

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO DE ESPECIALIZAÇÃO EM INTERNET DAS COISAS

PAULO JOSE DE CARVALHO NETO

**SISTEMA DE MONITORAMENTO DE TREINOS FÍSICOS EM
ACADEMIAS**

MONOGRAFIA DE ESPECIALIZAÇÃO

CURITIBA
2019

PAULO JOSE DE CARVALHO NETO

SISTEMA DE MONITORAMENTO DE TREINOS FÍSICOS EM ACADEMIAS

Monografia de Especialização, apresentada ao Curso de Especialização em Internet das Coisas, do Departamento Acadêmico de Eletrônica – DAELN, da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Especialista.

Orientador: Prof. Dr. Ohara Kerusauskas Rayel

CURITIBA
2019



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Curitiba

Diretoria de Pesquisa e Pós-Graduação
Departamento Acadêmico de Eletrônica
Curso de Especialização em Internet das Coisas



TERMO DE APROVAÇÃO

SISTEMA DE MONITORAMENTO DE TREINOS FÍSICOS EM ACADEMIAS

por

PAULO JOSE DE CARVALHO NETO

Esta monografia foi apresentada em 28 de Novembro de 2019 como requisito parcial para a obtenção do título de Especialista em Internet das Coisas. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dr. Ohara Kerusauskas Rayel
Orientador

Prof. M. Sc. Danillo Leal Belmonte
Membro titular

Prof. M. Sc. Omero Francisco Bertol
Membro titular

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

Dedico esse trabalho a minha família e amigos por sempre me apoiarem e acreditaram no meu potencial.

AGRADECIMENTOS

Certamente os agradecimentos são muitos, muitas pessoas participaram desse momento da minha vida. Aqui uma breve citação aos que estiveram mais presentes na realização desse trabalho.

Agradeço ao meu orientador Prof. Dr. Ohara, pela sabedoria com que me guiou nesta trajetória.

Aos amigos e colegas de curso.

A todo corpo docente do Departamento Acadêmico de Eletrônica.

E um grande agradecimento a minha família que esta presente ao meu lado em todos desafios que enfrentei até chegar nesse momento hoje.

RESUMO

CARVALHO NETO, Paulo José de. **Sistema de monitoramento de treinos físicos em academias**. 2019. 48 p. Monografia de Especialização em Internet das Coisas, Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2019.

O surgimento da Internet das Coisas ainda é muito recente, mas em tão pouco tempo já vem tendo um papel muito importante e impactante na sociedade. A Internet das Coisas já auxilia em muitas áreas do nosso cotidiano, presente em nossas casas, em nossos caminhos para o trabalho e também em nossas atividades físicas. Explorando mais essa área, hoje encontramos Internet das Coisas quando corremos, monitorando nossa velocidade, distância, tempo, batimentos cardíacos, etc.. A ideia desse projeto é utilizar a Internet das Coisas para diagnosticar os movimentos de um treino de musculação em uma academia. Neste sistema uma pessoa em que esteja realizando seu treinamento passa a monitorar seu desempenho em tempo real, pelo seu smartphone, detalhando os indicadores da execução de cada exercício que esta realizando. Especificamente serão monitoradas a quantidade de repetições de cada exercício, além de salvar cada exercício e ter um histórico de cada treino em que o sistema foi utilizado.

Palavras-chave: Internet das Coisas. Atividades físicas. Academias. Monitoramento. Tempo real.

ABSTRACT

CARVALHO NETO, Paulo José de. **Physical training monitoring system at gyms.** 2019. 48 p. Monografia de Especialização em Internet das Coisas, Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2019.

The emergence of the internet of things is still very recent, but in such a short time has already been playing a very important and impacting role in society. The internet of things already helps in many areas of our daily lives, present in our homes, in our ways to work and also in our physical activities. Exploring this area further, today we find internet of things as we run, monitoring our speed, distance, time, heartbeat, etc. The idea of this project is to use the Internet of Things to diagnose the movements of a bodybuilding workout in a gym. In this system a person who is conducting his training starts monitoring his performance in real time, by his smartphone, detailing the indicators of the execution of each exercise that he is performing. Specifically will be monitored the amount of repetitions of each exercise, save each exercise and have a history of each workout in which the system was used.

Keywords: Internet of Things. Physical activities. Gym. Monitoring. Real time.

LISTA DE FIGURAS

Figura 1 - Dispositivos conectados a internet.....	14
Figura 2 - Distribuição Internet das Coisas	14
Figura 3 - Fluxograma MQTT	16
Figura 4 - CoAP GET exemplo.....	17
Figura 5 - Modelo banco de dados relacional.....	18
Figura 6 - Representação chave/valor JSON	19
Figura 7 - Raspberry Pi	20
Figura 8 - CC2650 LAUNCHPAD.....	21
Figura 9 - Placa NodeMCU	21
Figura 10 - Exemplo sensores IoT	22
Figura 11 - Exemplo interfaces com o usuário	23
Figura 12 - Exemplo interfaces por voz.....	24
Figura 13 - Fluxograma funcionamento MQTT.....	27
Figura 14 - Esquema circuito eletrônico	29
Figura 15 - Dinâmica do sistema com pinos	30
Figura 16 - Fluxograma funcionamento aplicativo.....	32
Figura 17 - Estrutura JSON.....	33
Figura 18 - Diagrama conexão banco de dados.....	33
Figura 19 - Broker conectado.....	34
Figura 20 - Comunicação wi-fi e broker.....	35
Figura 21 - Protótipo dispositivo	36
Figura 22 - Coleção usuários Firebase	36
Figura 23 - Coleção trainings Firebase	37
Figura 24 - Coleção exercises Firebase.....	37
Figura 25 - Coleção rep Firebase.....	38
Figura 26 - Demonstração Tabs.....	39
Figura 27 - Tela login e Novo Usuário.....	40
Figura 28 - Autenticação Firebase	41
Figura 29 - Página inicial de treinos	41
Figura 30 - Tela de treino antes exercícios e depois.....	42
Figura 31 - Exemplo QR Code e informação.....	43
Figura 32 - Tela contador de repetições.....	43
Figura 33 - Tela sequência contador.....	44
Figura 34 - Sequência do movimento.....	44

LISTA DE TABELAS

Tabela 1 - Quantidade de dispositivos nos protocolos IPv4 e IPv6.....	15
Tabela 2 - Características e meios do projeto	25
Tabela 3 - Interligação NodeMCU e Sensor Ultrassônico	29
Tabela 4 - Conversão formato JSON para objeto	40

SUMÁRIO

1 INTRODUÇÃO	10
1.1 PROBLEMA.....	10
1.2 OBJETIVOS	11
1.2.1 Objetivo Geral	11
1.2.2 Objetivos Específicos	11
1.3 JUSTIFICATIVA	11
1.4 ESTRUTURA DO TRABALHO	12
2 METODOLOGIA	13
2.1 INTERNET DAS COISAS	13
2.2 TECNOLOGIAS IOT	15
2.2.1 Rede.....	15
2.2.2 Transporte	15
2.2.3 Aplicação.....	16
2.2.3.1 MQTT	16
2.2.3.2 CoAP	17
2.2.4 Banco de Dados.....	18
2.2.4.1 Banco de dados relacionais	18
2.2.4.2 Banco de dados não relacionais	18
2.2.5 Hardware	19
2.2.5.1 Raspberry PI	19
2.2.5.2 CC2550	20
2.2.5.3 NodeMCU.....	21
2.2.5.4 Sensores	22
2.2.6 Interface com o Usuário	23
3 DESENVOLVIMENTO DO PRODUTO	25
3.1 CRITÉRIOS DO PRODUTO	25
3.2 SELEÇÃO DA ESTRUTURA	25
3.3 DEFINIÇÃO DOS MEIOS	25
3.3.1 Comunicação	26
3.3.1.1 Broker.....	26
3.3.1.2 Clients	26
3.3.2 Dispositivo	27
3.3.3 Software	28
3.3.4 Hardware.....	28
3.3.5 Dinâmica	29
3.3.6 Interface com o Usuário	30
3.3.7 Banco de Dados.....	32
4 RESULTADOS OBTIDOS	34
4.1 BROKER	34
4.2 CLIENTS	35
4.3 BANCO DE DADOS	36
4.4 INTERFACE COM O USUÁRIO	38
5 CONCLUSÃO	45
REFERÊNCIAS	46

1 INTRODUÇÃO

A evolução da tecnologia está em ritmo acelerado nas últimas décadas, e um dos principais objetivos é melhorar a qualidade de vida da população. Uma das áreas em que existe mais desenvolvimento é a de saúde, seja para auxiliar médicos a tomarem conta de seus pacientes ou para as pessoas se prevenirem de doenças e levarem uma vida mais saudável.

Boa parte dos produtos desenvolvidos para este nicho do mercado se encaixa em dispositivos de Internet das Coisas (IoT, do inglês *Internet of Things*). Já existem, por exemplo sistemas de monitoramento implantados no corpo para controle de doenças, lentes de contato que medem níveis de glicose obtidos através do fluido lacrimal e enviam os dados para um smartphone. Além da prevenção de doenças, a área da saúde também engloba a área do bem-estar, que será o foco desse projeto. Já existem dispositivos de Internet das Coisas desenvolvidos para diversas atividades físicas como corrida, natação, futebol dentre outras, visando monitoramento de batimentos cardíacos, velocidade, distância, perda de calorias, etc. No entanto, para um treino de musculação por exemplo, onde um atleta pode ter um controle do treino que está executando e ter a possibilidade de compartilhar com um profissional, quantas repetições realizou em cada exercício, em quanto tempo e se o movimento estava correto, não se encontrou um dispositivo de IoT que atenda este mercado. Assim, este projeto visa preencher esta lacuna.

1.1 PROBLEMA

Em muitas academias os sistemas de monitoramento de treino ainda são muito precários, não tendo um acompanhamento de como os atletas desempenham seu treino. Hoje com a tecnologia já avançada é possível criar sistemas de monitoramento que auxiliam as pessoas no dia a dia. Essas tecnologias podem ser levadas para as academias para os usuários terem um controle do que estão fazendo nos exercícios, tirarem um maior proveito e alcançarem um maior desempenho dos seus treinos.

1.2 OBJETIVOS

Nesta seção são apresentados os objetivos geral e específicos do trabalho, relativos ao problema anteriormente apresentado.

1.2.1 Objetivo Geral

Desenvolver um dispositivo de monitoramento de treinos em academias, para que se possa controlar a quantidade de repetições realizadas em um determinado exercício.

1.2.2 Objetivos Específicos

Para atender ao objetivo geral neste trabalho de conclusão de curso de especialização os seguintes objetivos específicos serão abordados:

- Desenvolver sistema para contagem de repetições de determinado exercício em tempo real;
- Desenvolver aparelho para contagem de repetição;
- Desenvolver aplicativo para monitoramento em tempo real;
- Desenvolver banco de dados para histórico de treinos e sistemas utilizados;
- Aplicação de protocolo MQTT para integração das partes.

1.3 JUSTIFICATIVA

Apesar de academias novas hoje em dia possuírem muitos equipamentos modernos de última geração, a Internet das Coisas ainda não chegou completamente nesse cenário. A interação da academia com o seu frequentador pode ser muito maior com a tecnologia que existe hoje em dia.

O acompanhamento dos treinos dos clientes nas academias muitas vezes nem é acompanhado por um profissional, um cliente simplesmente pode chegar na academia, fazer o treino que quiser, com o peso que quiser e ir embora sem nem mesmo falar com um profissional da área. E isso é um problema por alguns fatores, como lesões por excesso de peso, lesões por executar o movimento de forma

errônea, ou mesmo por não obter o resultado esperado com o treino que está executando.

Levando a Internet das Coisas para dentro das academias é possível solucionar esses problemas, tendo o próprio cliente em suas mãos um meio de seguir e registrar seus treinos, e que um profissional possa ter acesso para auxiliá-lo sem mesmo ter interação dentro da academia. Isso pode assegurar a saúde do atleta, evitando que possam ocorrer lesões, além de ter a possibilidade de obter um maior desempenho em seus treinos.

1.4 ESTRUTURA DO TRABALHO

O trabalho apresentará a seguinte estrutura:

- Capítulo 1 – Introdução: Apresentado o tema, problema, objetivo geral, objetivos específicos e justificativa.
- Capítulo 2 – Metodologia: Apresentado os conceitos de Internet das Coisas e das tecnologias utilizadas no projeto.
- Capítulo 3 – Desenvolvimento do Produto: Apresentado os passos para o desenvolvimento do sistema proposto.
- Capítulo 4 – Resultados Obtidos: Apresentado o resultado final do projeto.
- Capítulo 5 – Conclusão: Apresentado possíveis ideias para trabalhos futuros.

2 METODOLOGIA

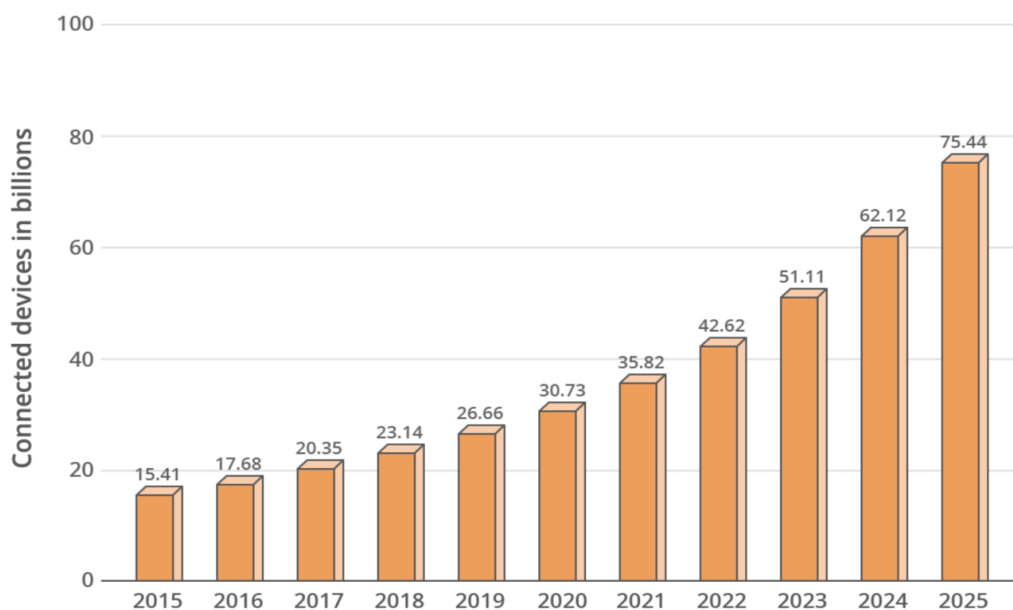
2.1 INTERNET DAS COISAS

A Internet das Coisas é uma estrutura global, que através da interconexão de dispositivos utilizando padrões de comunicações interoperáveis, provê serviços diversos, baseados em informações produzidas ou controladas no mundo físico por estas coisas interconectadas (ITU, 2012). O termo foi primeiramente utilizado em 1999 pelo grupo de pesquisa do *Massachusetts Institute of Technology* (MIT) Auto-ID Center. Kevin Ashton, um pesquisador do grupo que estudava conexões do RFID com a internet, de acordo com Lopez (2013), Ashton acreditava que se todos os objetos do dia a dia fossem equipados com identificadores e conectividade sem fio, eles poderiam se comunicar e ser gerenciados por computadores.

A Internet das Coisas surgiu com o objetivo de conectar vários objetos, visando facilitar e automatizar processos que são utilizados no dia a dia. Para que isso fosse possível, varias tecnologias como redes de sensores sem fio, sistemas embarcados, microcontroladores e comunicação de rede tiveram que evoluir para que cada vez mais dispositivos pudessem se conectar a internet e trazer a Internet das Coisas no dia a dia da população em diversas áreas de atuação. Tais dispositivos, com o devido gerenciamento e com a devida implementação de recursos de segurança, serão empregados nas mais diversas áreas, como por exemplo: automação residencial, transporte, educação, segurança, energia, entre outros (MIRAZ *et al.*, 2015).

A Internet das Coisas vem crescendo em um ritmo acelerado e o mercado está investindo cada vez mais nessa tecnologia. De acordo com Columbus (2018) uma pesquisa da *International Data Corporation* (IDC) afirma que até o ano de 2022 será investido até 1,2 trilhão de dólares no mundo todo em tecnologias de Internet das Coisas. Segundo pesquisa Bustamante (2019), por volta de 42 bilhões de dispositivos estarão conectados a internet até o ano de 2022, e podendo chegar até 75 bilhões até o ano de 2025. A Figura 1 mostra o gráfico de dispositivos conectados a internet ao longo dos anos.

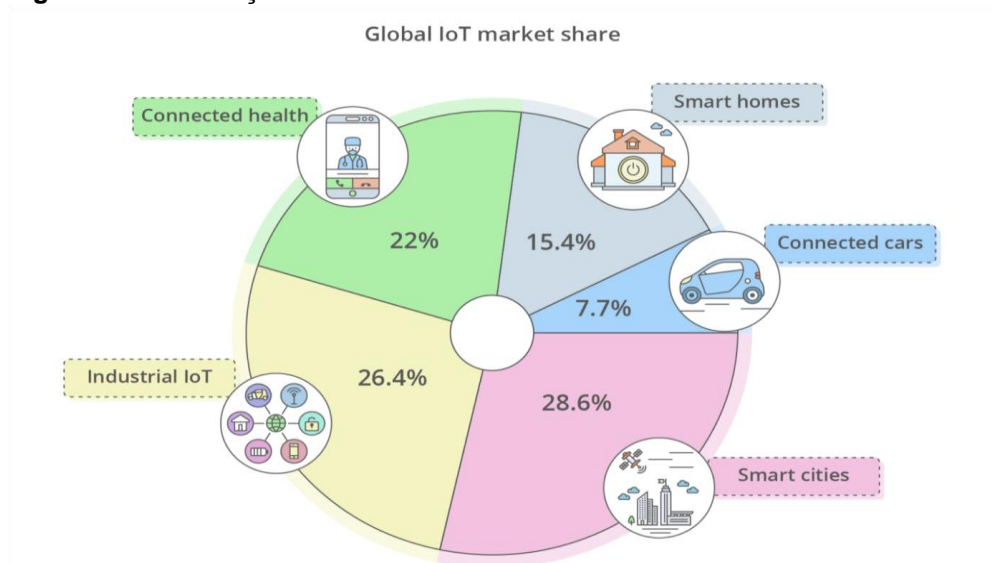
Figura 1 - Dispositivos conectados a internet



Fonte: Bustamante (2019).

Os números expressivos relacionados a IoT se devem as diversas possibilidades de aplicação, nas áreas mais heterogêneas é possível encontrar dispositivos que poderiam estar conectados a internet e que auxiliariam no desenvolvimento de algum trabalho. A Figura 2 mostra a distribuição da Internet das Coisas nas diversas áreas.

Figura 2 - Distribuição Internet das Coisas



Fonte: Bustamante (2019).

2.2 TECNOLOGIAS IOT

Esta sessão tem por objetivo apresentar algumas das tecnologias que são utilizadas no âmbito da Internet das Coisas.

2.2.1 Rede

Os protocolos mais utilizados hoje, não só para a Internet das Coisas, mas para qualquer tipo de aplicação na internet são o IPv4 e o IPv6. O IPv6 vem sendo implementado em substituição ao IPv4 visando principalmente endereçar um número muito maior de dispositivos.

A Tabela 1 mostra a discrepância de dispositivos possíveis de endereçamento IPv4 para o IPv6. O IPv4 possui um endereço de 32 bits dividido em 4 octetos representados em decimal. Já o IPv6 possui um endereço de 128 bits divididos em 8 grupos de 16 bits, representados por 4 dígitos em hexadecimal.

Tabela 1 - Quantidade de dispositivos nos protocolos IPv4 e IPv6

Protocolo	Número de dispositivos
IPv4	4×10^9
Ipv6	$3,4 \times 10^{38}$

Fonte: Autoria própria.

Assim, para dispositivos de IoT é fundamental a utilização de IPv6, já que existe disponibilidade de endereços válidos para todos os dispositivos.

2.2.2 Transporte

Existem dois principais meios de transporte sobre o IP, sendo eles TCP (*Transmission Control Protocol*) e UDP (*User Datagram Protocol*). O protocolo UDP oferece um serviço de entrega mínimo, e não garante a entrega da mensagem que está sendo enviada da origem para o destino, já o protocolo TCP oferece confiabilidade e entrega em sequência dos dados, garantindo que os mesmos chegaram ao destino (BRADEN, 1989).

Para evitar o overhead introduzido pelo TCP, dispositivos com restrição de energia e de poder de processamento (como é o caso dos dispositivos de IoT) frequentemente empregam UDP como seu protocolo de transporte.

2.2.3 Aplicação

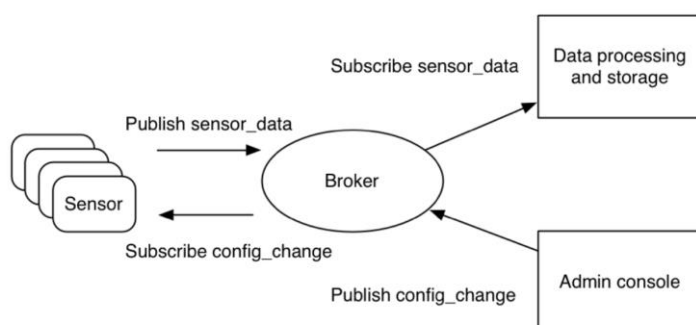
A camada de aplicação possui dois protocolos que são os mais conhecidos para a Internet das Coisas. São eles o MQTT e o CoAP. O MQTT é um protocolo que roda em cima da camada de transporte TCP, posteriormente foi criado também o MQTT-SN que funciona com o UDP. Já o CoAP foi projetado para operar com transporte UDP, pois é dedicado para dispositivos IoT.

2.2.3.1 MQTT

MQ Telemetry Transport (MQTT) foi desenvolvido por Andy Stanford-Clark da IBM em 1999 e é baseado em um modelo *publish/subscribe*, utilizado principalmente em dispositivos restritos com baixa largura de banda, alta latência ou não confiáveis (MQTT, 2019).

O modelo *publish/subscribe* foi criado para aplicações em que uma única mensagem seja distribuída para múltiplos usuários que tem interesse nesse tipo de mensagem, podendo manter o cliente que publicou a mensagem independente dos clientes que estão inscrevendo a mensagem, tornando a comunicação assíncrona. No protocolo MQTT os *Clients* podem publicar e se inscrever em tópicos e o *Broker* tem a função de distribuir a publicação para os *Clients* que estão inscritos nestes tópicos. O fluxograma do protocolo é mostrado na Figura 3 (YUAN. 2017).

Figura 3 - Fluxograma MQTT



Fonte: Yuan (2017).

Essa transmissão de mensagens é feita através de tópicos, onde qualquer cliente conectado ao *broker* pode inscrever-se em qualquer tópico do *broker*, e qualquer cliente também pode publica em qualquer tópico do *broker*, passando qualquer mensagem. Assim que feita a publicação, todos os *Clients* que se

inscreveram no tópicos que a mensagem foi publicada, vão receber as informações enviadas na mensagem.

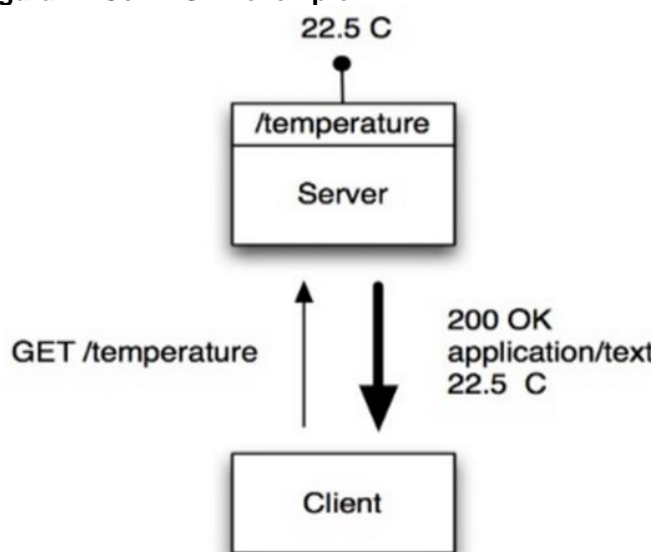
2.2.3.2 CoAP

O *Constrained Application Protocol* (CoAP) é um protocolo da camada de aplicação desenvolvido pela IETF Restful Environment para fornecer uma interface RESTful (HTTP) leve. *Representation State Format* (REST) é uma interface HTTP entre cliente e servidor (SALMAN, 2015).

O CoAP é um protocolo web de transferência para uso com nós e redes restritas da Internet das Coisas, é como a transferência de documentos em HTTP, porém para dispositivos reduzidos, podendo assim os sensores de baixa potência utilizarem o serviço RESTful (COAP, 2019).

Utilizando a transferência através de UDP o CoAP é dividido em duas subcamadas principais, mensagens e request/response. As mensagens são responsáveis pela confiabilidade e duplicação de mensagens, enquanto request/response é responsável pela comunicação. As mensagens, como no HTTP podem ser de GET, PUT, PUSH e DELETE, como no exemplo da Figura 4 onde um cliente faz uma requisição para saber a temperatura utilizando o método GET, e recebe a resposta com o valor da temperatura.

Figura 4 - CoAP GET exemplo



Fonte: Autoria própria¹.

¹ Notas de aula do Prof. Dr. Guilherme Luiz Moritz. Disponível em: <<http://paginapessoal.utfpr.edu.br/moritz>>. Acesso em: 25 out. 2019.

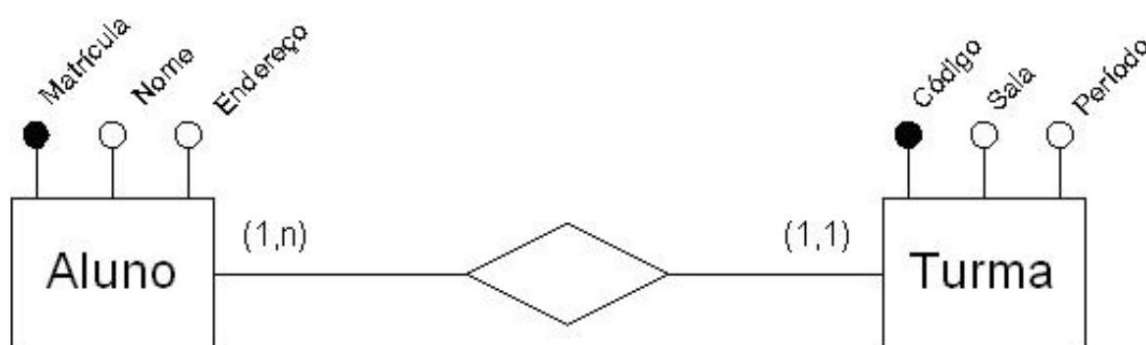
2.2.4 Banco de Dados

Um banco de dados pode ser relacional, que é do método tradicional que utiliza linguagem SQL (*Structured Query Language*) e o método não tradicional que seriam os bancos de dados que não utilizam a linguagem SQL.

2.2.4.1 Banco de dados relacionais

Nesse modelo os dados são armazenados em tabelas que possuem relações entre si. A linguagem SQL permite criar operações de leitura e escrita juntando dados de diferentes tabelas. Alguns exemplos de bancos de dados relacionais são MySQL, Oracle e SQLServer. Na Figura 5 é ilustrado um exemplo de relação entre duas tabelas.

Figura 5 - Modelo banco de dados relacional



Fonte: Rezende (2006).

2.2.4.2 Banco de dados não relacionais

Nos modelos não relacionais os dados são armazenados em pares de chave/valor e possuem um esquema flexível de armazenamento, podendo dar mais escalabilidade, velocidade e acesso aos dados. Normalmente os dados armazenados em um banco de dados não relacional estão em formato JSON (*JavaScript Object Notation*). Alguns exemplos de bancos de dados não relacionais utilizados hoje em dia são o Firebase, Redis e MongoDB. A Figura 6 demonstra uma chave/valor com os dados em JSON.

Figura 6 - Representação chave/valor JSON

Key	Document
1001	<pre>{ "CustomerID": 99, "OrderItems": [{ "ProductID": 2010, "Quantity": 2, "Cost": 520 }, { "ProductID": 4365, "Quantity": 1, "Cost": 18 }], "OrderDate": "04/01/2017" }</pre>
1002	<pre>{ "CustomerID": 220, "OrderItems": [{ "ProductID": 1285, "Quantity": 1, "Cost": 120 }], "OrderDate": "05/08/2017" }</pre>

Fonte: Microsoft Azure (2018).

2.2.5 Hardware

Em geral, os dispositivos de IoT são baseados em microcontroladores, esses por definição são como um computador em um único chip contém “processador, memória, periféricos de entrada e de saída, temporizadores, dispositivos de comunicação serial, dentre outros” (PENIDO, 2013).

Além disto, usualmente os dispositivos de IoT são equipados com sensores e atuadores que vão ser responsáveis pela interação com o mundo físico, obtendo e armazenando dados, processando informações e interagindo de acordo com funções específicas.

Esse capítulo vai apresentar alguns hardwares que são mais utilizados para as aplicações da Internet das Coisas.

2.2.5.1 Raspberry Pi

Lançado em 2012 pela Raspberry Pi Foundation, uma empresa com intuito de educar pessoas em computação e criar fácil acesso para a educação de computadores, o Raspberry Pi é um microcomputador completo em apenas uma placa. Tem portanto baixo custo, e é capaz de executar o sistema operacional Linux, além de oferecer uma serie de pinos GPIO (*General Purpose Input/Output*) que permite controlar componentes físicos (OPENSOURCE.COM, 2019).

O Raspberry Pi é o terceiro computador mais vendido no mundo (OPENSOURCE.COM, 2019). Possui praticamente o mesmo tamanho de um cartão de crédito, sendo necessário apenas conectá-lo via HDMI a uma TV ou monitor, utilizando teclado e mouse em suas portas USB. Desta forma é possível utilizá-lo para aprender a programar e criar projetos de eletrônica, além de realizar atividades de um *Desktop* normal. A Figura 7 ilustra uma placa Raspberry Pi.

Figura 7 - Raspberry Pi

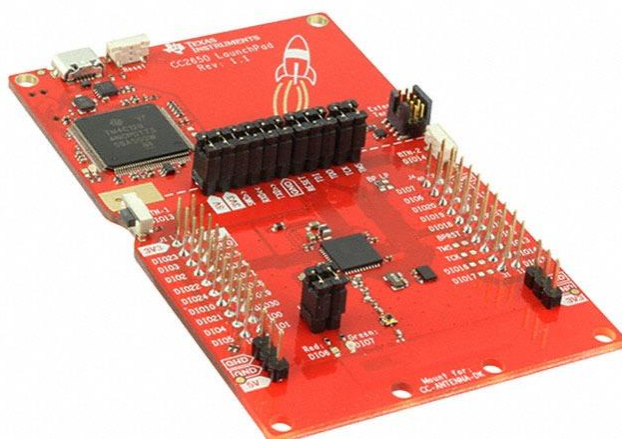


Fonte: Opensource.com (2019).

2.2.5.2 CC2550

Criado pela empresa *Texas Instruments* o CC2650 é um MCU (*Microcontroller Unit*) com conexão sem fio direcionado para *Bluetooth*, *ZigBee* e *6LoWPAN*. Dispositivo de baixo custo e baixo consumo de energia, permitindo operação com pequenas baterias. Possui um controlador de sensor de potência exclusivo, esse permite a coleta de dados analógicos e digitais do microcontrolador autonomamente enquanto o resto do Sistema está em *standby*. É portanto um ótimo dispositivo para a Internet das Coisas e ideal para suas diversas aplicações.

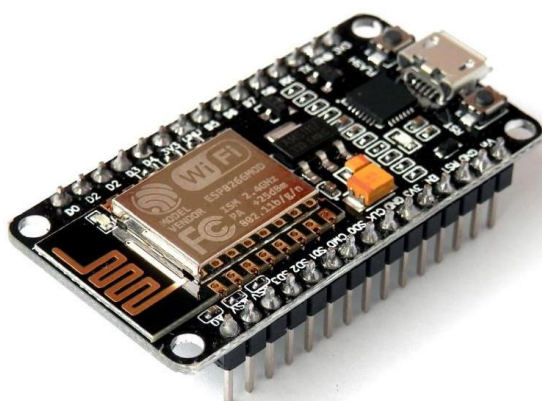
Uma das placas de desenvolvimento que incorpora o microcontrolador é conhecida como CC2650 LAUNCHPAD, nela encontra-se a pinagem GPIO para auxiliar na integração com outros dispositivos e a porta USB para a depuração do Código. Na Figura 8 tem-se a placa de desenvolvimento da *Texas Instruments* (TEXASINSTRUMENTS, 2015).

Figura 8 - CC2650 LAUNCHPAD

Fonte: Autoria própria.

2.2.5.3 NodeMCU

Criado pela empresa Espressif Systems no ano de 2013 o NodeMCU é uma plataforma livre da Internet das Coisas. A placa de desenvolvimento, apresentada na Figura 9, é baseada no módulo ESP-12E, que é um circuito integrado em alta escala que apresenta funcionalidades de um microcontrolador com uma estrutura de comunicação Wi-Fi dentro do chip (ESPRESSIF, 2019).

Figura 9 - Placa NodeMCU

Fonte: Autoria própria.

A placa inclui o módulo ESP8266, uma interface usb-serial e um regulador de tensão 3,3V. O modelo de programação é assíncrono e orientado a eventos, podendo ser programado em Lua, Python ou pela própria interface do Arduino, como o Raspberry Pi, também possui pinos GPIO que permite controlar componentes físicos com facilidade.

2.2.5.4 Sensores

Por definição sensor é um dispositivo que responde a um estímulo físico, como calor, luz, som, pressão, magnetismo ou um movimento particular, e transmite um impulso de resultado como medida ou operação de controle (SENSOR, 2019).

A leitura destes sensores é realizada em geral por um microcontrolador, e posteriormente disponibilizada ao usuário. A interface dos sensores com o controlador normalmente é feita por SPI (*Serial Peripheral Interface*) para sensores digitais e ADC (*Analog to Digital Converter*) para sensores analógicos. Quando a disponibilização desses dados ocorre via internet, pode-se caracterizar o conjunto como um dispositivo IoT. A Figura 10 ilustra alguns sensores IoT.

Figura 10 - Exemplo sensores IoT



Fonte: Autoria própria.

Segundo Theo (2018) alguns dos sensores mais utilizados são:

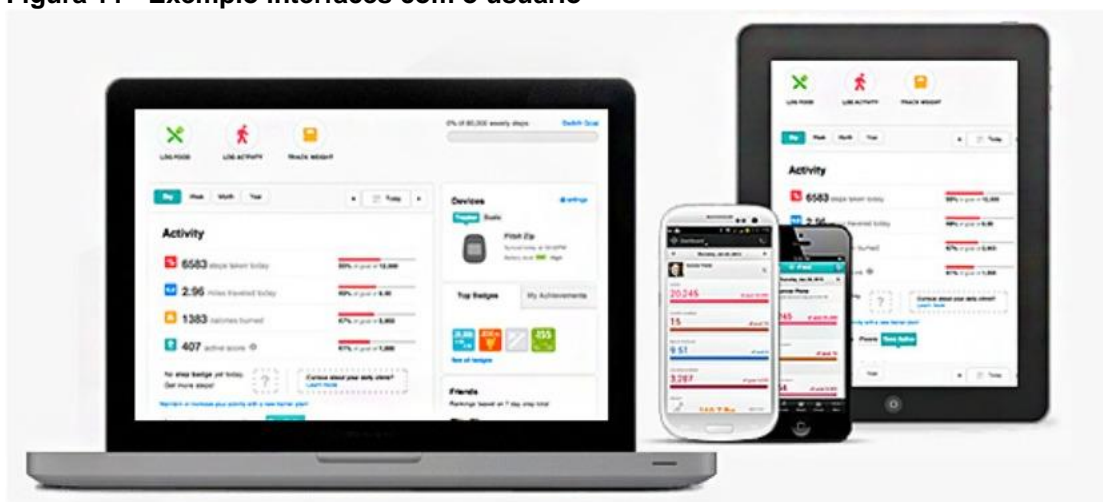
- sensores de temperatura, umidade e movimento
- sensores de gás, fumaça e pressão
- sensores de imagens
- acelerômetros
- sensores IR (*Infra-Red*)
- sensor de proximidade

2.2.6 Interface com o Usuário

Interface com o usuário é a interação humana com o computador, onde um usuário pode interagir com os dispositivos para cumprir tarefas e obter informações de seu estado. Uma interface com o usuário deve ser intuitiva, eficiente e amigável. Essa tecnologia vem crescendo rapidamente, há pouco tempo atrás a principal forma de interação com as máquinas era simplesmente por linhas de comando. Atualmente, existem interfaces gráficas (GUI, do inglês *Graphical User Interface*), onde o usuário pode obter várias informações com gráficos, botões, alertas e notificações para poder interagir com os dispositivos (LEVEREGE, 2019).

Para a Internet das Coisas a interface com o usuário, na maioria das vezes é utilizada para a visualização dos dados coletados pelos sensores, ou seja, um meio de visualização que seriam os aplicativos ou interfaces web, como observado na Figura 11.

Figura 11 - Exemplo interfaces com o usuário



Fonte: Leverage (2019).

Existem outros meios de comunicação entre usuários e máquinas, um exemplo é pela voz, onde é possível enviar comandos ou solicitar informações que serão obtidas através de inteligência artificial. Exemplos dessas interfaces, apresentados na Figura 12, são o Google Home (lado esquerdo) e Amazon Alexa (lado direito) (EVERYINTERACTION, 2019).

Figura 12 - Exemplo interfaces por voz



Fonte: EveryInteraction (2019).

3 DESENVOLVIMENTO DO PRODUTO

Neste capítulo serão demonstradas as tecnologias utilizadas para o desenvolvimento do projeto, explicando como foi executada cada etapa do Sistema.

O sistema consiste em um servidor que gerencia mensagens enviadas pelos dispositivos que, possuem um contador de repetições do exercício. Com o auxílio de um aplicativo de celular, o *Client* pode verificar o Contador de repetições em tempo real e quando finalizada o treino, os dados são salvos em um banco de dados.

3.1 CRITÉRIOS DO PRODUTO

Para o desenvolvimento do produto estipulou-se as seguintes funcionalidades.

- Aplicativo para interface com usuário
- Comunicação cliente/servidor
- Banco de dados para guardar informações
- Dispositivo para Contador de repetições

3.2 SELECÃO DA ESTRUTURA

A partir de cada uma das características citadas acima pode-se verificar o melhor componente ou meio para criar o sistema (Tabela 2).

Tabela 2 - Características e meios do projeto

Características	Meios
Interface com Usuário	Aplicativo celular
Comunicação	MQTT, RaspberryPI
Banco de dados	Firebase
Dispositivos (<i>Clients</i>)	NodeMCU, Sensor Ultrassônico

Fonte: Autoria própria.

3.3 DEFINIÇÃO DOS MEIOS

A partir da análise dos meios para a realização de cada uma das características foi possível desenvolver o sistema da seguinte forma.

3.3.1 Comunicação

Para a comunicação foi decidido utilizar a tecnologia do MQTT. Pelos motivos de dar confiabilidade na transferência dos dados, pois se trata de uma aplicação onde não se pode perder informações, pela facilidade de se aplicar o modelo de tópicos nos dispositivos e também pela facilidade de criar um servidor broker.

3.3.1.1 Broker

Para o desenvolvimento do Brokert MQTT foi utilizado um Raspberry Pi, nele é possível criar um broker na rede interna com apenas alguns simples comandos. O *broker* fica disponível para uso interno pelo endereço IPv4 de rede que o Rasperry Pi está utilizando, e também estará disponível para rede externa pelo endereço IPv6 que o Raspberry Pi está utilizando. Também foi configurado um endereço de DNS dinâmico para o Raspberry Pi. A configuração do DNS foi feita utilizando o site FreeDNS, que é uma forma gratuita de configuração de DNS.

O *broker* vai ser utilizado para gerenciar as mensagens nos tópicos que vem do dispositivo. No caso do Sistema desenvolvido nesse trabalho, cada dispositivo vai utilizar um tópico, definido por um id, para se comunicar com o broker. Ele então vai ser responsável para enviar as mensagens de “START” e “STOP” vindas dos aplicativo de celular para os dispositivos para iniciar e para o Contador e vai ser responsável para enviar as mensagens do contador vindas dos dispositivos para o aplicativo de celular.

3.3.1.2 Clients

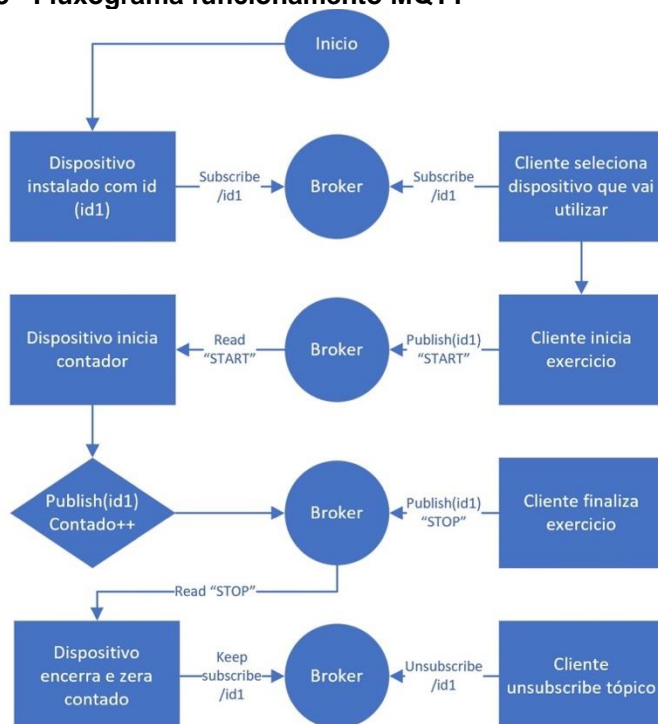
São dois tipos de *Clients* para esse sistema, um dos tipos de *Client* serão os dispositivos da aplicação e outro os de interface com o usuário.

Os dispositivos quando ligados vão estar sobrescrevendo o tópico com seus respectivos ids, assim qualquer mensagem que chegar no tópico, os dispositivos estarão escutando. Quando chegar no tópico mensagens com conteúdo “START”, o dispositivo vai iniciar o Contador, e quando chegar ao tópico mensagem com conteúdo de “STOP”, o dispositivo vai para o Contador. Depois de “lida” a mensagem “START”, o *Client* envia para o tópico cada iteração do Contador, para

que o Contador possa ser observado para todos que estão sobrescrevendo o tópic. Esse Contador é parado e zerado quando a mensagem “STOP” é escrita no tópic.

Os *Clients* de interface com o usuário funcionarão da seguinte forma, quando for iniciar um exercício novo, o *Client* vai sobrescrever no id do dispositivo que quer utilizar, assim o dispositivo e o usuário estarão escutando tudo que se envia para o mesmo tópic. Quando for dar início no exercício o usuário envia a mensagem “START” para o tópic, fazendo com que inicie o Contador. Como ele está sobrescrevendo o tópic, quando o Contador iniciar ele consegue obter o Contador em tempo real que esta sendo enviado pelo dispositivo. Assim que desejar parar o Contador, o usuário envia uma mensagem de “STOP”, fazendo com o Contador pare e zere para o próximo que utilizar aquele dispositivo. A Figura 13 demonstra o funcionamento em um fluxograma.

Figura 13 - Fluxograma funcionamento MQTT



Fonte: Autoria própria.

3.3.2 Dispositivo

Para a confecção do dispositivo foi utilizado um NodeMCU como microcontrolador e um sensor de distância para realizar o contador. O NodeMCU foi escolhido pelo motivo de ser de baixo custo, já ter bibliotecas de implementação de

MQTT *Client* pronta para se trabalhar e pela fácil linguagem de programação. O sensor ultrassônico de distância foi escolhido por ser a melhor forma de medir distâncias entre objetos, ser de baixo custo e também possuir biblioteca pronta para programação do sistema.

3.3.3 Software

Para a programação do dispositivo foi utilizada a IDE do arduino, nele o programa é dividido em duas partes, uma para Setup que realiza a função somente uma vez, e o *loop* onde o programa fica rodando interruptamente após o Setup. Na parte de Setup estão todas as configurações de conexão com a internet via *wi-fi* e conexão com o MQTT *Broker*. Na função de *loop* estão funcionalidades como de sobrescrever o tópico na qual o dispositivo está atuando, função de verificar a distância do sensor ultrassônico e de publicar no tópico quando o contador é incrementado.

O programa funciona da seguinte forma, quando iniciado ele conecta com a rede *wi-fi* que for selecionada no programa. Depois que conectado o *Client* vai se conectar com o broker a partir do endereço fornecido pelo DNS do Broker e vai sobrescrever o cliente no tópico com o id do dispositivo. Feito todas essas configurações o dispositivo aguarda uma mensagem no tópico com o conteúdo “START”, assim que recebido o programa inicializa o contador e começa a verificar a distância medida pelo sensor, caso a distância ultrapasse os 20cm o contador é incrementado 1 posição e é publicado no tópico. Esse processo vai acontecer até que uma mensagem de conteúdo “STOP” seja enviada para o tópico, fazendo com que o contador pare e zere e o programa volte a aguardar a mensagem de “START”. Um fluxograma abaixo demonstra o funcionamento do dispositivo.

3.3.4 Hardware

Para o hardware foi utilizado o NodeMCU com o sensor de distância ultrassônico. O sensor ultrassônico possui 4 pinos, VCC, GND, RX e TX onde são ligados no NodeMCU, respectivamente seguindo a relação apresentada na Tabela 3.

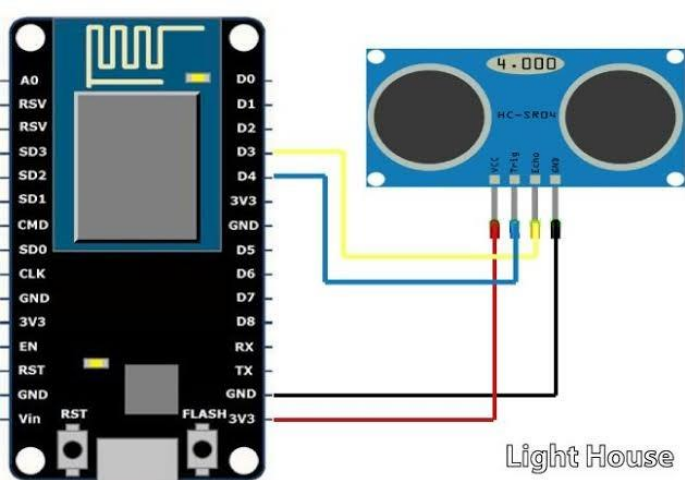
Tabela 3 - Interligação NodeMCU e Sensor Ultrassônico

NodeMCU	Sensor Ultrassônico
3.3V	VCC
TX	RX
RX	TX
GND	GND

Fonte: Autoria própria.

Também é possível observar na Figura 14 o diagrama esquemático do circuito na imagem abaixo.

Figura 14 - Esquema circuito eletrônico

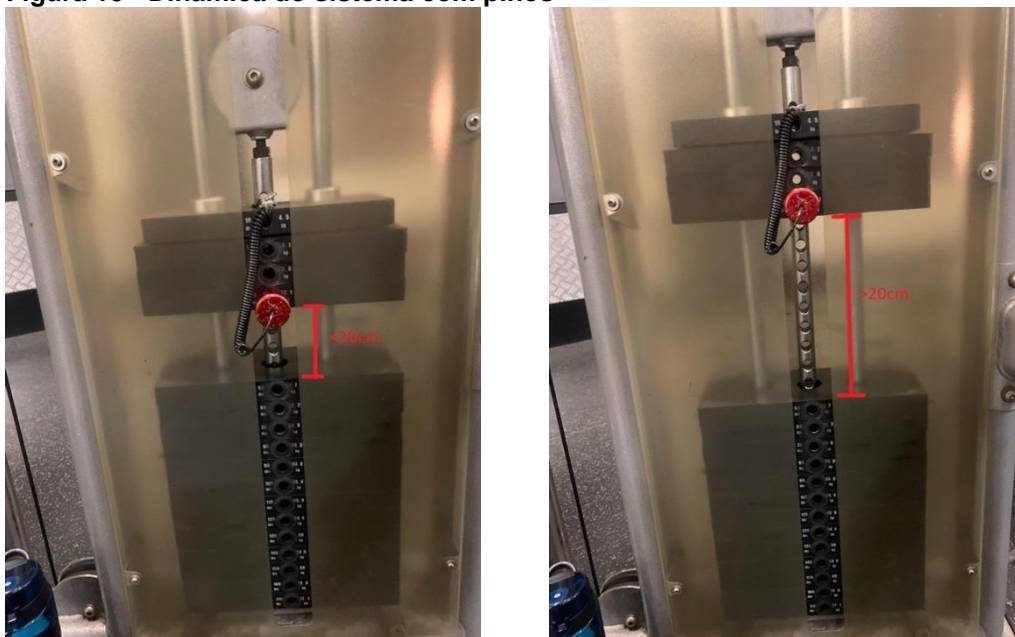


Fonte: Autoria própria.

3.3.5 Dinâmica

Os dispositivos serão acoplados em pinos travas de academia, serão necessários dois pinos para o dispositivo funcionar, o pino que leva o dispositivo e um pino auxiliar para fazer a medida da distância de um para o outro. Assim que iniciado o trabalho do contador, o dispositivo vai verificar a distância de um pino a outro, que no caso seria, se o atleta que está realizando o exercício exerceu o movimento de puxar ou empurrar os pesos, o sensor de distância vai ter um comprimento maior e nesse caso o atleta realizou uma repetição no aparelho, incrementando assim uma unidade ao contador. A Figura 15 abaixo demonstra como funciona a dinâmica.

Figura 15 - Dinâmica do sistema com pinos



Fonte: Autoria própria.

3.3.6 Interface com o Usuário

Para o desenvolvimento da interface com o usuário foi utilizado a linguagem Flutter da Google, essa linguagem foi escolhida pois permite criar aplicativos de celular tanto para Android quanto para iOS com apenas uma escrita de código, é uma linguagem fácil e já possui biblioteca pronta para integração com o banco de dados Firebase e MQTT *Client*.

O aplicativo é dividido em 4 partes, sendo autenticação e criação de usuário, treinos do perfil, exercícios do treino e Contador de repetições. Cada parte do aplicativo possui uma tela para a integração com o usuário.

A autenticação e criação do usuário é feita através do Firebase *Authentication*, onde o Sistema do Google já possui uma biblioteca que permite criar um usuário com senha e autenticar o mesmo. Quando o usuário é criado, seu perfil já é gravado diretamente no banco de dados. Para criar o perfil o usuário deve fornecer nome, email e senha, depois para fazer a autenticação só é necessário utilizar o email com a senha escolhida.

Assim que autenticado o aplicativo abre a página inicial, a página padrão inicial do aplicativo é onde se encontra os treinos já realizados pelo usuário, contendo a data de cada treino. A página também possui um botão flutuante para a criação de novos treinos. Outro modo de criar um novo treino é utilizando o menu

lateral e abrindo a aba de “current”, que seria a página onde apareceria o treino caso ele já tivesse sido iniciado, caso o treino ainda não tenha sido iniciado a página mostra um botão para iniciar o treino. Assim que algumas das opções de iniciar treino forem acionadas, dentro do perfil do usuário é gravado no banco de dados o novo treino que se inicia.

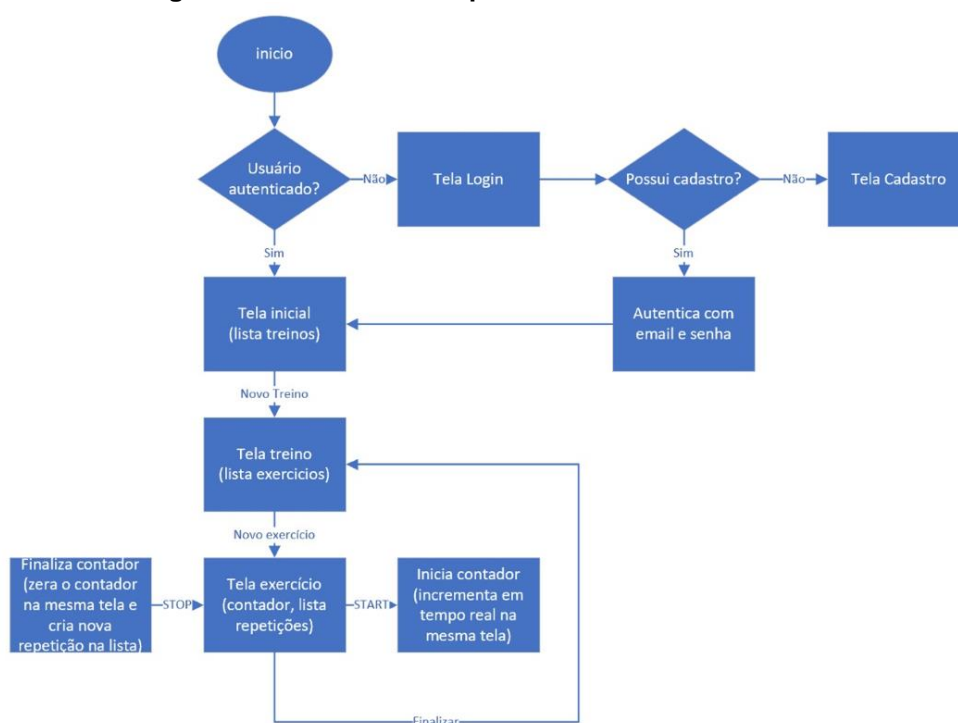
Depois de iniciado o treino, o aplicativo abre a tela do treino atual, nela irá aparecer os exercícios já realizados dentro daquele treino, um botão para começar um novo exercício e um botão para finalizar o treino. Os exercícios já realizados aparecem mostrando informações de quantas repetições foram realizadas e o peso que foi realizado o exercício. Se o usuário optar por finalizar o treino, ele volta para a tela inicial e se optar pela opção de realizar um novo exercício a tela de contador de repetições vai ser aberta e um novo exercício é criado no banco de dados.

Antes de entrar na página de exercício, o aplicativo precisa saber qual dispositivo ele está se conectando para mostrar o contador de repetições, então a câmera do aparelho é aberta para fazer o escaneamento de um QR Code. Cada máquina que vai utilizar o dispositivo, irá possuir um QR Code, esse QR Code vai passar as informações para o aplicativo de qual tópico MQTT que o dispositivo instalado na máquina está aguardando e publicando informações, para que o aplicativo possa sobrescrever o mesmo tópico e conseguir se comunicar com o dispositivo. Tendo o nome do tópico através do QR Code o aplicativo se conecta ao MQTT Broker e já sobrescreve o tópico do dispositivo. Para a geração do QR Code foi utilizado o site [the-qrcode-generator](http://the-qrcode-generator.com), as informações passadas para o aplicativo são enviadas em forma de *String*.

Quando aberta a tela de repetições é possível ver um campo de texto com informações do contador em tempo real, que no caso de não ter iniciado o exercício o valor é zero e quando iniciado ele é incrementado de acordo com a realização do exercício, dois botões de iniciar e finalizar cada série de repetições, uma tabela em que mostra cada série de repetições que já foram realizadas no exercício e um botão para salvar o exercício. Quando o botão iniciar é pressionado o aplicativo vai publicar a mensagem de “START” para o broker para iniciar o contador e vai criar a primeira repetição no banco de dados, a cada repetição é possível observar o contador em tempo real e assim que optar por finalizar o exercício é necessário pressionar o botão de encerrar, para que seja publicado a mensagem de “STOP” para o dispositivo e o contador ser finalizado e zerado, e para salvar as informações

da série no banco de dados, mostrando na tabela a primeira série de repetição. Quando o exercício for finalizado basta o usuário pressionar o botão de salvar o exercício que ele vai ser direcionado para a tela do treinamento que esta realizando. A Figura 16 mostra o fluxograma do funcionamento do aplicativo.

Figura 16 - Fluxograma funcionamento aplicativo



Fonte: Autoria própria.

3.3.7 Banco de Dados

Para o banco de dados foi escolhido utilizar um banco não relacional, essa escolha foi feita pela facilidade de manipular os dados. O banco escolhido para se trabalhar foi o Firebase da Google, que tem fácil integração com o Flutter, pois também é da Google.

Os dados serão gravados no banco de dados em formato JSON com a seguinte estrutura demonstrada na Figura 17.

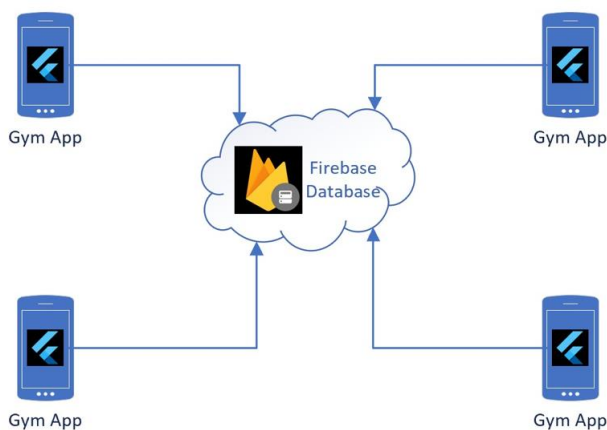
Figura 17 - Estrutura JSON

```
{
  "userID": "123",
  "name": "nameExample",
  "email": "emailExample",
  "admin": false,
  "trainings": {
    "trainingID": "123",
    "start": "startTime",
    "end": "endTime",
    "exercises": {
      "exerciseID": "123",
      "name": "exerciseNameExample",
      "rep": {
        "repID": "123",
        "times": "times",
        "weight": "weight"
      }
    }
  }
}
```

Fonte: Autoria própria.

Quando iniciado o aplicativo o cliente cria uma conta, e então é feita a primeira gravação dos dados, estando dentro de sua conta o usuário pode optar por criar um treino, feita essa escolha os dados de treino são gravados no banco. Iniciado o treino o usuário começa a fazer os exercícios, quando iniciado um exercício esse também é gravado no banco. E por fim cada repetição que é realizada no exercício é gravada o número de vezes realizado e o peso. A Figura 18 mostra o diagrama da conexão do aplicativo com o banco de dados.

Figura 18 - Diagrama conexão banco de dados



Fonte: Autoria própria.

4 RESULTADOS OBTIDOS

Esse capítulo tem por finalidade apresentar os resultados obtidos com a realização do trabalho.

Como resultado obteve-se um Sistema que auxilia nos treinos de atletas dentro das academias. Esse Sistema é composto por um aplicativo celular e um dispositivo que se comunicam através de um MQTT Broker por mensagens em tópicos. O Sistema também permite gravar um histórico de treinamentos para consultas, além de toda segurança oferecida no aplicativo pela autenticidade do usuário. O dispositivo é de baixo custo e pode ser aplicado em qualquer máquina da academia que possua travas de pesos. O dispositivo utiliza um sensor de distância para fazer a contagem de repetições do exercício que esta sendo realizado. Completando assim todos os objetivos citados.

4.1 BROKER

O broker foi instalado na rede local com o Raspberry PI conectado por um cabo *Ethernet* ao modem. Como demonstrado na Figura 19.

Figura 19 - Broker conectado



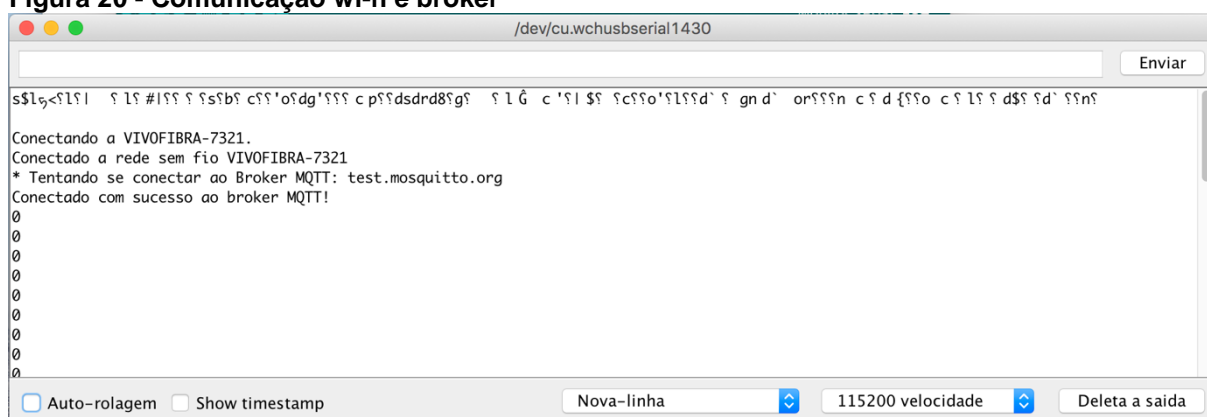
Foto: Autoria própria.

4.2 CLIENTS

Os clientes foram construídos em um conjunto de pinos travas de academias com o NodeMCU e o sensor de distância. O número de dispositivos é ilimitado, mas um protótipo foi construído para demonstração de como seriam cada um deles.

Quando o dispositivo é ligado ele conecta a rede *wi-fi* e ao broker MQTT, também é sobrescrito o tópico em que o dispositivo vai trabalhar, isso pode ser observado na serial do dispositivo como mostrado na Figura 20.

Figura 20 - Comunicação wi-fi e broker



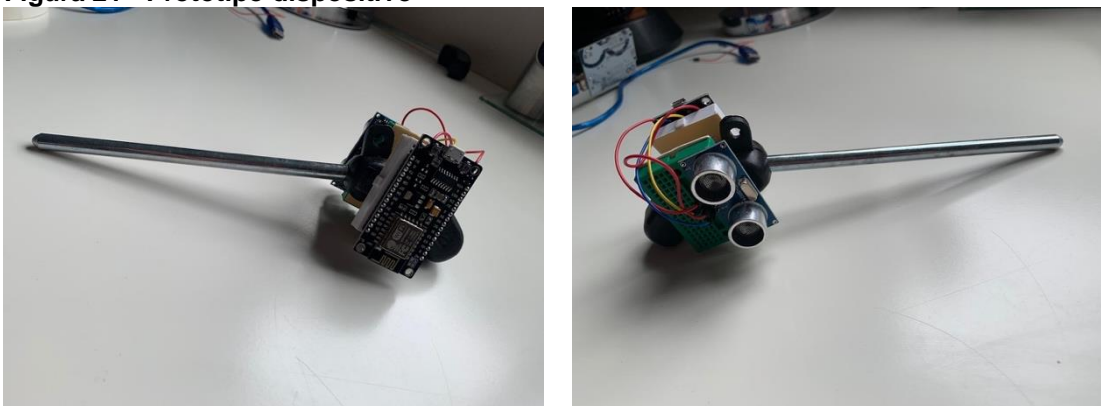
The image shows a serial terminal window titled "/dev/cu.wchusbserial1430". The terminal output displays the following text: "Conectando a VIVOFIBRA-7321.", "Conectado a rede sem fio VIVOFIBRA-7321", "* Tentando se conectar ao Broker MQTT: test.mosquitto.org", and "Conectado com sucesso ao broker MQTT!". Below this, there are several lines of the character "0". At the bottom of the window, there are controls for "Auto-rolagem", "Show timestamp", "Nova-linha", "115200 velocidade", and "Deleta a saida".

Fonte: Autoria própria.

O protótipo foi criado em um *protoboard* para interligar o controlador com o sensor, nele os pinos são conectados com a alimentação do sensor no pino de 3,3V do NodeMCU, o terra do sensor no terra do NodeMCU, o pino de RX na entrada D7 do controlador e o pino TX na entrada D8 do controlador. Como demonstrado na figura abaixo.

O conjunto montado no *protoboard* foi adicionado ao pino trava para a finalização do dispositivo, para isso em um pino ficou acoplado o *protoboard* enquanto o outro recebeu um papel para aumentar a área de alcance do sensor de distância, para que quando for feita a medida o sensor não tenha erros de medição. O protótipo é ilustrado na Figura 21.

Figura 21 - Protótipo dispositivo

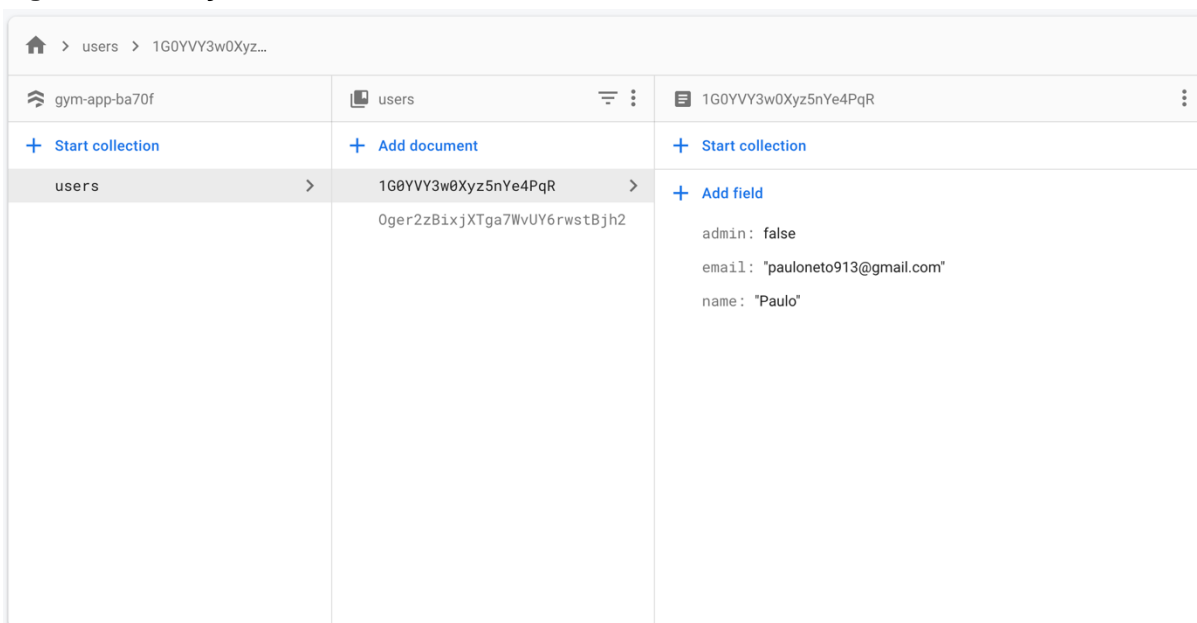


Fonte: Autoria própria.

4.3 BANCO DE DADOS

O banco de dados foi criado utilizando o Firebase. Para isso foi utilizada uma conta do Google. Dentro do *console* foi criado um projeto com o nome de *gym-app* e adicionado um *database*, onde foram adicionados os dados. A primeira coleção sempre é criada quando feita a autenticação de um novo usuário, e a partir dela é criado os documentos para aquele usuário. Cada usuário recebe um id único que cria um novo documento com seu nome. Nesse documento são gravados os campos de email, nome e se o usuário é administrador ou não. Como mostra a Figura 22.

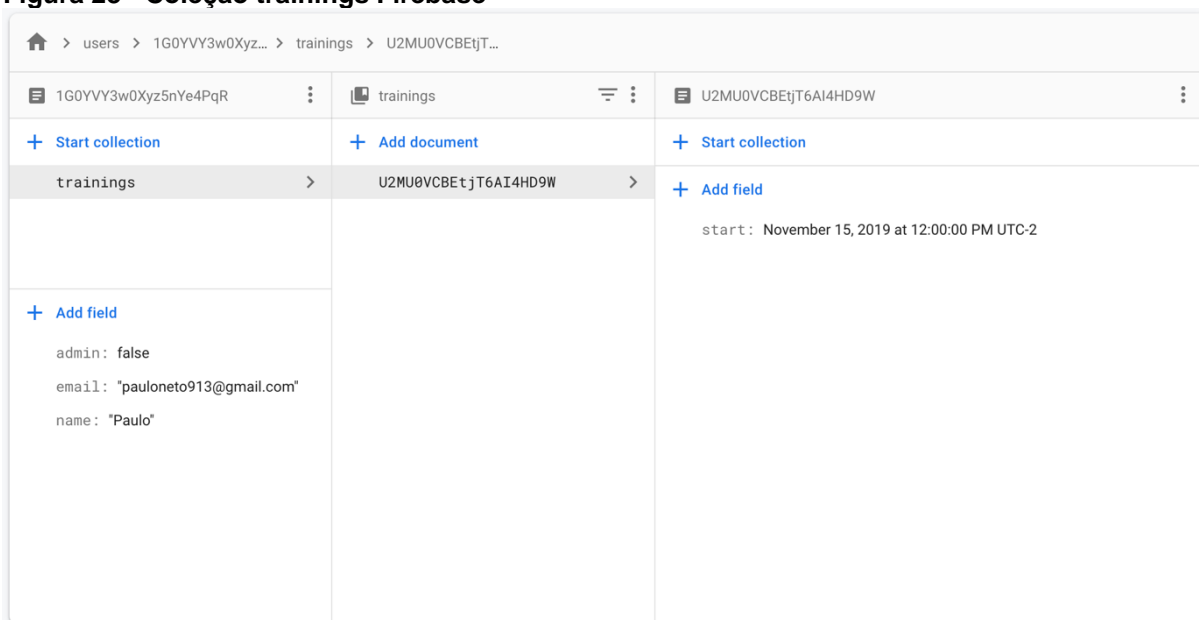
Figura 22 - Coleção usuários Firebase



Fonte: Autoria própria.

Quando criado o primeiro treino é criada uma coleção “trainings”, e nele o primeiro treino leva um id único. Neste documento é gravado o campo de quando o treino foi iniciado, como mostra a Figura 23.

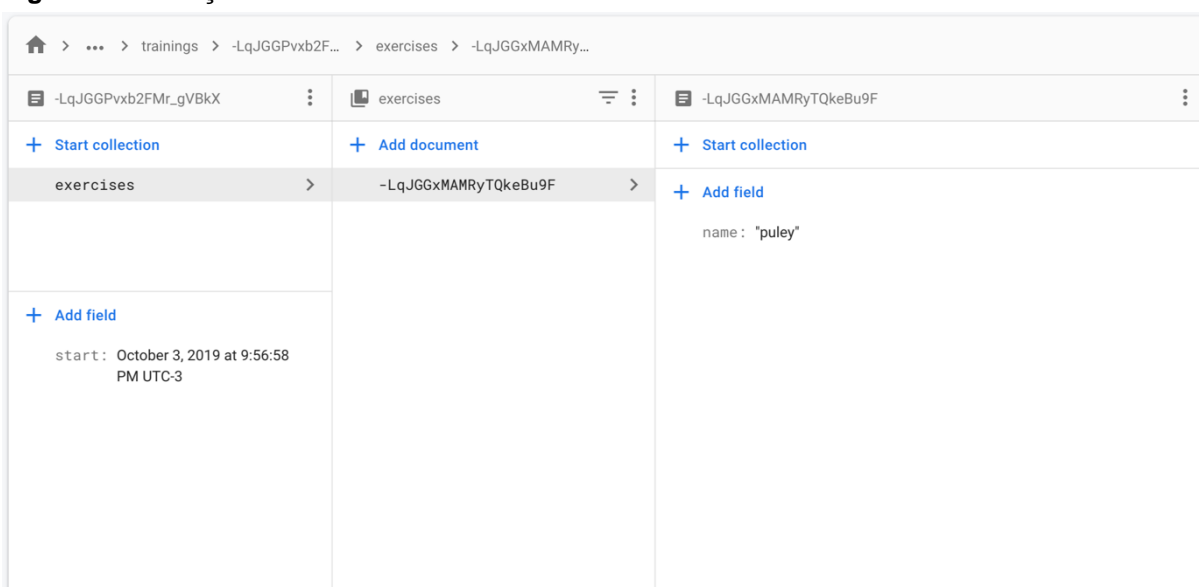
Figura 23 - Coleção trainings Firebase



Fonte: Autoria própria.

Quando iniciado um exercício, é criado uma coleção “exercise”, com um documento com id único. E no documento é gravado o campo de nome do exercício. Como mostra a Figura 24.

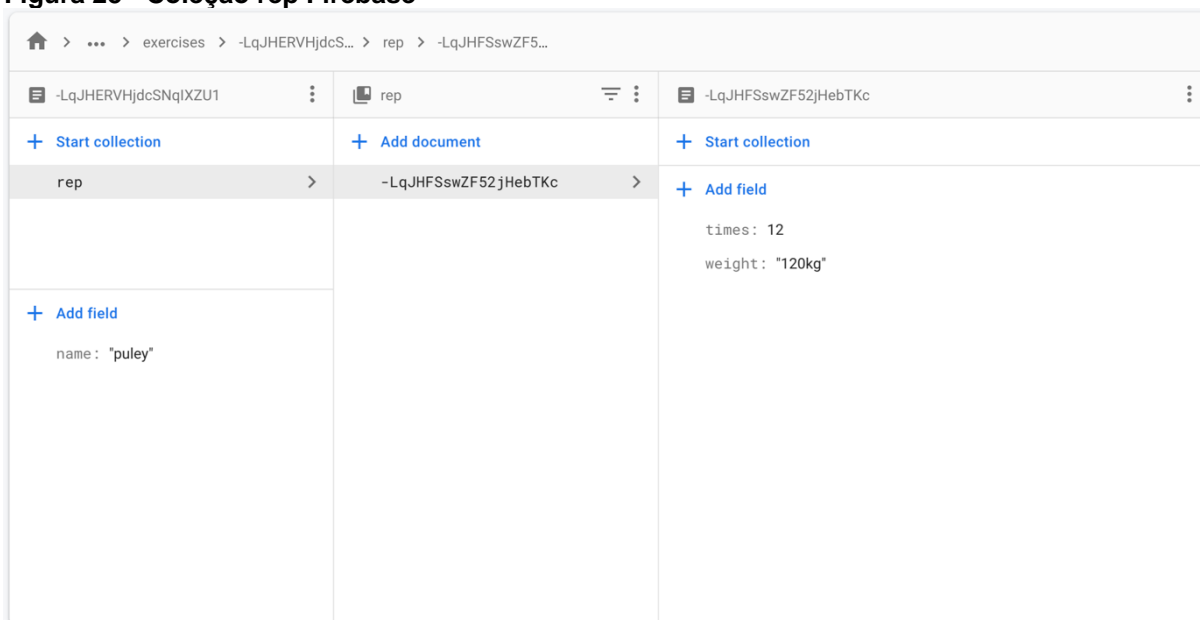
Figura 24 - Coleção exercises Firebase



Fonte: Autoria própria.

Quando iniciado uma nova repetição é criado uma coleção “rep”, com um documento de id único. Nele são gravados campos de peso e vezes que a repetição foi executada. Como mostrado na Figura 25.

Figura 25 - Coleção rep Firebase



Fonte: Autoria própria.

Quando um novo documento é criado ele é gravado em sua respectiva coleção, de acordo com cada usuário.

4.4 INTERFACE COM O USUÁRIO

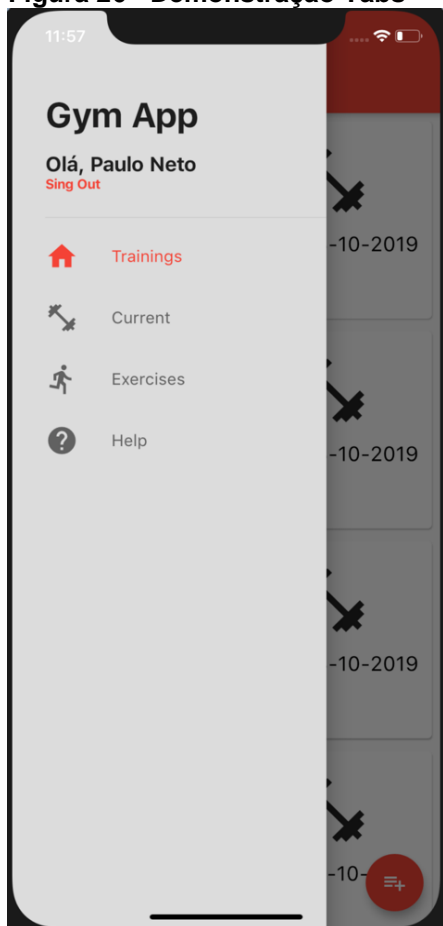
O aplicativo desenvolvido com a ferramenta Flutter tem a possibilidade de ser instalado tanto em iOS quanto em Android, porem para esse trabalho ele foi instalado somente em um iPhone Xr.

Com uma linguagem muito simples de se trabalhar, o aplicativo tem uma interface muito amigável, sendo de muito limpa e de simples compreensão. Para a construção, o projeto foi dividido em pacotes, onde cada classe corresponde a uma página, *Tab* ou dado.

Uma página do aplicativo utiliza vários métodos para passar informações para o usuário, utilizando tabelas, botões, caixas de textos. Para o aplicativo desenvolvido, foi criada a página inicial com as *tabs*, página de treino, de exercícios, página do contador de repetições, página de login e página de novo usuário.

Uma *Tab* é utilizada para trazer informações em cima de uma página, onde o usuário consegue facilmente através de um menu trocar de uma *tab* para outra. No caso do aplicativo aqui desenvolvido, uma página de início é criado com quatro *tabs*, sendo elas de início, *current*, suporte. Como mostrado na Figura 26.

Figura 26 - Demonstração Tabs



Fonte: Autoria própria.

Os dados são convertidos do formato *JSON* (Tabela 4) que vem dos documentos do banco de dados para um formato definido de acordo com a necessidade. Por exemplo, quando coletadas as informações do banco de dados de um treino, os dados vindos do banco com o documento de id único são transformados para um tipo "Training", esse novo tipo de objeto contém as informações do treino como os exercícios, horários de início e fim. E esse objeto é utilizado para mostrar as informações na tela.

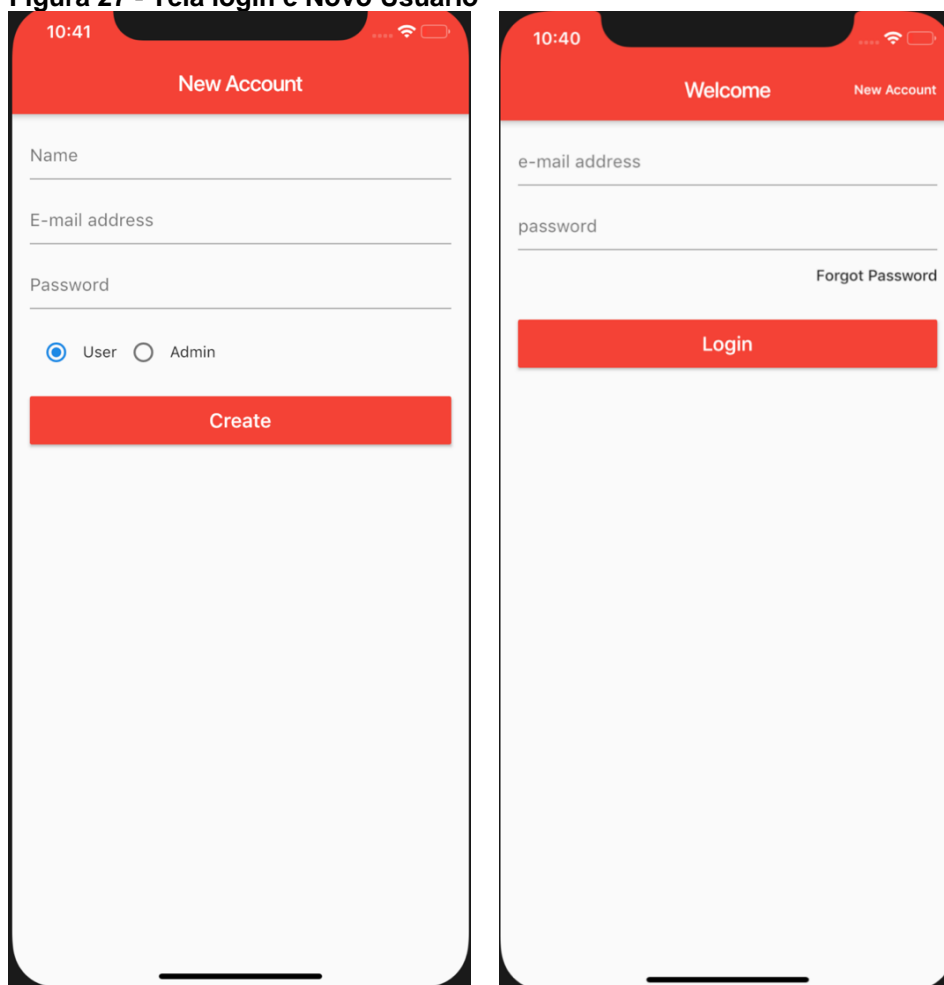
Tabela 4 - Conversão formato JSON para objeto

Formato JSON	Formato Objeto
<pre> "rep":{ "repID": "123", "times": "times", "weight": "weight" } </pre>	<pre> rep.id rep.times rep.weight </pre>

Fonte: Autoria própria.

A autenticação é feita através das páginas de login e nova conta. A princípio se o usuário já estiver logado essas páginas nem são utilizadas no aplicativo, somente se o usuário sair de sua conta ou se for a primeira vez em que esta usando o aplicativo, a única maneira de acesa a página de criar novo usuário é através da página de *login*. As páginas são mostradas na Figura 27.

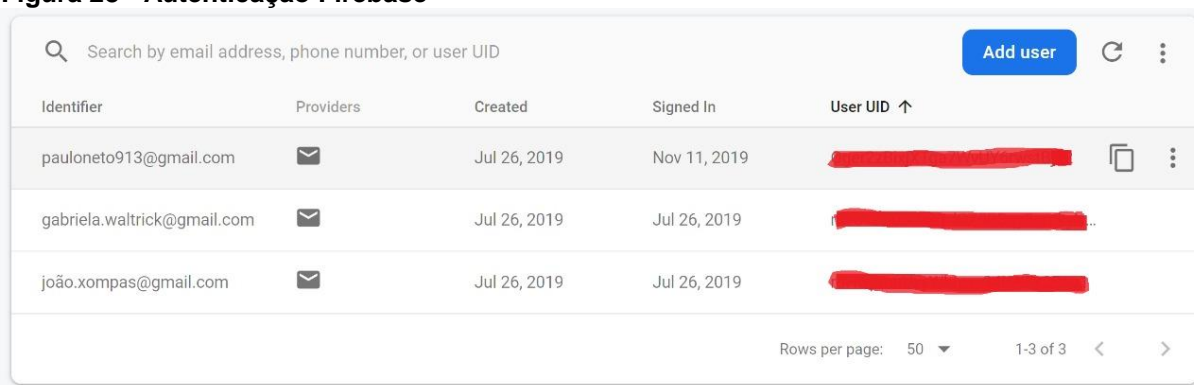
Figura 27 - Tela login e Novo Usuário



Fonte: Autoria própria.

No Firebase a autenticação é feita automaticamente através do UID único para cada usuário, e quando criada uma nova conta um novo id para o email criado é relacionado. Como visto na Figura 28 um exemplo de emails autenticados para teste do aplicativo.

Figura 28 - Autenticação Firebase

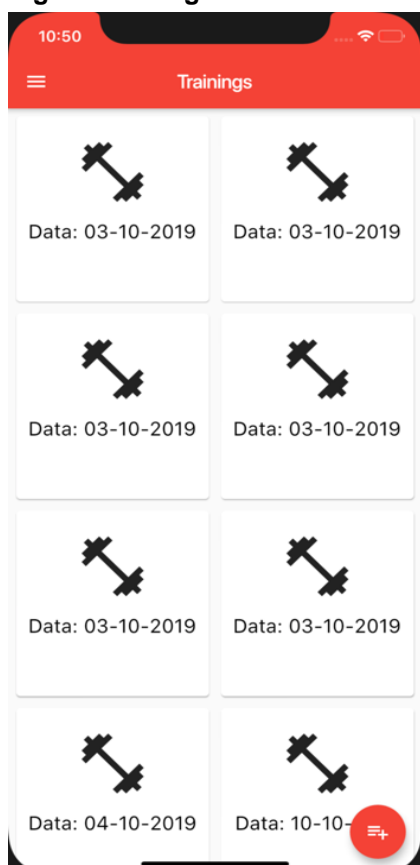


Identifier	Providers	Created	Signed In	User UID ↑
pauloneto913@gmail.com	✉	Jul 26, 2019	Nov 11, 2019	[Redacted]
gabriela.waltrick@gmail.com	✉	Jul 26, 2019	Jul 26, 2019	[Redacted]
joão.xompas@gmail.com	✉	Jul 26, 2019	Jul 26, 2019	[Redacted]

Fonte: Autoria própria.

Depois de feita a autenticação o usuário vai para a página inicial, que seria a página de treinos, como demonstra a Figura 29.

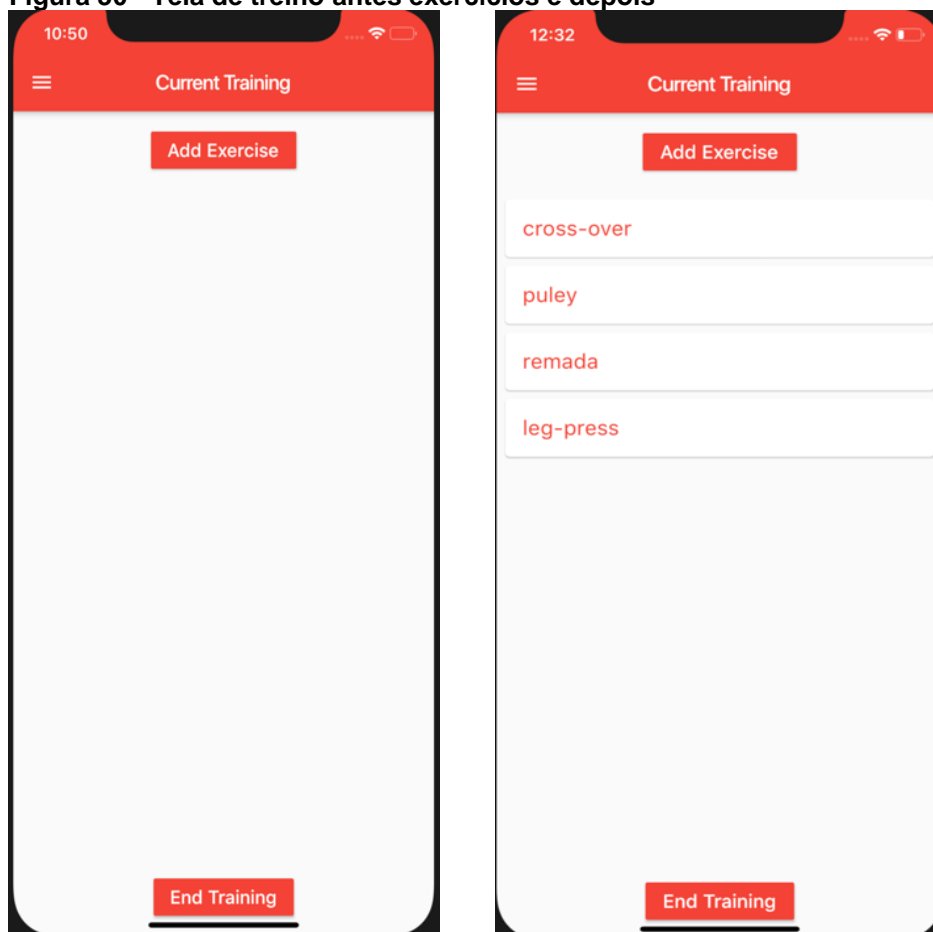
Figura 29 - Página inicial de treinos



Fonte: Autoria própria.

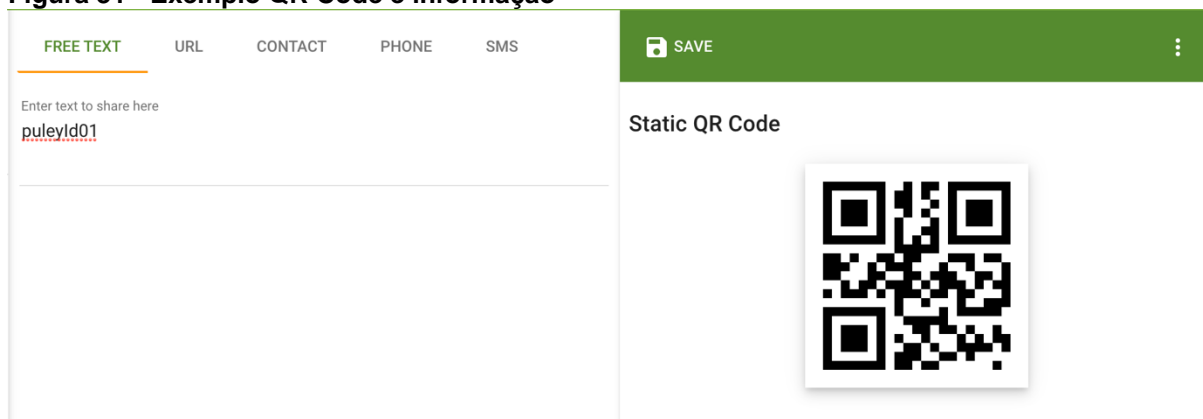
Quando iniciado um novo treino o aplicativo vai para a página deste treino que esta sendo realizado, onde são demonstrados os exercícios que já foram realizados. Como demonstra a Figura 30.

Figura 30 - Tela de treino antes exercícios e depois



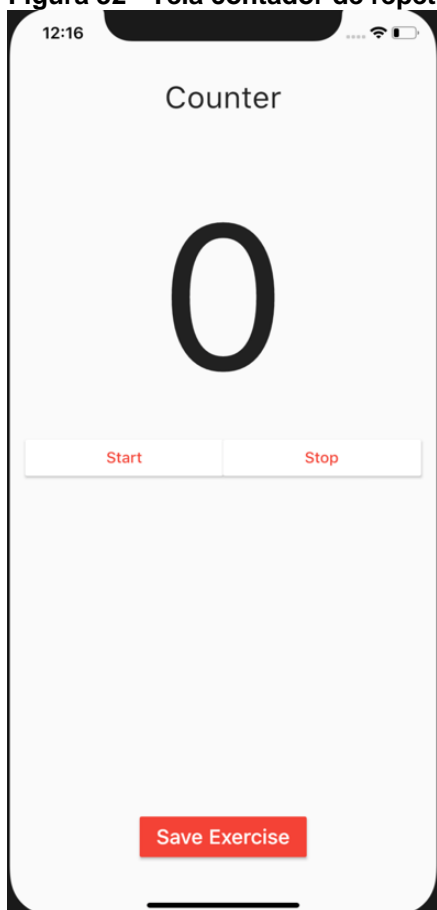
Fonte: Autoria própria.

Quando iniciado um novo exercício, é aberta a camera do celular para escanear o QR Code da máquina em que vai ser executado o exercício, que traz a informação de qual tópico o *Client* vai se inscrever. A Figura 31 mostra um QR Code com suas informações.

Figura 31 - Exemplo QR Code e informação

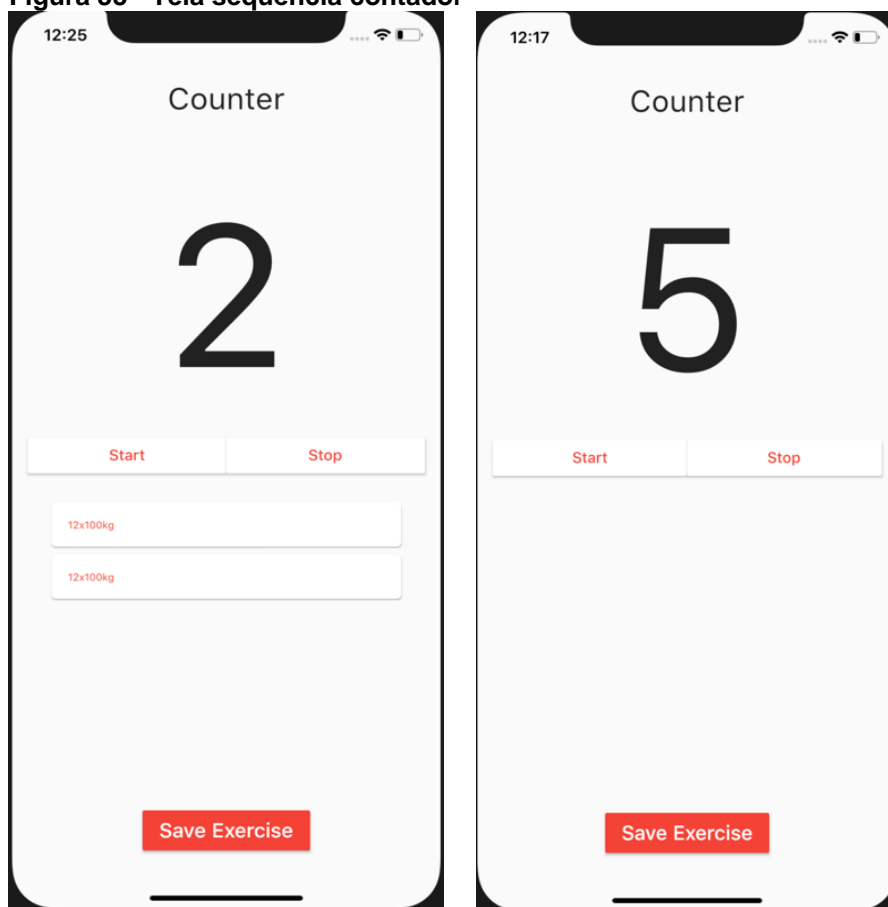
Fonte: Autoria própria.

Assim que escaneado o QR Code o aplicativo abre a página do contador de repetições, da forma que está demonstrada na Figura 32.

Figura 32 - Tela contador de repetições

Fonte: Autoria própria.

De acordo com cada repetição que é realizada o contador é incrementado em tempo real e assim que cada repetição for finalizada é salvo na tabela abaixo do contador. A Figura 33 demonstra essa sequência.

Figura 33 - Tela sequência contador

Fonte: Autoria própria.

A Figura 34 mostra um exemplo do movimento em que faz gerar a incrementação do contador.

Figura 34 - Sequência do movimento

Fonte: Autoria própria.

5 CONCLUSÃO

Com o desenvolvimento desse trabalho foi possível criar um sistema que seja capaz de auxiliar atletas em seus treinos dentro de academias. Possibilitando-os a acompanhar em tempo real a quantidade de repetições em que esta desenvolvendo um exercício que utiliza o dispositivo proposto nesse projeto. Tendo o acompanhamento de seus treinos, o atleta pode ter uma melhora na saúde e aumentar o seu desempenho na prática de exercícios. E tendo esses dados gravados, também é possível mostrar para um profissional mais qualificado para ter diferentes opiniões, um acompanhamento e auxílio em seus treinos. O sistema proposto como citado em seus respectivos objetos foi criado com uma interface para o usuário em um aplicativo móvel para acompanhamento e tempo real, uma comunicação dos dispositivos utilizando um protocolo de comunicação da Internet das Coisas e um banco de dados para gravar informações dos treinos já realizados. O dispositivo é capaz de contar as repetições dos exercícios em tempo real através de sensores acoplados nos pinos travas das máquinas das academias. Para melhor acompanhamento é pensado também para trabalhos futuros o desenvolvimento maior do sistema de acompanhamento criando ferramentas para ajudar na postura do desenvolvimento de cada exercício e ter a possibilidade de um profissional qualificado acompanhar o treino de seus atletas dentro do próprio aplicativo.

REFERÊNCIAS

BRADEN, R. **Requirements for internet hosts: Communication layers**. Network Working Group. Internet Engineering Task Force. Publicado em: out. 1989. Disponível em: <<https://tools.ietf.org/html/rfc1122#page-77>>. Acesso em: 15 nov. 2019.

BUSTAMANTE, J. **IoT statistics**. Copyright© 2019, iPropertyManagement.com. Disponível em: <<https://ipropertymanagement.com/iot-statistics>>. Acesso em: 17 out. 2019.

COAP. **CoAP: RFC 7252 constrained application protocol**. Disponível em: <<https://coap.technology/>>. Acesso em: 01 nov. 2019.

COLUMBUS, L. **2018 roundup of Internet Of Things forecasts and market estimates**. Copyright©2019 Forbes Media LLC. Publicado em: 13 dez. 2018. Disponível em: <<https://www.forbes.com/sites/louiscolumbus/2018/12/13/2018-roundup-of-internet-of-things-forecasts-and-market-estimates/#749c255d7d83>>. Acesso em: 24 out. 2019.

ESPRESSIF. **ESP6266EX**. Copyright@ 2019 ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD. Disponível em: <<https://www.espressif.com/en/products/hardware/esp8266ex/overview>>. Acesso em: 29 out. 2019.

EVERYINTERACTION. **User interface**. Copyright© 2019 Every Interaction. Disponível em: <<https://www.everyinteraction.com/definition/user-interface/>>. Acesso em: 30 out. 2019.

ITU. **Recommendation ITU-T Y.2060: Overview of the Internet of Things**. International Telecommunication Union (ITU). Copyright© ITU 2013. Publicado em: jun. 2012. Disponível em: <>. Acesso em: 20 out. 2019.

LEVEREGE. **Introduction to UIs & UX for IoT**. Copyright© Leverage LLC 2019. Disponível em: <<https://www.leverage.com/iot-ebook/ui-and-ux-design-iot>>. Acesso em: 30 out. 2019.

LOPEZ, M. **Uma introdução à Internet da Coisas (IoT)**. Copyright© 2018 Lopez Research LLC. Publicado em: nov. 2013. Disponível em: <https://www.cisco.com/c/dam/global/pt_br/assets/brand/iot/iot/pdfs/lopez_research_an_introduction_to_iiot_102413_final_portuguese.pdf>. Acesso em: 16 out. 2019.

MICROSOFT AZURE. **Dados não relacionais e NoSQL**. Copyright© 2019 Microsoft. Publicado em: 11 fev. 2018. Disponível em: <<https://docs.microsoft.com/pt-br/azure/architecture/data-guide/big-data/non-relational-data>>. Acesso em: 05 nov. 2019.

MIRAZ, M. H. *et al.* **A review of Internet of Things (IoT), Internet of Everything (IoE) and Internet of Nano Things (IoNT)**. 2015 Intert Technologies and Applications, ITA 2015 ITU. Overview of the internet of things. ITU-T Std. Y.2060, n. Y.2060, jun. 2012. Disponível em: <<https://ieeexplore.ieee.org/document/7317398/>>. Acesso em: 16 out. 2019.

MQTT. **FAQs**. MQTT.org, 2019. Disponível em: <<http://mqtt.org/faq>>. Acesso em: 23 out. 2019.

OPENSOURCE.COM. **What is a Raspberry Pi?** Copyright ©2019 Red Hat, Inc. Disponível em: <<https://opensource.com/resources/raspberry-pi>>. Acesso em: 29 out. 2019.

PENIDO, É. de C. C. **Microcontroladores**. Édilus de Carvalho Castro Penido. Ronaldo Silva Trindade. Ouro Preto: Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais; Santa Maria: Universidade Federal de Santa Maria, Colégio Técnico Industrial de Santa Maria. Rede e-Tec Brasil, 2013. Disponível em: <http://estudio01.proj.ufsm.br/cadernos/ifmg/tecnico_automacao_industrial/microcontroladores.pdf>. Acesso em: 20 out. 2019.

REZENDE, R. **Conceitos fundamentais de banco de dados**. DEVMEDIA: Rio de Janeiro. 2006. Disponível em: <<https://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649>>. Acesso em: 25 out. 2019.

SALMAN, T. **Internet of Things protocols and standards: 5.4 CoAP**. Orientação Prof. Raj Jain. Washington University in St. Louis. Última modificação em: 30 nov. 2015. Disponível em: <https://www.cse.wustl.edu/~jain/cse570-15/ftp/iiot_prot/#CoAP>. Acesso em: 28 nov. 2019.

SENSOR. **Dicionário online do Merriam-Webster**. Disponível em: <<https://www.merriam-webster.com/dictionary/sensor>>. Acesso em: 30 out. 2019.

TEXASINSTRUMENTS. **CC2650 SimpleLink Multistandard Wireless MCU**. Copyright© 1995-2019 Texas Instruments Incorporated. Publicado em: fev. 2015. Disponível em: <<http://www.ti.com/lit/ds/symlink/cc2650.pdf>>. Acesso em: 30 out. 2019.

THEO, S. **Importance of sensors In the Internet of Things**. **Eletronicsforu**, Copyright© 2019 - EFY Group. Publicado em: 28 mai. 2018. Disponível em: <<https://electronicsforu.com/technology-trends/tech-focus/iot-sensors>>. Acesso em: 30 out. 2019.

YUAN, M. **Getting to know MQTT**: Why MQTT is one of the best network protocols for the Internet of Things. IBM Developer. Publicado em: 12 mai. 2017. Disponível em: <<https://developer.ibm.com/articles/iot-mqtt-why-good-for-iot/#>>. Acesso em: 23 out. 2019.