

CENTRO UNIVERSITÁRIO DO INSTITUTO MAUÁ DE TECNOLOGIA

Escola de Engenharia Mauá

Engenharia Eletrônica

CHRISTIAN KODAIRA GOÊS

LUCAS GASPAR VETTORAZZO

THALLES DEMÉTRIO FERREIRA SANTOS

YGOR PARADA FRAZÃO

Aplicativo de Navegação e telemetria para o HPA usando IoT

São Caetano do Sul

2022

CHRISTIAN KODAIRA GOÉS
LUCAS GASPAR VETTORAZZO
THALLES DEMÉTRIO FERREIRA SANTOS
YGOR PARADA FRAZÃO

Aplicativo de navegação e telemetria para o HPA usando IoT

Trabalho de Conclusão de Curso apresentado à Escola de Engenharia Mauá do Centro Universitário do Instituto Mauá de Tecnologia como requisito parcial para a obtenção do título de Engenheiro Eletrônico e Elétrico.

Orientador: Eng. Me. Fernando de Almeida Martins

Área de concentração: Engenharia Eletrônica.

São Caetano do Sul

2022

Góes, Christian

Aplicativo de navegação e telemetria para o HPA usando IoT. / Christian Kodaira Góes, Lucas Gaspar Vettorazzo, Thalles Demétrio Ferreira Santos, Ygor Parada Frazão. — São Caetano do Sul : CEUN-IMT, 2022.

100 p.

Trabalho de Conclusão de Curso – Escola de Engenharia Mauá do Centro Universitário do Instituto Mauá de Tecnologia, São Caetano do Sul, SP, 2022.

Orientador(a): Eng. Me. Fernando de Almeida Martins

1. IoT. 2. HPA. 3. Aplicativo mobile. 4. microcontrolador. I. Vettorazzo, Lucas. II. Santos, Thalles. III. Frazão, Ygor. IV. Instituto Mauá de Tecnologia. Escola de Engenharia. V. Título.

CHRISTIAN KODAIRA GOÉS
LUCAS GASPAR VETTORAZZO
THALLES DEMÉTRIO FERREIRA SANTOS
YGOR PARADA FRAZÃO

Aplicativo de navegação e telemetria para o HPA usando IoT

Trabalho de Conclusão de Curso aprovado pela Escola de Engenharia Mauá do Centro Universitário do Instituto Mauá de Tecnologia como requisito parcial para a obtenção do título de Engenheiro Eletrônico e Elétrico.

Banca avaliadora:

Engenheiro Mestre Fernando de Almeida Martins

Engenheiro Doutor Joseph Youssif Saab Junior

Engenheiro Doutor Bruno Galelli Chierregatti

São Caetano do Sul, 05 de Dezembro de 2022.

Dedicamos este trabalho a todos, em especial, aos nossos familiares e amigos que fizeram parte desta jornada.

AGRADECIMENTOS

Agradecemos, primeiramente, ao Centro de Pesquisas em específico a Divisão de Eletrônica e Telecomunicações do Instituto Mauá de Tecnologia e ao nosso orientador Me. Fernando de Almeida Martins que junto do Rogério Cassares Pires, sempre estiveram presentes para sanar dúvidas durante a evolução do projeto além do apoio no desenvolvimento do *software* e *hardware*.

Durante o desenvolvimento do projeto foi nos permitido conhecer e interagir com inúmeras pessoas dedicadas e inteligentes. Demonstramos nossa plena gratidão a cooperação do Dr. Joseph Youssif Saab Junior, na qual sua participação nos ajudou na criação de soluções.

Agradecemos também ao corpo docente do Instituto Mauá de Tecnologia, os quais nos deram a base provendo conhecimento e ajuda quando necessário além de fazerem parte de momentos memoráveis durante esses 5 anos de formação, em especial, aos mestres e doutores do tronco de elétrica e eletrônica.

Nossos sinceros agradecimentos aos nossos familiares e amigos que estiveram presentes nessa trajetória nos dando apoio, torcendo e vibrando com nossas conquistas.

“A estrada à frente é tão longa quanto você a fizer.

Faça a viagem valer a pena.”

Jon Bon Jovi.

RESUMO

O trabalho de conclusão de curso se propôs a coletar informações em tempo real da saúde do piloto do HPA (*Human Powered Aircraft*) como batimento cardíaco e glicemia, além de criar um sistema de navegação do usuário durante atividades físicas realizando aquisição de dados dos sinais vitais, desempenho e de navegação como posição, velocidade, altitude, potência e localização. Tais dados coletados foram transmitidos por protocolo *Bluetooth Low Energy* com microcontrolador ESP32 e essas informações foram exibidas por meio de um aplicativo *mobile* de autoria própria em forma de gráficos e mapas. A partir da aquisição, todos os dados foram concentrados no aplicativo *mobile* e enviados por meio de um protocolo de comunicação MQTT e representados em uma página Web.

Palavras-chaves: IoT; HPA; aplicativo mobile; microcontrolador.

ABSTRACT

The undergraduate final project proposes to collect real-time information on the health of the HPA (Human Powered Aircraft) pilot, such as heart rate and blood glucose besides creating an user navigation during physical activities, performing data acquisition of vital signs, performance and navigation such as position, speed, altitude, power and location, such data were transmitted by Bluetooth Low Energy's protocol with ESP32 microcontroller and this information were displayed through a mobile application of our own authorship in the form of graphs and maps. After the acquisition, all data were concentrated in the mobile application and sent by an MQTT communication protocol and presented on a web page.

Keywords: IoT; HPA; app mobile; microcontroller.

LISTA DE FIGURAS

Figura 1 – Arquitetura de dados de um sistema utilizando IoT	33
Figura 2 – Aplicativo Strava	35
Figura 3 – Aplicativo Zwift	35
Figura 4 – Aplicativo TrainingPeaks	36
Figura 5 – Demonstração da mecânica do HPA	37
Figura 6 – representação do GATT	40
Figura 7 – SPI comunicação	41
Figura 8 – Esquema do protocolo de comunicação I ² C	42
Figura 9 – Representação do Start-Bit e do Stop-Bit.....	42
Figura 10 – Esquema do protocolo de comunicação UART	43
Figura 11 – Exemplo da sequência de bits da comunicação UART	43
Figura 12 – Estrutura da publicação por MQTT.....	44
Figura 13 – Modos de utilização do NFC	45
Figura 14 – Exemplo de tubo de pitot de um avião	48
Figura 15 – Alterações na glicemia antes e depois do treino de musculação	51
Figura 16 – Método para calcular a frequência cardíaca	52
Figura 17 – Diagramação de blocos do Node-RED	55
Figura 18 – Exemplo de código HTML	56
Figura 19 – Exemplificação do sistema de API	58
Figura 20 – Editor de código.....	58
Figura 21 – Arduino IDE	59
Figura 22 – Exemplo de um circuito desenhado no Proteus.....	60
Figura 23 – HPA	60
Figura 24 – Pitch, Roll e Yaw.....	62
Figura 25 – Fluxograma de informações de sensores conectados por cabo ao ESP-32	64
Figura 26 – Fluxograma das informações de sensores BLE	65
Figura 27 – Diagrama dos sensores de navegação do celular.....	66
Figura 28 – Fluxograma dos dados a serem enviados para interface Web	67
Figura 29 – Cinta cardíaca	68
Figura 30 – Medidor de potência	69

Figura 31 – Dinamômetro	70
Figura 32 – Sensor de glicose	70
Figura 33 – Medidor de pressão relativa	71
Figura 34 – Sensor de ultrassom	72
Figura 35 – Leitor RFid	73
Figura 36 – Microcontrolador ESP32	73
Figura 37 – Esquema elétrico do sistema embarcado	75
Figura 38 – Captura de tela do aplicativo NRF-connection do sensor de batimento cardíaco	76
Figura 39 – Modelo 3D da PCB feito no Proteus	81
Figura 40 – PCB física com os componentes do sistema	81
Figura 41 – Apresentação 3D caixa Patola	82
Figura 42 – Vista superior do sistema	82
Figura 43 – Sistema final montado	83
Figura 44 – Página de Login e Home	83
Figura 45 – Página Home com conexão estabelecida e dashboards em tempo Real	84
Figura 46 – Tela de configuração, navegação e localização.	85
Figura 47 – Página Web dos dados recebidos	86
Figura 48 – Node-Red.....	87
Figura 49 – Protótipo montado para teste do sistema de aquisição de dados	88
Figura 50 – Em teste (EUREKA)	88
Figura 51 – Suporte do dispositivo móvel.....	89

LISTA DE TABELA

Tabela 1 – Comparação entre os microcontroladores Arduino UNO, Arduino MEGA 2560, ESP32 e ESP8266	53
Tabela 2 – Características Técnicas- iGPSPORT HR40	68
Tabela 3 – Características Técnicas do Dura-Ace 9000	69
Tabela 4 – Característica técnicas da trainer Elite Turbo Muin	70
Tabela 5 – Características Técnicas do sensor de glicemia	71
Tabela 6 – Características Técnicas do sensor de pressão relativa.....	71
Tabela 7 – Características Técnicas do sensor de ultrassom	72
Tabela 8 – Características Técnicas do leitor RFID/NFC	73
Tabela 9 – Características do ESP32	74

LISTA DE ABREVIATURAS E SIGLAS

API – *Application Programming Interface*

Bit – *Binary Digit*

BLE – *Bluetooth Low energy*

CGM – *Continuous glucose monitoring*

ECG – *Eletrocardiograma*

FGM – *flash glucose monitoring*

GATT – *Generic Attribute*

GPIO – *General Purpose Input/Output*

GPS – *Global Positioning System*

HPA – *Human Power Aircraft*

HTML – *HyperText Markup Language*

IDE – *Integrated Development Environment*

iOS – *iPhone Operating System*

IoT – *Internet of Things*

JSON – *JavaScript Object Notation*

MEMS – *Micro Electro Mechanical Systems*

MQTT – *Message Queuing Telemetry Transport*

NFC – *Near Field Communication*

P2P – *Peer-to-Peer*

PCB – *Printed Circuit Board*

RAM – *Random Access Memory*

RF – *Rádio Frequência*

ROM - *Read Only Memory*

SCL – *Serial Clock Line*

SDA – *Serial Data*

SPI – *Serial Peripheral Interface*

TWI – *Two Wire Interface*

UART – *Universal Asynchronous Receiver/Transmitter*

UTC – *Universal Time Coordinated*

UUID – *Universal Unique Identifier*

LISTA DE SÍMBOLOS

V – Diferença de Potencial medida em Volts.

P_{total} – Pressão absoluta medida em Pascal.

P_{estática} – Pressão estática medida em Pascal.

γ – Densidade do ar em kg/m³

mg – Miligrama.

dL - Decilitro.

bpm – Batimento por minuto.

h – Hora.

W – Watts.

s – Segundo.

rpm – Rotação Por Minuto.

kg – Quilograma.

kPA – Quilo Pascal.

mA – Miliampere.

kHz – Quilo-Hertz

μ A – Microampere.

mW - milliwatti

PR – Pressão Relativa medida em Pascal.

v – Velocidade em metros por segundo.

Ω – Ohm.

cm – Centímetro.

SUMÁRIO

1	INTRODUÇÃO	32
1.1	IOT SPORTS	32
1.2	IOT NA SAÚDE	33
1.3	CONHECENDO O MERCADO ATUAL DE APLICATIVOS FITNESS	34
1.3.1	STRAVA	34
1.3.2	ZWIFT	35
1.3.3	TRAININGPEAKS.....	36
1.4	HPA.....	36
1.5	OPORTUNIDADE E APLICAÇÃO	37
2	OBJETIVO	38
3	REVISÃO BIBLIOGRAFICA	39
3.1	DEFINIÇÕES, PROTOCOLOS E INTERFACES	39
3.1.1	BIT E BYTE	39
3.1.2	BLUETOOTH LOW ENERGY	39
3.1.3	SPI.....	40
3.1.4	I ² C	41
3.1.5	UART	43
3.1.6	MQTT	44
3.1.7	NFC	44
3.2	SENSORES DO SISTEMA DE NAVEGAÇÃO	45
3.2.1	GIROSCÓPIO	45
3.2.2	ACELERÔMETRO.....	46
3.2.3	MAGNETÔMETRO.....	46
3.2.4	BARÔMETRO	46
3.2.5	TUBO DE PITOT.....	47
3.2.6	MEDIDOR DE POTÊNCIA.....	48
3.2.7	DINAMÔMETRO	48
3.2.8	SENSOR ULTRASSÔNICO	49
3.3	SENSORES DE ACOMPANHAMENTO DO USUÁRIO.....	49
3.3.1	SENSOR DE GLICEMIA.....	49
3.3.2	SENSOR DE FREQUÊNCIA CARDÍACA	51
3.4	SISTEMA EMBARCADO.....	52
3.4.1	ESP32	52
3.5	APLICATIVO MOBILE	53
3.5.1	ANDROID	53
3.5.2	IOS	54
3.5.3	REACT NATIVE.....	54
3.5.4	NODE-RED	54
3.5.5	HTML	55
3.5.6	JAVASCRIPT	56
3.5.7	TYPESCRIPT	56
3.5.8	ANGULAR.....	56
3.6	API	57
3.6.1	JSON	57

3.6.2	TIMESTAMP.....	57
3.6.3	GEOLOCATION.....	57
3.7	AMBIENTE DE DESENVOLVIMENTO E FERRAMENTAS.....	58
3.7.1	VISUAL STUDIO CODE.....	58
3.7.2	ARDUINO IDE.....	59
3.7.3	GIT E GITHUB.....	59
3.7.4	PROTEUS.....	59
3.8	HUMAN POWERED AIRCRAFT.....	60
3.9	NAVEGAÇÃO.....	61
3.9.1	PITCH, ROLL E YAW.....	61
4	MATERIAIS E MÉTODOS.....	63
4.1	MÉTODO.....	63
4.1.1	FLUXOGRAMA DAS INFORMAÇÕES DO SISTEMA EMBARCADO.....	63
4.1.2	FLUXOGRAMA DAS INFORMAÇÕES OBTIDAS POR SENSORES COM COMUNICAÇÃO BLE.....	64
4.1.3	DIAGRAMA DAS INFORMAÇÕES DOS DADOS DE NAVEGAÇÃO.....	65
4.1.4	FLUXOGRAMA DE INFORMAÇÕES DA INTERFACE WEB.....	66
4.2	SENSORES.....	67
4.3	CARACTERÍSTICAS DOS SENSORES UTILIZADOS.....	67
4.3.1	CINTA CARDÍACA.....	68
4.3.2	MEDIDOR DE POTÊNCIA.....	69
4.3.3	DINAMÔMETRO.....	69
4.3.4	MEDIDOR DE GLICEMIA.....	70
4.3.5	SENSOR DE PRESSÃO RELATIVA.....	71
4.3.6	SENSOR DE ULTRASSOM.....	72
4.3.7	LEITOR RFid/NFC.....	72
4.4	ESP32.....	73
4.5	SOFTWARES UTILIZADOS.....	74
5	DESENVOLVIMENTO.....	75
5.1	HARDWARE.....	75
5.1.1	SISTEMA EMBARCADO.....	75
5.1.2	COMUNICAÇÃO BLE DO ESP32 COM OS SENSORES BLUETOOTH.....	76
5.1.3	AQUISIÇÃO DE DADOS DO SENSOR DE GLICOSE.....	77
5.1.4	AQUISIÇÃO DE DADOS DO SENSOR ULTRASSOM.....	77
5.1.5	AQUISIÇÃO DE DADOS DO SENSOR DE PRESSÃO RELATIVA.....	77
5.2	SOFTWARE DO APLICATIVO MÓVEL.....	78
5.2.1	PÁGINA HOME.....	79
5.2.2	PÁGINA NAVEGAÇÃO.....	79
5.2.3	PÁGINA LOCALIZAÇÃO.....	79
5.2.4	GRÁFICOS.....	79
5.2.5	COMUNICAÇÃO MQTT.....	80
5.3	SERVIÇO WEB.....	80
6	RESULTADOS E DISCUSSÃO.....	81
6.1	<i>HARDWARE.....</i>	<i>81</i>
6.2	<i>APLICATIVO MOBILE.....</i>	<i>83</i>
6.2.1	NODE-RED.....	86
6.3	IMPLEMENTAÇÃO E TESTE DO PROJETO.....	87

6.4	INTEGRAÇÃO DO SISTEMA AO HPA	89
7	CONCLUSÕES.....	90
8	SUGESTÃO DE CONTINUAÇÃO	91
9	REFERÊNCIAS.....	92

1 INTRODUÇÃO

Nos últimos anos, a utilização da Internet das Coisas (IoT) vem inovando a maneira de dispositivos se conectarem entre si. Esses dispositivos podem ser desde equipamentos do dia a dia até instrumentos industriais complexos, em que todas as informações coletadas são enviadas para a rede de Internet, a fim de melhorar a tomada de decisões. Além disso, o custo de incorporar sensores e objetos de pequeno a grande porte a uma interface de usuário com conectividade e processamento de dados diminuiu consideravelmente, possibilitando um maior acesso a todo tipo de público e, conseqüentemente, se popularizando.

Um interessante exemplo seriam os dispositivos inteligentes para automatizar residências, como a Alexa e o Google Home, que estão ajudando as pessoas a viver e a trabalhar de forma mais eficiente. Além disso, a tecnologia IoT se tornou essencial para os negócios, pois fornece às empresas uma visão em tempo real de como os seus sistemas realmente se comportam, contribuindo com informações sobre tudo, desde o desempenho das máquinas até a cadeia de suprimentos e as operações de logística (SAP, 2021).

Dentre os benefícios quando se utiliza IoT, destaca-se as empresas que automatizam processos e diminuem os custos trabalhistas, além de reduzir o desperdício e melhorar a prestação de serviços, tornando mais barata a fabricação e entrega de mercadorias ganhando força à medida que mais empresas perceberem o potencial dos dispositivos conectados (RED HAT, 2019).

1.1 IOT SPORTS

Dentro do mundo esportivo inúmeras soluções IoT são utilizadas a fim de realizar medidas preditivas e de desempenho cruciais para o cenário competitivo, sendo possível comparar cada atleta com o melhor dele e assim traçar um plano de ação individual. Os dados provenientes dos sensores podem otimizar a evolução do atleta fazendo o esporte alcançar um novo nível de inteligência, estatística e tecnologia.

Na seleção de vôlei, por exemplo, os atletas utilizam uma faixa na cintura que marca a quantidade e a altura dos saltos e a partir de um conjunto de sensores que fornecem dados da partida é possível auxiliar a comissão técnica no volume e no direcionamento dos treinamentos (KNOPLUCH CAROL, 2017).

O uso de IoT no esporte não se estende exclusivamente ao mundo competitivo. A Sony, por exemplo, criou o Advagym que digitaliza a experiência da academia. Essa aplicação possui três diferentes usuários: o cliente, o *personal trainer* e a academia. Com isso, o cliente publica a sua performance de treino e o *personal trainer* pode criar exercícios personalizados para ele a distância. Do ponto de vista da academia, os dados podem analisados para melhorar a própria infraestrutura baseada no uso dos clientes. O diagrama explicado anteriormente é representado pela Figura 1.(HÅKANSSON, 2019).

Figura 1 – Arquitetura de dados de um sistema utilizando IoT

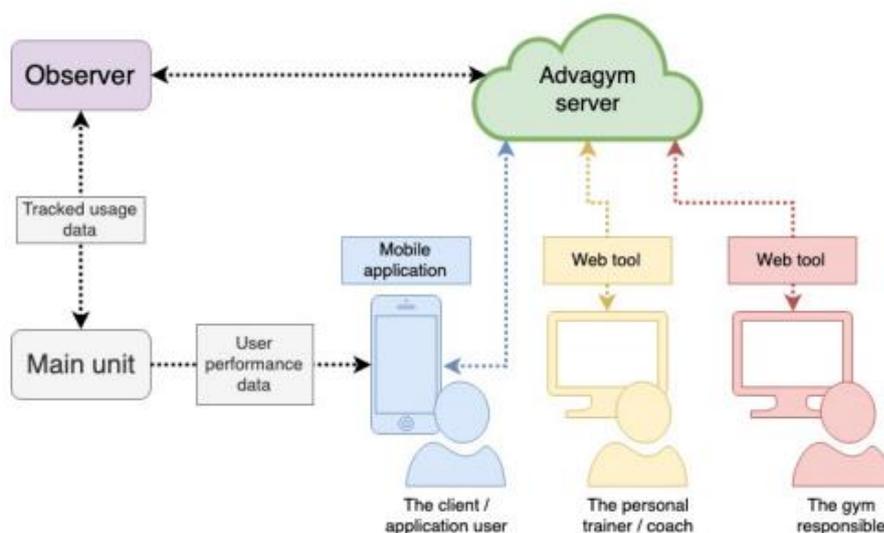


Figure 2, illustration of Advagym's IoT data flow to different users.

Fonte:(HÅKANSSON, 2019)

Outro marco interessante do uso da análise de dados no esporte foi o ocorrido no baseball, após Billy Beane e Peter Brand começarem a realizar a contratação e a mudança de posições de jogadores baseado em estatísticas, melhorando o desempenho do time e popularizando as estatísticas no mundo esportivo. A história até se tornou um filme chamado "*Moneyball: the Art of Winning an Unfair Game*" (MILLER BENNETT, 2012).

1.2 IOT NA SAÚDE

O uso de IoT no esporte também pode contribuir para a detecção de lesões e de outros problemas de saúde, como problemas cardíacos e de hipoglicemia. Os testes de fadiga e de direção, por exemplo, revelam alguns indicativos que predizem uma lesão por meio de dados provenientes

de um sistema IoT, podendo assim realizar o afastamento do atleta antes que algum problema ocorra (SANTIAGO PAULO, 2019).

No Brasil, cerca de 13 milhões brasileiros possuem diabetes e diante dessa situação o monitoramento do nível de glicose no sangue se mostrou uma aplicação interessante e necessária para a saúde. O OpenAPS, por exemplo, é uma aplicação projetada para e manter o nível glicose estável em ambientes hospitalares (GHORAYEB NABIL, 2013).

Já no cotidiano o controle da glicose é essencial principalmente durante o exercício, em que a quantidade no sangue pode diminuir bruscamente. Os sensores de glicose como os fabricados pela marca FreeStyle permitem a medição e o acompanhamento a qualquer momento através da aproximação de um *smartphone* com leitor NFC ao sensor.

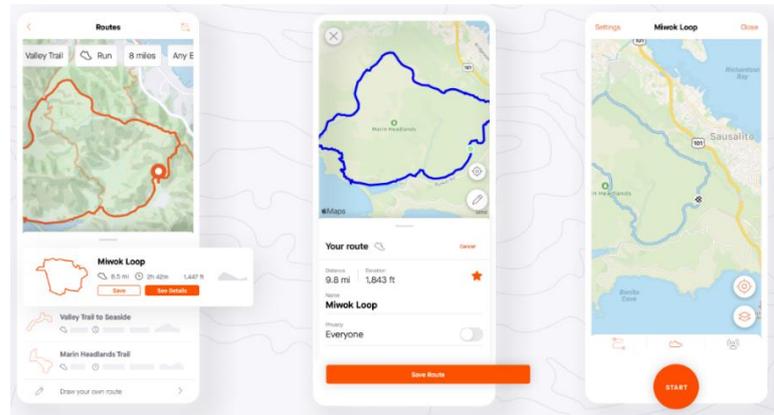
1.3 CONHECENDO O MERCADO ATUAL DE APLICATIVOS FITNESS

Observando o cenário atual, 2022, há alguns aplicativos fitness com diferentes aplicações e modelos de negócio, cada um deles voltados a uma experiência diferente ao consumidor por apresentar características principais distintas.

1.3.1 *Strava*

O Strava é um serviço americano que realiza o rastreamento de exercícios físicos feitos tanto por ciclistas, como por corredores (Figura 2). Nesse serviço os usuários podem publicar as suas performances em uma rota específica e compará-las com a de outras pessoas. O aplicativo monitora dados como distância percorrida, velocidade média, máxima e mínima, rota no mapa e até os batimentos cardíacos dependendo dos sensores que o usuário dispõe ou da atividade que ele seleciona.

Figura 2 – Aplicativo Strava



Fonte: (STRAVA, 2021)

Por ser uma rede social, o Strava também permite a conexão com amigos e novas pessoas, podendo trocar dicas de treinos e se desafiarem, estimulando o desenvolvimento pessoal e do esporte.

1.3.2 Zwift

O Zwift é um aplicativo de lazer *multiplayer online* para ciclismo e corrida em que os usuários podem interagir e competir. O aplicativo tem a proposta de realizar um *Gamefication* da prática desses exercícios através do uso de sensores para medir a potência e a velocidade, exemplificado na Figura 3, promovendo uma experiência ao usuário mais iterativa, o qual pode escolher rotas de ciclismo de diferentes partes do mundo.

Figura 3 – Aplicativo Zwift



Fonte:(ZWIFT, 2021)

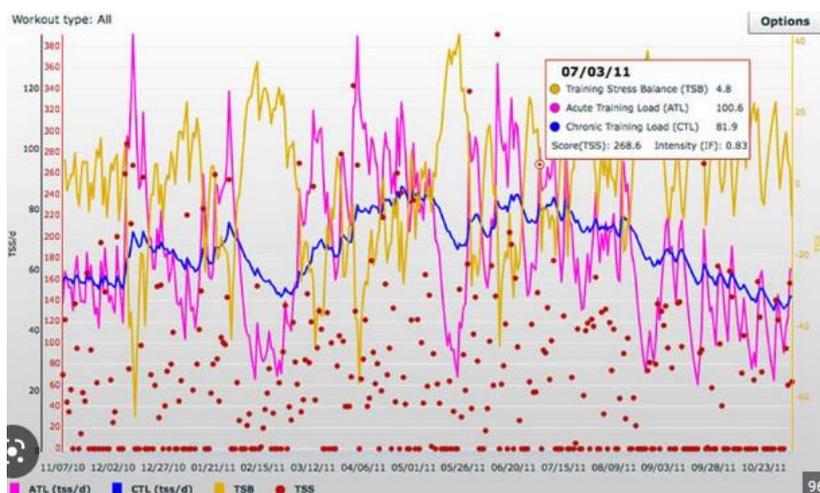
Com uma experiência do usuário é muito mais imersiva do que seria apenas pedalando em casa, assim o consumidor é estimulado a realizar a prática de exercícios físicos.

1.3.3 TrainingPeaks

O aplicativo TrainingPeaks tem funcionalidades que não se estendem somente a corrida e ciclismo; abrange diferentes esportes, como triátlon, natação e até esportes de inverno, a princípio utilizado por atletas mais avançados. Apesar dessas diferenciadas modalidades, também é acessível para iniciantes.

Dentre as funcionalidades, destaca-se a criação de treinos e a sua performance detalhada em gráficos e relatórios, *Figura 4*, com uma amostragem impressionante de dados. O aplicativo calcula a fadiga, o estresse e o esforço, parâmetros essenciais para que um especialista consiga personalizar os treinos dos usuários com precisão.

Figura 4 – Aplicativo TrainingPeaks



Fonte:(TRAININGPEAKS, 2020)

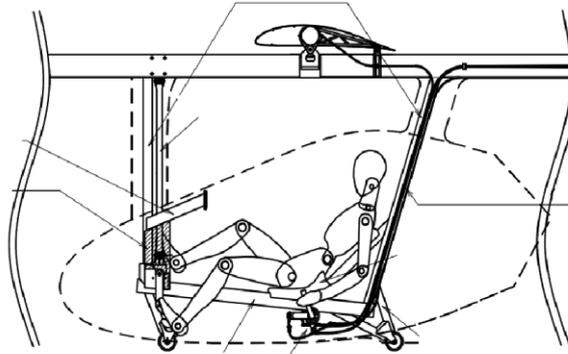
O TrainingPeaks possui uma abordagem única no que diz respeito a construção e apresentação de gráficos e relatórios algo que possibilita o alcance de um novo nível competitivo e de performance.

1.4 HPA

O HPA, *Human Powered Aircraft*, é definido como uma aeronave de propulsão humana, ou seja, apenas com a potência introduzida pelo piloto é possível realizar o voo. A aeronave possui mecânica semelhante à de uma bicicleta (Figura 5) isto é, a potência gerada deve ser capaz de

sustentar o voo, a partir da compensação da relação entre peso e potência, sem o auxílio de energias externas ou armazenadas (JEX; MITCHELL, 1982).

Figura 5 – Demonstração da mecânica do HPA



Fonte:(NEXTZPLUS, 2011)

1.5 OPORTUNIDADE E APLICAÇÃO

A grande oportunidade da pesquisa é a criação de um aplicativo que contenha a apresentação de dados do desempenho físico do usuário em conjunto com parâmetros do sistema de navegação da aeronave, permitindo que o piloto do HPA e a sua equipe obtenham um acompanhamento em tempo real.

A telemetria é essencial para medir e verificar as condições do usuário da aeronave. Parâmetros como potência e velocidade mínimas são de suma importância para manter a aeronave em condições ideais de voo.

O tratamento e apresentação de dados relativos ao sistema de navegação do HPA como informações sobre posição, velocidade, inclinação, altitude, potência e localização que contribuirão para monitorar a movimentação e o dinamismo do atleta e sua equipe.

2 OBJETIVO

O objetivo principal deste trabalho de conclusão de curso foi coletar dados de desempenho físico do piloto do HPA através de sensores, de potência, cardíaco e de glicemia, apresentando-os em tempo real, além adquirir informações do HPA a partir de sensores como acelerômetro, giroscópio, GPS, magnetômetro e barômetro. Estes dados foram exibidos por meio de um aplicativo *mobile* e uma interface Web de autoria própria representados em forma numérica e gráfica.

3 REVISÃO BIBLIOGRAFICA

3.1 DEFINIÇÕES, PROTOCOLOS E INTERFACES

As seguintes definições, protocolos e interfaces foram utilizados no decorrer do projeto:

3.1.1 *Bit e Byte*

Bit é a menor unidade de informação que pode ser armazenada ou transmitida. É geralmente usada na computação, eletrônica e teoria da informação. Um bit pode assumir somente 2 valores: 0 ou 1.

Já um *byte* é definido como sendo 8 *bits* e é normalmente usado para especificar o tamanho ou a quantidade de memória ou a capacidade de armazenamento de um dispositivo. Em toda comunicação digital há a utilização deste conceito para mensurar o tamanho da mensagem durante as comunicações digitais (GOGONI RONALDO, 2019).

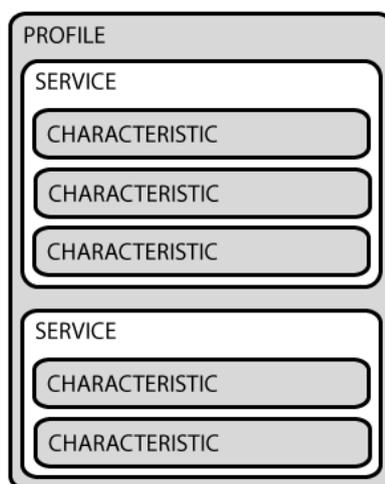
3.1.2 *Bluetooth Low Energy*

O BLE, *Bluetooth Low Energy*, também conhecido como *Bluetooth Smart* é um protocolo de rede sem fio desenvolvido para dispositivos que necessitam de um baixo consumo de energia, como equipamentos médicos, sensores industriais, comunicadores e dispositivos inteligentes; vital em aplicações que utilizam pequenas quantidades de dados periódicos, além de somente funcionar quando a conexão é iniciada, permanecendo com a operação suspensa enquanto não há pareamento, resultando em um baixo consumo de energia.

O protocolo trabalha com diferentes topologias de rede como a *Peer-to-Peer* (P2P), que é a comunicação direta entre dois dispositivos, *server* e o *client*. O *server* anuncia a outros dispositivos que está disponível para comunicação e detém os dados que devem ser lidos. Por outro lado, o *client* é quem faz a conexão com o dispositivo e recebe os dados. Já a rede *mesh* é a união e a comunicação de diversos aparelhos com o intuito de trocar informação. Nessa topologia, o dispositivo *master* pode se conectar com diversos dispositivos *slave* simultaneamente em consequência do protocolo usar 48 *bits* de endereçamento.

Em relação ao funcionamento da tecnologia BLE, o GATT (*Generic Attribute*) define a estrutura hierárquica do comportamento do BLE, isto é, a maneira como os dispositivos transferem os dados. Ele utiliza dos conceitos de *service* e *characteristic*, ambos são descritos por um UUID (*Universal Unique Identifier*), que é um número exclusivo de até 128 bits. O *service* é o ponto de hierarquia mais alto reunindo um conjunto de informações, já o outro conceito, *characteristic*, sempre deve pertencer a um *servisse*, Figura 6, o qual localiza o dado real que deve ser transmitido. (WISINTAINER, 2018)

Figura 6 – representação do GATT



Fonte:(TOWNSEND, 2014)

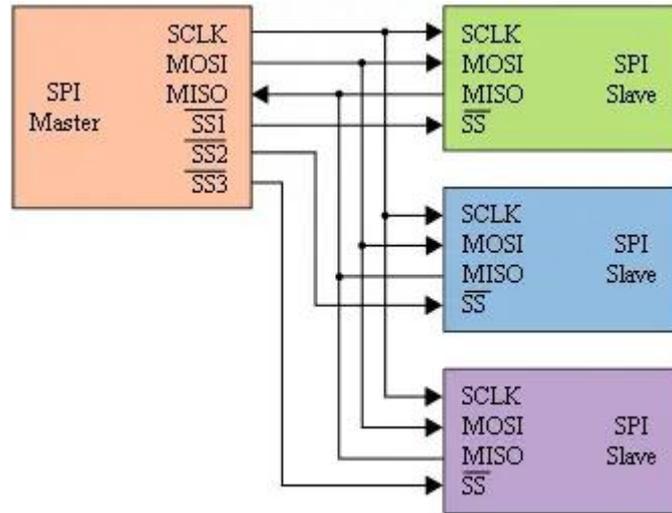
3.1.3 SPI

A interface SPI (*Serial Peripheral Interface*) é um protocolo de comunicação serial síncrona desenvolvido pela Motorola, e é amplamente utilizada para conexões entre sensores, microcontroladores e telas. O modo de comunicação é o *full-duplex* que, em outras palavras, possibilita enviar e receber dados de forma simultânea. O princípio básico desse comunicador é o *Shift-Register*, que usa um registrador paralelo para converter os dados em sinais seriais usando um sinal de *clock* para o sincronismo. A interface de comunicação SPI permite a conexão de múltiplos *slaves* a um único *master*, que é o responsável por gerar o sinal de *clock*.

Esse protocolo também é chamado por barramento de quatro fios que representam, respectivamente, a conexão do *master* para o *slave*, a conexão do *slave* para o *master*, o sinal de *clock* e a seleção de *slave*, que tem a função parecida com o *chip select*, sendo responsável

por selecionar os inúmeros *slaves* dentro da rede criada pela interface SPI, exemplificado na *Figura 7*.

Figura 7 – SPI comunicação



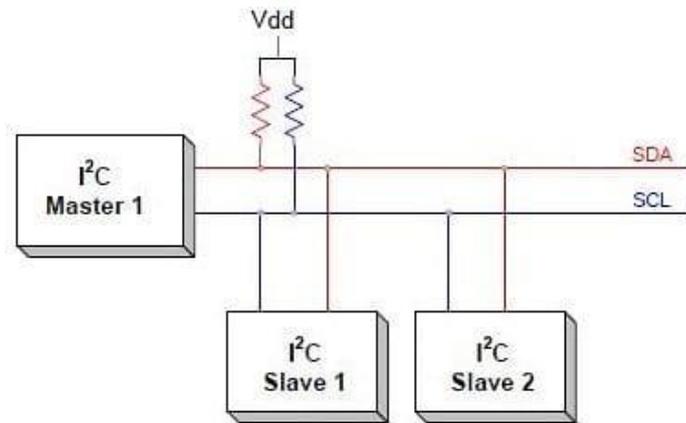
Fonte:(SACCO FRANCESCO, 2015)

3.1.4 I²C

A interface I²C é um barramento de comunicação serial síncrono desenvolvido pela Philips Semiconductor, também conhecido como TWI, *Two Wire Interface*, que utiliza o padrão *master* (Unidade que coordena os periféricos) e *slave* (Figura 8). Cada um desses escravos tem um endereço individual de 7 *bits* que permite identificação para que o mestre se comunique de forma individual com cada um deles.(TEIXEIRA EVANDRO, 2017)

A comunicação é feita a partir de dois fios, o SDA (*Serial Data*) e o SCL (*Serial Clock Line*). O SDA é um sinal bidirecional que permite a troca de informações entre o mestre e o escravo de forma não simultânea, já o SCL é um sinal de *clock* gerado pelo mestre a fim de garantir o sincronismo do barramento. Ambos os sinais fazem uso da interface de dreno aberto que significa a existência de um transistor para chavear o sinal em 0 volt (GND). Para chavear o sinal em 1 volt (Vdd), utiliza-se a resistor de *pull-up*. Esse método garante que não tenha curta circuito durante um possível erro no protocolo (SACCO FRANCESCO, 2015).

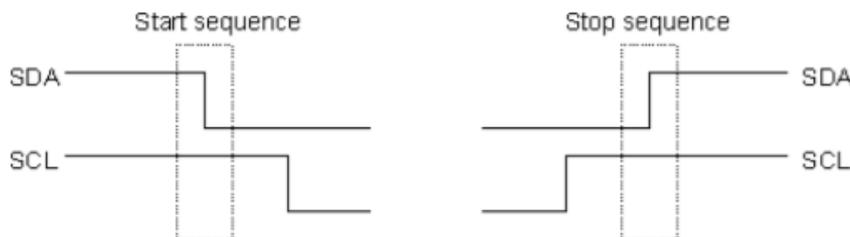
Figura 8 – Esquema do protocolo de comunicação I²C



Fonte : (TEIXEIRA EVANDRO, 2017)

O início da comunicação por meio da interface I²C, *start-bit*, acontece quando o SCL está em nível lógico alto e o SDA tem uma borda de descida; o encerramento da comunicação acontece quando o SCL está em nível lógico alto e o outro sinal tem uma borda de subida, conforme a Figura 9 (TEIXEIRA EVANDRO, 2017).

Figura 9 – Representação do Start-Bit e do Stop-Bit



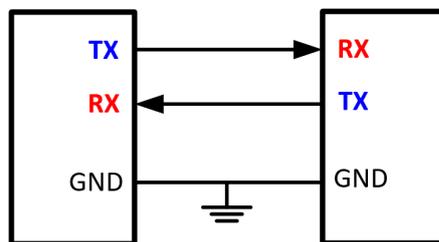
Fonte: (CAMARA ROMULO, 2022)

Após a sequência de inicialização demonstrada acima, o mestre escreve o endereço do dispositivo com o qual deseja se comunicar enquanto o sinal de *clock* fica oscilando de acordo com a frequência estabelecida. Depois que endereço foi escrito, é necessário enviar um *bit* a fim de informar se é escrita, 0, ou leitura, 1. A seguir, a informação deve ser enviada de acordo com o estipulado, estabelecendo a comunicação (GUIMARÃES FÁBIO, 2018).

3.1.5 UART

A interface UART (*Universal Asynchronous Receiver/Transmitter*) é um protocolo de comunicação que envolve um receptor e um transmissor universal que acontece de forma assíncrona. Essa interface trabalha de forma *full-duplex* como representado na Figura 10 permitindo a comunicação simultânea entre os dispositivos pois os dados trafegam em meios diferentes. (REZENDE RONIERE, 2019)

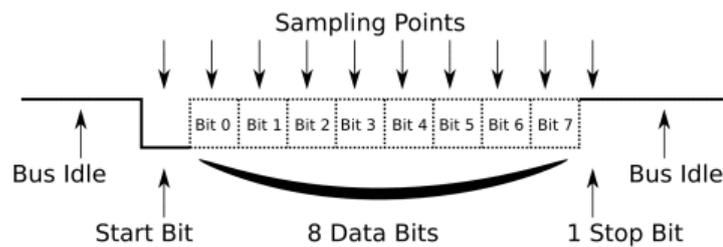
Figura 10 – Esquema do protocolo de comunicação UART



Fonte: (REZENDE RONIERE, 2019)

Como se trata de uma comunicação assíncrona, é necessário que os dispositivos conectados na rede tenham a mesma taxa de transmissão de dados, denominado de *baud rate*. Quando não há transmissão de dados, os dois canais permanecem em nível lógico alto e para a iniciar a comunicação é necessário que o *start-bit* ocorra com a mudança do estado lógico. Logo em seguida, é enviado um *byte* de informação com sequência do menos para o mais significativo. Para encerrar a comunicação deve ser enviado um *stop-bit* de nível lógico alto. Um exemplo de da sequência de *bit* está representado na Figura 11. (MENDONÇA HÉLIO, 2018).

Figura 11 – Exemplo da sequência de bits da comunicação UART

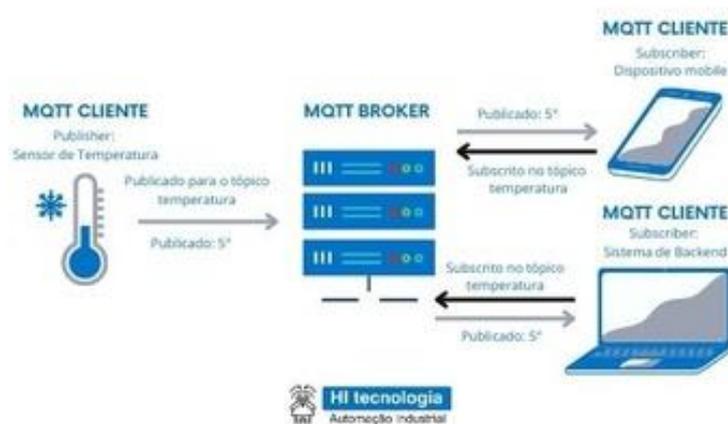


Fonte: (MENDONÇA HÉLIO, 2018)

3.1.6 MQTT

O protocolo de comunicação MQTT (*Message Queuing Telemetry Transport*), vem se tornando um dos principais protocolos para a implementações em IoT, Figura 12, visto que ele foi projetado para estabelecer uma comunicação assíncrona para que um dispositivo possa enviar ou receber apenas os dados solicitados e não sobrecarregar o servidor com diversas requisições (FERNANDES, 2021).

Figura 12 – Estrutura da publicação por MQTT



Fonte:(FERNANDES, 2021)

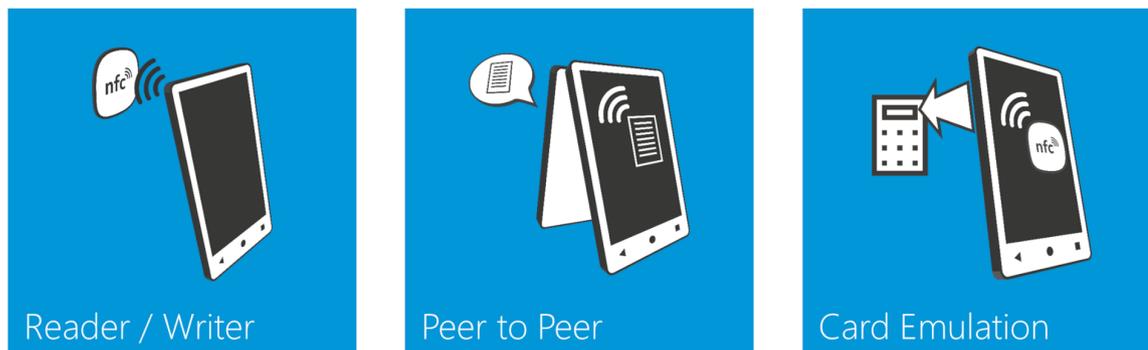
As funções de *subscribe* e *publish* permitem que dispositivos possam inscrever-se e/ou publicarem em um servidor de MQTT, denominado Broker MQTT. Por essa razão, quando um dispositivo publica as informações em um tópico, conectado em um determinado Broker, os dispositivos conectados a esse Broker e inscritos em um mesmo tópico recebem todas as informações publicadas. Dessa forma, a maior vantagem deste protocolo é a economia de banda de Internet.

3.1.7 NFC

O NFC (*Near Field Communication*), significa uma comunicação por proximidade de campo e é uma tecnologia que permite a troca de dados sem fios por meio de campos magnético. O seu funcionamento consta de um sistema de comunicação que inclui duas partes separadas: um *chip* leitor de NFC e uma *tag* NFC. O *chip* leitor fornece energia e envia comandos NFC para a *tag*, que processa a informação antes de enviar uma resposta específica.

Quando o NFC é combinado com um microcontrolador que possui suporte a múltiplos protocolos e recursos de radiofrequência (RF), ele pode ser utilizado em três diferentes modos: *Read/Write*, *Peer-to-Peer* e *Card Emulation* (Figura 13).(STMICROELECTRONICS, 2021).

Figura 13 – Modos de utilização do NFC



Fonte:(JAKL, 2018)

- *Reader/Write*: o modo leitor/gravador trata da comunicação de um leitor habilitado para NFC com uma *tag* NFC com o fim de ler ou gravar dados da ou para a *tag*.
- *Peer-to-Peer*: o modo *Peer-to-Peer* permite que dois dispositivos habilitados para NFC troquem informações como um contato, uma mensagem de texto ou quaisquer outros tipos de dados.
- *Card Emulation*: a informação armazenada pode ser lida como se fosse um *tag* NFC

3.2 SENSORES DO SISTEMA DE NAVEGAÇÃO

Abaixo foram descritos os sensores referentes ao sistema de navegação do HPA.

3.2.1 Giroscópio

O giroscópio é um aparato utilizado para medir e corrigir a direção de determinado dispositivo, sendo capaz de ser esboçado pelos fundamentos do momento angular, que mede a quantidade de rotação que um objeto recebe em relação a sua massa e a sua forma.

Os giroscópios são normalmente arquitetados com outras tecnologias. Um exemplo é o giroscópio MEMS (*Micro Electro Mechanical Systems*), que está sendo usado neste trabalho, e tem como base a tecnologia de estruturas vibrantes e a mecânica quântica (PASSARO et al., 2017).

3.2.2 Acelerômetro

O acelerômetro é um sensor eletromecânico utilizado para mensurar as forças que possuem a mesma direção que a aceleração do corpo estudado causada pela gravidade, pelo movimento ou pela vibração.

Basicamente o acelerômetro é um sensor capacitivo, que possui duas microestruturas próximas fazendo com que exista uma capacitância entre elas, que muda ao ser aplicada uma força à estrutura. Esse dispositivo é aplicado em inúmeros campos de estudo, como detecção de terremotos, vibrações em motores, observação do comportamento de animais e para manipulação de celulares (BARATO, 2014).

3.2.3 Magnetômetro

O magnetômetro é um dispositivo com foco em medir forças e direções de campos magnéticos, como a medição do campo magnético da Terra. Um exemplo de aplicação é o uso do magnetômetro para a segurança, como em detectores de metais nos aeroportos. (GRACE, 2018)

Ultimamente, através da tecnologia os magnetômetros estão diminuindo o seu tamanho de forma a serem incorporados a circuitos integrados e, ao mesmo tempo, tem diminuído o seu preço. Dessa maneira podem ser utilizados como bússolas eletrônicas em inúmeras aplicações, como sistema de navegação de drones ou em *smartphones*. (RIPKA, 2007)

3.2.4 Barômetro

Os sensores de pressão barométrica são por sua vez responsáveis pela medição de pressão, em geral é aplicado a líquidos ou gases, gerando um sinal elétrico como resposta a uma pressão aplicada sobre ele. (BARATO, 2014)

Esses sensores são utilizados indiretamente em diversas aplicações, realizando a medição de alguma grandeza dependente diretamente da pressão, por exemplo a altitude, com uma utilidade direta na aviação.

3.2.5 *Tubo de Pitot*

É um instrumento de medição criado por Henri Pitot usado para aferir medida de vazão em fluidos através da velocidade, é constituído por um tubo com uma abertura em sua extremidade que é colocada na direção oposta ao fluxo do fluido. Geralmente o tubo de Pitot é acoplado a um transmissor diferencial de pressão que fornecerá um sinal referente à diferença de pressão e a partir da equação de Bernoulli é possível obter a velocidade, equação 1.

Equação 1 – Equação da velocidade baseada na equação de Bernoulli

$$v = \sqrt{\frac{2(P_{total} - P_{estática})}{\gamma}}$$

Seu funcionamento em aeronaves acontece da seguinte forma, a pressão que entra pelo orifício do tubo é a pressão atmosférica estática, Figura 14, quando o avião está parado, já quando o avião está em movimento, a pressão dinâmica é admitida no orifício, aumentando a diferença de pressão evidenciando assim uma velocidade diferente de zero.(CULTURA AERONÁUTICA, 2011)

Figura 14 – Exemplo de tubo de pitot de um avião



Fonte: (CULTURA AERONÁUTICA, 2011)

3.2.6 Medidor de potência

Os medidores de potência, em geral, possuem pequenos extensômetros que medem a mínima deformação de uma peça, como o pedivela, cubo e ou pedal, essa deformação ocorre quando o ciclista pedala, então o medidor é capaz de traduzir esse valor em torque, que multiplicado pela cadência (rotação), resulta em Watts.(VARGAS GABRIEL, 2018)

3.2.7 Dinamômetro

O dinamômetro é um equipamento capaz de medir torque e potência, esse sensor é composto por um eixo de um motor comum de corrente contínua conectado a uma carga variável, amperímetro e um voltímetro. Quando o motor gira ele funciona como gerador comum ou dínamo, quando opera sem carga, o dínamo gera uma certa tensão e nenhuma corrente.

Nessa condição a potência transferida pelo sistema mecânico ao eixo do motor é nula e ele pode girar praticamente sem esforço algum, porém quando conectado a uma carga é exigido uma circulação de uma corrente, ou seja, o motor que funciona como dínamo deve gerar energia para alimentar esta carga.(INSTITUTO NEWTON C. BRAGA, 2012)

Quanto mais corrente exige a carga, menor é a sua resistência, por seguinte mais força é necessária para girar o eixo do motor. O dinamômetro é composto também por um disco estroboscópico, que possibilita a medida de velocidade de rotação.

3.2.8 Sensor Ultrassônico

É um dispositivo muito utilizado na indústria para medição de distância, tem como funcionamento a emissão de uma onda sonora de alta frequência e a medição do tempo levado para a recepção do eco, produzido quando a onda se choca com um objeto capaz de refletir o som. Para realizar o cálculo da distância, baseado no tempo de resposta e velocidade do som é utilizado a equação 2.

Equação 2 – Cálculo da distância utilizando o ultrassom.

$$\text{Distância} = (\text{Tempo de echo} * \text{Velocidade do som})/2$$

O sensor ultrassônico apresenta uma detecção confiável e sem contato, independente da textura, cor ou formato do objeto a ser identificado, o que inclui itens transparentes. A presença de névoa, poeira ou sujeira também não influencia no resultado da detecção.(BERTULUCCI, 2016)

3.3 SENSORES DE ACOMPANHAMENTO DO USUÁRIO

Abaixo está a descrição dos sensores referentes a saúde do usuário.

3.3.1 Sensor de glicemia

Os sensores de glicemia adquirem os seus dados através do líquido intersticial, solução aquosa e transparente encontrada entre as células, que informa o nível de glicose. Existem dois tipos de sensores os FGM (Monitoramento de glicose instantânea) e os CGM (Monitoramento de glicose contínua), sendo que na primeira a glicemia é medida a cada minuto, porém os dados só são apresentados quando o leitor é aproximado do sensor, já o CGM está ligado a um transmissor que envia continuamente os dados para um dispositivo.(BARROS MÁRIO, 2018)

3.3.1.1 DIABETES

A Diabetes mellitus, conhecida popularmente por diabetes, a qual pode ser acompanhada por sensores de glicemia é uma doença crônica causada pela produção insuficiente de insulina ou pela não resposta do organismo em relação à insulina que é o hormônio responsável por

transformar a glicose no sangue em energia para o organismo, a principal forma de verificar se alguma pessoa está com diabetes é medindo a quantidade de glicose no sangue.

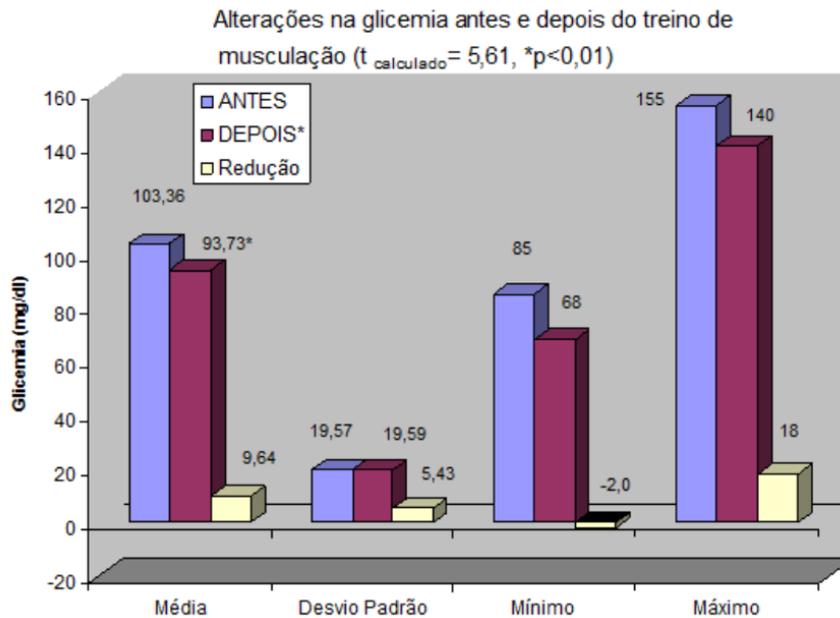
Ao fazer o exame de glicemia em jejum, o valor referência de diabetes, segundo a Cerpe é de igual ou superior a 126 mg/dL, outro valor de referência é de 200 mg/dL após a refeição.

Para sua prevenção, uma boa prática seria atividades físicas, pois o trabalho muscular consome glicose mesmo com a insulina baixa. O organismo, quando entra em exercício, pode aumentar de 7 a 20 vezes a gasto da glicose durante a hora inicial dependendo da dinâmica do exercício. (GALAN, 2021)

Portanto, atividades físicas ampliam a corrente sanguínea muscular, fazendo a insulina agir mais rapidamente e gerando um aumento da absorção de glicose. Porém, é preciso atentar-se a intensidade e duração do exercício, pois exercícios de alta intensidade fazem aumentar a produção de adrenalina, na qual faz a quebra do glicogênio, o que gera o aumento da glicemia no sangue. (PORTLANDCLINIC, 2015)

A Universidade de Fortaleza fez uma análise sobre a influência dos exercícios de musculação nos níveis de glicemia de indivíduos de 3ª idade. A metodologia da pesquisa teve como objetivo medir a glicemia de um grupo de idosos já praticantes de musculação antes e depois da atividade física. Esse exercício era composto de 6 sessões com 2 minutos de descanso cada e com duração total de aproximadamente 50 minutos. (LIMA; PINHEIRO, 2006)

Figura 15 – Alterações na glicemia antes e depois do treino de musculação



Fonte: (LIMA; PINHEIRO, 2006)

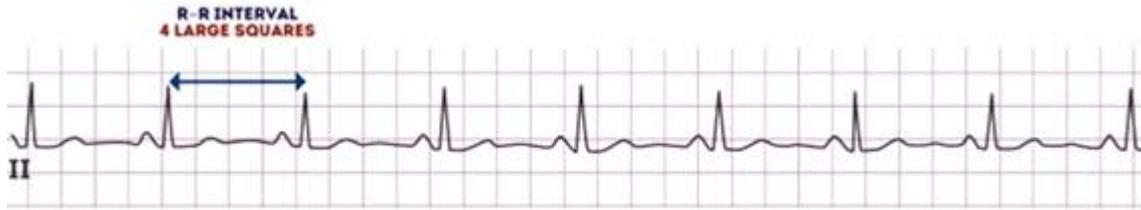
A *Figura 15* ilustra a comparação das medições antes e após o exercício. Em resumo, foi demonstrado que esse treinamento provocou uma redução da glicemia e, conseqüentemente, melhorou a sensibilidade a insulina e a captação de glicose do organismo.

3.3.2 *Sensor de frequência cardíaca*

Sensores de frequência cardíaca medem quantas vezes o coração pulsou em um certo período. Existem alguns tipos no mercado, dentre eles destacam-se o estetoscópio, o medidor de frequência óptico e o medidor de sinais elétricos do coração. Os mais comuns de serem utilizados no âmbito esportivo são os que fazem ECG (eletrocardiograma). (POLAR, 2021)

Sensores desse tipo são muito confiáveis e precisos, a medição é feita a partir da análise dos sinais, em que de maneira simplificada, a quantidade de picos de sinal durante um intervalo de tempo, *Figura 16*, determinam o valor resultante. (CAPSISTEMA, 2021)

Figura 16 – Método para calcular a frequência cardíaca



Fonte:(CAPSISTEMA, 2021)

3.4 SISTEMA EMBARCADO

Abaixo está a descrição do microcontrolador que tem a função de integrar todos os sensores:

3.4.1 ESP32

O ESP32 é um microcontrolador fabricado pela empresa Chinesa ESPRESSIF, foi projetado para possuir um bom desempenho de potência em conjunto com robustez, versatilidade e confiabilidade em diversas aplicações como *mobile*, *wearable* e a IoT. Ele apresenta *Wi-Fi* de 2,4 GHz e *bluetooth* integrado, além de ter baixo consumo de energia. (ESPRESSIF, 2021)

No mercado possui outros tipos de microcontroladores, como o Arduino UNO, Arduino MEGA 2560 e o ESP 8266. Conforme a Tabela 1 comparando o ESP32 com os outros modelos é possível perceber que ele em quesito de *hardware* se sobressai, possuindo um conversor digital analógico, distinguindo-se dos outros, possui mais memória RAM, 260 vezes maior que a do Arduino, e ROM, 112 vezes maior que a do Arduino MEGA 2560.

Tabela 1 – Comparação entre os microcontroladores Arduino UNO, Arduino MEGA 2560, ESP32 e ESP8266

Descrição	Arduino UNO	Arduino MEGA 2560	ESP32	ESP8266
Entrada Regulada (VIN)	7 ~ 12V	7 ~ 12V	5 ~ 9V	5 ~ 9V
Frequência de Operação:	0 ~ 16 MHz	0 ~ 16MHz	80MHz ~ 240MHz	80MHz ~ 160MHz
Memória FLASH:	32KB	256 KB	4MB	4MB
Memória RAM/SRAM:	2KB	8 KB	520KB	36kB
Memória ROM/EEPROM	4KB	4 KB	448KB	64KB
Pinos de I/O:	23 pinos com 6 PWM	54 pinos com 14 PWM	34 pinos com 16 PWM	13 pinos com 9 PWM
Conversores ADC (Analogico para Digital):	6 ADC com 10-bit de resolução (1024 bits)	16 ADC com 10-bit de resolução (1024 bits)	18 ADC com 12-bit de resolução (4096 bits)	1 ADC com 10-bit de resolução 1024 bits)
Conversores DAC (Digital para Analógico):	Nenhum	Nenhum	2 DAC com 8-bit de resolução (256 bits)	Nenhum
WiFi:	Somente com Shield	Somente com Shield	2,4 GHz, 802.11 b/g/n/e/i (802.11n até 150 Mbps)	2.4GHz com suporte a WPA e WPA2, 802.11 b/g/n, P2P e soft-AP
Bluetooth:	Não Possui	Não Possui	Bluetooth Low Energy v4.2 (BLE)	Não Possui
Interfaces de Módulos	I2C, SPI, UART	I2C, SPI, UART	SPI, SDIO, CAN, UART, I2S e I2C	SPI, SDIO, UART, I2S e I2C.

Fonte:(OLIVEIRA JAILSON, 2019)

Para o projeto é necessário ter um microcontrolador que possua os protocolos de comunicação I2C, SPI e UART, que são utilizados para comunicação com os sensores já descritos; Por possuir uma boa quantidade de GPIO'S, o ESP32 se demonstra como melhor opção, o que permiti adicionar mais sensores bem como seu alto poder de processamento permite trabalhar com eles de forma mais rápida.

3.5 APLICATIVO MOBILE

O aplicativo *mobile* pode ser construído em diferentes linguagens de programação para diferentes sistemas operacionais, como Android e iOS.

3.5.1 Android

Android é um sistema operacional baseado no núcleo Linux, projetado e usado especialmente em dispositivos eletrônicos móveis, como celulares e tablets. (ANDROID, 2022)

3.5.2 *IOS*

O iOS é um sistema operacional móvel da empresa Apple, desenvolvido para celulares da marca que é apenas executado em *hardwares* da marca.(APPLE, 2022)

3.5.3 *React Native*

React Native é um *framework* que possibilita o desenvolvimento de diversas aplicações *mobile*, que são multiplataforma (Android e iOS) utilizando apenas Javascript, linguagem de programação interpretada e estruturada de alto nível, baseada em React, com foco em criações de front-end. A preferência pelo uso do react native em comparação a outros frameworks é sua grande capacidade de trabalhar com propriedades nativas do smartphone além de ser utilizada por empresas já consolidadas no mercado como Instagram, Facebook, Tesla, Discord e Airbnb. (META, 2022)Aplicação Web

Em aplicações IoT, é muito comum o uso de soluções Web para que os dados possam ser visualizados e tratados em qualquer lugar, nesse contexto diversas tecnologias podem ser utilizadas.

3.5.4 *Node-RED*

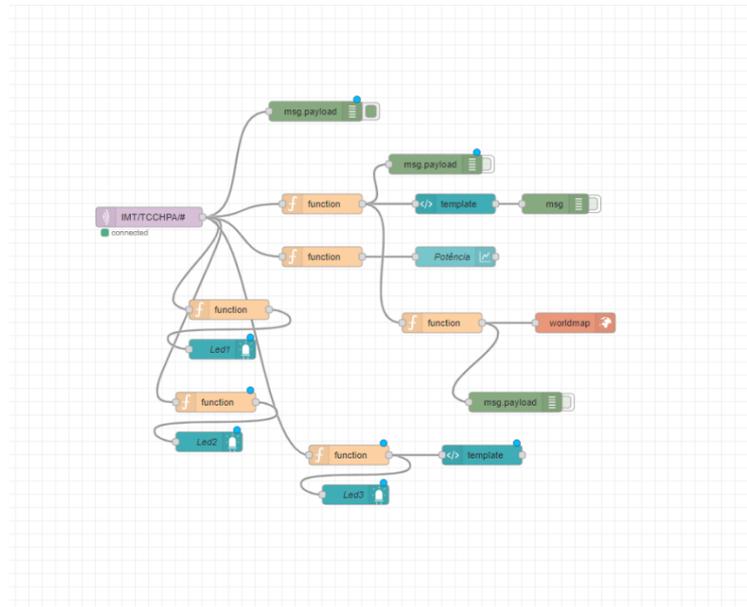
O Node-Red é uma ferramenta de desenvolvimento, criada pela IBM *Emerging Technology* com o intuito de comunicar diversos dispositivos de hardware, permitindo assim a interação com serviços online e APIs, além de ser capaz de tornar mais simples a implementação de sistemas embarcados. (NODE-RED, 2022)

A principal característica da tecnologia Node-Red é sua simplicidade, na qual, é uma programação *low code* tendo seu editor localizado no navegador, além de ser *Open Source*, Código Aberto, assim como o *linux*, permitindo assim uma gama de facilidades em virtude de sua comunidade gigantesca de profissionais. Tendo como código base o Node JS, a execução de eventos se torna mais rápida e robusta aproveitando ao máximo seu potencial. (NODE-RED, 2022)

É composto de “nós”, onde é possível clicar, arrastar e depois soltar na tela, estes nós oferecem diferentes funções (Figura 17):

- **Nós de entrada:** Inserção dos dados em uma determinada aplicação;
- **Nós de saída:** Envio de dados para fora de uma aplicação;
- **Nós de processamento:** Estes nós que permitem a manipulação dos dados com o intuito de disponibilizar os dados tratados ou criados para um próximo estágio.

Figura 17 – Diagramação de blocos do Node-RED



Fonte: Autor próprio

Quando eles são conectados, criam uma estrutura denominada fluxo, com informações de nó para nó. (VEIGA, 2022)

3.5.5 HTML

A HTML é uma linguagem de marcação utilizada para criar páginas Web, esses documentos criados são interpretados por diversos navegadores, como o Google Chrome, Microsoft Edge e Mozilla Firefox. A HTML usa diferentes marcações para diferentes elementos de texto, como imagem, título, parágrafo, entre outros. Na Figura 18, segue um exemplo de uma estrutura HTML padrão, na qual dentro da *tag head*, normalmente é definido os itens de cabeçalho e no *body*, todas as marcações que constroem os elementos visuais das páginas Web. (W3 SCHOOLS, 2022)

Figura 18 – Exemplo de código HTML

```
<html>
<head>
  <title>Telemetria HPA</title>

  <script src="https://cdn.tailwindcss.com"></script>
  <script src="/resources/node-red-chartjs/chart.min.js"></script>
  <script src="/resources/node-red-chartjs/chartjs-plugin-annotation.min.js"></script>
  <script src="jquery-3.6.0.min.js"></script>

  <style>

    /*Main background*/
    body {
      background: -webkit-linear-gradient(
        55deg,
        #006494 0%,
        #38b6ff 100%
      );
    }

    /*Top bar*/
    body.nr-dashboard-theme md-toolbar,
    body.nr-dashboard-theme md-content md-card {
      background-color: transparent !important;
      color: #FFFFFF !important;
    }
  </style>
</head>
</html>
```

Fonte: Própria Autor

3.5.6 JavaScript

JavaScript é uma linguagem leve, interpretada e baseada em objetos, também conhecida como a linguagem de script de páginas Web, mas também pode ser usada no *back-end* em ambientes como o Apache e o Node.js. (MDN CONTRIBUTORS, 2022)

3.5.7 TypeScript

TypeScript é uma linguagem de programação de código aberto desenvolvida pela Microsoft, baseada em JavaScript, porém é adicionado tipagem estática à linguagem, isto é, as variáveis possuem tipos mais específicos, como números e caracteres de texto. (TYPESCRIPTLANG, 2022)

3.5.8 Angular

Angular é uma plataforma de aplicações web de código-fonte aberto e *front-end* baseado em Typescript liderado pelo Google, a plataforma surgiu do AngularJS, que tinha todo o código baseado em JavaScript. (ANGULAR, 2022)

3.6 API

A Interface de Programação de Aplicações, cuja sigla em inglês é API permite que o usuário utilize um aplicativo, software ou até uma simples planilha, consultando, alterando e armazenando dados de diversos sistemas, sem que ele precise acessá-los diretamente. Dessa forma, a API possibilita automatizar e agilizar processos e assim criar funcionalidades. (ZENVIA, 2020)

3.6.1 JSON

O JSON é uma estrutura de formato de texto que tem a função de transferir informações entre sistemas, embora tenha se originado da linguagem JavaScript ele não é uma linguagem de programação. Pode ser utilizada em diferentes sistemas de programação e seus dados no formato JSON devem ser estruturados por meio de uma coleção de pares com nome e valor ou uma lista ordenada de valores, a estrutura é amplamente utilizada na comunicação Web.(SOUZA, 2020)

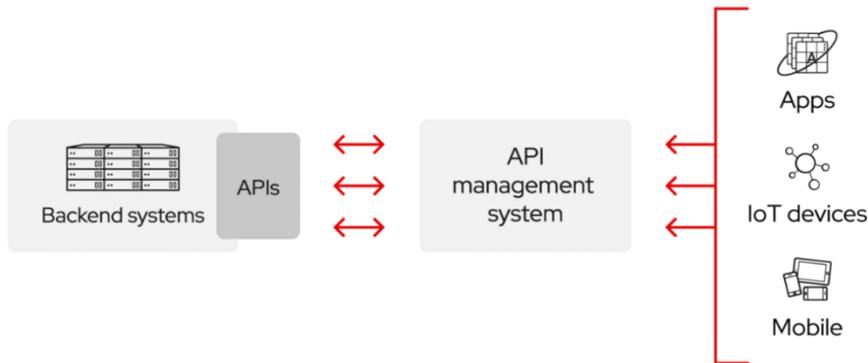
3.6.2 Timestamp

O *Timestamp* é uma representação de um instante único, um ponto específico na linha do tempo, e seu valor corresponde a uma determinada quantidade de tempo decorrida a partir de um instante inicial. Esse instante inicial é chamado Unix Epoch que se refere a 1° de janeiro de 1970, à meia noite, em UTC. O valor retornado por timestamp é em segundos, sendo ele o mesmo para o mundo todo. (HKOTSUBO, 2019).

3.6.3 Geolocation

O Geolocation API é uma ferramenta que permite encontrar dispositivos com grande precisão em qualquer lugar da superfície terrestre, para isso, se baseia em sinais de torres de celulares, GPS e WiFi que são detectados pelo dispositivo móvel (Figura 19). (RED HAT, 2017)

Figura 19 – Exemplificação do sistema de API



Fonte: (RED HAT, 2017)

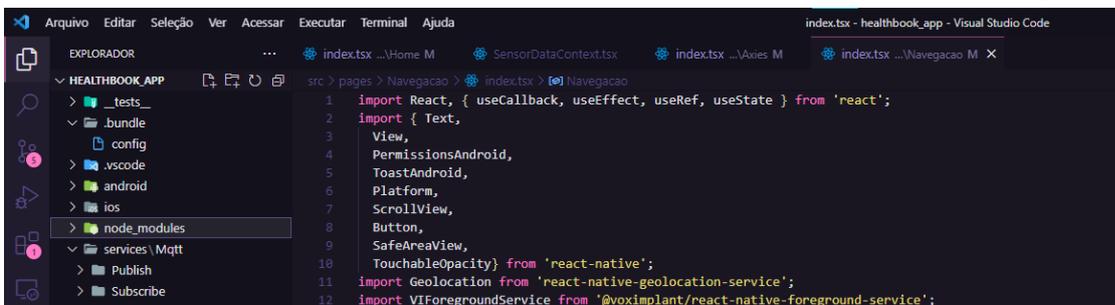
3.7 AMBIENTE DE DESENVOLVIMENTO E FERRAMENTAS

Em projetos para a criação de software geralmente são utilizados ambientes de desenvolvimento integrado que são ferramentas que facilitam a criação e compilação do código. Além disso, para manter o projeto organizado e facilitar o desenvolvimento de múltiplos usuários, ferramentas de versionamento de código são essenciais.

3.7.1 Visual Studio Code

O Visual Studio Code, Figura 20, é um editor de código-fonte desenvolvido pela Microsoft para diferentes sistemas operacionais, dentre as ferramentas disponibilizadas nele, destaca-se o realce da sintaxe, a complementação inteligente de código e principalmente o suporte para depuração.(VISUAL STUDIO, 2022)

Figura 20 – Editor de código.

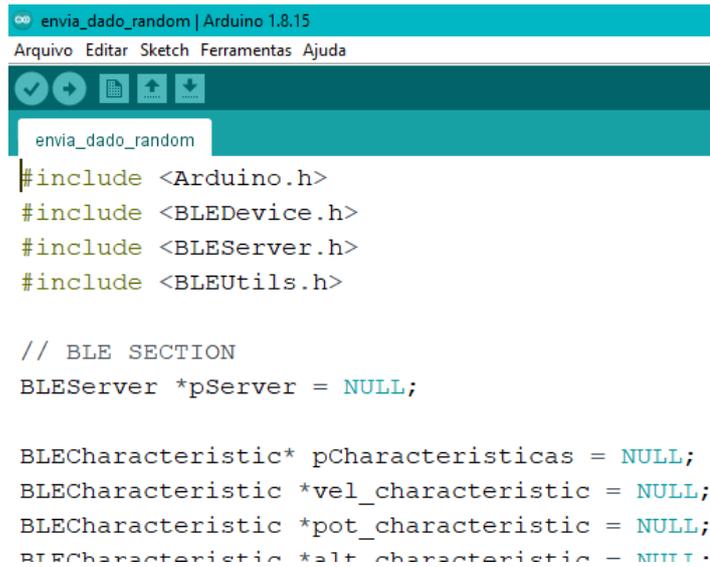


Fonte Próprio Autor

3.7.2 *Arduino IDE*

É um ambiente de desenvolvimento construído em C e C++ (Figura 21), sendo utilizado para desenvolver e gravar os códigos diretamente no microcontrolador Arduino ou em placas de terceiros como o ESP32. (ARDUINO, 2022)

Figura 21 – Arduino IDE



```
envia_dado_random | Arduino 1.8.15
Arquivo Editar Sketch Ferramentas Ajuda

envia_dado_random

#include <Arduino.h>
#include <BLEDevice.h>
#include <BLEServer.h>
#include <BLEUtils.h>

// BLE SECTION
BLEServer *pServer = NULL;

BLECharacteristic* pCharacteristics = NULL;
BLECharacteristic *vel_characteristic = NULL;
BLECharacteristic *pot_characteristic = NULL;
BLECharacteristic *alt_characteristic = NULL;
```

Fonte:Próprio Autor

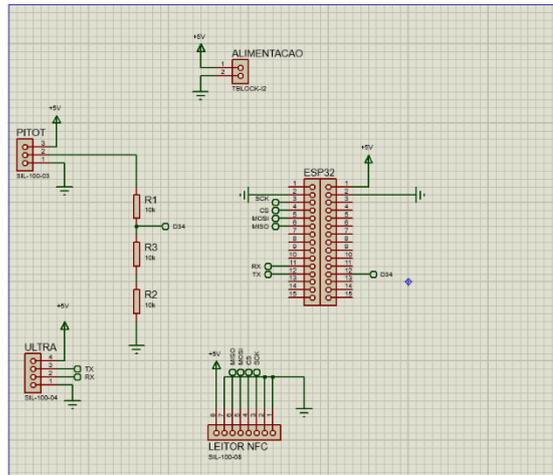
3.7.3 *Git e GitHub*

Git é um sistema de controle de versões distribuído que funciona localmente e serve para registrar o histórico de edições de qualquer tipo de arquivo. Já o GitHub é uma plataforma de hospedagem de código-fonte que normalmente atua em conjunto com o Git. (ANDREI, 2022)

3.7.4 *Proteus*

Proteus é um software de simulação e design de circuitos impressos, Figura 22, muito utilizado em laboratórios eletrônicos para design de circuito elétrico e eletrônico, pois há uma grande variedade de microcontroladores em seu inventário. Ele pode ser encontrado em várias configurações, dependendo do tamanho dos projetos que estão sendo produzidos e dos requisitos para simulação de microcontroladores.

Figura 22 – Exemplo de um circuito desenhado no Proteus



Fonte:Próprio Autor

Também é encontrado em universidades ao redor do mundo todo, auxiliando no ensino eletrônico, design integrado e layout de circuitos impressos para alunos, além de possuir recursos para simular virtualmente projetos de IoT.(PROTEUS, 2021)

3.8 HUMAN POWERED AIRCRAFT

O Human Powered Aircraft, aeronave de propulsão humana representado na Figura 23 é definido como uma aeronave que utiliza exclusivamente da potência produzida pelo ser humano como forma de propulsão, de maneira que tal potência pode ser gerada pelo movimento das pernas e/ou braços do ser humano. (GOMES et al., 2018)

Há 40 anos atrás, os HPAs eram relativamente frágeis, voavam apenas em ambientes sem vento, já as aeronaves que estão sendo projetadas atualmente podem lidar com ventos fracos. A sua construção funcional, não é uma tarefa fácil, pois com uma envergadura de cerca de 25 metros e peso total de pouco mais de 30 quilos são necessários materiais ultraleves e precisa técnica de construção. (JEX; MITCHELL, 1982)

Figura 23 – HPA



Fonte:(SUSU,2022)

O controle de um avião leve e de voo lento exige muita habilidade, além disso, o piloto deve gerar energia suficiente para decolar e permanecer no ar, que gira em torno de 400W para uma pessoa de estatura média. À medida que os projetos avançam, torna-se menos necessário que o piloto seja um atleta altamente treinado.(GOMES et al., 2018)

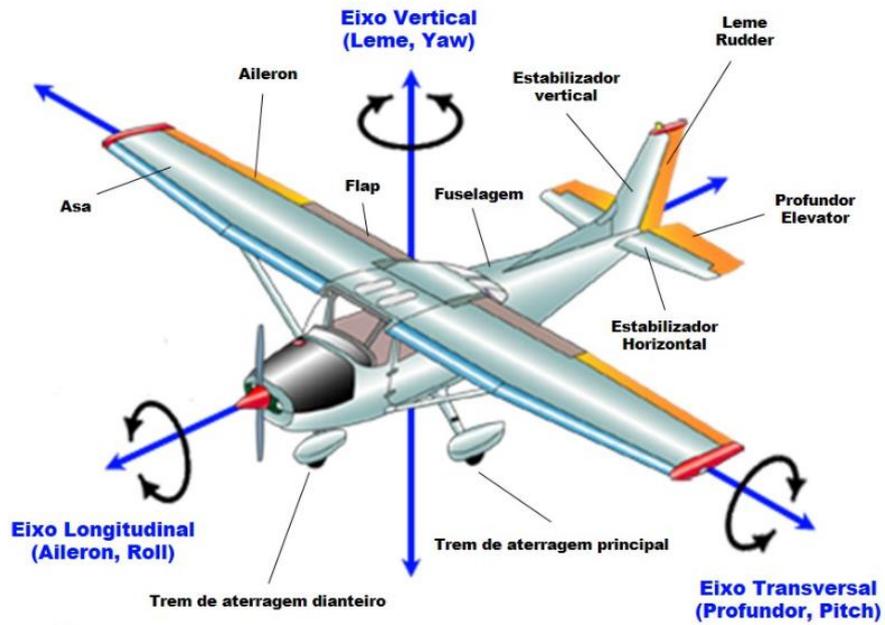
3.9 NAVEGAÇÃO

A telemetria da navegação é extremamente importante para qualquer aeronave, dentre os dados captados destacam-se a posição, o *pitch*, *roll* e *yaw* que são parâmetros essenciais para a realização de qualquer voo.

3.9.1 *Pitch, Roll e Yaw*

Essas variáveis são importantes parâmetros controlados durante o voo e suas medidas são baseadas na angulação do eixo do avião realizadas para o controle do voo. O *pitch* trata da medida do deslocamento angular em relação ao eixo lateral, isto é, o movimento do eixo horizontal, perpendicular ao eixo longitudinal. O movimento do eixo vertical perpendicular ao eixo longitudinal é chamado de *yaw*, este está muito associado também ao uso de bússolas, indicando a posição dos pontos cardeais. Já o *roll* está associado ao movimento em torno do eixo horizontal e na direção do eixo longitudinal, ou seja, ao deslocamento relacionado a rolagem, todos esses eixos estão ilustrados na Figura 24 (NOVAGE, 2021)

Figura 24 – Pitch, Roll e Yaw



Fonte: (NOVAGE, 2021)

4 MATERIAIS E MÉTODOS

A seguir é descrito os materiais e métodos utilizados neste trabalho:

4.1 MÉTODO

O trabalho foi segmentado e estruturado entre *hardware* e *software* de forma a simplificar e modularizar o sistema. No desenvolvimento do *hardware* destaca-se a captura e centralização das informações no microcontrolador ESP32, através da aquisição dos dados por diferentes sensores, com diferentes protocolos. Já na parte de software, destaca-se a criação de interfaces Web e *mobile*, além do uso de sensores do próprio dispositivo móvel. Para cada um deles, foi definido diferentes fluxos de informação que facilitaram a resolução da arquitetura do sistema.

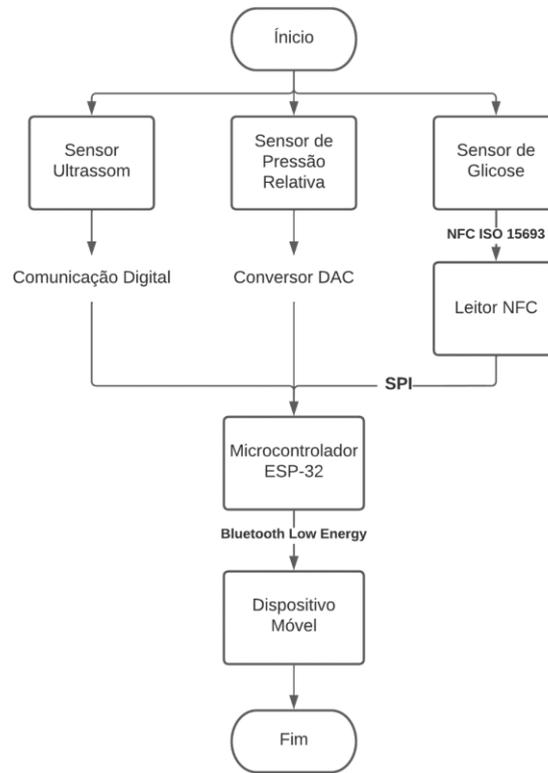
4.1.1 Fluxograma das informações do sistema embarcado

Para a aquisição de informações de velocidade, glicose e ultrassom foram utilizados os seguintes meios de comunicação, respectivamente: conversor analógico digital, SPI e comunicação digital.

Para informação de velocidade, o dado analógico capturado pelo sensor transmitido para um conversor analógico-digital, na qual é tratado através do código no microcontrolador. Já as informações de glicose são enviadas primeiro para um leitor NFC e posteriormente chegam no ESP32 através do protocolo de comunicação SPI. A captura da informação da altura foi implementada através de um sensor de ultrassom que transmite o dado via comunicação digital.

O fluxograma das informações do sistema embarcado está representado a seguir:

Figura 25 – Fluxograma de informações de sensores conectados por cabo ao ESP-32



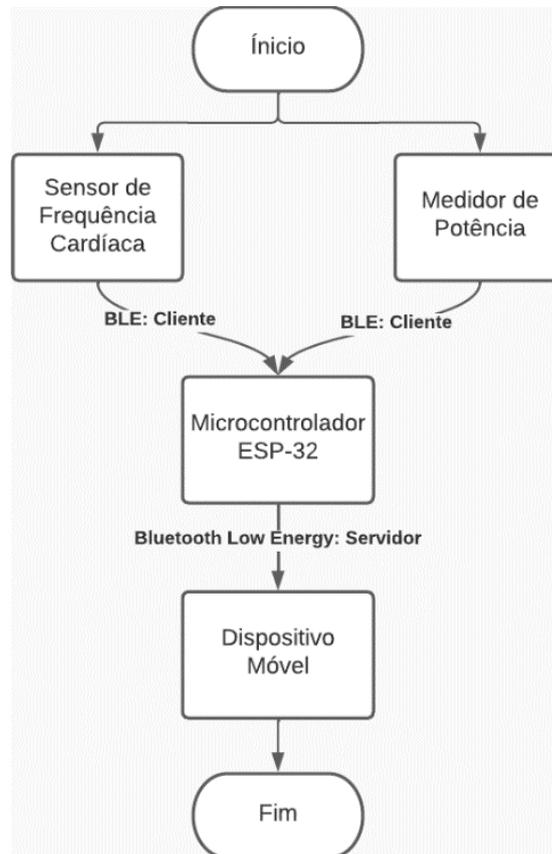
Fonte: Próprio Autor

4.1.2 Fluxograma das informações obtidas por sensores com comunicação BLE

Tanto o sensor de frequência cardíaca, quanto o medidor de potência, transmitem seus dados através do protocolo de comunicação *Bluetooth Low Energy*. Nesse caso, eles funcionam como *server* e o ESP32 como *client*, pois ele recebe e trata as informações transmitidas; O ESP32 precisa ser configurado de modo a aceitar múltiplas conexões e funcionar como *client* e *server*.

O fluxograma das informações obtidas por sensores com comunicação *BLE* está representado a seguir:

Figura 26 – Fluxograma das informações de sensores BLE



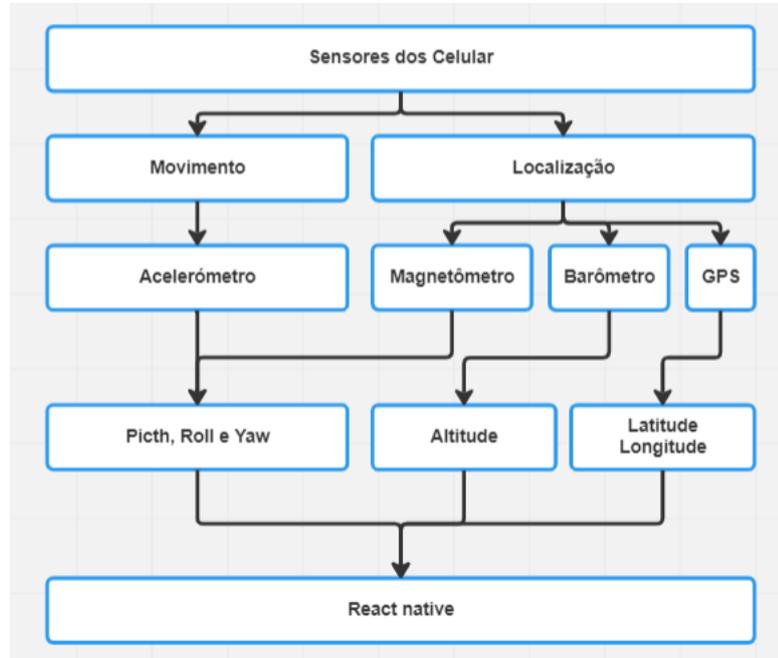
Fonte: Próprio Autor

4.1.3 Diagrama das informações dos dados de navegação

A melhor opção encontrada foi utilizar a API de geolocalização usada pelo Google Maps, que a partir de torres de celulares e dados móveis oferece com precisão a localização do HPA, assim como a informação da velocidade por exemplo. A comunicação utilizada pela Geolocation API é o HTTPS e a função Post. Tanto a mensagem enviada quanto a resposta do request recebida pelo servidor são mensagens do tipo JSON, com a informação de latitude e longitude, além da precisão, que difere dependendo das torres de celulares próximas. (GOOGLE, 2022)

No caso de utilizar os próprios recursos do dispositivo móvel, o diagrama seria da seguinte forma:

Figura 27 – Diagrama dos sensores de navegação do celular.



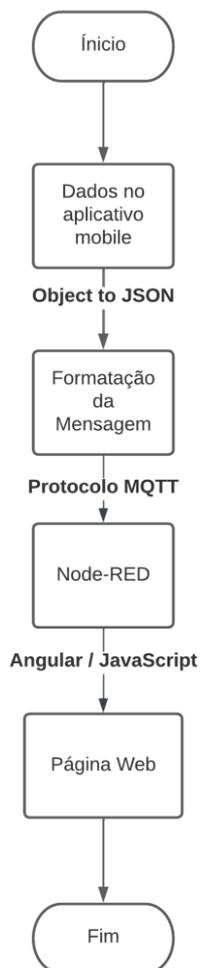
Fonte: Próprio Autor

4.1.4 Fluxograma de informações da Interface Web

Todos os dados centralizados no aplicativo *mobile* são transmitidos via MQTT para o Node-RED, através do formato de mensagem JSON, essas informações são manipuladas e exibidas em uma página Web, construída com Angular e JavaScript, a partir de mapas e gráficos.

O fluxograma das informações da interface Web está representado a seguir:

Figura 28 – Fluxograma dos dados a serem enviados para interface Web



Fonte: Próprio Autor

4.2 SENSORES

Para o desenvolvimento do projeto foi necessário utilizar os sensores descritos a seguir:

4.3 CARACTERÍSTICAS DOS SENSORES UTILIZADOS

Esta é seção se propõem há apresentar as especificações técnicas dos sensores utilizados no projeto.

4.3.1 Cinta Cardíaca

O sensor destinado a função da leitura dos batimentos cardíacos foi a Cinta iGPSPORT HR40 (Figura 29).

Figura 29 – Cinta cardíaca



Fonte:(IGSPORT, 2022)

- Unidade de frequência cardíaca: bpm
- Precisão da frequência cardíaca: +/- 1pbm
- Escala de frequência cardíaca: 40-240bpm

A Tabela 2 detalha as características técnicas da cinta cardíaca.

Tabela 2 – Características Técnicas- iGPSPORT HR40

Cinta Cardíaca- iGPSPORT HR40		
Tempo de Bateria	1000	h
Potência	0,62	W
Potência stand-by	-	W
Tempo Start	-	s
Tempo de Duração da Bateria	340	h
Protocolos de comunicação	Bluetooth BLE e ANT+	-
Tensão de alimentação	3	V

Fonte: (IGSPORT, 2022)

4.3.2 Medidor de Potência

O sensor destinado a função de medir a potência do ciclista o foi Stages Power Shimano's Dura-Ace 9000(Figura 30).

Figura 30 – Medidor de potência



Fonte:(STAGES CYCLING, 2020)

A Tabela 3 detalha as características técnicas do medidor de potência.

Tabela 3 – Características Técnicas do Dura-Ace 9000

Medidor de Potência-Stages Power Shimano's Dura-Ace 9000		
Tempo da Bateria	200	h
Faixa de Potência	0-2500	W
Precisão da medição de Potência	± 1,5	%
Faixa de Cadência	20-220	rpm
Precisão da medição de cadência	± 2	%
Protocolos de comunicação	ANT+, Bluetooth BLE	-

Fonte:(STAGES CYCLING, 2020)

4.3.3 Dinamômetro

O dinamômetro trainer elite turbo muin, destinado medir a potência do ciclista, foi utilizado na apresentação para fins explicativos (Figura 31).

Figura 31 – Dinamômetro



Fonte: (ELITE, 2020)

A Tabela 4 detalha as características técnicas do medidor de potência.

Tabela 4 – Característica técnicas da trainer Elite Turbo Muin

TRAINER ELITE TURBO MUIN		
Compatível	ANT+ e BLE	
Peso	17	Kg
Potência máxima a 60 Km/h	2940	W
Dados	Potência, cadência e velocidade	

Fonte:(ELITE, 2020)

4.3.4 Medidor de Glicemia

O sensor destinado a função de medir a glicose do ciclista foi o sensor da empresa FreeStyle (Figura 32).

Figura 32 – Sensor de glicose



Fonte: (SENSOR FREESTYLE, 2015)

A Tabela 5 detalha as características técnicas do medidor de potência.

Tabela 5 – Características Técnicas do sensor de glicemia

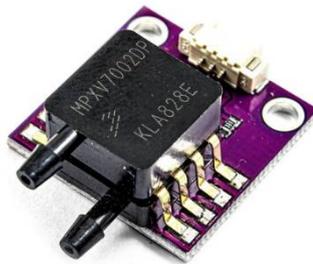
Sensor Freestyle Libre		
Tempo da Bateria	14	dias
Taxa de leitura	60	s
Armazena de dados	8	H

Fonte:(SENSOR FREESTYLE, 2015)

4.3.5 *Sensor de pressão relativa*

O sensor destinado a função de medir da pressão relativa foi o MPXV7002DP (Figura 33).

Figura 33 – Medidor de pressão relativa



Fonte:(DAMIRCHI, 2019)

A Tabela 6 detalha as características técnicas sensor de pressão relativa.

Tabela 6 – Características Técnicas do sensor de pressão relativa

MPXV7002DP		
Tensão de alimentação	5	V
Corrente de alimentação	10	mA
Range de pressão	-2 a 2	kPa
Precisão	2,5	%

Fonte:(SEMICONDUCTORS, 2017)

4.3.6 Sensor de ultrassom

O sensor destinado a função de medir a altura foi o JSN-SR04T (Figura 34).

Figura 34 – Sensor de ultrassom



Fonte:(USINAINFO, 2020)

A Tabela 7 detalha as características técnicas sensor de ultrassom.

Tabela 7 – Características Técnicas do sensor de ultrassom

JSN-SR04T		
Tensão de alimentação	3 a 5	V
Corrente de operação	<8	mA
Frequência do ultrassom	40	KHZ
Precisão	+/- 1	Cm

Fonte:(JAHANKIT, 2020)

4.3.7 Leitor RFid/NFC

Sensor destinado a realizar a leitura RFid (Figura 35).

Figura 35 – Leitor RFid



Fonte:(MIKROW, 2022)

A Tabela 8 detalha as características técnicas do leitor RFID/NFC.

Tabela 8 – Características Técnicas do leitor RFID/NFC

Leitor RFID/NFC		
Interface	UART, SPI, GPIO	
Tensão de Alimentação	3,3	V
Corrente Stand-by	20 - 80	μ A
Potência	55	mW

Fonte:(MIKROE, 2017)

4.4 ESP32

O microcontrolador destinado a aquisição de dados dos sensores foi o ESP32 (Figura 36):

Figura 36 – Microcontrolador ESP32



Fonte:(OLIVEIRA EULER, 2017)

A Tabela 9 detalha as características técnicas do microcontrolador ESP32.

Tabela 9 – Características do ESP32

ESP32		
Corrente	20 ~ 68	mA
Corrente Stand-by	1,00	uA
Tensão de alimentação	5 ~ 9	V
GPIO'S	34	
Conversores analógico digital	18	
Conversores digital analógico	2	
Protocolos de comunicação	UART, SPI, I2C, CAN, Bluetooth e WiFi 2,4Ghz	

Fonte:(ESPRESSIF, 2021)

4.5 SOFTWARES UTILIZADOS

O Software do dispositivo móvel foi desenvolvido no editor de código-fonte Visual Studio Code em React Native, o versionamento de código foi feito através do GitHub e Git de forma que várias partes do aplicativo pudessem ser desenvolvidas simultaneamente e depois integradas.

Já o Software do microcontrolador foi desenvolvido na plataforma Arduino IDE com bibliotecas compatíveis com o ESP32, foram feitos códigos para as diferentes funções propostas e depois os mesmos foram integrados.

5 DESENVOLVIMENTO

O desenvolvimento do trabalho foi dividido em duas partes principais: *hardware* e *software*. No caso do *hardware*, o código final contém a comunicação com diferentes protocolos e executa diferentes funções, como a comunicação e a calibração das medidas elétricas obtidas. Já no *software* são recebidas todas as variáveis do controlador e exibidas de diferentes formas, tanto na Web, como no dispositivo móvel.

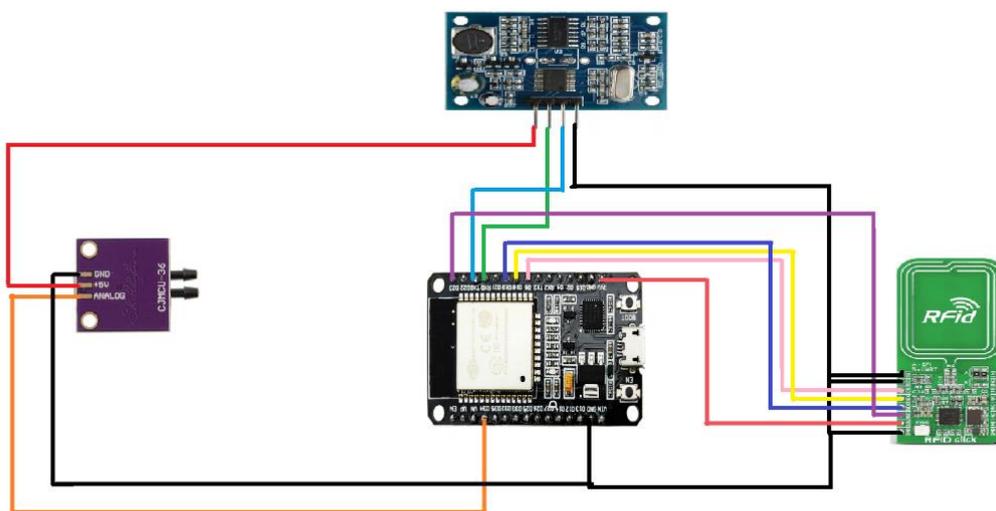
5.1 HARDWARE

O *hardware* implementado foi setorizado em diferentes partes, que contemplam a sua construção física e o software do microcontrolador.

5.1.1 Sistema embarcado

Para unir o Hardware foi desenvolvido um protótipo para aquisição de dados em que o componente principal é o ESP32, responsável por obter as informações dos sensores; O microcontrolador em questão possui 34 GPIOs (*General Purpose Input/Output*), na qual, são utilizados para diferentes propósitos e protocolos, por exemplo o sensor de pressão relativa necessita de um pino lógico com conversor analógico digital como pode ser observado na *Figura 37* em conjunto com a placa de NFC e o sensor de ultrassom.

Figura 37 – Esquema elétrico do sistema embarcado

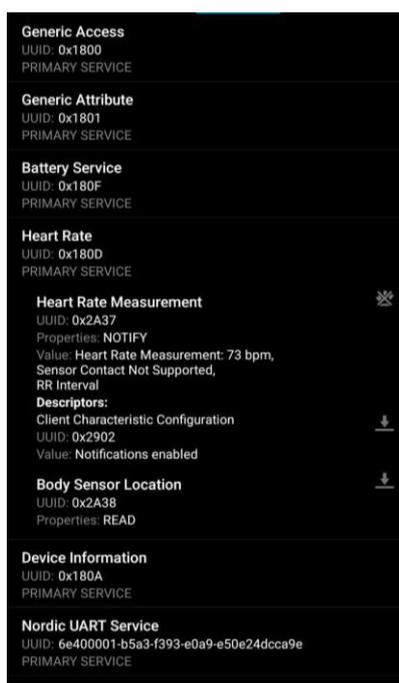


Fonte: Próprio Autor

5.1.2 Comunicação BLE do ESP32 com os sensores Bluetooth

Para conectar aos sensores de batimento cardíaco e ao medidor de potência é necessário o uso do *Bluetooth Low Energy*. O ESP32 precisa trabalhar no modo *client* e *server*, o primeiro modo é responsável por adquirir e guardar os dados dos sensores, já o segundo transmite para o dispositivo móvel.

Figura 38 – Captura de tela do aplicativo NRF-connection do sensor de batimento cardíaco



Fonte: Próprio Autor

Para receber os dados de frequência cardíaca e potência, o microcontrolador trabalha no modo *client*, na qual, é imprescindível a configuração do nome do *bluetooth*, do serviço e da característica para ocorrer a conexão e a aquisição do dado de forma exata. A cinta de frequência cardíaca tem como nome de *bluetooth* “HR40-40176044”, pela *Figura 38* é possível observar que esse dispositivo possui mais de um tipo de serviço como a bateria e o batimento cardíaco, no caso o serviço escolhido foi o da frequência cardíaca que possui o UUID “180D” e característica “2A37”.

5.1.3 Aquisição de dados do sensor de glicose

O sensor escolhido para a aquisição de dados de glicose do piloto foi o Freestyle Libre, pois, apesar de ser invasivo, é discreto e simples o suficiente para não atrapalhar a pilotagem, além de requisitar os dados de glicose por longos períodos.(FREESTYLE, 2020)

Por outro lado, não existe oficialmente um modo de fazer uma amostragem contínua da glicose com o Freestyle, pois a fabricante disponibiliza apenas um *scan* momentâneo através de NFC para a transmissão dos dados salvos da glicemia.

Dessa forma, foi necessário transformar o sistema Freestyle Libre em um sistema de monitoramento com intervalos de tempo constante de glicose. Isso significa que não é preciso realizar a leitura manualmente, pois o leitor NFC envia o dado para o ESP32 que posteriormente publica no dispositivo móvel.

5.1.4 Aquisição de dados do sensor ultrassom

Diferente dos sensores *bluetooth*, o sensor ultrassônico necessita de uma conexão física, no ESP32 a conexão deve ser feita através da comunicação serial, nesse modo de trabalho o cálculo da distância é feito no próprio sensor e transmitido para o ESP32. O dado vem fragmentado em 4 bits, a distância em milímetros está presente no primeiro e segundo *byte* sendo necessário realizar um deslocamento a esquerda de 8 bits do primeiro.

O circuito desenvolvido foi feito pelo Centro de Pesquisas, em específico, a Divisão de Eletrônica e Telecomunicações do Instituto Mauá de Tecnologia, na qual, foi utilizado uma placa de sensor ultrassônico, *Figura 34*, e foi adicionado um resistor de $47k\Omega$ no R27, habilitando o modo serial automático que envia o dado referente à distância a cada 100 milissegundos. (DET, 2020)

5.1.5 Aquisição de dados do sensor de pressão relativa

O sensor analógico de pressão relativa emite um sinal que varia de 0 a 5V, referente a conversão da pressão, este por sua vez é diretamente proporcional. Como o ESP32 trabalha somente com sinais digitais é necessário um conversor analógico digital de 12 bits.

Como a grandeza física a ser medida é a velocidade é necessária realizar algumas transformações:

Primeiro o sinal digital precisa ser transformado para respectivo valor em pressão a transformação está representada na equação 3.

Equação 3 – Conversão digital pela resolução

$$PR = \text{Leitura do sensor} - \frac{1023}{2}$$

Depois esse valor é aplicado na equação a seguir, obtendo assim a velocidade relativa.

Equação 4 – Equação de Velocidade baseada na equação de Bernoulli

$$v = \sqrt{\frac{2(PR)}{\gamma}}$$

5.2 SOFTWARE DO APLICATIVO MÓVEL

O aplicativo, como mencionado, foi desenvolvido em React Native, com uma arquitetura semelhante a aplicativos Web desenvolvidos como *Single Page Application*, onde as páginas são estruturadas em pilhas, na qual há um elemento que contém outros, ou seja, há um sistema de hierarquia entre os elementos do aplicativo.

A navegação do aplicativo é feita a partir de uma biblioteca conhecida como Stack Navigator, na qual todas as páginas do aplicativo ficam dentro desse elemento e entre elas é feito um esquema de rotas para que seja possível a navegação entre elas.

O aplicativo foi estruturado em três páginas principais, a Home, que contém um resumo dos dados que estão sendo recebidos ao aplicativo, a de localização que informa a localização atual do usuário e a de navegação que contém parâmetros do voo da aeronave.

5.2.1 *Página Home*

Na página Home, é feita a conexão *Bluetooth* do dispositivo móvel com o ESP32 a partir do nome do dispositivo e do serviço, a informação já é recebida separadamente de forma a identificar quais dados estão sendo recebidos e quais não, além de diminuir o tráfego de dados, caso não haja dispositivos conectados.

O programa foi configurado de modo a escanear os dispositivos *bluetooth* e ao encontrar o nome do dispositivo configurado, realizar o pareamento. Após parear, é realizado a conexão com um ou vários serviços para assim receber diferentes informações. Dentro da página Home também há todas as rotas para as diferentes páginas do aplicativo.

5.2.2 *Página Navegação*

Na página de Navegação, foi utilizado o magnetômetro e acelerômetro do próprio dispositivo móvel, a partir deles, é possível obter as informações do *pitch*, *roll* e *yaw*. É importante ressaltar que alguns dispositivos móveis não possuem esses sensores. Como esses valores dependem da posição em que o dispositivo foi colocado, foi necessário a criação de uma tela, em que pode ser configurado também a posição inicial do dispositivo, assim a partir de manipulações matemáticas foi obtida a posição real da aeronave.

5.2.3 *Página Localização*

Na página de Localização, foi utilizado a biblioteca do Google conhecida como Geolocation, nela é programado para que seja recebida a localização do dispositivo móvel através de uma requisição a API do Google em um intervalo de tempo conhecido. São várias as informações recebidas a cada request, porém as mais utilizadas no aplicativo é a de latitude e longitude, para esse serviço, o aplicativo precisa estar conectado à Internet.

5.2.4 *Gráficos*

O aplicativo realiza a criação de gráficos da frequência cardíaca e da glicose do usuário, além da potência que está sendo atribuída à aeronave, cada um desses gráficos são exibidos pontos de um intervalo de tempo limitado.

5.2.5 Comunicação MQTT

Todos os dados que podem ser visualizados no aplicativo são transmitidos via protocolo MQTT em um intervalo de tempo específico. Nele o dispositivo móvel é um MQTT *Client* que funciona como *Publisher*, ou seja, publica dados no formato JSON à um determinado MQTT *Broker*. O *software* foi configurado para ter autenticação, onde foi definido um determinado tópico para publicar as informações, para obter um maior desempenho e velocidade, foi definido uma qualidade de serviço de zero, assim não há uma comunicação entre o *Broker* e o *Client* para saber se o dado realmente foi enviado.

Esse serviço exigiu que todos os serviços de dados do aplicativo fossem realizados em pilhas com ordens de hierarquia mais elevadas, para que assim os dados pudessem ser acessados em todas as páginas, para esse recurso foi utilizado um *Hook* do React Native, conhecido como *UseContext* que possibilita que dados sejam trafegados por diferentes telas.

5.3 SERVIÇO WEB

Após publicar os dados no MQTT *Broker* através do celular, foi programado no Node-RED um MQTT *Client* com a função de *Subscriber*, assim todas as informações que chegam ao *Broker* são passadas a clientes que estão inscritos em um determinado tópico. Essa mensagem recebida está no formato JSON, então ela é segmentada e transformada em um objeto, todos esses dados são enviados a um *dashboard*, com gráficos, *labels* e mapas.

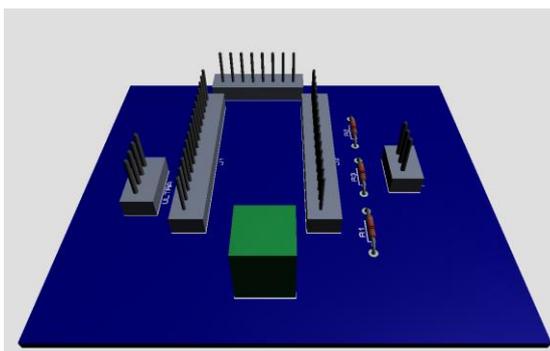
6 RESULTADOS E DISCUSSÃO

Como resultado do trabalho, foi possível obter um aplicativo *mobile*, um protótipo e uma página Web para visualização remota, estes por sua vez se comunicam via BLE e fazem a publicação através do MQTT.

6.1 HARDWARE

O protótipo final contempla o ESP32, sensor de pressão relativa, leitor NFC, sensor ultrassom com suas conexões realizadas em uma PCB, esta por sua vez foi desenvolvida via software Proteus. O resultado do modelo 3D pode ser observado na *Figura 39*.

Figura 39 – Modelo 3D da PCB feito no Proteus



Fonte:Próprio Autor

A imagem a seguir, *Figura 40*, exibe o resultado da impressão feita a partir do layout desenvolvido, em seguida foi realizado as conexões físicas necessárias com microcontrolador.

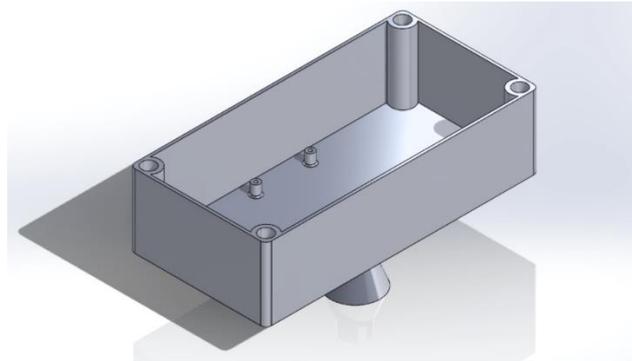
Figura 40 – PCB física com os componentes do sistema



Fonte:Próprio Autor

Com o hardware desenvolvido foi possível estimar o tamanho ideal de uma caixa para alocar uma área total de 50 cm² compostos por placas PCB 6 x 6 cm, ultrassom 4 x 2,5 cm e a do tubo de Pitot 2 x 2 cm, a fim de ajustar as dimensões para que a parte física não possuísse um peso desnecessário, a apresentação 3D da caixa é representada na *Figura 41*.

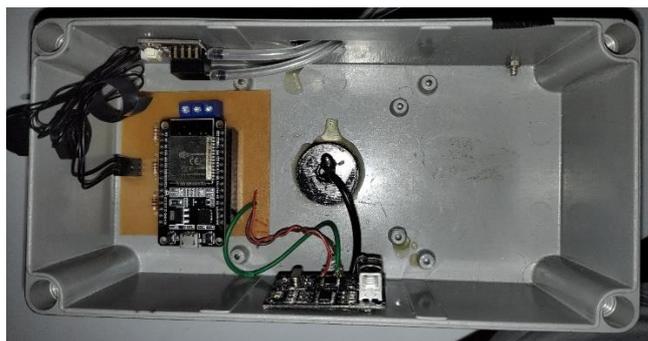
Figura 41 – Apresentação 3D caixa Patola



Fonte: Própria Autor

Por questões de disponibilidade foi utilizado uma caixa Patola PBL-200 com 70x100x200mm para construção do protótipo, Figura 42, vinculada a ela foi inserida um cone referente ao sensor de ultrassom.

Figura 42 – Vista superior do sistema



Fonte: Próprio Autor

O resultado final pode ser observado na Figura 43, o qual foi realizado a furação para alocar o tubo de pitot de forma que a medição sofresse menos interferencia do fluxo de ar impactado pela aerodinâmica da caixa.

Figura 43 – Sistema final montado



Fonte:Próprio Autor

6.2 APLICATIVO MOBILE.

A primeira página que o usuário tem acesso é a área de *login* podendo, futuramente, ter uma base de dados individual para cada usuário cadastrado. Em sequência é possível visualizar a página principal do aplicativo, o qual possui a estrutura de conexão, visualização e configuração das demais páginas.

Na página Home, Figura 44, é possível visualizar toda a estrutura principal do aplicativo, na qual, se encontra as funções de pareamento *bluetooth* e configuração, além de exibir os ícones dos dispositivos como batimento cardíaco, glicose, velocidade, navegação, GPS e potência.

Figura 44 – Página de Login e Home

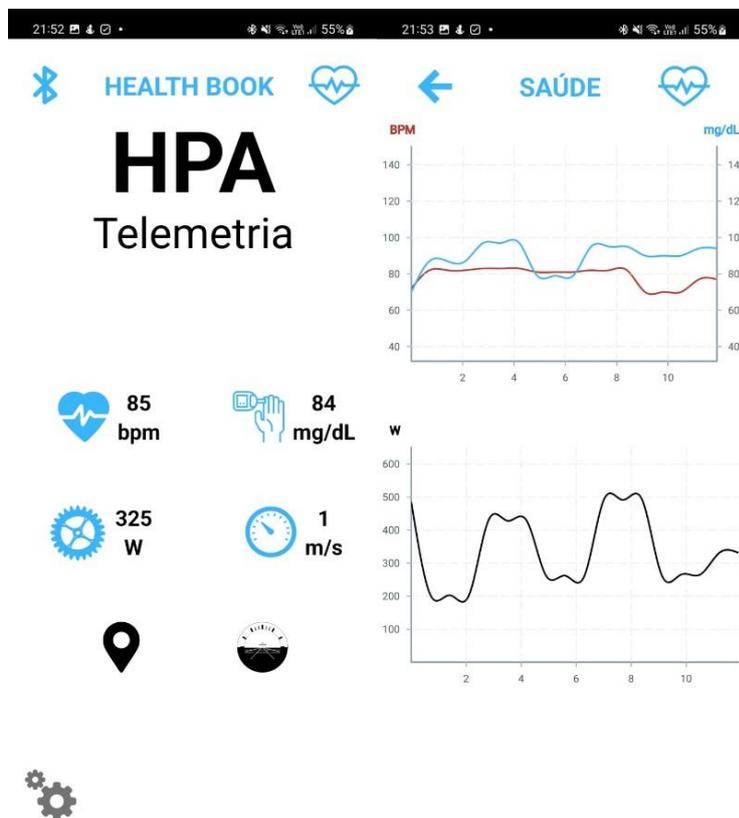


Fonte:Próprio Autor

Quando é estabelecido comunicação *bluetooth* com o ESP32 os ícones que antes eram da cor preta, indicando desconexão, ficam azul simbolizando que a conexão foi estabelecida com sucesso, *Figura 45*, logo em seguida ao lado do ícone deve ser amostrado os valores em tempo real adquiridos e enviados pelo ESP32.

Ao selecionar qualquer um dos ícones apresentado na *Home* uma nova página será aberta, pressionando qualquer um dos 4 ícones centrais, em azul, será exibido um dashboard em tempo real, *Figura 45*, o qual o piloto pode visualizar seu desempenho conforme o tempo.

Figura 45 – Página Home com conexão estabelecida e dashboards em tempo Real



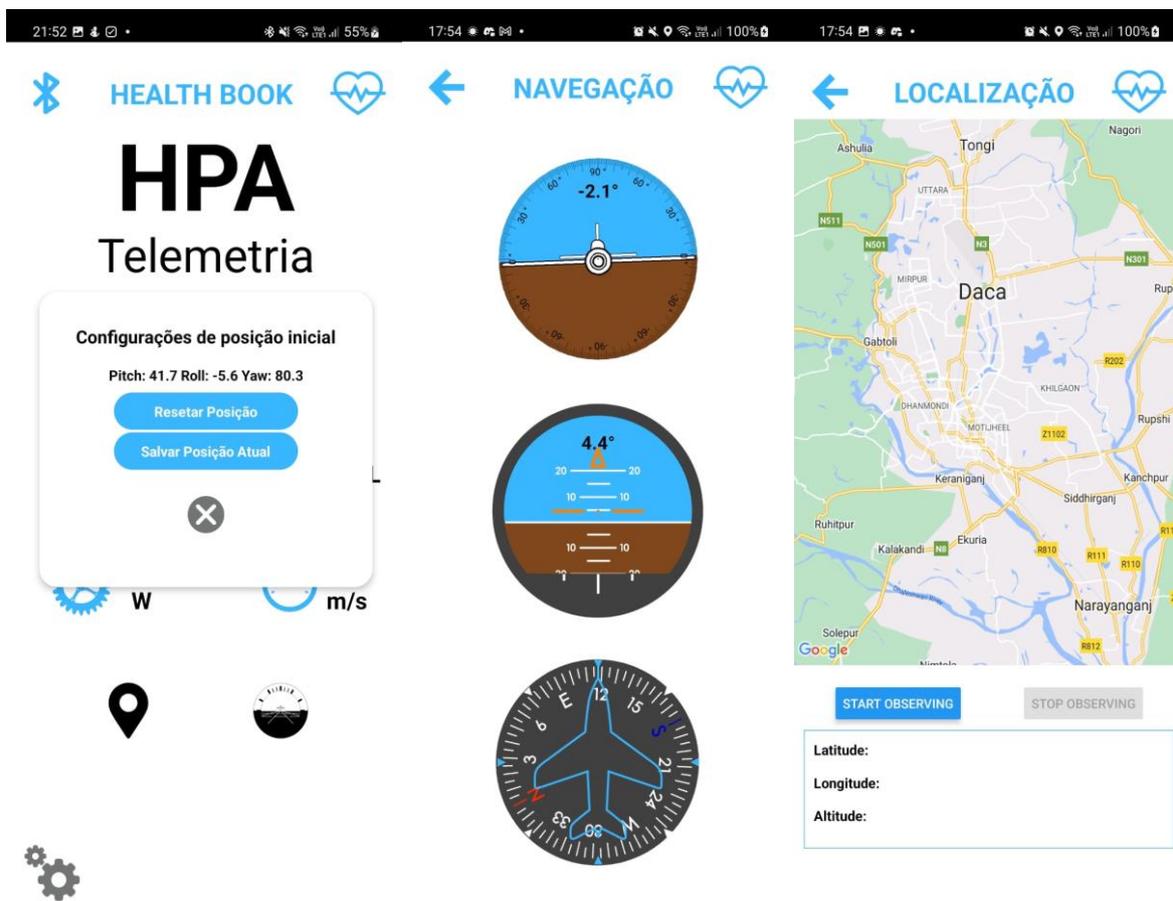
Fonte:Próprio Autor

Ainda na tela *Home* o ícone, em cinza, representa a configuração de posição, ao ser selecionado abre-se uma *modal*, possibilitando que o usuário salve a posição inicial do celular, função importante para a tela de navegação, na qual, pode-se fixar o *smartphone* em qualquer posição, assim fornecendo as condições iniciais de *pitch*, *roll* e o *yaw* para a navegação da aeronave.

Selecione o ícone de navegação, abaixo da velocidade, é exibido a estrutura de navegação, *Figura 46*, o qual o piloto consegue visualizar a posição relativa do celular/aeronave no espaço, simulando um sensor de horizonte artificial, instrumento encontrado em aeronaves, calculando o *pitch*, *roll* e *yaw* a partir do acelerômetro e magnetômetro do próprio *smartphone*.

No ícone ao lado da navegação se encontra a localização exibindo informações como altitude, latitude e longitude e exibe sua posição no mapa.

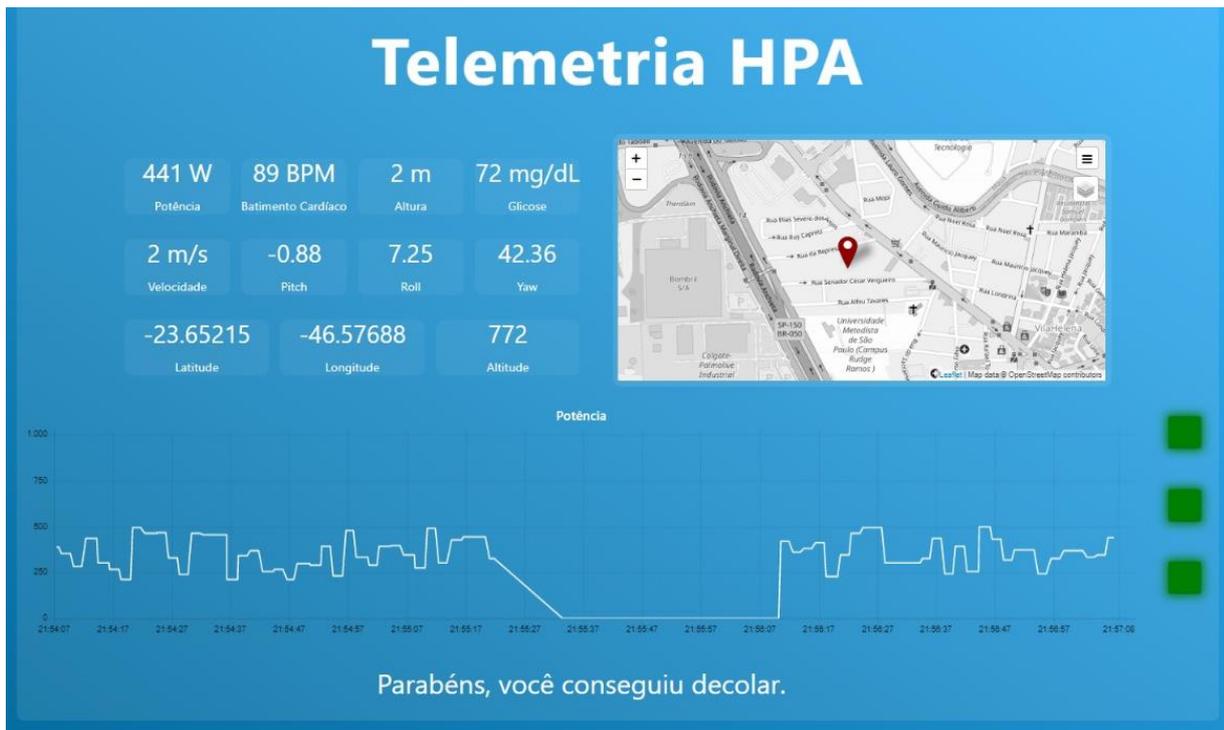
Figura 46 – Tela de configuração, navegação e localização.



Fonte: Próprio Autor

Enquanto o piloto navega a aeronave, HPA, também é possível que sua equipe acompanhe suas condições físicas e de pilotagem, coletando todas as informações do aplicativo e a reexibindo em tempo real em *dashboards*, *Figura 47*, via uma página Web.

Figura 47 – Página Web dos dados recebidos



Fonte: Próprio Autor

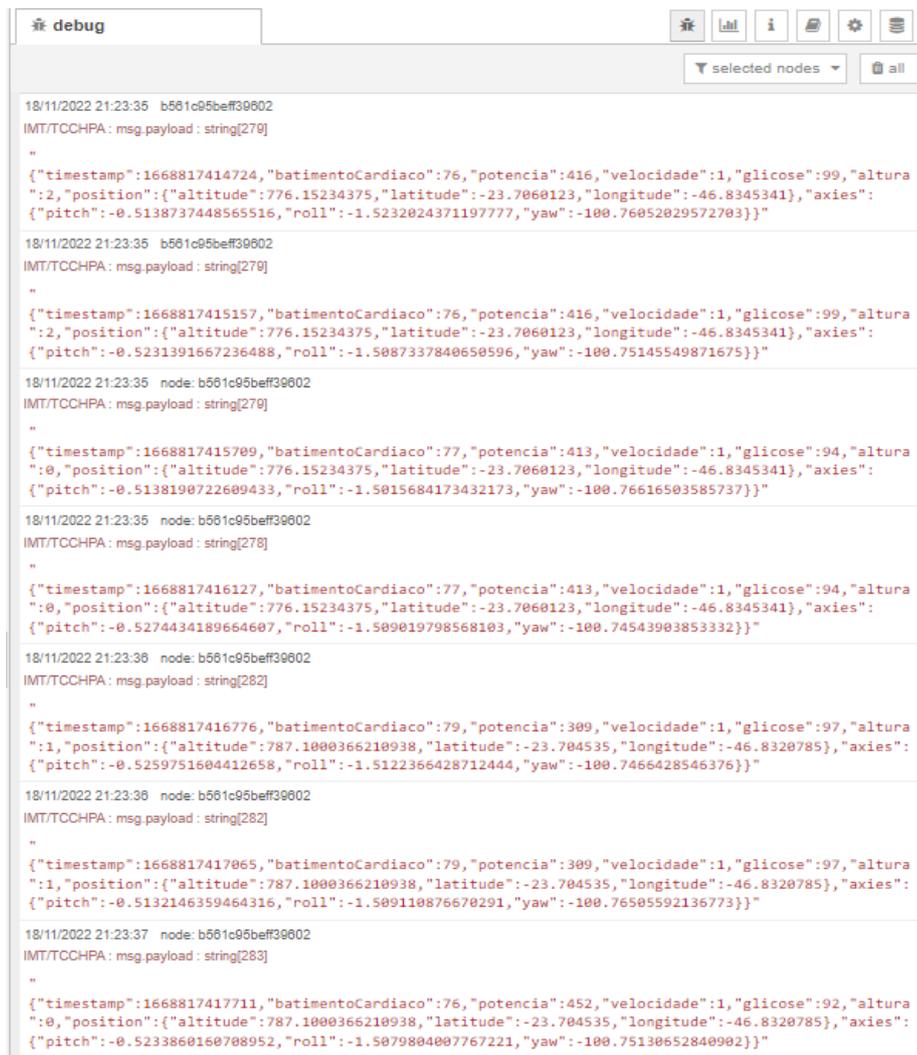
6.2.1 Node-Red

Os dados que chegam via MQTT estão estruturados na formatação JSON compostos por timestamp, batimento cardíaco, potência, velocidade, glicose, altura, altitude, latitude, longitude, pitch, roll e yaw. Após a postagem, essa mensagem é tratada e enviada para a página Web utilizando a própria plataforma do Node-Red (Figura 48).

Segue um exemplo do pacote de dados que chega via MQTT:

```
"{"timestamp":1668817463747,"batimentoCardiaco":79,"potencia":280,"velocidade":1,"glicose":94,"altura":0,"position":{"altitude":787.1000366210938,"latitude":-23.704535,"longitude":-46.8320785},"axes":{"pitch":-0.541333241859175,"roll":-1.5167295421309224,"yaw":-100.7242791592217}}"
```

Figura 48 – Node-Red



The image shows a screenshot of the Node-Red web interface. At the top, there is a 'debug' tab. Below it, there are several icons for actions like refresh, zoom, help, and settings. A dropdown menu shows 'selected nodes' and 'all'. The main area displays a list of messages received from a device. Each message includes a timestamp, a node ID, and a payload. The payload is a JSON object containing various sensor readings such as 'timestamp', 'batimentoCardiaco', 'potencia', 'velocidade', 'glicose', 'altura', 'position', and 'axies'.

```
18/11/2022 21:23:35 b561c95beff39002
IMT/TCCHPA: msg.payload: string[279]
"
{"timestamp":1668817414724,"batimentoCardiaco":76,"potencia":416,"velocidade":1,"glicose":99,"altura":2,"position":{"altitude":776.15234375,"latitude":-23.7060123,"longitude":-46.8345341},"axies":{"pitch":-0.5138737448565516,"roll":-1.5232024371197777,"yaw":-100.76052029572703}}

18/11/2022 21:23:35 b561c95beff39002
IMT/TCCHPA: msg.payload: string[279]
"
{"timestamp":1668817415157,"batimentoCardiaco":76,"potencia":416,"velocidade":1,"glicose":99,"altura":2,"position":{"altitude":776.15234375,"latitude":-23.7060123,"longitude":-46.8345341},"axies":{"pitch":-0.5231391667236488,"roll":-1.5087337840650596,"yaw":-100.75145549871675}}

18/11/2022 21:23:35 node: b561c95beff39002
IMT/TCCHPA: msg.payload: string[279]
"
{"timestamp":1668817415709,"batimentoCardiaco":77,"potencia":413,"velocidade":1,"glicose":94,"altura":0,"position":{"altitude":776.15234375,"latitude":-23.7060123,"longitude":-46.8345341},"axies":{"pitch":-0.5138190722609433,"roll":-1.5015684173432173,"yaw":-100.76616503585737}}

18/11/2022 21:23:35 node: b561c95beff39002
IMT/TCCHPA: msg.payload: string[278]
"
{"timestamp":1668817416127,"batimentoCardiaco":77,"potencia":413,"velocidade":1,"glicose":94,"altura":0,"position":{"altitude":776.15234375,"latitude":-23.7060123,"longitude":-46.8345341},"axies":{"pitch":-0.5274434189664607,"roll":-1.509019798568103,"yaw":-100.74543903853332}}

18/11/2022 21:23:36 node: b561c95beff39002
IMT/TCCHPA: msg.payload: string[282]
"
{"timestamp":1668817416776,"batimentoCardiaco":79,"potencia":309,"velocidade":1,"glicose":97,"altura":1,"position":{"altitude":787.1000366210938,"latitude":-23.704535,"longitude":-46.8320785},"axies":{"pitch":-0.5259751604412658,"roll":-1.5122366428712444,"yaw":-100.7466428546376}}

18/11/2022 21:23:36 node: b561c95beff39002
IMT/TCCHPA: msg.payload: string[282]
"
{"timestamp":1668817417065,"batimentoCardiaco":79,"potencia":309,"velocidade":1,"glicose":97,"altura":1,"position":{"altitude":787.1000366210938,"latitude":-23.704535,"longitude":-46.8320785},"axies":{"pitch":-0.5132146359464316,"roll":-1.509110876670291,"yaw":-100.76505592136773}}

18/11/2022 21:23:37 node: b561c95beff39002
IMT/TCCHPA: msg.payload: string[283]
"
{"timestamp":1668817417711,"batimentoCardiaco":76,"potencia":452,"velocidade":1,"glicose":92,"altura":0,"position":{"altitude":787.1000366210938,"latitude":-23.704535,"longitude":-46.8320785},"axies":{"pitch":-0.5233860160708952,"roll":-1.5079804007767221,"yaw":-100.75130652840902}}


```

Fonte: Próprio Autor

6.3 IMPLEMENTAÇÃO E TESTE DO PROJETO

Após montagem de um sistema de validação de protótipo, composto por um dinamômetro *bluetooth*, além dos sensores descritos anteriormente com o intuito de simular o motor do HPA, foi possível testar o *software* e *hardware* construídos.

A *Figura 49* mostra o desenvolvimento de um protótipo de motor do HPA.

Figura 49 – Protótipo montado para teste do sistema de aquisição de dados



Fonte: Próprio Autor.

A *Figura 50* representa um usuário utilizando o sistema de teste.

Figura 50 – Em teste (EUREKA)



Fonte: Próprio Autor.

A nível de software, tanto a aplicação mobile, quanto a Web funcionaram dentro do esperado. A tela de navegação operou em todos os dispositivos móveis testados, os gráficos foram construídos sempre que os sensores estavam conectados, já o serviço de localização que fazia *requests* na API do Google e a transmissão de dados via MQTT ficaram limitas aos dados móveis do celular, pois são recursos que utilizam internet.

Entretanto no sistema do microcontrolador, a comunicação BLE apresentou algumas inconsistências. Em ambientes abertos e com poucos sinais, o sistema funcionou muito bem com o ESP32 recebendo e transmitindo dados, porém durante o EUREKA, devido a grande quantidade de televisões *Smart TV*, celulares e *smartwatches*, ao escanear e parear a procura de dispositivos *bluetooth* como a cinta cardíaca, dinamômetro e celular, a conexão se mostrou lenta e instável.

Uma alternativa encontrada foi a conexão dos sensores *bluetooth* diretamente com o celular, tornando-a mais estável, pois o dispositivo móvel aparentou ter um sistema de conexão BLE mais robusto que o microcontrolador.

Uma limitação identificada no sistema projetado foi a multi-conexão BLE, pois ao estabelecer conexão entre dois dispositivos clientes ao mesmo tempo, o pareamento com um deles ou até mesmo os dois era perdido, vale ressaltar que o microcontrolador conseguia escanear e estar pareado com um servidor simultaneamente.

Por último, a placa encontrada para fazer a leitura do sensor de glicose que possuía o protocolo de leitura NFC correto apresentou algumas inconsistências ao fazer a leitura dos dados tanto na conexão via SPI, como na UART. Ao fazer a requisição dos dados, a placa apenas respondia ao comando de ECHO, que verifica a disponibilidade de conexão entre placa e o chip.

6.4 INTEGRAÇÃO DO SISTEMA AO HPA

A *Figura 51* representa a idealização da fixação do suporte do dispositivo móvel ao *cockpit* do HPA.

Figura 51 – Suporte do dispositivo móvel



Fonte:Próprio Autor

7 CONCLUSÕES

Com base no que foi apresentado e nos resultados obtidos, a estrutura desenvolvida durante este projeto atendeu ao objetivo de criar uma interface funcional de navegação e de monitoramento IoT para o HPA. O sistema utilizou vários protocolos, linguagens de programação e *hardwares* que estão em alta no mercado como aplicativo feito com o framework em React Native, microcontrolador ESP32 e Node-RED.

Tanto o aplicativo móvel quanto a interface Web exibiram os dados de maneira fácil e direta, características essenciais considerando que o usuário do HPA estará em voo e com movimentação limitada, para isso foi implementada uma interface setorizada e visual contendo três tópicos principais, a navegação, localização e a conectividade dos sensores, com intuito de ter rápido acesso a todas as telas.

A comunicação via MQTT do celular com servidor permitiu que a aquisição dos dados pelos sensores fosse acessada e exibida pela equipe que acompanha o piloto do HPA, a fim de ter mais segurança e assertividade da real condição do piloto e da aeronave com uma interface que contém todas as informações críticas mostradas em uma única tela.

Estes sistemas possui uma grande capacidade de empreendimento por utilizar um microcontrolador popular e de baixo-custo aliado a sensores encontrados no próprio *Smartphone* bem como um sistema arquitetado de forma modular com uma base robusta permitindo uma fácil integração com novas tecnologias e dispositivos.

8 SUGESTÃO DE CONTINUAÇÃO

Em relação ao hardware, a fim de diminuir o peso do protótipo será necessário a compactação, em especial o layout da caixa, já ao sistema microcontrolado há a necessidade de implementar o uso de outra biblioteca *bluetooth low energy* com multi-conexão que permita a conexão de forma estável e gerenciamento de inúmero dispositivos, uma vez que a atual não possui esse suporte, além disso, recomenda-se o uso do Visual Studio com *plugin Espressif* como IDE pois a fabricante do ESP32 provém mais suporte.

Para a implantação do sensor de glicose, seria interessante uma pesquisa específica para esse tema devido a importância e necessidade de assertividade além de um aprofundamento maior do estudo sobre os protocolos de comunicação NFC, principal empecilho na implementação com *software e hardware*.

Em relação ao *software* seria interessante que as informações do envio de dados via Web fossem encaminhados para um banco de dados, permitindo que cada usuário consiga acessar suas próprias informações trazendo históricos das suas atividades de forma individual. Por fim é relevante implementar com os dados da localização a criação da rota a ser representada em tempo real no mapa do aplicativo.

9 REFERÊNCIAS

ANDREI, L. **O Que é GitHub e Como Usá-lo - Guia Básico**. Disponível em: <<https://www.hostinger.com.br/tutoriais/o-que-github/>>. Acesso em: 18 nov. 2022.

ANDROID. **Android | A plataforma que redefine o impossível**. Disponível em: <https://www.android.com/intl/pt-BR_br/>. Acesso em: 20 nov. 2022.

ANGULAR. **Angular**. Disponível em: <<https://angular.io/>>. Acesso em: 12 nov. 2022.

APPLE. **iOS | Apple Developer Documentation**. Disponível em: <<https://developer.apple.com/documentation/packagedescription/platform/ios/>>. Acesso em: 20 nov. 2022.

ARDUINO. **Arduino IDE 1 Installation (Windows) | Arduino Documentation | Arduino Documentation**. Disponível em: <<https://docs.arduino.cc/software/ide-v1/tutorials/Windows>>. Acesso em: 20 nov. 2022.

BARATO, B. **Projeto de um sistema de controle para veículos aéreos não tripulados**. , 2014.

BARROS MÁRIO. **O que são sensores de glicemia? - Blog WinSocial**. Disponível em: <<https://blog.winsocial.com.br/tecnologia-sensor-de-glicemia/>>. Acesso em: 17 jan. 2022.

BERTULUCCI, C. **Sensor Ultrassom Industrial: Saiba como Especificar - Citisystems**. Disponível em: <<https://www.citisystems.com.br/sensor-ultrassom/>>. Acesso em: 20 nov. 2022.

CAMARA ROMULO. **PROTOCOLO I2C RESUMO**. 2022.

CAPSISTEMA. **Compreendendo os sensores de ECG e como programá-los para diagnosticar várias condições médicas - Cap Sistema**. Disponível em: <<https://capsistema.com.br/index.php/2021/07/06/compreendendo-os-sensores-de-ecg-e-como-programa-los-para-diagnosticar-varias-condicoes-medicadas/>>. Acesso em: 24 jan. 2022.

CULTURA AERONÁUTICA. **Cultura Aeronáutica: Tubo de pitot: como funciona?** Disponível em: <<http://culturaaeronautica.blogspot.com/2011/04/tubo-de-pitot-como-funciona.html>>. Acesso em: 20 nov. 2022.

DAMIRCHI, M. **Interfacing MPXV7002 Airspeed Sensor with Arduino - Electropeak.** Disponível em: <<https://electropeak.com/learn/interfacing-mpxv7002-airspeed-sensor-with-arduino/>>. Acesso em: 20 nov. 2022.

DET. **Sensor para Nível de Rio**,2020

ELITE. elite-it.com. 2020.

ESPRESSIF. ESP32 Series Datasheet Including. 2021.

FERNANDES, N. **O que é o protocolo MQTT? | HI Tecnologia.** Disponível em: <<https://www.hitecnologia.com.br/blog/o-que-e-protocolo-mqtt/>>. Acesso em: 25 jan. 2022.

FREESTYLE. **Medidor de Glicose FreeStyle Libre.** Disponível em: <<https://www.freestyle.abbott/br-pt/home.html>>. Acesso em: 20 nov. 2022.

GALAN, B. **Controle glicêmico através da atividade física: é possível?** Disponível em: <<https://beecorp.com.br/controle-glicemico-atraves-da-atividade-fisica-e-possivel/>>. Acesso em: 19 jan. 2022.

GHORAYEB NABIL. **Diabetes: cuidados que o portador deve ter antes e durante o exercício - eu atleta | globoesporte.com.** Disponível em: <<http://ge.globo.com/eu-atleta/saude/noticia/2013/04/diabetes-cuidados-que-o-portador-deve-ter-antes-e-durante-o-exercicio.html>>. Acesso em: 16 jan. 2022.

GOGONI RONALDO. **O que é Bluetooth? [tudo sobre a conexão sem fio] – Tecnoblog.** Disponível em: <<https://tecnoblog.net/responde/o-que-e-bluetooth/>>. Acesso em: 16 jan. 2022.

GOMES, M. et al. **CAROLINE COLAÇO SPINA GUILHERME MATTOS TAVARES LUCIANO NERES CORREIA PAES**, 2018

GRACE, R. Magnetometer Technology. **Space Microsystems and Micro/nano Satellites**, p. 341–360, 2018.

GUIMARÃES FÁBIO. **I2C - Mundo Projetado**. Disponível em: <<https://mundoprojetado.com.br/i2c/>>. Acesso em: 22 jan. 2022.

HÅKANSSON, J. **Interaction with IoT data to help users train smarter Utilizing IoT data & Velocity Based Training**, 2019. (Nota técnica).

HKOTSUBO. **O que é o timestamp? | hkotsubo_programming_blog**. Disponível em: <<https://hkotsubo.github.io/blog/2019-05-02/o-que-e-timestamp>>. Acesso em: 18 nov. 2022.

IGSPORT. **HR40 | iGPSport**. Disponível em: <<https://www.igpsport.com/pt/hr40>>. Acesso em: 6 jun. 2022.

INSTITUTO NEWTON C. BRAGA. **Como funciona o dinamômetro (MEC127)**. Disponível em: <<https://www.newtoncbraga.com.br/index.php/como-funciona/5453-mec127.html>>. Acesso em: 20 nov. 2022.

JAHANKIT. **JSN-SR04T-2.0**. Disponível em: <https://www.usinainfo.com.br/index.php?controller=attachment&id_attachment=723>. Acesso em: 20 nov. 2022.

JAKL, A. **NFC Tags, NDEF and Android (with Kotlin)**. Disponível em: <<https://www.andreasjakl.com/nfc-tags-ndef-and-android-with-kotlin/>>. Acesso em: 19 nov. 2022.

JAVASCRIPT. **JavaScript.com**. Disponível em: <<https://www.javascript.com/>>. Acesso em: 18 nov. 2022.

JEX, H. R.; MITCHELL, D. G. Stability and Control of the Gossamer Human-Powered Aircraft By Analysis and Flight Test. **NASA Contractor Reports**, 1982.

KNOPLOCH CAROL. **Melhor que o sutiã: cinto com chip monitora saltos e deixa seleção de vôlei livre de lesões - Jornal O Globo.** Disponível em: <<https://oglobo.globo.com/esportes/melhor-que-sutia-cinto-com-chip-monitora-saltos-deixa-selecao-de-volei-livre-de-lesoes-21865170>>. Acesso em: 16 jan. 2022.

LABSIS. **Proteus.** Disponível em: <<https://www.labsis.com.br/index.php/produtos/proteus>>. Acesso em: 20 nov. 2022.

LIMA, H.; PINHEIRO, M. **Nível de glicose no pré e pós-exercício de musculação em indivíduos da 3a idade.pdf.** , 2006.

MDN CONTRIBUTORS. **JavaScript | MDN.** Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acesso em: 12 nov. 2022.

MENDONÇA HÉLIO. **UART | Hélio Sousa Mendonça.** Disponível em: <<https://paginas.fe.up.pt/~hsm/docencia/comp/uart/>>. Acesso em: 22 jan. 2022.

META. **React Native · Learn once, write anywhere.** Disponível em: <<https://reactnative.dev/>>. Acesso em: 5 jun. 2022.

MÉTODO PROGRAMAR. **O que é Git e GitHub?** Disponível em: <<https://metodoprogramar.com.br/o-que-e-git-e-github/>>. Acesso em: 18 nov. 2022.

MIKROE. RFid click board TM Schematic v100. p. 2017, 2017.

MIKROW. **RFid Click Board - MikroElektronika.** Disponível em: <<https://www.mikroe.com/rfid-click>>. Acesso em: 6 jun. 2022.

MILLER BENNETT. **O Homem Que Mudou o Jogo (2011) - IMDb.** Sony Pictures, , 17 fev. 2012.

NEXTZPLUS. **NextzPlus.** , 2011.

NODE-RED. **Node-RED.** Disponível em: <<https://nodered.org/>>. Acesso em: 20 nov. 2022.

NOVAGE. **Eixos de controlo da aeronave Roll, Yaw e Pitch-NOVAGEO - Soluções em Sistemas de Informação Geográfica.** Disponível em: <https://www.novageo.pt/novageo/displayArticles?numero=38167&eixos_controlo_aeronave_roll_yaw_pitch>. Acesso em: 14 nov. 2022.

OLIVEIRA EULER. **Conhecendo o NodeMCU-32S ESP32 - BLOG MASTERWALKER SHOP.** Disponível em: <<https://blogmasterwalkershop.com.br/embarcados/esp32/conhecendo-o-nodemcu-32s-esp32>>. Acesso em: 22 jan. 2022.

OLIVEIRA JAILSON. **Arduino, ESP32 e ESP8266 – Comparação – XProjetos.** Disponível em: <<https://xprojetos.net/arduino-esp32-e-esp8266-comparacao/>>. Acesso em: 19 jan. 2022.

PASSARO, V. M. N. et al. Gyroscope technology and applications: A review in the industrial perspective. **Sensors (Switzerland)**, v. 17, n. 10, 2017.

PIMENTA, R. **Android vs. iOS: veja as principais diferenças! - Geek Blog.** Disponível em: <<https://geekblog.com.br/android-vs-ios-veja-as-principais-diferencas/>>. Acesso em: 18 nov. 2022.

POLAR. **Como monitorar a frequência cardíaca | Polar Brasil.** Disponível em: <<https://www.polar.com/br/smart-coaching/polar-heart-rate-measurement-technology>>. Acesso em: 24 jan. 2022.

PORTLANDCLINIC. **Why Do I Have Low Blood Sugars After Exercise ?**

PROTEUS. **PCB Design and Circuit Simulator Software.** Disponível em: <<https://www.labcenter.com/>>. Acesso em: 19 nov. 2022.

RAVOOF, S. **A Melhor Maneira de Aprender HTML de Graça (E Porque Você Deve) - Kinsta.** Disponível em: <<https://kinsta.com/pt/blog/aprender-html/>>. Acesso em: 18 nov. 2022.

RED HAT. **O que é API?** Disponível em: <<https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces>>. Acesso em: 25 jan. 2022.

RED HAT. **O que é IoT? - Internet das Coisas**. Disponível em: <<https://www.redhat.com/pt-br/topics/internet-of-things/what-is-iot>>. Acesso em: 18 nov. 2022.

REZENDE RONIÈRE. **Interface Gráfica em C# para Comunicação Serial (UART) - Embarcados**. Disponível em: <<https://www.embarcados.com.br/interface-grafica-para-uart/>>. Acesso em: 22 jan. 2022.

RIPKA. **Magnetometer - an overview | ScienceDirect Topics**. Disponível em: <<https://www.sciencedirect.com/topics/engineering/magnetometer>>. Acesso em: 26 jan. 2022.

SACCO FRANCESCO. **Módulo Tiny RTC I2C - Parte 1 - Embarcados**. Disponível em: <<https://www.embarcados.com.br/modulo-tiny-rtc-i2c-parte-1/#Comunicacao-I2C>>. Acesso em: 22 jan. 2022.

SANTIAGO PAULO. **Uma tecnologia que pode revolucionar a ciência do esporte: IoT – Jornal da USP**. Disponível em: <<https://jornal.usp.br/atualidades/uma-tecnologia-que-pode-revolucionar-a-ciencia-do-esporte-iot/>>. Acesso em: 16 jan. 2022.

SAP. **O que é IoT e como funciona? | SAP Insights**. Disponível em: <<https://www.sap.com/brazil/insights/what-is-iot-internet-of-things.html>>. Acesso em: 18 nov. 2022.

SEMICONDUCTORS, N. MPXV7002 Integrated Silicon Pressure Sensor - Data sheet. 2017.

SENSOR FREESTYLE. **Sensor Freestyle**. Disponível em: <<https://www.freestyle.abbott/br-pt/freestyle-libre-sensor.html>>. Acesso em: 18 jan. 2022.

SOUZA, I. **Afinal, o que é JSON e para que ele serve? Descubra agora!** Disponível em: <<https://rockcontent.com/br/blog/json/>>. Acesso em: 20 nov. 2022.

STAGES CYCLING. **Stages Power L - Shimano Dura Ace 9000 POWER METER - Factory Install | Stages Cycling**. Disponível em: <https://stagescycling.com/en_us/stages-power-l-shimano-dura-ace-9000-power-meter-factory-install>. Acesso em: 16 jan. 2022.

STMICROELECTRONICS. **NFC Technology - Essentials & Insights**. Disponível em: <https://www.st.com/content/st_com/en/support/learning/essentials-and-insights/connectivity/nfc.html>. Acesso em: 19 nov. 2022.

STRAVA. **Recursos Strava | Monitoramento em GPS, Mapas, Análises, Desafiar amigos, Encontrar melhores corridas e pedaladas**. Disponível em: <<https://www.strava.com/features>>. Acesso em: 16 jan. 2022.

SUSU. **SUSU - Human Powered Aircraft**. Disponível em: <<https://www.susu.org/groups/human-powered-aircraft>>. Acesso em: 18 nov. 2022.

TEIXEIRA EVANDRO. **Biblioteca de software de I2C para FRDM-KL25Z - Embarcados**. Disponível em: <<https://www.embarcados.com.br/biblioteca-i2c-para-frdm-kl25z/>>. Acesso em: 22 jan. 2022.

TOWNSEND, K. **GATT | Introduction to Bluetooth Low Energy | Adafruit Learning System**. Disponível em: <<https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt>>. Acesso em: 6 jun. 2022.

TRAININGPEAKS. **Training Software and Apps For Endurance Athletes**. Disponível em: <<https://www.trainingpeaks.com/athlete-features/>>. Acesso em: 16 jan. 2022.

TYPESCRIPTLANG. **TypeScript: JavaScript With Syntax For Types**. Disponível em: <<https://www.typescriptlang.org/>>. Acesso em: 12 nov. 2022.

USINAINFO. **JSN-SR04T Sensor Ultrassônico a Prova D'água para Arduino - Usinainfo**. Disponível em: <<https://www.usinainfo.com.br/sensor-ultrassonico/sensor-ultrassonico-jsn-sr04t-a-prova-d-agua-modulo-para-arduino-4704.html>>. Acesso em: 20 nov. 2022.

VARGAS GABRIEL. **Medidores de potência: o que saber antes de comprar - parte 1 - Bikemagazine**. Disponível em: <<https://www.bikemagazine.com.br/2018/11/medidores-de-potencia-o-que-saber-antes-de-comprar-parte-1/>>. Acesso em: 16 jan. 2022.

VEIGA, F. **(6) 10 Motivos que fazem no Node-RED sua melhor opção de Low-Code | LinkedIn**. Disponível em: <<https://www.linkedin.com/pulse/10-motivos-que-fazem-node-red-sua-melhor-opção-de-low-code-veiga/?originalSubdomain=pt>>. Acesso em: 20 nov. 2022.

VISUAL STUDIO. **Visual Studio Code - Code Editing. Redefined**. Disponível em: <<https://code.visualstudio.com/>>. Acesso em: 18 nov. 2022.

W3 SCHOOLS. **HTML Tutorial**. Disponível em: <<https://www.w3schools.com/html/>>. Acesso em: 12 nov. 2022.

WISINTAINER, M. **ESP32 Bluetooth Low Energy (BLE) no Arduino IDE | Tutoriais Aleatórios para Nerds**. Disponível em: <<https://randomnerdtutorials.com/esp32-bluetooth-low-energy-ble-arduino-ide/>>. Acesso em: 6 jun. 2022.

ZENVIA. **O que é API, para que serve e como usar? | Zenvia**. Disponível em: <<https://www.zenvia.com/blog/apis-entenda-o-que-sao-e-como-funcionam/>>. Acesso em: 26 jan. 2022.

ZWIFT. **Get Training Plans & Workouts | Zwift**. Disponível em: <<https://www.zwift.com/training-plans>>. Acesso em: 16 jan. 2022.