

Done uploading.

```
** Verified OK **
** Resetting Target **
shutdown command invoked
```

15 BLE\_Nano2 on COM1

Figura 29 - Trecho do código “SimpleChat”

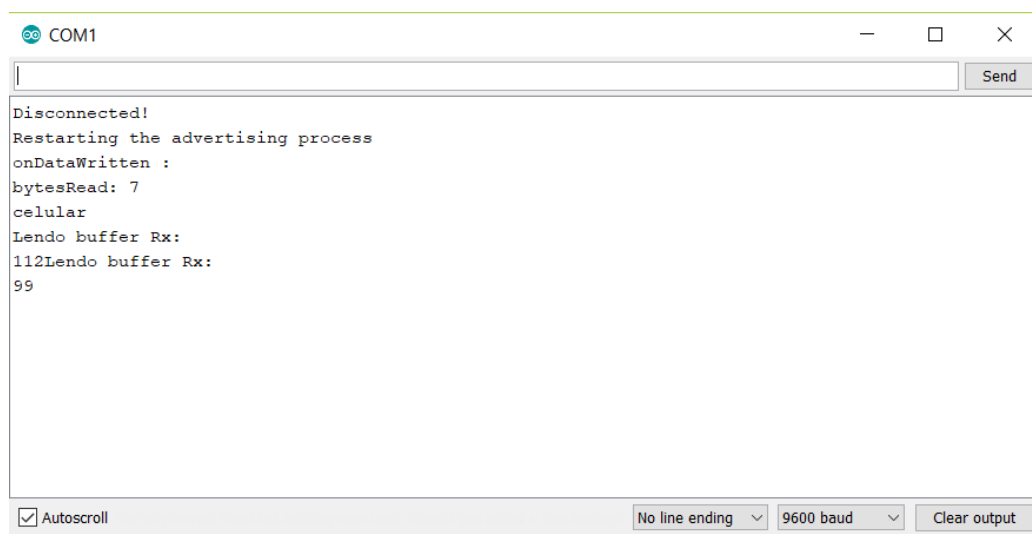


Figura 30 - Ferramenta “Serial Monitor” mostrando os dados enviados e recebidos

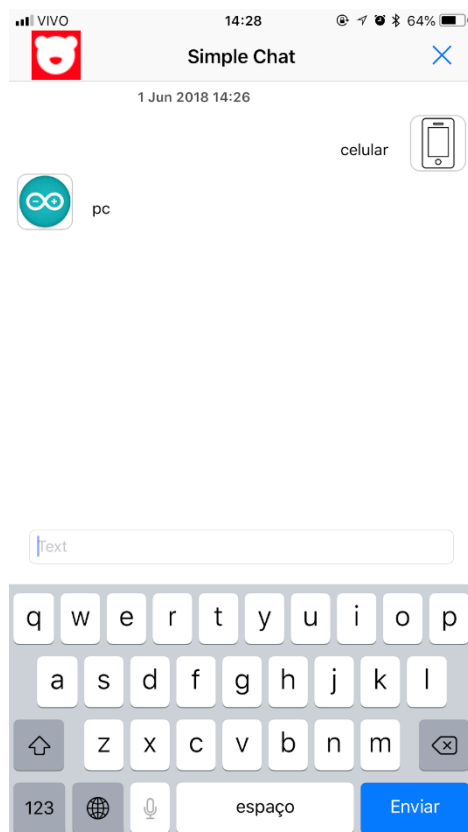


Figura 31 - Interface do dispositivo móvel

#### 4.2.2 Teste do Funcionamento do sensor de vibração com o envio para o celular

Esta parte do experimento foi dividida em duas partes, onde consistia em atingir os seguintes objetivos:

1. Receber a informação de variação de tensão nos sensores
2. Enviar dados a cada 1s do BLE para o celular

Após atingidos estes dois objetivos, o passo seguinte será integrar um passo com o outro, ou seja, o RedBear BLE nano deve receber os dados obtidos no sensor, e, enviar, periodicamente estes dados, para o celular.

Então, para atingir o objetivo 1, foi utilizado o seguinte código:



```
const int PIEZO_PIN = A3; // Piezo output

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  // Read Piezo ADC value in, and convert it to a voltage
  int piezoADC = analogRead(PIEZO_PIN);
  float piezoV = piezoADC / 1023.0 * 3.3;
  delay(250);
  Serial.println(piezoV); // Print the voltage.
}
```

Figura 32 - Declaração das variáveis

Onde *PIEZO\_PIN = A3*, está configurando o pino A3 como entrada para receber a saída do redBear, de acordo com já apresentadas anteriormente. O *Serial.begin(9600)* está configurando o valor de BaudRate dos dados na porta serial. A BaudRate é a taxa na qual a informação é transferida em um canal de comunicação. No contexto da porta serial, "9600 baud" significa que a porta serial é capaz de transferir no máximo 9600 bits por segundo. E, então, é criada uma variável para recebimento dos dados obtidos na porta A3, do tipo float, e dividido por  $1023 * 3.3$ , pois o RedBear trabalha com 10 bits, ou seja, 1024 valores, e tensão máxima trabalhada por ele, de acordo com o manual técnico é 3.3V. Logo para ajustar a variação de tensão é feita a conversão por meio deste cálculo, aparecendo as variações já na unidade Volts. O delay foi utilizado para que apresente os resultados a cada 1s. Terminando a parte lógica, seguimos para a parte física descrita a seguir.



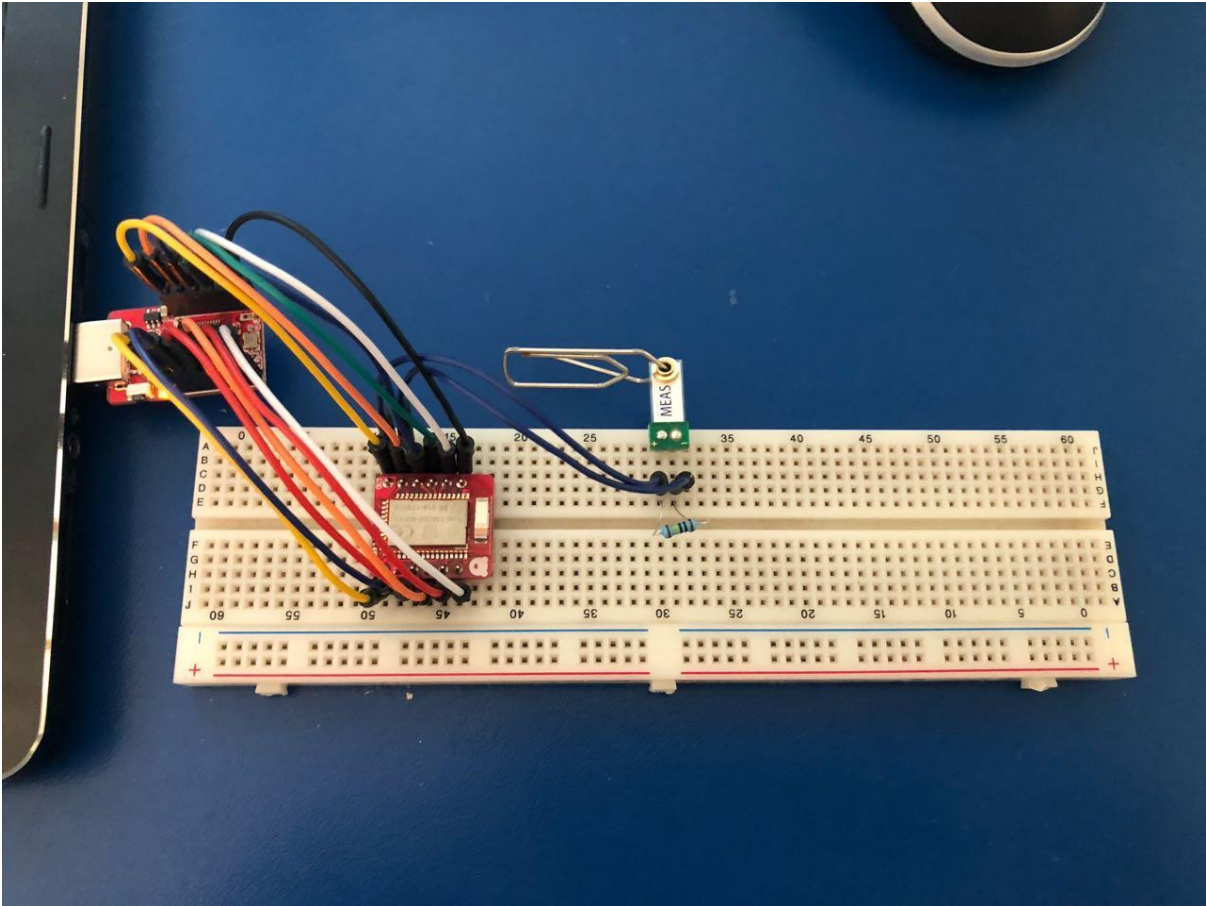


Figura 33 - Montagem do Circuito utilizando o RedBear e o Sensor MiniSense 100

Não foi encontrado um *circuit designer* mais eficiente para conter o circuito com o RedBear e o sensor MEAS. Esta dificuldade vem do difícil encaixe do sensor em jumpers fêmeas da forma de alimentação do RedBear.

O Daplink é conectado à porta USB do computador e ao RedBear, funcionando como alimentação e interface de programação do RedBear. Seguindo o datasheet, são colocados dois jumpers, sendo um o GND e o outro a porta A3, que são colocados em paralelos com o sensor horizontal e o resistor de  $1\text{M}\Omega$  (ver Figura 33). Também é colocado um paper clip preso ao orifício do sensor, com o objetivo de aumentar a sensibilidade do mesmo adicionando uma massa ao sensor.

O resistor de  $1\text{M}\Omega$  é utilizado com base no manual técnico e cálculos teóricos, onde  $R = \frac{1}{2\pi fC}$  indica que, para um resultado de maior precisão é necessário um resistor na ordem de mega ( $10^6$ ).

Testes realizados com o sensor vertical, utilizando o mesmo circuito descrito anteriormente, indicam que os movimentos de rotação no sensor aumentam a sensibilidade da variação de tensão obtida conforme resultados mostrados a seguir.

## Resultados:

Os dados foram coletados variando a posição, exercendo impacto (“pancada com o dedo da mão”) no sensor e utilizando o monitoramento do arduino IDE (ver Figura 34).

### 1. Média de tensão com impacto ou choque (3.15V)

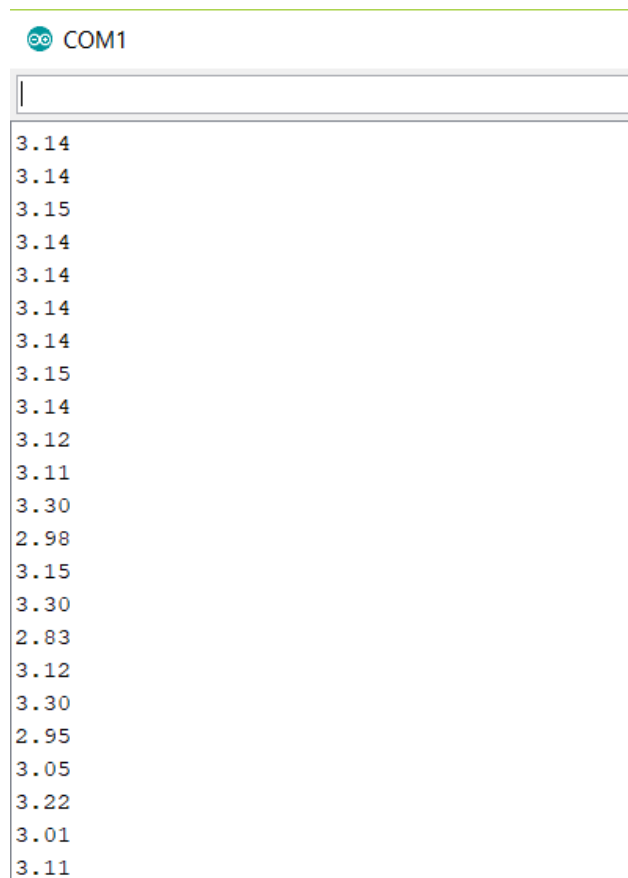


Figura 34 - Resultado com movimentos feitos no sensor

Para realizar este envio de dados periodicamente, do BLE para o celular, foi criado um *Ticker de 1s*. A interface do *Ticker* é usada para configurar uma interrupção recorrente para chamar repetidamente uma função a uma taxa especificada, ou seja, a cada 1s, tempo que foi determinado, ele atualiza o valor e envia para o celular. A interface *Timeout* é usada para configurar uma interrupção para chamar uma função após um delay especificado. O programa em si não armazena os valores coletados, pois a memória do RedBear é limitada e seria desperdício de espaço, portanto, o valor

da variação do sensor será coletado e enviado, em tempo real, para o aplicativo no dispositivo los, funcionando assim como um streaming.

```
BLE          ble;
Timeout      timeout;
Ticker       ticker1s; // criando ticker para atualizar o valor a cada 1s
```

Figura 35 - Criação dos objetos

Na Figura 35 é possível observar a criação do objeto BLE e do Ticker que será utilizado para atualizar a leitura do valor a cada 1s.

```
void atualizaSensor() {
  // uint16_t valordosensor = analogRead(A2);
  // funcao converte valor
  static uint16_t valordosensor=0;
  valordosensor++;
  Serial.println(valordosensor);
  ble.updateCharacteristicValue(characteristic2.getValueAttribute().getHandle()

}
```

Figura 36 - Função Atualiza Sensor

É criada uma função chamada *atualizaSensor* (ver Figura 36) que recebe o valor lido no pino A2, onde ficará o sensor, e colocará o valor medido em Volts, na variável *valordosensor* e, então, será enviado a cada 1s para o celular.

```
void setup() {
  // put your setup code here, to run once
  Serial.begin(9600);
  Serial.attach(uart_handle);

  //Inicializando o ticker
  ticker1s.attach(atualizaSensor, 1);

  //Definindo pino A2 como input
  pinMode(A2, INPUT);
```

Figura 37 - Definição do Tempo do Ticker

Neste trecho de código que é chamada a função *atualizaSensor* é definido o tempo do Ticker, através da linha *ticker1s.attach(atualizaSensor, 1)*, que, neste caso, será de 1s.

#### 4.2.3 Passos para acoplar na raquete e testar o choque

Com a necessidade de algo mais compacto, utilizamos uma mini protoboard e foi desenvolvida uma caixa de material adaptável para armazenar todo o sistema e assim acoplar na raquete (ver Figura 39).

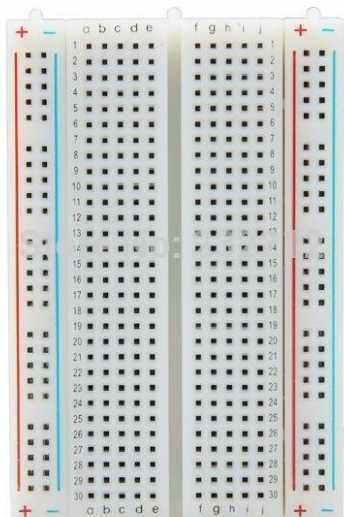


Figura 38 - Mini Protoboard utilizada [29]

Pensadas em várias formas de acoplar na raquete, a escolhida foi no centro da corda da raquete, para conseguir medir com precisão o exato choque que está sendo feito.

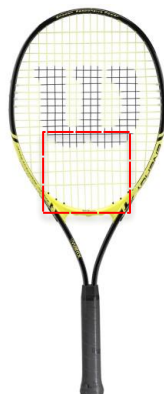


Figura 39 - Localização do Sensor na Raquete [30]

#### 4.2.4 Substituição do micro controlador e teste de precisão do choque

Nesta etapa do projeto, o objetivo era integrar o micro controlador com o celular, por meio do software Blynk. Para isso, foram obtidos diversos códigos fontes da biblioteca disponibilizada pelos desenvolvedores da plataforma Blynk. Ao utilizar os códigos fontes destinados à integração do Blynk ao micro controlador RedBear Nano v2, não estava sendo realizada a conexão entre os dispositivos. Após diversos testes, estudos, análises e alterações no código, o real problema foi detectado: as versões do Bluetooth Low Energy utilizadas no RedBear e pelo software eram incompatíveis, e, por isso, não é possível a conexão entre ambas. Ao mensurar a importância do software para o projeto, e, para um futuro desenvolvimento comercial, decidiu-se por alterar o micro controlador para outro mais complexo e com mais recursos: o ESP32.

Ao realizar tal alteração, obteve-se a conexão via Bluetooth entre a placa e o software, permitindo o acesso aos dados e o tratamento dos mesmos via celular. Após a integração, o circuito, apresentado na *Figura 36*, foi montado. Este é o circuito utilizado para a capturar e transmitir os dados do impacto da bola de tênis na raquete. Os dados são tratados na unidade de Volts, devido à capacidade piezelétrica do sensor minisense 100, e comparados os choques. O circuito é composto do ESP32, 2 jumpers, sendo um para conectar o ground na parte negativa do sensor, e, o outro para capturar os dados do sensor e tratá-los no micro controlador. Também é utilizado um resistor de 1M Ohm, que ajusta a precisão do sensor, e, por último, o sensor.

O teste foi iniciado por meio da conexão via Bluetooth do ESP32 e o aparelho Iphone 8 plus, utilizando um dos widgets do aplicativo Blynk no celular. Após a conexão, foram adicionados widgets de painéis, leds para detecção de batida, e uma tabela para armazenamento dos valores dos impactos entre a bola e a raquete de tênis. Todos estes recursos foram acoplados no Blynk, via celular, e no ESP32, via código, por meio do Arduino IDE. O código em questão pode ser acessado no *Apêndice 2*, no final deste documento. Por fim, foram realizadas batidas de diferentes forças no sensor para o teste de funcionamento da integração e da captação dos dados pelo sensor, tais dados que estão distribuídos entre valores de 0.0V a 3.3V, de acordo com as especificações do ESP32.

Os resultados obtidos deste teste estão apresentados na Figura 41, e, de acordo com os objetivos esperados, pode-se ser considerado uma etapa concluída com sucesso, pois, os valores variaram conforme a força exercida ao tocar no sensor.

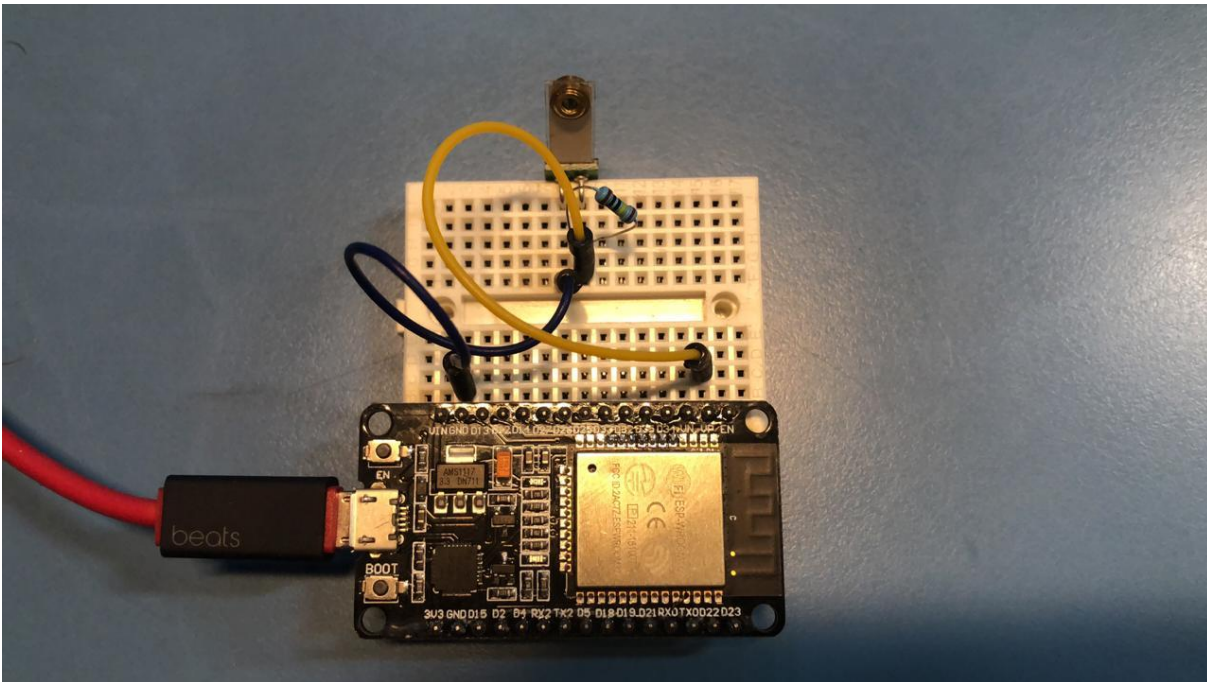


Figura 40 - Circuito montado com o ESP32

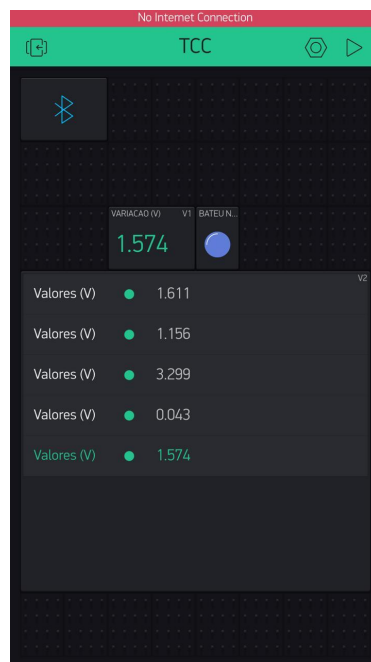


Figura 41 - Valores da Voltagem impressos em formato de tabela no aplicativo Blynk

Com o intuito de melhorar o circuito proposto, e, facilitar a identificação de resultados falsos positivos, ou seja, movimentos que o sensor capta choque onde, na verdade, não houve nenhum, foi acoplado um módulo sensor buzzer de 5V, onde este, é alimentado por um pino de 3.3V do ESP32, seu GND é conectado à um GND comum, e, o pino I/O é conectado ao pino 32 do ESP32, que, de acordo com o programa, recebe o dado do sensor e aciona o buzzer quando existe um choque na raquete. Desta forma, permitiu-se a identificação de falsos positivos e mecanismos para diminuí-los, como por exemplo variando a posição do sensor minisense, alteração de jumpers, diminuição de ruídos e outros. Para reduzir o tamanho do circuito e transmitir os dados para o celular, foi utilizado uma bateria externa de 5V para alimentar o ESP32, e, por fim, o circuito pôde ser inserido na raquete para início dos testes práticos com a bola de tênis. Tal circuito está explicitado na Figura 42.

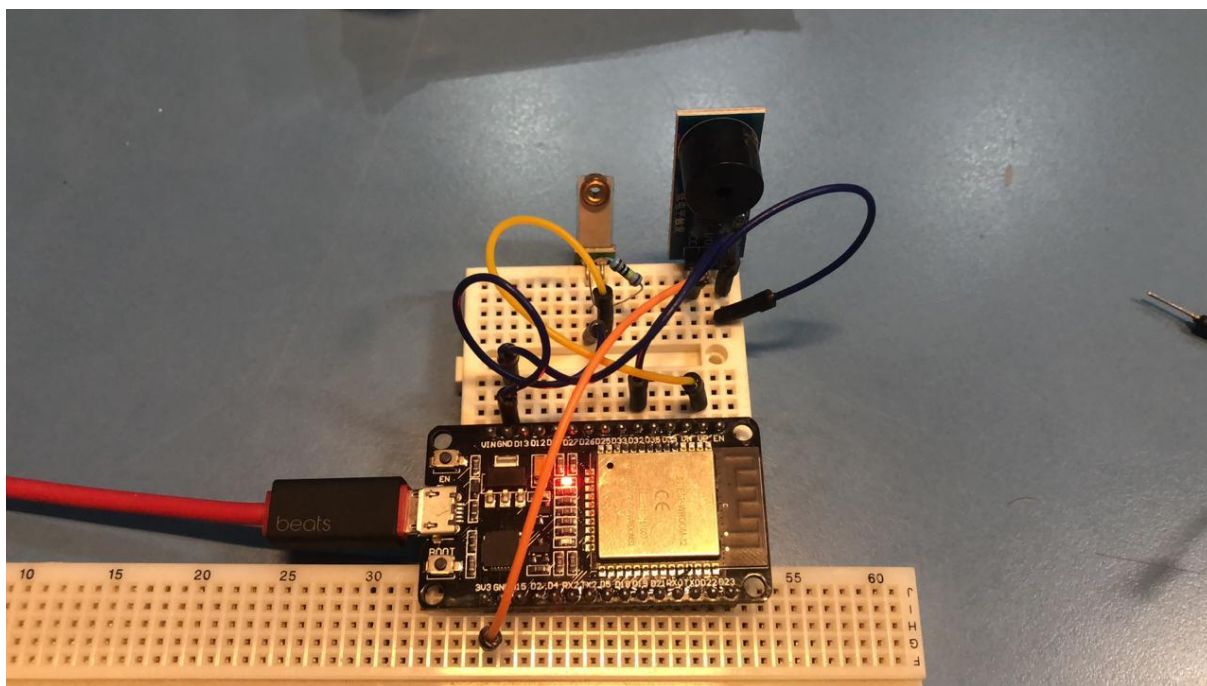


Figura 42 - Circuito com o ESP32 e o buzzer

#### 4.2.5 Dispositivo acoplado na raquete e primeiros resultados

O passo seguinte no experimento foi acoplar o dispositivo no aro da raquete de Tênis Head Youtek Graphene Speed Pro 18 x 20 com o sensor atado nas cordas da raquete. A posição do sensor preso à corda da raquete foi determinada de forma empírica, ou seja, a posição era variada e a bola arremessada, e analisando os

resultados dos choques em cada posição, determinou-se que esta posição da Figura 39 era a que melhor representava os resultados dos choques. O que é coerente com a teoria, pois, a força do impacto recebido no meio da raquete é dissipada em todas as direções de forma linear.

Neste experimento, foi realizado da seguinte forma:

- 2 pessoas, uma arremessando a bola e outra segurando a raquete com o dispositivo e o sensor no aro
- A raquete foi segurada de forma como um jogador de tênis a utiliza
- A distância entre as pessoas era de 1m
- A pessoa que arremessa a bola varia a força nas escalas fraca, média e forte
- A raquete recebia o impacto do arremesso, e o sensor captava o choque, mostrando na tela do Iphone 8 Plus, pelo Blynk, o impacto do do choque em Volts
- O experimento foi repetido 3 vezes, sendo 5 batidas de nível fraco, 5 batidas de nível médio e 5 batidas de nível forte, de forma intercaladas.

Os resultados foram satisfatórios no sentido que, ao variar a força de arremate, os valores apresentados no Blynk pelo sensor, também variaram da mesma forma. Entretanto, foram identificados empecilhos e desafios, como por exemplo, ao colocar o sensor no centro da raquete, o valor obtido no sensor era o valor máximo. Apesar da coerência dos resultados obtidos, os valores não eram precisos, devido à dificuldade humana de equilibrar a força do arremesso, e, também, a dificuldade de quantificação da força de arremate. É um aspecto impreciso as denominações de fraco, médio e forte. Outro resultado obtido a partir deste experimento foi de que o sensor só capta valores quando existe o choque entre a bola e a raquete, ou seja, quando a raquete é movimentada pelo jogador, sem que exista o choque, o sensor não apresenta nenhum valor. A demonstração da captação dos valores obtidos está apresentada na Figura 44 e Figura 45.

Pela Figura 45, é possível observar a presença dos 3 níveis de forças descritos: fraco, médio e forte. Os dois primeiros valores da tabela e o último, 1.611V, 1.156 e 1.574V, são forças médias de choque da bola de tênis com a raquete. O terceiro valor, de 3.299 é o valor máximo de tensão descrito pelo sensor, o que denota a força forte. E, por último, o valor restante, que é de 0.043, representa a força fraca.





Figura 43 - Raquete com o sensor encaixado na rede



Figura 44 - Valores sendo impressos na tabela com o sensor na rede

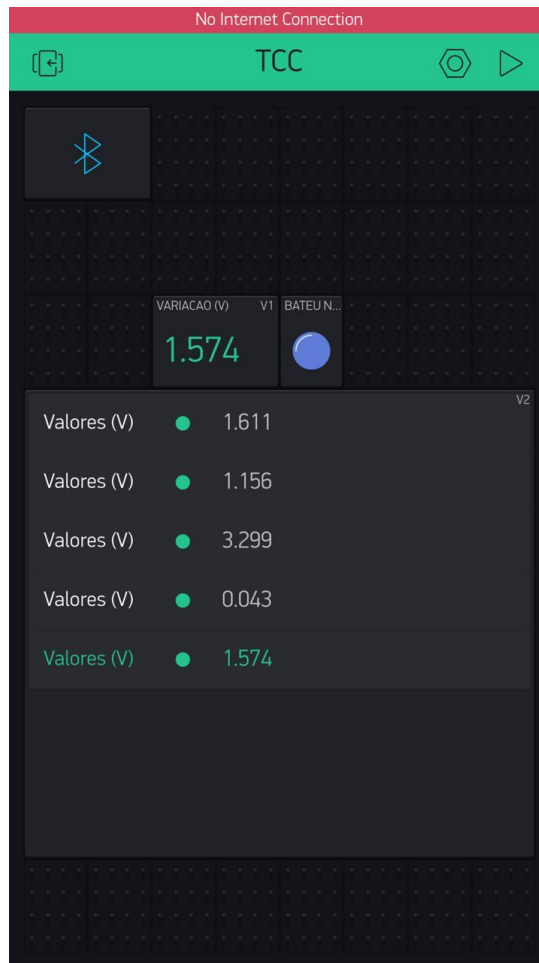


Figura 45 - Valores apresentados no app Blynk

#### 4.2.6 Aperfeiçoamento do experimento

Neste experimento, o principal objetivo era aperfeiçoar os resultados anteriores, isto é, possuir um maior controle do ambiente analisado e obter resultados mais precisos em relação aos anteriores. Para isso, foi feita um controle de todas as medidas utilizadas durante o experimento. E com base nos resultados obtidos, foram criadas faixas de valores que possam determinar a força do choque, em fraca ou forte.

A metodologia utilizada para esta etapa do projeto consiste em posicionar a raquete de forma que a bola de tênis seja arremessada e ocorra a colisão no centro da raquete, para que a força seja distribuída de forma linear em todas as direções, inclusive na que o sensor está presente. Para isso, foi utilizado um banco, de 30,5 cm, onde a raquete fica em cima deste, onde foi feita a marcação para a posição ideal para ocorrer o choque descrito, conforme mostra a Figura 46. Também foram colocadas fitas isolantes atrás da raquete, para que o choque ocorra de forma natural,

sem a intervenção da força de reação, quando há alguém segurando a raquete, ou seja, a raquete recebe o choque e continua vibrando, e, a barreira criada com fita isolante posicionada atrás da raquete, vibra na mesma frequência da raquete, que está apresentada na Figura 47. Para controlar a força utilizada, e, buscar um padrão e a precisão de possuir aproximadamente a mesma força no arremesso, criou-se um pêndulo. Este pêndulo foi criado utilizando uma barra de porta, de comprimento de 61 cm, que foi fixado em 194 cm em relação ao chão. Então, foi colocada uma corda com comprimento de 100 cm presa à bola de tênis, e, em relação à barra de porta, foi fixada à 36,5 cm em relação ao início da barra. Por último, foram determinados que, a força fraca seria a altura de 34 cm em relação ao ponto mais baixo da bola no pêndulo, que é de 82 cm, e a força forte seria a altura de 112 cm em relação a este mesmo ponto. A posição do sensor se manteve a mesma do experimento descrito na seção 4.2.5, que é na diagonal inferior esquerda próximo ao centro da raquete, mostrado na Figura 48. Foram realizadas 30 batidas no total para cada força, sendo estas analisadas via celular, pelo app Blynk, e também pelo computador, via Arduino IDE, para a analisar a convergência de valores e para a constituição de gráficos acerca dos choques ocorridos. Sintetizando as informações, têm-se:

- Altura do banco: 30,5 cm
- Altura da barra em relação ao chão: 194 cm
- Comprimento da barra utilizada para criar o pêndulo: 61 cm
- Comprimento do pêndulo: 100 cm
- Altura da bola no ponto mais baixo do pêndulo: 82 cm
- Altura para força fraca: 116 cm
- Altura para força forte: 194 cm
- Peso da bola: 50g
- Peso da raquete com o aparelho incluído: 441g



Figura 46 - Demarcação da posição ideal da raquete

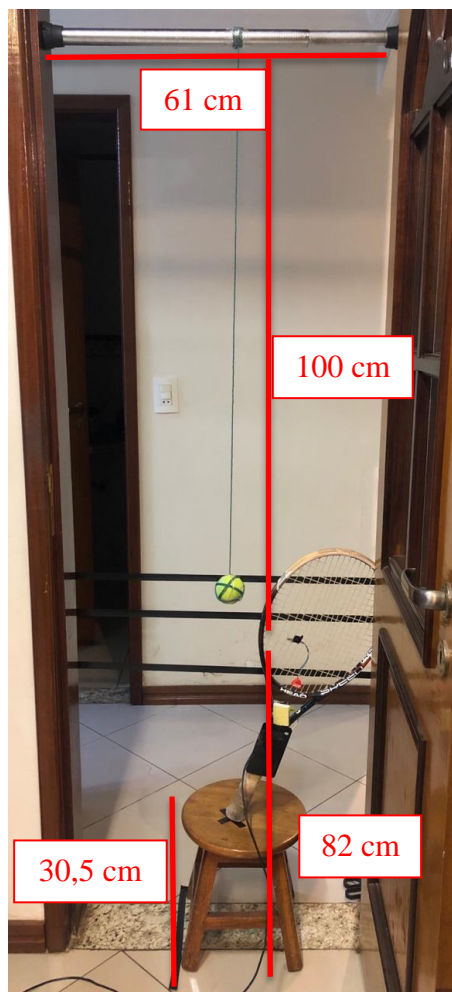


Figura 47 - Protótipo do experimento



Figura 48 - Posição do sensor



Figura 49 - Peso da bola de tênis

Após criado e documentado todo ambiente, foram iniciadas as análises via Blynk e via Arduino IDE, como mostra a Figura 50. Por limitação do Arduino IDE, que só permite observar o Serial Plotter, que é um gerador de gráfico, ou o Serial Monitor, que apresenta os valores obtidos pelo sensor, decidiu-se observar primeiramente os gráficos gerados para as duas forças: a fraca e a forte. Por isso, este experimento foi dividido em duas etapas, a primeira, buscando analisar as diferenças gráficas entre as forças fracas e fortes, sendo esta, uma combinação de aproximadamente 5

arremessos de cada altura. A segunda etapa consiste na análise quantitativa, ou seja, dos valores obtidos durante os arremessos das alturas determinadas, sendo esta, repetida 30 vezes em cada altura para obter a melhor aproximação possível.

Pelo serial plotter, foi possível observar o comportamento do choque das vibrações seguintes após o choque, pois, com a barreira de fita isolante atrás da raquete, quando ocorre o choque a raquete atinge a barreira e ambas vibram na mesma frequência até a raquete ser colocada em repouso novamente. Por meio do gráfico da Figura 50, percebe-se pequenas variações até a primeira onda observada, estas variações devem-se à pequenos ruídos do circuito e a fita que foi utilizada para fixar o sensor na corda. Pelo gráfico, pode-se inferir que o choque principal são as ondas com as cristas mais altas, e as menores, ao seu redor, são vibrações de lóbulos secundários após o choque da bola com a raquete de tênis. Também é possível perceber a escala das ondas, em Volts, que é um valor muito baixo, que será analisado na seção posterior pelo serial monitor e pelo Blynk. E quando não há presença de choque da bola com a raquete, o gráfico permanece muito próximo de zero, existindo pequenas variações devido ao ruído.

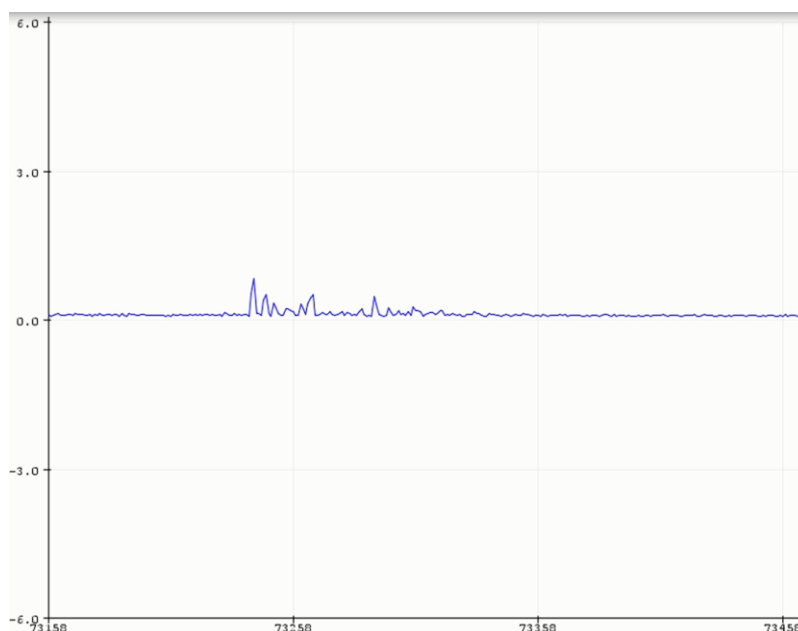


Figura 50 - 4 choques arremessados com a força fraca (Segundos x Volts)

Na segunda etapa do experimento, repetiu-se o modelo anterior, entretanto, para a altura que representa a força forte, que é 112 cm mais alto que o ponto mais

baixo do pêndulo, e a bola de tênis foi lançada 5 vezes. O resultado obtido pelo serial plotter está representado na Figura 51, e pôde-se perceber que a crista das ondas está muito mais alta em relação à altura fraca, ou seja, o choque possui uma intensidade muito maior e chega bem próximo ao valor máximo do range apresentado, que é 3.3V. Também é notório que os lóbulos secundários são maiores e demoram mais para decair, até ficar próximo de 0 novamente.

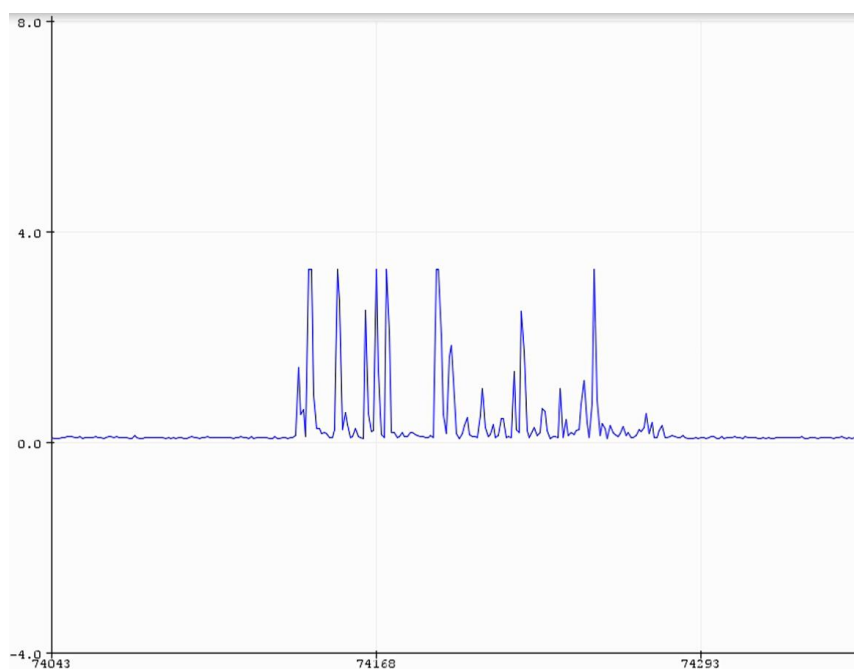


Figura 51 - 5 choques arremessados com a força forte (Segundos x Volts)

Por fim, foram realizados 30 arremessos, valor que permitiu dar uma melhor precisão dos resultados para a força fraca, e, após, mais 30 arremessos para a força forte, com intuito de analisar e comparar os resultados obtidos pelo serial monitor e pelo aplicativo Blynk. Após a coleta destes dados, foram transferidos para uma planilha e calculada a média das forças afim de conseguir distinguir faixas de valores relacionadas com o grau da força utilizado no arremesso. Para isso, foram realizados 30 arremessos ininterruptos, podendo conter falhas humanas e instrumentais, como por exemplo a bola de tênis não atingir exatamente o mesmo ponto da raquete nos 30 lançamentos, pequena variação de posição do sensor na corda e pequena variação na posição da raquete em cima do banco utilizado. Todos esses fatores podem comprometer a precisão dos resultados, e, o experimento buscou minimizar o máximo

possível tais erros. Alguns valores do resultado obtido estão presentes na Figura 52. A tabela descrita está presente no *Apêndice 3*.

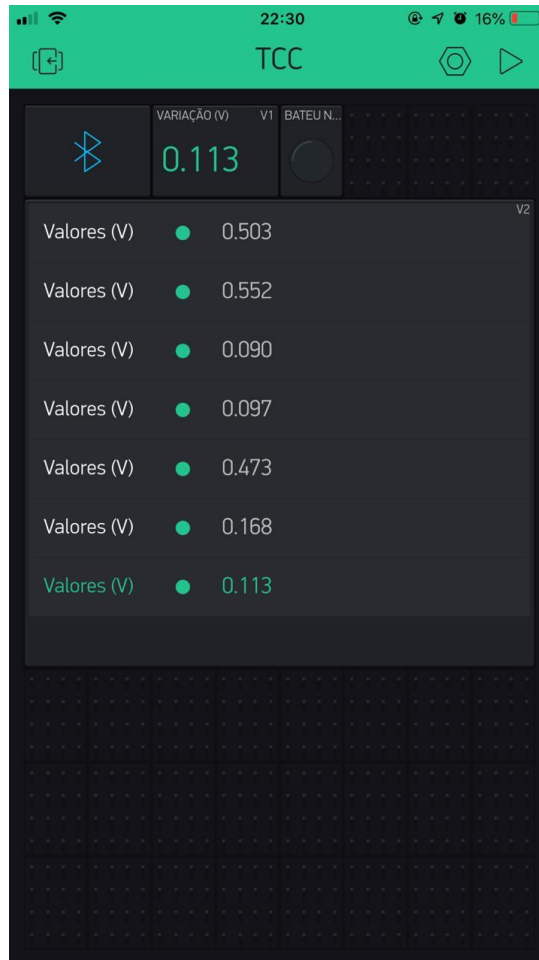


Figura 52 - Valores para força fraca no aplicativo Blynk

Por meio da tabela descrita no *Apêndice 3*, calculou-se a média dos 30 valores obtidos para a força fraca e 40 valores para força forte. Este valor de 40 foi escolhido para se ter uma melhor precisão dos resultados. Sendo elas:

- Média da força fraca em 30 arremessos: 0,344V
- Média da força forte em 30 arremessos: 3,100V

Podendo inferir que um arremesso fraco ficaria entre 0 e 1.1V e que um arremesso forte ficaria entre 2.14V a 3.3V.

Tabela 1 - Dados Resultantes com a batida fraca e forte



<b>Jogada</b>	<b>Força Baixa (V)</b>	<b>Força Alta (V)</b>
1	0,22	3,30
2	0,12	3,30
3	0,1	3,30
4	0,15	3,04
5	1,98	3,30
6	0,25	3,30
7	0,09	2,32
8	0,45	3,30
9	0,26	3,30
10	0,46	3,06
11	0,10	3,30
12	0,09	3,30
13	0,30	2,92
14	0,08	2,78
15	0,50	3,13
16	0,55	3,30
17	0,09	3,30
18	0,10	3,30
19	0,47	3,30
20	0,17	2,57
21	0,11	2,14
22	1,08	3,30
23	0,58	3,01
24	0,10	3,30
25	0,19	3,30
26	0,96	3,30
27	0,10	3,12
28	0,09	2,23
29	0,17	3,30
30	0,30	3,30
<b>Média</b>	0,344483	3,100667
<b>Desvio Padrão</b>	0,406761	0,34627

#### 4.2.7 Repetição do experimento adicionando 3 massas de 3g

Neste experimento, com intuito de analisar os efeitos da adição de massas na raquete, buscou-se repetir o experimento explicitado na seção 4.2.6, de forma que o ambiente fosse praticamente o mesmo, levando em considerações erros humanos e

experimentais. Portanto, de acordo com o estudo citado em [17], A massa adicionada à alça (ver Figura 3) geralmente reduz as vibrações da alça, independentemente do ponto de impacto. A massa adicionada a um ponto de n de vibração tem efeito na frequência de vibração, apenas um pequeno efeito na amplitude e pode ser utilmente adicionada a raquetes muito leves e rígidas para aumentar o poder da raquete sem afetar a baixa amplitude de vibração destas raquetes. Logo, para este experimento, a metodologia foi a mais parecida possível com a seção citada.

Nesta metodologia, foi utilizada a mesma bola de tênis, de massa de aproximadamente 50g, conforme apresentado na Figura 49, um banco, de 30,5 cm, onde a raquete fica posicionada na posição X, também presente na Figura 46. Foram colocadas fitas isolantes atrás da raquete, para que o choque ocorra naturalmente, sem a força de reação, e, para que permita o Arduino IDE detectar as vibrações seguintes ao choque da raquete com a bola. A barra de porta utilizada, a mesma do experimento anterior, de comprimento de 61 cm, foi fixada em 194 cm em relação ao chão. Tal barra, que possui uma corda de 100 cm, preso à bola, possui a função de atuar como pêndulo para a análise do choque da bola com a raquete. Seguindo os mesmos critérios do experimento descrito na seção 4.2.6, determinou-se que a força fraca seria a altura de 34 cm em relação ao ponto mais baixo da bola no pêndulo, que é 82 cm, e a força forte seria a altura de 112 cm em relação a este mesmo ponto. A posição do sensor é a mesma da Figura 48. Foram realizados 30 arremessos no total para cada força, sendo estas analisadas simultaneamente pelo app Blynk e pelo Arduino IDE. Ou seja, a única mudança foi na alteração da massa da raquete, onde adicionou-se 3 fitas de 3g cada uma, da fabricante Babolat, que estão apresentadas na Figura 53. As fitas foram posicionadas nas laterais na raquete e na parte superior, para que houvesse a distribuição de peso de forma equilibrada, conforme a fundamentação teórica apresentada neste trabalho. O resultado da raquete com a adição dos pesos está explicitado na Figura 55 e Figura 57. Resumindo, tem-se:

- Altura do banco: 30,5
- Altura da barra em relação ao chão: 194 cm
- Comprimento da barra utilizada para criar o pêndulo: 61 cm
- Comprimento do pêndulo: 100 cm
- Altura da bola no ponto mais baixo do pêndulo: 82 cm
- Altura para a força fraca: 116 cm

- Altura para a força forte: 194 cm
- Peso da bola: 50g
- Peso da raquete com o aparelho incluído com adição das fitas: 450g

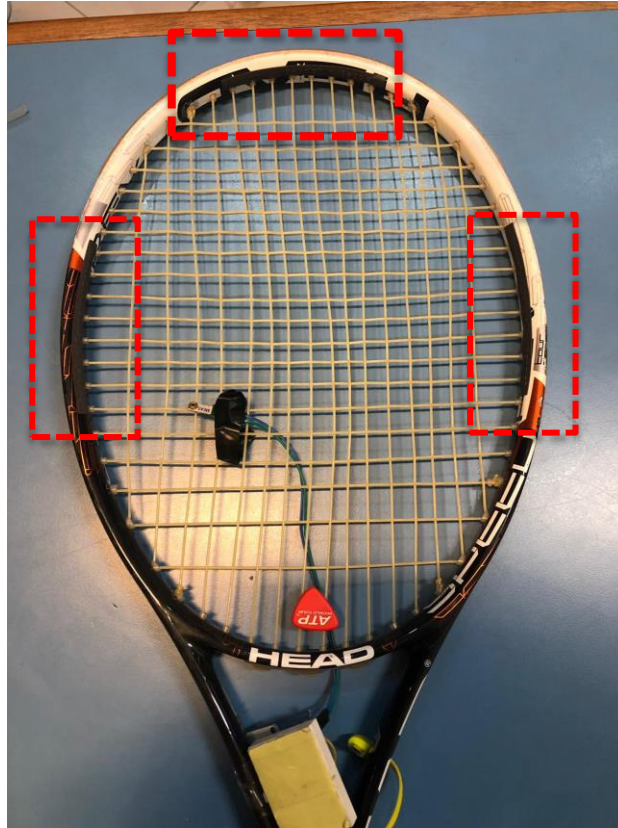


Figura 53 - Posicionamento das fitas de 3g

Após a adição das fitas na raquete, o ambiente foi recriado de forma muito semelhante ao dos resultados da seção 4.2.6, para que a análise comparativa tivesse o menor número possível de variáveis mutáveis, como por exemplo: uma leve alteração na posição do sensor, lugar onde ocorre o choque da bola na raquete, pequena variação na posição do banco, pequena variação da superfície de contato do sensor com a corda da raquete, variação na área de contato da fita isolante ao fixar o sensor e outros. Todos estes fatores, que possuem um nível de dificuldade alto de controle, influenciam nos resultados apresentados. Essas influências foram percebidas empiricamente.

Portanto, ao buscar ao máximo reduzir tais fatores, montou-se o ambiente para realização do experimento, apresentado na Figura 54. Então, utilizando uma fita métrica, mediu-se a altura que a bola de tênis deveria ser solta, tanto para a força

fraca quanto para a força forte, que eram respectivamente 116 cm e 194 cm. Por fim, marcou-se estas alturas e os testes foram iniciados, sendo 30 arremessos para cada força.

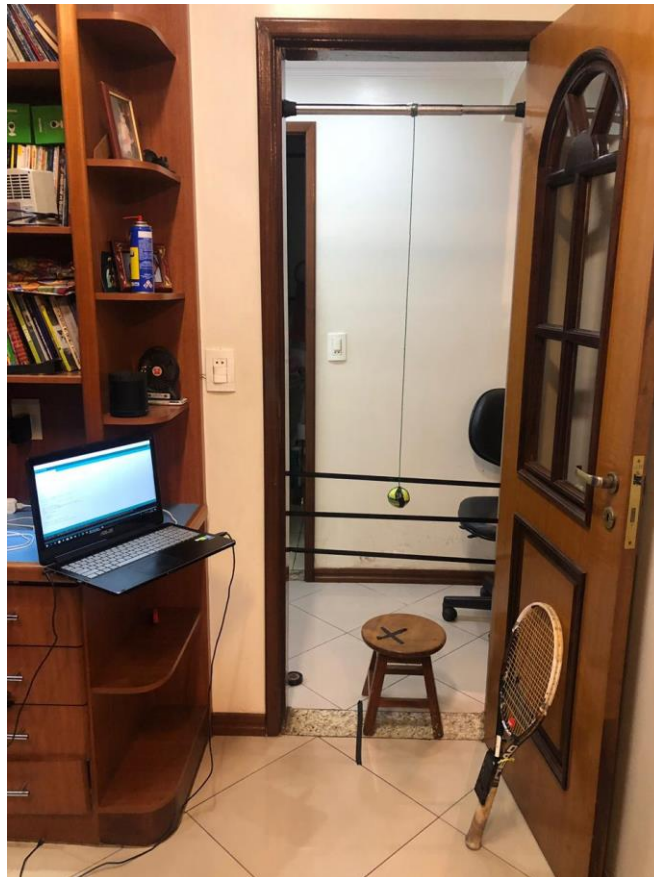


Figura 54 - Ambiente semelhante ao do teste da seção 4.2.6

Seguindo a mesma linha de raciocínio do experimento anterior, foram analisados, primeiramente, o impacto dos choques fracos na raquete de tênis, e observados por meio do Serial Plotter, recurso disponibilizado pelo Arduino IDE obtendo-se o resultado na Figura 55. Foram realizados 5 arremessos da altura fraca, de 116 cm, que resultaram no gráfico da figura descrita. Pode-se observar a redução considerável do impacto da bola de tênis com a raquete, se comparado com o gráfico da Figura 50. Também é notório as mudanças nos lóbulos secundários, onde a raquete demora um pouco mais para perder a força do impacto, ou seja, há presença de mais ondas secundárias. As vibrações seguintes ao choque demoram um tempo maior para perder sua energia. Tais resultados corroboram com a teoria esperada, pois, ao adicionar massa a raquete, espera-se uma maior estabilidade nas jogadas de

tênis em uma partida, portanto, espera-se que não perca muita energia durante a jogada.

A próxima etapa foi executar 30 arremessos, ainda da altura de força fraca, para capturar os resultados numéricos via Arduino IDE e app Blynk. Os resultados obtidos estão expressos nas Figura 56 e Tabela 2. Os resultados provam que realmente existe uma estabilidade muito maior, tendo em vista que os valores foram muito mais próximos entre si do que os valores apresentados na Tabela 1. Pode-se inferir que houve redução na amplitude média do impacto, e, que as vibrações secundárias, obtidas por meio da barreira de fita isolante atrás da raquete, foram perdendo energia em um tempo maior. Então, com base neste teste, pode-se afirmar que a adição de massa na raquete facilitou uma estabilização dos choques. Os valores apresentados na Tabela 2, possuem uma média e a desvio padrão muito menor, de 0.17 e 0.08 respectivamente, o que denota uma precisão maior do teste realizado.

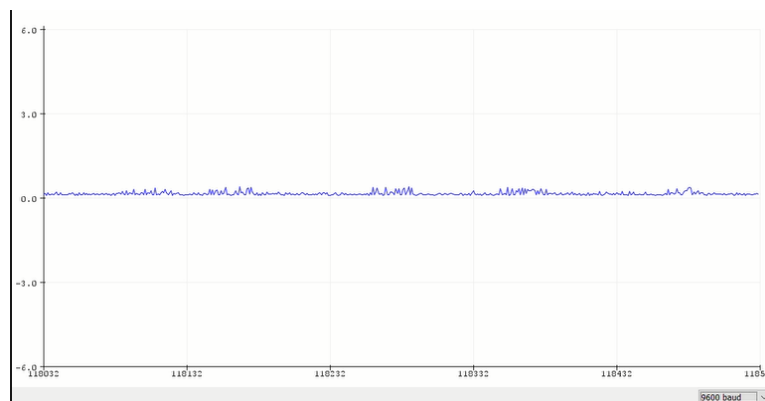


Figura 55 - 5 batidas fracas com o acréscimo de peso de 9g (Segundos x Volts)

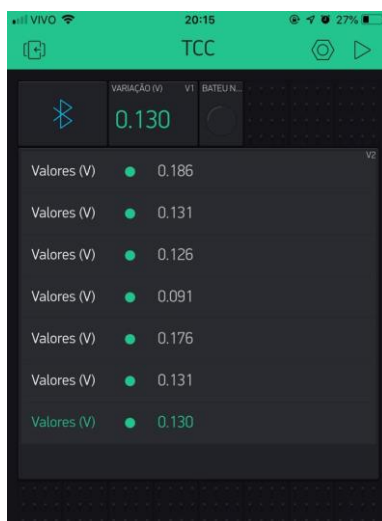


Figura 56 - Captura de valores dos choques

A etapa seguinte foi a análise do choque para a força forte, que é da altura de 194 cm, realizada por meio de 5 batidas da bola com a raquete de tênis, o que produziu o gráfico explicitado na Figura 57. É importante ressaltar que, apenas nesta etapa, não inclui o estágio do teste com os 30 arremessos, foi realizado um filtro para analisar apenas as ondas que possuíam amplitude maior do que 2V. Isso deve-se ao fato de que, com a estabilidade proporcionada pela adição das massas, o Serial Plotter estava capturando diversas ondas dos lóbulos secundários, e, com isso dificultando a análise e a visualização de cada batida realizada. Mas pode-se afirmar que a força forte tem um efeito sobre a raquete e com tensões secundárias consequentes da batida mais amenizadas.

Por fim foram realizados os 30 arremessos da altura da força forte, sem nenhum filtro, e os valores obtidos estão presentes na Tabela 2. Por meio da tabela citada, percebe-se que a média dos choques para a força forte também diminuiu, foi de 3.10 V para 2.99 V, e o desvio padrão da mesma forma, de 0.34 V para 0.29 V, o que sugere uma estabilização, ainda que de menor impacto em relação à comparação para a força fraca.

Após o estudo do impacto foi realizado o estudo da vibração que será descrito no próximo item.

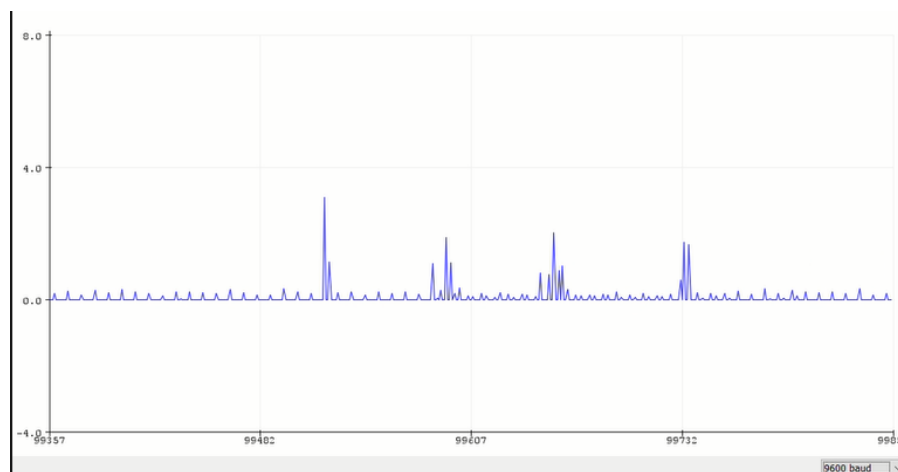


Figura 57 - 5 batidas fortes com o acréscimo de peso de 9g (Segundos x Volts)

Tabela 2 - Dados resultantes com a batida fraca e forte e o acréscimo de 9g

Batidas	Fraco (V)	Forte (V)
1	0,10	2,59
2	0,08	3,13
3	0,14	3,30

4	0,11	3,30
5	0,17	2,90
6	0,48	3,29
7	0,18	2,89
8	0,11	3,30
9	0,13	3,30
10	0,12	3,30
11	0,09	3,30
12	0,23	2,53
13	0,19	2,64
14	0,15	2,88
15	0,11	2,90
16	0,16	2,51
17	0,12	3,30
18	0,15	3,30
19	0,18	2,78
20	0,21	2,71
21	0,18	2,53
22	0,19	2,65
23	0,14	3,15
24	0,15	3,30
25	0,19	2,75
26	0,13	3,07
27	0,09	2,77
28	0,18	2,85
29	0,13	3,26
30	0,36	3,30
<b>Média</b>	0,17	2,992667
<b>Desvio Padrão</b>	0,080462	0,294395

#### 4.2.8 Estudo da vibração com os valores da batida forte

Realizou-se este estudo somente com a força denominada forte neste estudo (ver item 4.2.7) já que é a força mais encontrada em jogos de tênis amadores ou profissionais. Após a coleta dos dados conforme descrito no item anterior, aplicou-se a transformada de Fourier nos gráficos obtidos com o sensor para estudo da frequência da vibração utilizando o código de programa que se encontra no Apêndice

4 utilizado no software MatLab. As amostras foram de 100 pontos durante o tempo de 1 segundo, obtidos de cada gráfico da batida em volts.

Foram realizados dois estudos: (a) batida forte sem a massa adicionada e (b) batida forte com massa de 3g adicionada, como mostrado na Figura 58, Figura 59, Figura 60 e Figura 61:

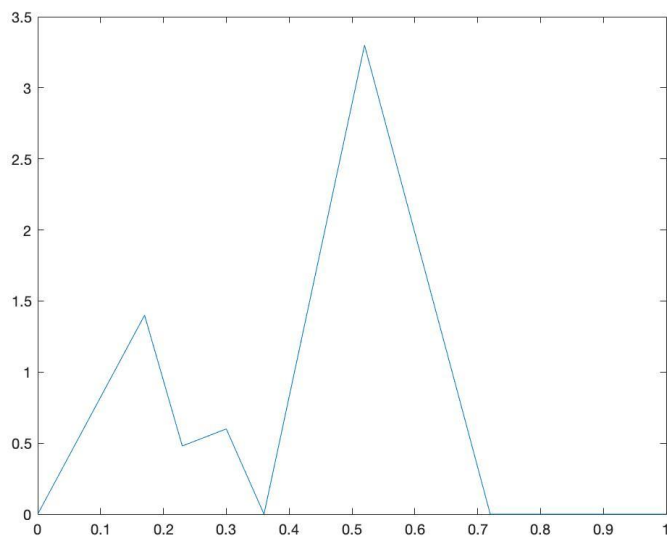


Figura 58 - Gráfico em Volts da batida forte sem massa (Segundos x Volts)

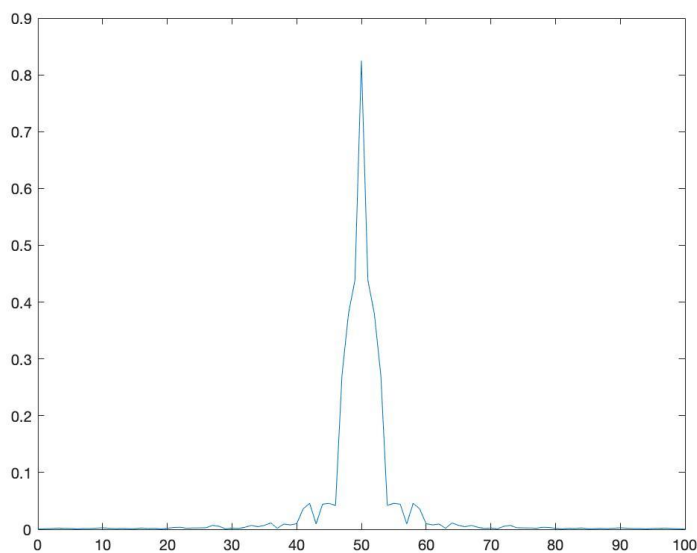


Figura 59 - Gráfico de Fourier da batida forte sem massa (Hz x Amplitude)



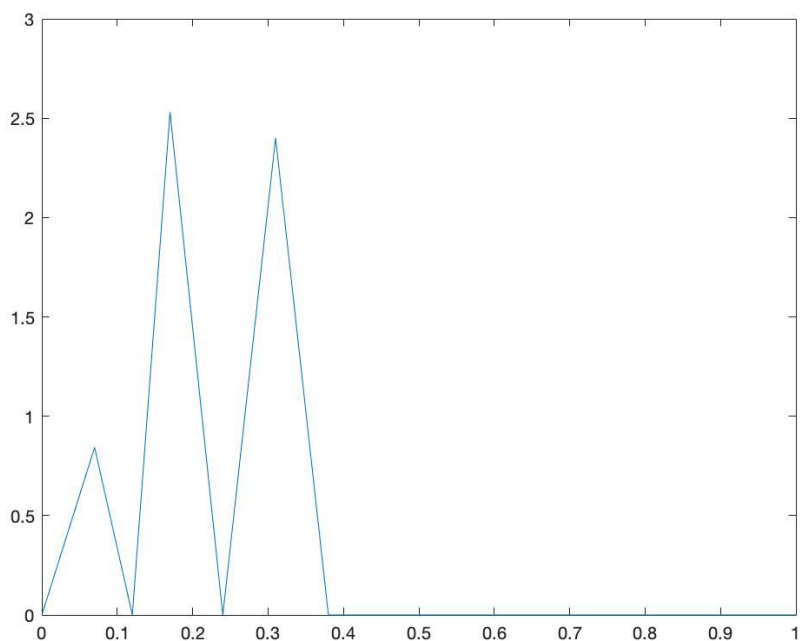


Figura 60 - Gráfico da batida forte com massa (Segundos x Volts)

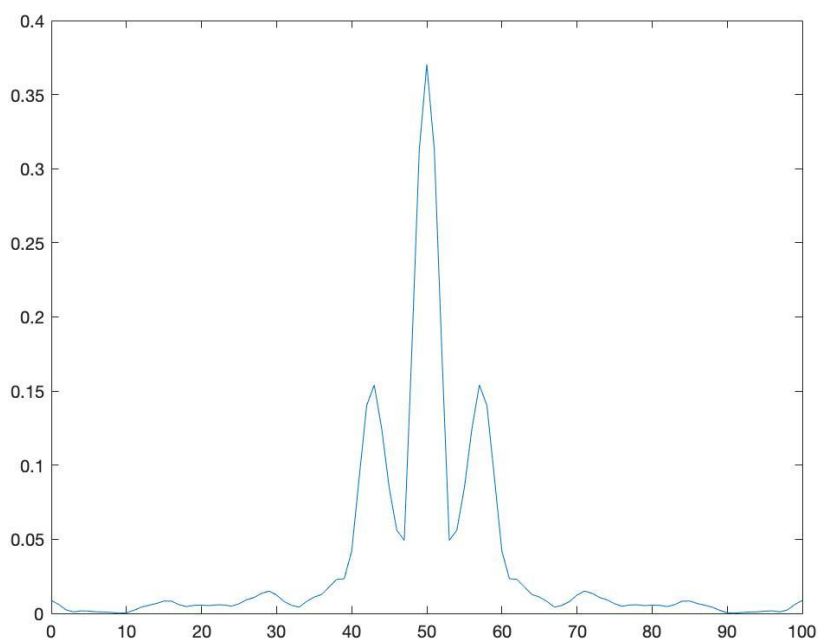


Figura 61 - Gráfico de Fourier da batida forte com massa (Hz x Amplitude)

Foram obtidos os seguintes dados:

Tabela 3 - Valores calculados após a análise com a transformada de Fourier

Valores	Raquete sem massa	Raquete com massa	Varição com o adicional de massa
<b>Média da Amplitude</b>	0,0367	0,0337	- 8,17%

<b>Máxima Amplitude</b>	0,8252	0,3704	- 55,11%
<b>Desvio Padrão</b>	0,1177	0,0672	-
<b>Potência do Choque</b>	1,5204 W	0,5657 W	- 62,79%
<b>Duração do impacto (saindo de 0V e voltando até 0V)</b>	0,72s	0,38s	- 47,22%

Tabela 4 - Valores das amplitudes calculados pela Transformada de Fourier

<b>Amplitude – Raquete sem Massa</b>	<b>Amplitude – Raquete Com massa</b>
0,000365	0,008795
0,001082	0,00594
0,001435	0,002271
0,002213	0,001026
0,001863	0,001696
0,001631	0,001512
0,000828	0,001036
0,001524	0,000961
0,001471	0,000634
0,00205	0,000167
0,00268	0,000393
0,001969	0,002087
0,00134	0,004213
0,001771	0,005493
0,00127	0,006693
0,001126	0,008315
0,002414	0,008225
0,001482	0,005887
0,001824	0,00456
0,000848	0,005475
0,001794	0,005565
0,003319	0,005253
0,003622	0,005752
0,001888	0,00571
0,002469	0,004774
0,002553	0,006521
0,002772	0,009121
0,00697	0,010761
0,00533	0,013516
0,000821	0,015063
0,002382	0,012348
0,001556	0,007956
0,003619	0,005389
0,006899	0,004186
0,004573	0,008051

0,00697	0,011024
0,011204	0,012728
0,001836	0,017839
0,00929	0,023005
0,007779	0,023333
0,009867	0,041975
0,03604	0,091681
0,045781	0,140325
0,00951	0,154095
0,04428	0,124336
0,045684	0,084578
0,042036	0,05609
0,271276	0,049256
0,379348	0,176815
0,438464	0,312962
0,8252	0,37038
0,438464	0,312962
0,379348	0,176815
0,271276	0,049256
0,042036	0,05609
0,045684	0,084578
0,04428	0,124336
0,00951	0,154095
0,045781	0,140325
0,03604	0,091681
0,009867	0,041975
0,007779	0,023333
0,00929	0,023005
0,001836	0,017839
0,011204	0,012728
0,00697	0,011024
0,004573	0,008051
0,006899	0,004186
0,003619	0,005389
0,001556	0,007956
0,002382	0,012348
0,000821	0,015063
0,00533	0,013516
0,00697	0,010761
0,002772	0,009121
0,002553	0,006521
0,002469	0,004774
0,001888	0,00571
0,003622	0,005752
0,003319	0,005253

0,001794	0,005565
0,000848	0,005475
0,001824	0,00456
0,001482	0,005887
0,002414	0,008225
0,001126	0,008315
0,00127	0,006693
0,001771	0,005493
0,00134	0,004213
0,001969	0,002087
0,00268	0,000393
0,00205	0,000167
0,001471	0,000634
0,001524	0,000961
0,000828	0,001036
0,001631	0,001512
0,001863	0,001696
0,002213	0,001026
0,001435	0,002271
0,001082	0,00594
0,000365	0,008795

Analisando a Tabela 3 e a Tabela 4, pode-se observar que a amplitude máxima, o desvio padrão, a duração do impacto e a média da amplitude reduziram depois que a massa foi adicionada, mostrando que a raquete se torna mais estável. Como já explicitado pelo estudo referenciado em [17] o acréscimo de massa causa um pequeno efeito na amplitude de vibração e grande redução da duração da vibração da raquete, fazendo com que esta esteja mais estável. Este estudo chegou a este mesmo resultado como pode-se ver na tabela 3.

Novas tecnologias com a Kinectic Q sugerem a eliminação dos choques e redução da vibração em 43% também com adição de massas as raquetes [25]. Neste estudo, apesar de que foi realizado em um ambiente controlado e com modelo diferente de raquete e de bola, também conseguiu-se uma ótima redução do choque, 62,79%, e da duração da vibração, 47,22%, mostrando com isso que realmente a adição de massas a raquetes de tênis diminui o choque da bola e reduz a vibração, aspectos muito importantes quando se fala da saúde do braço de uma tenista, com suas frequentes lesões devido ao jogo de tênis.

Tabela 5 - Valores calculados após a análise da transformada de Fourier e priorização da frequência fundamental

	Sem Massa	Com Massa
Média frequência (somente com a frequência harmônica)	0,0374	0,0244

Tabela 6 - Valores das amplitudes calculados pela Transformada de Fourier sem as frequências secundárias

Amplitude – Raquete sem Massa	Amplitude – Raquete Com massa
0,000365	0,008795
0,001082	0,00594
0,001435	0,002271
0,002213	0,001026
0,001863	0,001696
0,001631	0,001512
0,000828	0,001036
0,001524	0,000961
0,001471	0,000634
0,00205	0,000167
0,00268	0,000393
0,001969	0,002087
0,00134	0,004213
0,001771	0,005493
0,00127	0,006693
0,001126	0,008315
0,002414	0,008225
0,001482	0,005887
0,001824	0,00456
0,000848	0,005475
0,001794	0,005565
0,003319	0,005253
0,003622	0,005752
0,001888	0,00571
0,002469	0,004774
0,002553	0,006521
0,002772	0,009121
0,00697	0,010761
0,00533	0,013516
0,000821	0,015063
0,002382	0,012348
0,001556	0,007956

0,003619	0,005389
0,006899	0,004186
0,004573	0,008051
0,00697	0,011024
0,011204	0,012728
0,001836	0,017839
0,00929	0,023005
0,045684	0,084578
0,042036	0,05609
0,271276	0,049256
0,379348	0,176815
0,438464	0,312962
0,8252	0,37038
0,438464	0,312962
0,379348	0,176815
0,271276	0,049256
0,009867	0,041975
0,007779	0,023333
0,00929	0,023005
0,001836	0,017839
0,011204	0,012728
0,00697	0,011024
0,004573	0,008051
0,006899	0,004186
0,003619	0,005389
0,001556	0,007956
0,002382	0,012348
0,000821	0,015063
0,00533	0,013516
0,00697	0,010761
0,002772	0,009121
0,002553	0,006521
0,002469	0,004774
0,001888	0,00571
0,003622	0,005752
0,003319	0,005253
0,001794	0,005565
0,000848	0,005475
0,001824	0,00456
0,001482	0,005887
0,002414	0,008225
0,001126	0,008315
0,00127	0,006693
0,001771	0,005493
0,00134	0,004213

0,001969	0,002087
0,00268	0,000393
0,00205	0,000167
0,001471	0,000634
0,001524	0,000961
0,000828	0,001036
0,001631	0,001512
0,001863	0,001696
0,002213	0,001026
0,001435	0,002271
0,001082	0,00594
0,000365	0,008795

Pelos gráficos e pelos dados apresentados na Tabela 5 e Tabela 6, pode-se observar que na frequência de vibração da raquete sem massa, não apresentava frequências secundárias e na raquete com massa já apresenta as frequências secundárias mais elevadas e a harmônica diminuída. Priorizando apenas a frequência fundamental e não medindo as frequências secundárias, pode-se observar que a média das amplitudes diminui ainda mais, com isso, tendo em vista que a potência do impacto diminuiu quando adicionada a massa, e a média das amplitudes das frequências observando apenas a frequência harmônica foi também menor, podemos dizer que a vibração foi diminuída e a raquete tende a se tornar mais estável com o acréscimo de massa.

## 5. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Diante dos estudos realizados ao longo do projeto e dos resultados discutidos na seção 4 deste documento, pode-se inferir que as conclusões do trabalho foram semelhantes aos dos estudos comparados na seção 1. O que sugere que esta linha de pesquisa e de experimento estão indo em direção correta como a da tecnologia Kinetic Q em raquetes de tênis citada na seção 4.

Esta linha de estudo sugere que a adição e/ou remoção de pequenas massas, distribuídas de forma a equilibrar o peso da raquete, estão intrinsecamente relacionadas ao poder de choque realizado em contato com a bola de tênis. Estas podem estabilizar, aumentar ou diminuir o choque inicial e as vibrações seguintes da corda após o choque de acordo com tais massas. Esse tipo de estudo é de extrema importância para atletas de alto rendimento, como os tenistas Novak Djokovic, Maria Sharapova, Roger Federer, Serena Williams e outros. O protótipo desenvolvido permitiu esta conexão entre esporte e tecnologia, podendo ser usado como ferramenta de análise esportiva na palma da mão, via celular, utilizando o aplicativo Blynk. Cada detalhe é importante para atletas deste nível, seja para aumentar sua eficiência dentro de quadra, para prevenir lesões ou até mesmo para escolha da raquete para disputar campeonatos. Este trabalho contribui para um leque de estudos dentro da ciência do esporte, principalmente se aliado com a medicina e biotecnologia do esporte para estudos de prevenção de lesões.

Entretanto, mesmo com a conclusão em direção ao caminho correto, foram enfrentadas diversas dificuldades para obtenção destes resultados, e a precisão dos mesmos, foi satisfatória, mas não ideal. Os desafios enfrentados esbarram em limitações financeiras, dificuldades mecânicas e também na dificuldade de um ambiente ideal. Por exemplo, o sensor minisense 100, talvez não fosse sensor mais indicado para análise destes choques. Houve uma grande dificuldade de entrelaçar os jumpers utilizados em meio as cordas da raquete e para a fixação do sensor na raquete de forma que capte os valores, pois a fita do sensor deveria estar em contato com a corda da raquete, e, ao utilizar a fita isolante para fixar, muitos dos resultados eram influenciados por este. Por isso, neste sentido, para estudos posteriores,



recomenda-se pensar a respeito de novas formas de fixação do sensor na raquete e, até mesmo, em outros tipos de sensores. Talvez algum sensor que consiga captar os choques através do cabo da raquete e não por meio das cordas. Outro desafio foi o tamanho do dispositivo desenvolvido para encaixe entre os cabos da raquete. É possível reduzir o tamanho do dispositivo, seja utilizando microcontroladores menores ou baterias menores, no projeto em questão foi descartado o RedBear Nano V2 devido à incompatibilidade bluetooth, que era uma das premissas do trabalho, obter os resultados via bluetooth no celular. Por isso optou-se pelo ESP32, entretanto perdeu-se no tamanho do microcontrolador e na bateria utilizada, que era uma bateria externa de 5V ao invés da bateria-moeda. Também é válido ressaltar que devido ao tamanho do dispositivo, houve uma alteração no peso final da raquete com o dispositivo acoplado. É um outro ponto a se observar. Por último, os ambientes de teste descritos neste trabalho foram os mais controlados possíveis dentro do orçamento proposto para o projeto, mas notou-se uma dificuldade para arremessar a bola de tênis no centro da raquete da mesma forma e sempre no mesmo ponto, tais parâmetros podem alterar resultados. É uma dificuldade que foi imposta pelo ambiente e pelo erro humano.

Sugere-se também a realização deste estudo com outros modelos de raquetes com características como tamanho da cabeça e peso diferentes, além deste estudo com diferentes valores de massas.

Este estudo foi feito em um ambiente controlado com forças diferentes e tempos de impacto diferente, por isso, os valores de frequências, duração de impacto e potência são diferentes dos outros estudos e servem apenas para comparação qualitativa e, conseqüente análise do acréscimo de massa em uma raquete.

Portanto conclui-se, por meio deste estudo, que o acréscimo de massa a raquete realmente produz um efeito na vibração, reduzindo-a, e produz um efeito na amplitude como mostrado pelos gráficos da Transformada de Fourier gerados através dos valores de tensão medidos pelo sensor. Sendo assim, este acréscimo é muito importante para a diminuição de lesões, pois, quanto menor a vibração na raquete, menor é a vibração transmitida para o pulso do atleta, conservando sua saúde física.

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

- [1]. Shamsaa Hilal Al. Hosni, "Bluetooth Low Energy: A Survey", 2017
- [2]. <https://medium.com/@avigezerit/bluetooth-low-energy-on-android-22bc7310387a>
- [3]. João Martins, "How to implement a FFT algorithm", 2005 <<https://www.codeproject.com/Articles/9388/How-to-implement-the-FFT-algorithm>>
- [4]. Seyed Mahdi Darroudi, "Bluetooth Low Energy Mesh Networks: A Survey", 2017
- [5]. Shahid Raza, "Bluetooth Smart: An Enabling Technology for the Internet of Things", 2015
- [6]. J. Nieminen, T. Savolainen, M. Isomaki, B. Patil, Z. Shelby, e C. Gomez, "IPv6 over Bluetooth Low Energy. Working Draft", 2015.
- [7]. Dimitrios Sikeridis, "BLEBeacon: A Real-Subject Trial Dataset from Mobile Bluetooth Low Energy Beacons", 2018
- [8]. Inês Maria Saraiva Ramos, "Versatilidade da Tecnologia Beacon: As suas potencialidades num Museu de Portugal", 2017
- [9]. Isabelle Rogowski, Thomas Creveaux, Sylvain Triquigneaux, Pierre Macé, Fabien Gauthier, Violaine Sevrez, "Tennis Racket Vibrations and Shock Transmission to the Wrist during Forehand Drive", 2015
- [10]. Saul Teukolsky and William H. Press, "Numerical Recipes in C", 1986
- [11]. Weisstein, Eric W. "Danielson-Lanczos Lemma." From MathWorld--A Wolfram Web Resource. < <https://mathworld.wolfram.com/danielson-lanczoslemma.html> >
- [12]. <http://twu.tennis-warehouse.com/cgi-bin/vibfrequency.cgi>
- [13]. <https://www.maximintegrated.com/en/app-notes/index.mvp/id/505>
- [14]. <https://www.sparkfun.com/products/retired/14070>
- [15]. [http://www.alwayslearn.com/DFT%20and%20FFT%20Tutorial/DFTandFFT\\_FF\\_T\\_DanielsonLanczosLemma.html](http://www.alwayslearn.com/DFT%20and%20FFT%20Tutorial/DFTandFFT_FF_T_DanielsonLanczosLemma.html)
- [16]. [https://cdn.sparkfun.com/datasheets/loT/esp32\\_datasheet\\_en.pdf](https://cdn.sparkfun.com/datasheets/loT/esp32_datasheet_en.pdf)
- [17]. <https://www.sparkfun.com/products/12786>
- [18]. Rod Cross, "Factors affecting the vibration of tennis racquets", 2015
- [19]. Brody H (1989) Vibration damping of tennis rackets. Journal of Sports Biomechanics, 5:293-296
- [20]. <https://www.blynk.cc/>
- [21]. Renato Fiorentino Vieira, "Conhecendo a piezoelectricidade, uma nova forma de geração de energia", 2017
- [22]. [https://www.sparkfun.com/datasheets/Sensors/Flex/MiniSense\\_100.pdf](https://www.sparkfun.com/datasheets/Sensors/Flex/MiniSense_100.pdf)
- [23]. [http://www.huinfinito.com.br/arduino/969-modulo-sensor-buzzer-5v.html?search\\_query=buzzer&results=9](http://www.huinfinito.com.br/arduino/969-modulo-sensor-buzzer-5v.html?search_query=buzzer&results=9)
- [24]. <https://opentextbc.ca/physicstestbook2/chapter/the-simple-pendulum/>
- [25]. <http://http://www.prokennex.com.br/>
- [26]. <https://www.sparkfun.com/products/338>
- [27]. <https://www.sparkfun.com/products/13883>
- [28]. <https://portal.vidadesilicio.com.br/conhecendo-o-esp32/>
- [29]. <https://www.filipeflop.com/produto/protoboard-400-pontos/>
- [30]. <https://www.tennisexpress.com/tennis-racquets>

# APÊNDICES

## Apêndice 1

```
#include <nRF5x_BLE_API.h> // chamando a biblioteca do nRF
#include "arduinoFFT.h"

#define DEVICE_NAME      "BLE_TCC"
#define TXRX_BUF_LEN    20
#define SAMPLES 128      //Must be a power of 2
#define SAMPLING_FREQUENCY 1000 //Hz, must be less than 10000 due to ADC

arduinoFFT FFT = arduinoFFT();

unsigned int sampling_period_us;
unsigned long microseconds;

double vReal[SAMPLES];
double vImag[SAMPLES];
const int PIEZO_PIN = A3; // Recebe informacao do pino A3 e coloca em PIEZO em valores

BLE          ble;
Timeout      timeout;
Ticker       ticker1s; // criando ticker para atualizar o valor a cada 1s
static uint8_t rx_buf[TXRX_BUF_LEN];
static uint8_t rx_buf_num;
static uint8_t rx_state=0;

// The uuid of service and characteristics
static const uint8_t service1_uuid[] = {0x71, 0x3D, 0, 0, 0x50, 0x3E, 0x4C, 0x75, 0xBA,
0x94, 0x31, 0x48, 0xF1, 0x8D, 0x94, 0x1E};
static const uint8_t service1_tx_uuid[] = {0x71, 0x3D, 0, 3, 0x50, 0x3E, 0x4C, 0x75, 0xBA,
0x94, 0x31, 0x48, 0xF1, 0x8D, 0x94, 0x1E};
static const uint8_t service1_rx_uuid[] = {0x71, 0x3D, 0, 2, 0x50, 0x3E, 0x4C, 0x75, 0xBA,
0x94, 0x31, 0x48, 0xF1, 0x8D, 0x94, 0x1E};
static const uint8_t uart_base_uuid_rev[] = {0x1E, 0x94, 0x8D, 0xF1, 0x48, 0x31, 0x94,
0xBA, 0x75, 0x4C, 0x3E, 0x50, 0, 0, 0x3D, 0x71};

uint8_t tx_value[TXRX_BUF_LEN] = {0,};
uint8_t rx_value[TXRX_BUF_LEN] = {0,};

// Initialize value of chars
GattCharacteristic characteristic1(service1_tx_uuid, tx_value, 1, TXRX_BUF_LEN,
GattCharacteristic::BLE_GATT_CHAR_PROPERTIES_WRITE |
GattCharacteristic::BLE_GATT_CHAR_PROPERTIES_WRITE_WITHOUT_RESPONSE );
GattCharacteristic characteristic2(service1_rx_uuid, rx_value, 1, TXRX_BUF_LEN,
GattCharacteristic::BLE_GATT_CHAR_PROPERTIES_NOTIFY);
GattCharacteristic *uartChars[] = {&characteristic1, &characteristic2};
GattService      uartService(service1_uuid, uartChars, sizeof(uartChars) /
sizeof(GattCharacteristic *));

void disconnectionCallback(const Gap::DisconnectionCallbackParams_t *params) {
```

```

Serial.println("Disconnected!");
Serial.println("Restarting the advertising process");
ble.startAdvertising();
}

void gattServerWriteCallBack(const GattWriteCallbackParams *Handler) {
  uint8_t buf[TXRX_BUF_LEN];
  uint8_t index;
  uint16_t bytesRead = TXRX_BUF_LEN;

  Serial.println("onDataWritten : ");
  if (Handler->handle == characteristic1.getValueAttribute().getHandle()) {
    ble.readCharacteristicValue(characteristic1.getValueAttribute().getHandle(), buf,
&bytesRead);
    Serial.print("bytesRead: ");
    Serial.println(bytesRead, HEX);
    for(index=0; index<bytesRead; index++) {
      Serial.write(buf[index]);
    }
    Serial.println("");
  }
}

void m_uart_rx_handle() { //update characteristic data
  ble.updateCharacteristicValue(characteristic2.getValueAttribute().getHandle(), rx_buf,
rx_buf_num);
  memset(rx_buf, 0x00,20);
  rx_state = 0;
}

void atualizaSensor() {
  // uint16_t valordosensor = analogRead(A2);
  // funcao converte valor
  static uint16_t valordosensor=0;
  // valordosensor++;
  // Serial.println(valordosensor);
  ble.updateCharacteristicValue(characteristic2.getValueAttribute().getHandle(), (uint8_t
*)&valordosensor,5); // atualizando a caracteristica do ble (lendo valor do sensor a cada 1s)
}

void uart_handle(uint32_t id, SerialIrq event) { /* Serial rx IRQ */
  if(event == RxIrq) {
    if(rx_state == 0) {
      rx_state = 1;
      timeout.attach_us(m_uart_rx_handle, 100000);
      rx_buf_num=0;
    }
  }
  while(Serial.available()) {

    Serial.println("Lendo buffer Rx: ");
    /* if(rx_buf_num < 20) {
      rx_buf[rx_buf_num] = Serial.read();
    }
  }
}

```

```

        Serial.print(rx_buf[rx_buf_num]);
        rx_buf_num++;
    }
    else {
        Serial.read();
    }
    /*
    loop();
    */
}
}
}

void setup() {
    // put your setup code here, to run once
    Serial.begin(9600);
    Serial.attach(uart_handle);

    sampling_period_us = round(1000000*(1.0/SAMPLING_FREQUENCY));
}

//Inicializando o ticker
ticker1s.attach(atualizaSensor, 1);

//Definindo pino A2 como input
pinMode(A2,INPUT);

ble.init();
ble.onDisconnection(disconnectionCallBack);
ble.onDataWritten(gattServerWriteCallBack);

// setup adv_data and srp_data
ble.accumulateAdvertisingPayload(GapAdvertisingData::BREDR_NOT_SUPPORTED);
ble.accumulateAdvertisingPayload(GapAdvertisingData::SHORTENED_LOCAL_NAME,
    (const uint8_t *)"TXRX", sizeof("TXRX") - 1);

ble.accumulateAdvertisingPayload(GapAdvertisingData::COMPLETE_LIST_128BIT_SERVICES_IDS,
    (const uint8_t *)uart_base_uuid_rev, sizeof(uart_base_uuid_rev));

// set adv_type
ble.setAdvertisingType(GapAdvertisingParams::ADV_CONNECTABLE_UNDIRECTED);
// add service
ble.addService(uartService);
// set device name
ble.setDeviceName((const uint8_t *)"Simple Chat");
// set tx power, valid values are -40, -20, -16, -12, -8, -4, 0, 4
ble.setTxPower(4);
// set adv_interval, 100ms in multiples of 0.625ms.
ble.setAdvertisingInterval(160);
// set adv_timeout, in seconds
ble.setAdvertisingTimeout(0);
// start advertising
ble.startAdvertising();
Serial.println("Advertising Start!");
}

```

```

void loop() {
  ble.waitForEvent();

  /*SAMPLING*/
  for(int i=0; i<SAMPLES; i++)
  {
    microseconds = micros(); //Overflows after around 70 minutes!

    //int piezoADC = analogRead(PIEZO_PIN);
    // float piezoV = piezoADC/ 1023.0 * 3.3; //3.3V e a maxima voltagem do redBear

    vReal[i] = analogRead(PIEZO_PIN); // Joga o valor em Volts para ser devolvido em Hz
    // Serial.println(i);
    // Serial.println(vReal[i]);
    vImag[i] = 0;

    while(micros() < (microseconds + sampling_period_us)){
    }
  }

  /*FFT*/
  FFT.Windowing(vReal, SAMPLES, FFT_WIN_TYP_HAMMING, FFT_FORWARD);
  FFT.Compute(vReal, vImag, SAMPLES, FFT_FORWARD);
  FFT.ComplexToMagnitude(vReal, vImag, SAMPLES);
  double peak = FFT.MajorPeak(vReal, SAMPLES, SAMPLING_FREQUENCY);

  /*PRINT RESULTS*/
  // Serial.println(peak); //Print out what frequency is the most dominant.

  for(int i=0; i<(SAMPLES/2); i++)
  {
    /*View all these three lines in serial terminal to see which frequencies has which
    amplitudes*/

    Serial.println((i * 1.0 * SAMPLING_FREQUENCY) / SAMPLES, 1);
    Serial.print(" ");
    Serial.println(vReal[i], 1); //View only this line in serial plotter to visualize the bins

    rx_buf[rx_buf_num] = Serial.read();
    Serial.print(rx_buf[rx_buf_num]);

  }

  delay(1000); //Repeat the process every second OR:
  //while(1); //Run code once
}

```

---

## Apêndice 2

```
#define BLYNK_PRINT Serial

#define BLYNK_USE_DIRECT_CONNECT

#include <BlynkSimpleEsp32_BLE.h>
#include <BLEDevice.h>
#include <BLEServer.h>

#define ANALOG_PIN_0 36
float analog_value = 0.0;

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "a2d10ad60d6e441e922935d67a246274";
BlynkTimer timer;
int rowIndex = 0;

void setup()
{
  // Debug console
  Serial.begin(9600);

  Serial.println("Waiting for connections...");

  //clean table at start
  Blynk.virtualWrite(V2, "clr");

  Blynk.begin(auth);
}

void loop()
{
  Blynk.run();
  timer.run();
}
```

```

analog_value = analogRead(ANALOG_PIN_0); // recebe o valor do pino do ESP
float voltage_value = analog_value*(3.3/4096.0); /*(3.3/4096.0);
//Serial.println(voltage_value);

if (voltage_value > 0.08) // if para mostrar o valor em volts na tela
{
  Serial.println(voltage_value);
  Blynk.virtualWrite(V1,voltage_value);
  Blynk.virtualWrite(V2, "add",rowIndex, voltage_value, "Valores (V)",voltage_value);
  Blynk.virtualWrite(V2, "pick", rowIndex);
  rowIndex++;
  //Serial.println(rowIndex);

  if(rowIndex==8){
    Blynk.virtualWrite(V2, "clr");
    rowIndex=0;}
}

if(analog_value > 0) //acender led se bater no sensor
{
  Blynk.virtualWrite(V0,analog_value); // acende led se bater na rede
}

}

```

---



## Apêndice 3

```
#include <nRF5x_BLE_API.h> // chamando a biblioteca do nRF
#include "arduinoFFT.h"

#define DEVICE_NAME          "BLE_TCC"
#define TXRX_BUF_LEN        20
#define SAMPLES 128          //Must be a power of 2
#define SAMPLING_FREQUENCY 1000 //Hz, must be less than 10000 due to ADC

arduinoFFT FFT = arduinoFFT();

unsigned int sampling_period_us;
unsigned long microseconds;

double vReal[SAMPLES];
double vImag[SAMPLES];
const int PIEZO_PIN = A3; // Recebe informacao do pino A3 e coloca em PIEZO em valores

BLE          ble;
Timeout      timeout;
Ticker       ticker1s; // criando ticker para atualizar o valor a cada 1s
static uint8_t rx_buf[TXRX_BUF_LEN];
static uint8_t rx_buf_num;
static uint8_t rx_state=0;

// The uuid of service and characteristics
static const uint8_t service1_uuid[] = {0x71, 0x3D, 0, 0, 0x50, 0x3E, 0x4C, 0x75, 0xBA,
0x94, 0x31, 0x48, 0xF1, 0x8D, 0x94, 0x1E};
static const uint8_t service1_tx_uuid[] = {0x71, 0x3D, 0, 3, 0x50, 0x3E, 0x4C, 0x75, 0xBA,
0x94, 0x31, 0x48, 0xF1, 0x8D, 0x94, 0x1E};
static const uint8_t service1_rx_uuid[] = {0x71, 0x3D, 0, 2, 0x50, 0x3E, 0x4C, 0x75, 0xBA,
0x94, 0x31, 0x48, 0xF1, 0x8D, 0x94, 0x1E};
static const uint8_t uart_base_uuid_rev[] = {0x1E, 0x94, 0x8D, 0xF1, 0x48, 0x31, 0x94,
0xBA, 0x75, 0x4C, 0x3E, 0x50, 0, 0, 0x3D, 0x71};

uint8_t tx_value[TXRX_BUF_LEN] = {0,};
uint8_t rx_value[TXRX_BUF_LEN] = {0,};

// Initialize value of chars
GattCharacteristic characteristic1(service1_tx_uuid, tx_value, 1, TXRX_BUF_LEN,
GattCharacteristic::BLE_GATT_CHAR_PROPERTIES_WRITE |
GattCharacteristic::BLE_GATT_CHAR_PROPERTIES_WRITE_WITHOUT_RESPONSE );
GattCharacteristic characteristic2(service1_rx_uuid, rx_value, 1, TXRX_BUF_LEN,
GattCharacteristic::BLE_GATT_CHAR_PROPERTIES_NOTIFY);
GattCharacteristic *uartChars[] = {&characteristic1, &characteristic2};
GattService      uartService(service1_uuid, uartChars, sizeof(uartChars) /
sizeof(GattCharacteristic *));

void disconnectionCallBack(const Gap::DisconnectionCallbackParams_t *params) {
    Serial.println("Disconnected!");
    Serial.println("Restarting the advertising process");
    ble.startAdvertising();
}
```

```

void gattServerWriteCallBack(const GattWriteCallbackParams *Handler) {
  uint8_t buf[TXRX_BUF_LEN];
  uint8_t index;
  uint16_t bytesRead = TXRX_BUF_LEN;

  Serial.println("onDataWritten : ");
  if (Handler->handle == characteristic1.getValueAttribute().getHandle()) {
    ble.readCharacteristicValue(characteristic1.getValueAttribute().getHandle(), buf,
&bytesRead);
    Serial.print("bytesRead: ");
    Serial.println(bytesRead, HEX);
    for(index=0; index<bytesRead; index++) {
      Serial.write(buf[index]);
    }
    Serial.println("");
  }
}

void m_uart_rx_handle() { //update characteristic data
  ble.updateCharacteristicValue(characteristic2.getValueAttribute().getHandle(), rx_buf,
rx_buf_num);
  memset(rx_buf, 0x00,20);
  rx_state = 0;
}

void atualizaSensor() {
  // uint16_t valordosensor = analogRead(A2);
  // funcao converte valor
  static uint16_t valordosensor=0;
  // valordosensor++;
  // Serial.println(valordosensor);
  ble.updateCharacteristicValue(characteristic2.getValueAttribute().getHandle(), (uint8_t
*)&valordosensor,5); // atualizando a caracteristica do ble (lendo valor do sensor a cada 1s)
}

void uart_handle(uint32_t id, SerialIrq event) { /* Serial rx IRQ */
  if(event == RxIrq) {
    if(rx_state == 0) {
      rx_state = 1;
      timeout.attach_us(m_uart_rx_handle, 100000);
      rx_buf_num=0;
    }
  }
  while(Serial.available()) {

    Serial.println("Lendo buffer Rx: ");
    /* if(rx_buf_num < 20) {
      rx_buf[rx_buf_num] = Serial.read();

      Serial.print(rx_buf[rx_buf_num]);
      rx_buf_num++;
    }
  }
}

```

```

    else {
        Serial.read();
    } */
    loop();
}
}
}

void setup() {
    // put your setup code here, to run once
    Serial.begin(9600);
    Serial.attach(uart_handle);

    sampling_period_us = round(1000000*(1.0/SAMPLING_FREQUENCY));
}

//Inicializando o ticker
ticker1s.attach(atualizaSensor, 1);

//Definindo pino A2 como input
pinMode(A2,INPUT);

ble.init();
ble.onDisconnection(disconnectionCallBack);
ble.onDataWritten(gattServerWriteCallBack);

// setup adv_data and srp_data
ble.accumulateAdvertisingPayload(GapAdvertisingData::BREDR_NOT_SUPPORTED);
ble.accumulateAdvertisingPayload(GapAdvertisingData::SHORTENED_LOCAL_NAME,
    (const uint8_t *)"TXRX", sizeof("TXRX") - 1);

ble.accumulateAdvertisingPayload(GapAdvertisingData::COMPLETE_LIST_128BIT_SERVICES_IDS,
    (const uint8_t *)uart_base_uuid_rev, sizeof(uart_base_uuid_rev));

// set adv_type
ble.setAdvertisingType(GapAdvertisingParams::ADV_CONNECTABLE_UNDIRECTED);
// add service
ble.addService(uartService);
// set device name
ble.setDeviceName((const uint8_t *)"Simple Chat");
// set tx power, valid values are -40, -20, -16, -12, -8, -4, 0, 4
ble.setTxPower(4);
// set adv_interval, 100ms in multiples of 0.625ms.
ble.setAdvertisingInterval(160);
// set adv_timeout, in seconds
ble.setAdvertisingTimeout(0);
// start advertising
ble.startAdvertising();
Serial.println("Advertising Start!");
}

void loop() {
    ble.waitForEvent();

    /*SAMPLING*/

```

```

for(int i=0; i<SAMPLES; i++)
{
    microseconds = micros(); //Overflows after around 70 minutes!

    //int piezoADC = analogRead(PIEZO_PIN);
    // float piezoV = piezoADC/ 1023.0 * 3.3; //3.3V e a maxima voltagem do redBear

    vReal[i] = analogRead(PIEZO_PIN); // Joga o valor em Volts para ser devolvido em Hz
    // Serial.println(i);
    // Serial.println(vReal[i]);
    vImag[i] = 0;

    while(micros() < (microseconds + sampling_period_us)){
    }
}

/*FFT*/
FFT.Windowing(vReal, SAMPLES, FFT_WIN_TYP_HAMMING, FFT_FORWARD);
FFT.Compute(vReal, vImag, SAMPLES, FFT_FORWARD);
FFT.ComplexToMagnitude(vReal, vImag, SAMPLES);
double peak = FFT.MajorPeak(vReal, SAMPLES, SAMPLING_FREQUENCY);

/*PRINT RESULTS*/
// Serial.println(peak); //Print out what frequency is the most dominant.

for(int i=0; i<(SAMPLES/2); i++)
{
    /*View all these three lines in serial terminal to see which frequencies has which
amplitudes*/

    Serial.println((i * 1.0 * SAMPLING_FREQUENCY) / SAMPLES, 1);
    Serial.print(" ");
    Serial.println(vReal[i], 1); //View only this line in serial plotter to visualize the bins

    rx_buf[rx_buf_num] = Serial.read();
    Serial.print(rx_buf[rx_buf_num]);

}

    delay(1000); //Repeat the process every second OR:
//while(1); //Run code once
}

```

## Apêndice 4

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Geração da transformada de Fourier a partir dos gráficos que representam
```

```
% a batida forte sem e com massa respectivamente.
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
clc
```

```
clear
```

```
% Geração do eixo do tempo
```

```
tempo = [0:0.01:1];
```

```
% Tensão sem massa
```

```
tensaos = [0
```

```
0.082352941
```

```
0.164705882
```

```
0.247058824
```

```
0.329411765
```

```
0.411764706
```

```
0.494117647
```

```
0.576470588
```

```
0.658823529
```

```
0.741176471
```

```
0.823529412
```

```
0.905882353
```

0.988235294  
1.070588235  
1.152941176  
1.235294118  
1.317647059  
1.4  
1.2466665  
1.0933332  
0.9399999  
0.7866666  
0.6333333  
0.48  
0.49714286  
0.51428572  
0.53142858  
0.54857144  
0.5657143  
0.58285716  
0.60000002  
0.5  
0.4  
0.3  
0.2  
0.1  
0  
0.20625

0.4125  
0.61875  
0.825  
1.03125  
1.2375  
1.44375  
1.65  
1.85625  
2.0625  
2.26875  
2.475  
2.68125  
2.8875  
3.09375  
3.3  
3.135  
2.97  
2.805  
2.64  
2.475  
2.31  
2.145  
1.98  
1.815  
1.65  
1.485

1.32

1.155

0.99

0.825

0.66

0.495

0.33

0.165

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0



```
0  
0  
0  
0  
0  
0  
0  
0  
0  
0  
0  
0  
];
```

```
tensaos = tensaos';
```

```
figure(1)
```

```
plot(tempo,tensaos)
```

```
% Transformada da tensão sem massa
```

```
Y = fft(tensaos);
```

```
Y = fftshift(Y);
```

```
Y = abs(Y);
```

```
Fs = 100;
```

```
df = Fs/100;
```

```
freq = (0:df:Fs);
```

```
figure(2)
```

plot(freq,Y/Fs)

% Tensão com massa

tensaoc = [0

0.120428571

0.240857143

0.361285714

0.481714286

0.602142857

0.722571429

0.843

0.6744

0.5058

0.3372

0.1686

0

0.506

1.012

1.518

2.024

2.53

2.168571429

1.807142857

1.445714286

1.084285714

0.722857143

0.361428571

0

0.342857143

0.685714286

1.028571429

1.371428571

1.714285714

2.057142857

2.4

2.057142857

1.714285714

1.371428571

1.028571429

0.685714286

0.342857143

0

0

0

0

0

0

0

0

0

0

0





```
tensaoc = tensaoc';
```

```
figure(3)
```

```
plot(tempo,tensaoc)
```

```
% Transformada da tensão com massa
```

```
Yc = fft(tensaoc);
```

```
Yc = fftshift(Yc);
```

```
Yc = abs(Yc);
```

```
Fs = 100;
```

```
df = Fs/100;
```

```
freq = (0:df:Fs);
```

```
figure(4)
```

```
plot(freq,Yc/Fs)
```

```
% Obtenção das potências
```

```
pot = sum((Y/Fs).^2);
```

```
pot2 = sum((Yc/Fs).^2);
```