UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

Vitor Hideoshi Inazaki de Assis

Desenvolvimento de aplicativo web para dinamômetro de preensão palmar

São Carlos
2017

Vitor Hideoshi Inazaki de Assis

Desenvolvimento de aplicativo web para dinamômetro de preensão palmar

Monografia apresentada ao Curso de Engenharia Elétrica com ênfase em Eletrônica, da Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Alberto Cliquet Junior

São Carlos

2017

Assis, Vitor Hideoshi Inazaki de
A848d Desenvolvimento de aplicativo web para dinamômetro de preensão palmar / Vitor Hideoshi Inazaki de Assis; orientador Alberto Cliquet Junior. São Carlos, 2017.

Monografia (Graduação em Engenharia Elétrica com ênfase em Eletrônica) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2017.

1. Aplicativo web. 2. Força de preensão palmar. 3. Célula de carga. 4. Protocolo MQTT. 5. Esp8266. 6. Micropython. 7. Flask. 8. Bokeh. I. Título.

FOLHA DE APROVAÇÃO

Nome: Vitor Hideoshi Inazaki de Assis

Título: "Dinamômetro de preensão palmar com aplicativo web"

Trabalho de Conclusão de Curso defendido e aprovado em 22 / 11 / 2017,

com NOTA 7,5 (sete , cinco), **pela Comissão Julgadora:**

Prof. Titular Alberto Cliquet Júnior - Orientador - SEL/EESC/USP

Dr. Renato Varoto - Pós-doutorado/ UNICAMP

Mestre Renata Manzano Maria - UNICAMP

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado Rogério Andrade Flauzino

DEDICATÓRIA

À minha mãe Sandra e a toda minha família que, com muito carinho e apoio, não mediram esforços para que eu concluísse mais esta etapa da minha vida.

AGRADECIMENTOS

Aos meus avós Mitiko e Masatuki (in memoriam), por todo o suporte dado à minha formação.

Aos meu familiares, pelo apoio em todos estes anos de graduação.

Ao professor Cliquet, meu orientador, pelo incentivo e a oportunidade de desenvolver este trabalho.

Ao Dr. Renato Varoto, pelos auxílios e conselhos dados ao longo do projeto.

À sociedade paulista, cuja contribuição financia esta universidade e custeou meus estudos.

RESUMO

ASSIS, V. H. I. **Desenvolvimento de aplicativo web para dinamômetro de preensão palmar**. 2017. 87 f. Trabalho de Conclusão de Curso (Engenharia Elétrica com ênfase em Eletrônica) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2017.

A medição da força de preensão palmar (FPP) é uma forma de avaliar quantitativamente a força que uma pessoa é capaz de exercer com as mãos para pegar objetos. Sua importância está relacionada ao seu uso como ferramenta tanto de acompanhamento em tratamentos de reabilitação de pacientes com problemas de preensão palmar quanto de estudo da característica da preensão palmar em pessoas saudáveis. Este trabalho teve por objetivo desenvolver uma aplicação baseada na internet - aplicação web - para receber, armazenar em um servidor e disponibilizar graficamente as medidas de FPP coletadas ao longo do tempo da preensão por um dinamômetro de preensão palmar. Essa abordagem se utiliza do protocolo de troca de mensagens Message Queuing Telemetry Transport (MQTT) para desacoplar fisicamente o instrumento de medição do sistema que processa e gerencia as medidas coletadas. Isso porque esse protocolo se baseia na arquitetura publicador-subscritor na qual o servidor MQTT atua como um intermediário na comunicação entre todos os instrumentos de FPP e a aplicação web. Quanto à aplicação, ela tem a vantagem de armazenar as informações das medidas em um banco de dados centralizado, o que permite que elas possam ser analisadas e compartilhadas por pesquisadores à distância que, por suas vezes, podem contribuir para aumentar a massa de dados do sistema e melhorar a qualidade das análises.

Palavras chave: Aplicativo web. Força de preensão palmar. Célula de carga. Protocolo MQTT. Esp8266. Micropython. Flask. Bokeh.

ABSTRACT

ASSIS, V. H. I. **Development of web application for hand grip dynamometer**. 2017. 87 f. Trabalho de Conclusão de Curso (Engenharia Elétrica com ênfase em Eletrônica) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2017.

The measurement of palmar grip strength (FPP) is a way of assessing quantitatively the strength that a person is able to exercise with his hands to pick up objects. Its importance is related to its use as a tool both for monitoring in rehabilitation treatment of patients with problems of palmar prehension and for a study of the characteristic of palmar prehension in healthy people. The objective of this work was to develop an internet based application - web application - to receive, to store into a server and to graphically display the FPP measures collected over the time of the prehension by a hand grip dynamometer. This approach uses the protocol of exchange Message Queuing Telemetry Transport (MQTT) to decouple physically the measuring instrument of the system that handles and manages the collected measures. This is because the protocol is based on the architecture publish-subscribe in which the server MQTT acts as an intermediary in the communication between all instruments of FPP and web application. With regard to the application, it has the advantage of storing measurement information in a centralized database, that allows them to be analyzed and shared by researchers remotely which, in turn, may contribute to increase the mass of system data and improve the quality of the analysis.

Key words: Web application. Hand grip strength. Load cell. MQTT protocol. Esp8266. Micropython. Flask. Bokeh.

LISTA DE FIGURAS

ÍNDICE DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Os tratamentos de recuperação de pacientes, bem como os estudos dos mesmos – em casos tais como de lesões medulares com comprometimento da força das mãos, problemas de perda muscular e óssea que afetam a capacidade de fazer força com as mãos em decorrência de idade avançada ou de doenças degenerativas tais como artrite reumatoide (AR)[1,2] ou infecciosas como a hanseníase[3], contusões ou mesmo lesões musculares por uso intenso das mãos e braços em práticas esportivas[4] – podem ser beneficiados pela medição da FPP pois ela proporciona a criação de um método de acompanhamento do histórico do tratamento e da capacidade de preensão palmar de pacientes.

Seguindo a tendência da Web 2.0, ou seja, a tendência de migrar para a internet os recursos e serviços que tradicionalmente são disponibilizados localmente nos computadores dos usuários – como, por exemplo, editores de texto e de planilhas, armazenamento em nuvem, etc. – a fim de tornar a própria internet uma plataforma de aplicação que fornece serviços e experiências como se fosse uma extensão do próprio computador do usuário, o projeto atual propõe desenvolver um aplicativo web para receber, processar, armazenar em um banco de dados e disponibilizar graficamente em um navegador de internet as medidas de FPP lidas e enviadas por um dinamômetro eletrônico. Com isso será possível desacoplar fisicamente do instrumento de medição o sistema que processa e gerencia as medidas coletadas com a finalidade de eliminar a necessidade de instalação de aplicações gráficas nos computadores de cada usuário que conduzir medições com um dinamômetro de FPP. Assim, enquanto o sistema de processamento das medidas fica centralizado em uma única máquina de um departamento de fisioterapia, por exemplo, muitos dinamômetros de FPP podem conectar-se a ele e ter suas medidas visualizadas por meio de um navegador de internet.

Como as aplicações baseadas na internet são de fácil aprendizagem e amplamente utilizadas atualmente, os usuários deste sistema não deverão ter problemas em utilizá-lo, pois ele não exige conhecimentos técnicos específicos. Ademais, a conveniência em se armazenar as informações das medidas em um banco de dados centralizado e acessível por meio da internet permite que tais informações possam ser analisadas e compartilhadas por pesquisadores à distância que, por suas vezes, podem contribuir para aumentar a massa de dados do sistema e melhorar a qualidade das análises realizadas.

1.1 FORÇA DE PREENSÃO PALMAR E SUA AVALIAÇÃO

A avaliação da FPP está inserida no contexto do estudo da força de preensão manual (FPM), que envolve a análise da força das mãos em todos os tipos de movimentos possíveis. No universo das avaliações, a avaliação pela FPP é a mais simples por envolver apenas o movimento de pegada da mão e isso pode ser facilmente medido por um dinamômetro de empunhadura.

Conforme as pesquisas bibliográficas consultadas indicam[1,4,5] e os trabalhos ilustram[2,3,5-8], a FPP é a medida mais referenciada na literatura e é realizada tradicionalmente por dinamômetros isométricos, sejam eles analógicos ou digitais[4]. Dentre os dinamômetros, o Jamar é reconhecido pela literatura como o instrumento padrão para medidas de FPP[1,4,5]. Ele é um instrumento tradicionalmente hidráulico analógico, figura 1, mas também possui modelos digitais, figura 2, que podem realizar até mesmo medidas de força máxima.

Figura 1 – Jamar hidráulico analógico.

Fonte: PhysioSupplies [20--?].

Figura 2 – Jamar digital.

Fonte: FysioSupplies B.V. [20--?].

Além dos hidráulicos, a categoria dos dinamômetros analógicos ainda possui os pneumáticos, que medem a pressão exercida pela FPP ao comprimir o ar por um bulbo do instrumento[1]. Quanto aos instrumentos digitais, eles podem ser do tipo que faz apenas medições de força máxima ou do tipo que faz medições de força ao longo do tempo em que a força é aplicada – Jamar Smart. Este último tipo de instrumento, ao permitir traçar curvas de força pelo tempo, possibilita que se estude variáveis da FPP que ocorrem ao longo do

intervalo da preensão[4,6] tais como valores de força mínima, média e máxima, capacidade de sustentação da força aplicada, fadiga, área sob a curva, etc[1,6].

Conforme avaliado no estudo de (FIGUEIREDO et al., 2007), muitos fatores podem influenciar os resultados das medições, a ponto de não se poder realizar comparações de medidas entre dois ou mais estudos de FPP. O principal fator está relacionado ao posicionamento dos corpos dos pacientes durante as medições. Justamente por isso, a Sociedade Americana de Terapeutas de Mão (ASHT, em inglês) definiu um protocolo que recomenda que as medições sejam feitas com os pacientes sentados em uma cadeira em posição ereta, com as pernas apoiadas no chão e mantendo os joelhos flexionados com ângulo de 90º, ombros em posição neutra, cotovelos em 90º e braço mantido suspenso no ar segurando o dinamômetro com o punho também em posição neutra. A figura 3 a seguir ilustra o posicionamento recomendado pelo protocolo da ASHT para medições de FPP.

Figura 3 – Posição recomendada pela ASHT para a medição da FPP.

Fonte: Nunes e Bruno (2012)[7].

O protocolo da ASHT ilustra a necessidade de se padronizar as medições para que as medidas realizadas possam ser agrupadas e comparadas entre diferentes populações, livres da influência da posição corporal. Entretanto, é preciso considerar que as próprias condições físicas dos pacientes submetidos às medições podem impedir que se satisfaça algum dos requisitos do protocolo padrão. Portanto, é preciso não apenas seguir um protocolo padronizado mas ponderar se todos as exigências de tal protocolo podem ser satisfeitas pelo

grupo de pacientes do estudo ou se será necessária alguma modificação do protocolo para que ele atenda às necessidades desse grupo em estudo.

Ainda existem outro fatores influenciadores para as medições de FPP, como o sexo dos pacientes, idade, condições de saúde, peso e altura, tamanho das mãos, dominância entre as mãos[1,5], que precisam ser ponderados sobre se devem ou não ser relevantes a determinado estudo.

O artigo de (MATHIOWETZ et al., 1985), com dados normativos de FPP, também recomenda que se realizem três medições e que a média entre elas é que deva ser usada como valor de FPP. Já o artigo (MATHIOWETZ, 1990) afirma não ter encontrado um consenso na literatura consultada sobre o intervalo de descanso adequado entre as medições, mas que, em sua investigação sobre o efeito da fadiga devido às três repetições, foi verificado que um descanso de 15 segundos entre as medidas é suficiente para se ignorar o efeito da fadiga, pois as medições com 60s de intervalo resultaram em uma diferença de menos de 0,5 kgf e isso pode ser desprezado.

Com relação aos dados normativos, o estudo de (MATHIOWETZ et al., 1985) revela em sua tabela 2 que o intervalo de FPP da população estadunidense de adultos homens saudáveis, com 20 anos ou mais, abrange valores de força entre 25,0kgf e 55,2kgf e que o intervalo das mulheres compreende valores entre 17,1kgf e 35,7kgf. Os dados normativos para a população brasileira foram levantados no estudo de (CAPORRINO et al., 1998), com adultos saudáveis com idades entre 20 e 59 anos. O intervalo de valores de força para os homens foi entre 38,2kgf e 46,3kgf e para as mulheres, entre 27,2kgf e 32,9kgf.

Embora haja literatura a respeito do instrumento adequado para medir a FPP e como deve ser o protocolo a ser adotado para essa medição, não foi possível encontrar um estudo de referência sobre a frequência de amostragem adequada para a conversão do sinal analógico da medida de FPP para um sinal digital - conversão AD. Os trabalhos consultados utilizam valores que variam entre 28Hz e 320Hz[2,6,7,8,12]. Portanto, o projeto atual deixa definido como padrão o valor de 200Hz para a frequência da conversão AD, embora permita que se possa variar a frequência entre 20Hz e 200Hz. Frequências de aquisição superiores a 200Hz se mostraram impraticáveis pelo sistema adotado pelo projeto devido ao tempo consumido pelas rotinas internas de gerenciamento do microcontrolador utilizado, bem como pelo próprio sistema de aquisição das medidas.

1.2 OBJETIVOS

O objetivo deste projeto é utilizar os conhecimentos adquiridos durante o curso de graduação para desenvolver um aplicativo web para receber, processar, armazenar em um banco de dados e disponibilizar graficamente em um navegador de internet as medidas de FPP lidas e enviadas por um dinamômetro eletrônico. Com essa abordagem, portanto, torna-se desnecessário o sistema de comunicação serial entre o dinamômetro e o computador que contém o software de processamento, assim como o próprio software LabVIEW utilizado por (NUNES; BRUNO, 2012), que representa um grande custo no projeto, tanto em termos de licença quanto de gestão e manutenção de versões executáveis para cada sistema operacional no qual possa estar instalado. Em vez disso, o projeto propõe que se instale o sistema da aplicação web em apenas um computador central para que os usuários acessem-no por um navegador de internet.

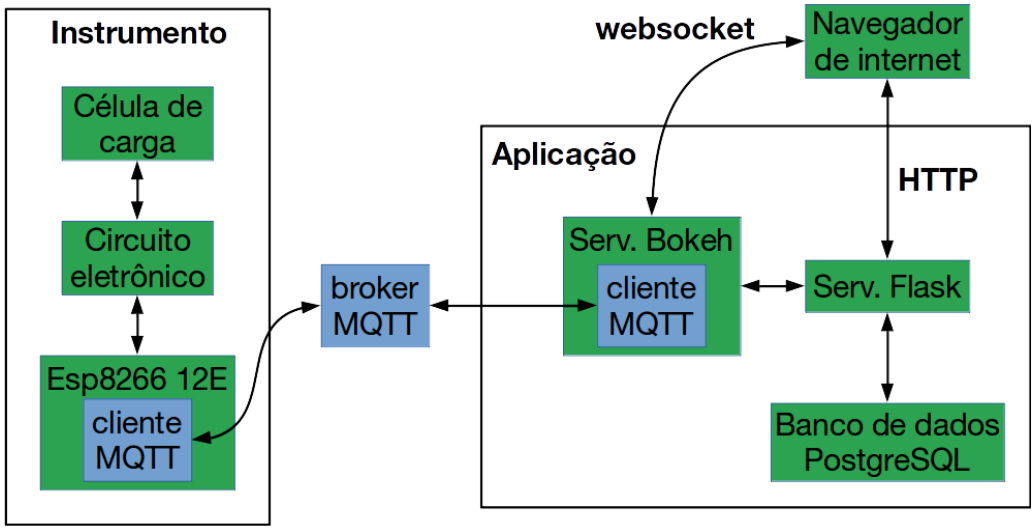
A construção do dinamômetro de FPP utilizado já foi trabalhada nos projetos anteriormente realizados (NUNES; BRUNO, 2012) e (HORITA, 2015) e por isso a célula de carga e a manopla são as mesmas. A proposta atual, contudo, modifica o circuito do instrumento para que ele possa ser alimentado por uma fonte de tensão entre 12V – como uma fonte de tomada ou uma bateria – e 24V. Nos trabalhos anteriores foram utilizadas duas baterias de 12V cada para obter uma tensão de operação de 24V.

Também é feito um circuito para dividir a tensão de alimentação em tensões simétricas que devem alimentar tanto o circuito de aquisição de sinal de medida quanto o módulo microcontrolador Esp8266, responsável por realizar a conversão do sinal analógico medido para um sinal digital e então enviá-lo para a aplicação web por meio de uma conexão de internet sem fio (Wi-Fi). A medida deve chegar à aplicação por mensagens enviadas com o protocolo MQTT.

Como resultado esperado, o dinamômetro deve ser facilmente operado e a aplicação web deve se comportar como uma extensão natural do instrumento, apresentando uma experiência de uso simples e intuitiva como a navegação por um website usual da internet.

1.3 PROJETO

Figura 4 – Arquitetura do sistema da aplicação web.



Fonte: autoria própria (2017).

O projeto é composto por três níveis de abstração. O primeiro envolve o nível eletrônico do trabalho e é responsável pelo circuito de instrumentação, que abrange desde os trabalhos com a célula de carga até os circuitos da fonte de alimentação de tensão e de leitura e condicionamento das medidas.

O segundo é um nível intermediário onde o foco é posto sobre o sistema gerenciado pelo microcontrolador Esp8266. Nessa etapa são feitas as conversões AD das medidas fornecidas pelo dinamômetro que então são enviadas por meio de mensagens MQTT para a aplicação web por meio de uma conexão Wi-Fi e por intermédio de um servidor broker MQTT, como ilustrado na figura 4.

A aplicação web compreende o alto nível. Elá é a responsável por se comunicar com o módulo Esp8266 para controlar a leitura AD das medidas de FPP e então recebê-las e processá-las, convertendo seus valores digitalizados de tensão em valores de força. Antes de a leitura acontecer, contudo, é preciso o dispositivo microcontrolador que seja registrado no sistema, por meio do seu código MAC, para que identifique o dinamômetro utilizado na medição. Também é preciso que o dinamômetro seja calibrado em software pelo sistema de calibração presente na aplicação web.

A aplicação também deve ser capaz de plotar, em um gráfico de fluxo de dados, as medidas recebidas em tempo real para que uma janela de dados úteis, dentro do gráfico de

fluxo de dados e a critério do usuário, possa ser selecionada a fim de que seja analisada ou salva em um banco de dados. Por fim, uma vez salva, a medida deve ficar disponível para ser plotada pela aplicação web conforme a necessidade do usuário.

Quatro partes principais compõem o sistema da aplicação web para que ela possa realizar tais processos, conforme pode ser observado na figura 4. São o broker Mosquitto e seus clientes MQTT, o banco de dados PostgreSQL, a aplicação da biblioteca web Flask e a aplicação da biblioteca de visualização gráfica interativa Bokeh.

A aplicação da biblioteca Flask é a responsável por gerir o acesso aos recursos da aplicação web, ou identificadores uniformes de recursos (URI), tanto entre o navegador de internet e a própria aplicação web quanto entre o servidor Bokeh e o banco de dados. Ficam nela os modelos de dados que definem os usuários, os dispositivos, as medidas e calibrações e as relações entre eles. É ela também a responsável pelo registro de usuários e por controlar seus acessos aos dados.

O controle de usuários abrange o cadastro dos mesmos no banco de dados e um sistema de recuperação de acesso, ambos com confirmação por e-mail. Um usuário pode visualizar somente os dados cadastrados por ele mesmo.

2 DESENVOLVIMENTO

2.1 CÓDIGO FONTE

O projeto das aplicações web e do microcontrolador foram desenvolvidos na distribuição linux Ubuntu 14.04 com o auxílio do programa git[13], que é uma ferramenta para gerenciar versões de código-fonte. Essa ferramenta possui um serviço online chamado GitHub[14] que armazena os códigos-fonte versionados dos projetos através de seus respectivos repositórios git. Ao todo 3 repositórios compõem o código-fonte do projeto e estão definidos na tabela 1 a seguir.

Tabela 1 – Repositórios do código-fonte do projeto.

Nome	Função	Endereço
tcc	instala e configura os sistemas do projeto	https://github.com/Corleo/tcc
webapp	sistemas da aplicação web	https://github.com/Corleo/webapp
webesp	sistema do microcontrolador Esp8266	https://github.com/Corleo/webesp

Fonte: autoria própria (2017).

O ambiente linux foi escolhido para o desenvolvimento do projeto porque apresenta um alto nível de automatização para configurar, baixar, instalar e atualizar programas através de scritps. As instruções e o script para a instalação do projeto, assim como as aplicações utilizadas por ele, estão disponíveis na página inicial do repositório *tcc*.

2.2 MICROCONTROLADOR ESP8266

O módulo Esp8266, figura 5, foi escolhido para compor o sistema uma vez que reúne em um microcontrolador características que dispensam a necessidade de se utilizar módulos externos a ele para realizar as tarefas de aquisição de dados, processamento, comunicação Wi-Fi e controle.

Figura 5 – Módulo Esp8266 12E.

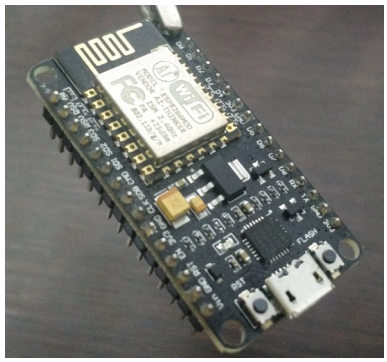
Fonte: Gearbest[15] [20--?].

Com dimensões de 24mm por 16mm o módulo ESP8266 12E, da fabricante chinesa Espressif, é um microcontrolador RISC de 32 bits alimentado com 3,3V que pode operar tanto em 80MHz quanto em 160MHz. Sua única entrada para conversão AD (ADC) é de 10 bits e opera com faixa de tensão de entrada entre 0V e 1V. Tem suporte às interfaces SPI, I²C, I²S e UART. Para a comunicação ele possui Wi-Fi no padrão 802.11 b/g/n, tem suporte para conexões TCP/IP e para autenticações WEP e WPA/WPA2. Quanto à memória, tem flash ROM reservada para programação que varia entre 512KB e 4MB, 64KB de memória RAM para instruções e 96 KB para dados - que diminuem para menos de 32KB livres para o usuário final em razão do uso de RAM pelo firmware e pelas pilhas de Wi-Fi e TCP/IP. Além de todos os recursos presentes seu custo é baixo, cerca de US$ 3.

Entretanto, como pode ser visto na figura 5, o módulo Esp8266 em si não contém os circuitos de interfaceamento USB-TTL, pinos e conectores, regulador de tensão para 3,3V e botões para controle de reset e de gravação de firmware. Como o kit de desenvolvimento NodeMcu Esp8266 possui esses circuitos, figura 6, ele foi utilizado no lugar do módulo Esp8266 a fim de facilitar a prototipagem do projeto. Por isso, uma vez terminada a prototipagem, o kit de desenvolvimento pode ser substituído pelo módulo Esp8266 em si.

Apenas será necessário se que faça as devidas conexões e soldagens de fios diretamente no módulo conforme as recomendações de seu datasheet.

Figura 6 – Kit de desenvolvimento NodeMcu.



Fonte: autoria própria (2017).

2.3 LINGUAGENS PYTHON E MICROPYTHON

As especificações do módulo Esp8266 tornaram-no muito popular entre desenvolvedores e entusiastas, chegando ao ponto de codificarem firmwares específicos para ele, além do padrão em C do fabricante, como o do Arduino IDE em uma linguagem baseada em C/C++, o NodeMcu em uma linguagem Lua para dispositivos embarcados, a eLua, e o firmware em micropython, uma versão da linguagem python para dispositivos embarcados.

Python[16] é uma linguagem de programação de código aberto, interpretada, orientada a objeto, de alto nível, com tipagem dinâmica, de sintaxe simples e de fácil aprendizagem com foco na legibilidade dos seus códigos escritos. Os arquivos escritos nessa linguagem possuem a extensão **.py*. Contudo, antes de ser executado de fato, o código fonte em python é compilado pelo interpretador python canônico, o *cpython*, em uma representação interna de codificação chamada *bytecode* cuja extensão é **.pyc*. Então o bytecode produzido é executado por uma máquina virtual python. Esse procedimento tem o objetivo de otimizar a execução do código python tanto ao traduzi-lo para uma representação mais próxima do código de máquina quanto ao evitar a necessidade de reinterpretação de um mesmo código python a partir de sua segunda execução caso já exista o arquivo bytecode.

Dada a facilidade de codificação que a linguagem proporciona, a grande quantidade e variedade de bibliotecas disponíveis, a existência de uma comunidade de usuários consolidada que contribui para a linguagem produzindo bibliotecas, extensões, documentações, etc., ela foi escolhida como a linguagem base da aplicação. Portanto, o firmware em micropython

1.9.1 foi escolhido para o módulo Esp8266 e o seu código de aplicação também foi escrito nessa linguagem. Já a aplicação web foi majoritariamente construída em python 2.7.

A linguagem micropython[17], por sua vez, foi desenvolvida para ser um compilador python completo com foco em dispositivos embarcados e com o máximo de compatibilidade com a linguagem python. Ela ainda conta com um prompt interativo (REPL) que permite escrever e executar comandos diretamente nos dispositivos embarcados e, mesmo assim, é leve o suficiente para ser executada em hardwares tão compactos quanto um com apenas 256KB de memória de programação e 16KB de RAM. Adicionalmente a essa compatibilidade com a versão python padrão, a linguagem embarcada traz módulos python específicos para os dispositivos que provêem acesso ao hardware em baixo nível como as portas de entrada e saída (IO), ADC, temporizadores, UART, etc.

Na versão embarcada os códigos em bytecode têm a extensão **.mpy*. Além disso, os arquivos e módulos python podem ser executados pelo interpretador ou diretamente da memória flash onde ficam armazenados, quando *congelados*, ou da maneira usual que é sendo carregados na memória RAM, compilados nessa memória para então serem executados usando a memória RAM. Assim, arquivos e módulos *frozen*, ou *congelados*, promovem uma redução do consumo de memória RAM da aplicação.

Para serem *congelados* os módulos e arquivos precisam ser adicionados aos respectivos diretórios de módulo e de *script* durante o processo de composição do firmware micropython. Já o processo de pré-compilação para bytecode pode ser feito de maneira avulsa usando uma ferramenta de compilação cruzada que é executada em um computador comum. Nesse último caso, a versão bytecode simplesmente deve ser transferida para o sistema de arquivos do dispositivo embarcado.

Também existem compiladores micropython para sistemas Unix, Windows e MacOS. Esse é um recurso de grande utilidade para otimizar a prototipagem de código para os dispositivos embarcados pois evita o trabalho repetitivo de se transferir o código do computador para o dispositivo a cada modificação para testar a validade do mesmo. Contudo, como essas versões de compiladores não são para sistemas embarcados elas não fornecem e nem compreendem os módulos micropython que dão acesso ao hardware. Como solução de contorno, portanto, devem ser criadas funções que emulem o comportamento esperado do hardware para que a prototipagem seja realizada nestes sistemas.

2.4 APLICAÇÃO WEB

2.4.1 Servidor Mosquitto e cliente Paho

Mosquitto[18] é um projeto de código aberto da Fundação Eclipse que implementa um servidor broker para o protocolo de mensagens MQTT nas versões 3.1 e 3.1.1. Ele é um software leve e de fácil instalação e foi escolhido como o servidor broker dessa aplicação. Sua comunicação com a aplicação web é feita com um cliente MQTT implementado na própria aplicação por meio de uma biblioteca python do projeto de código aberto PAHO[19], também da Fundação Eclipse.

As telas de calibração e de medição de dados, na aplicação web, interagem com os dispositivos Esp8266 através da comunicação MQTT. Por isso, cada uma tem seu próprio cliente MQTT implementado.

2.4.2 Protocolo MQTT

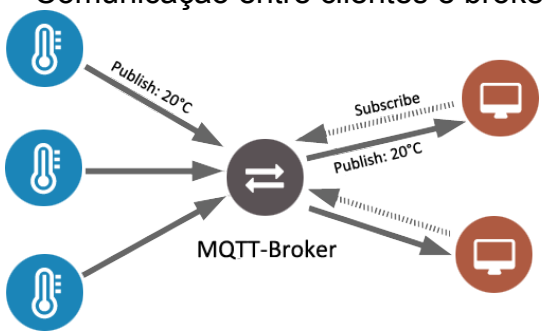
Message Queuing Telemetry Transport (MQTT) é um protocolo de transporte de mensagens em redes de computadores que usa a arquitetura *publicador-subscritor.* De código aberto, apresenta baixo consumo de banda e de potência, é leve, simples e desenhado para ser de fácil implementação. Justamente por isso sua principal aplicação está associada à comunicação entre máquinas - *Machine-to-Machine* (M2M).

O protocolo MQTT é baseado no protocolo TCP/IP assim como o HTTP, mas se diferencia deste pelo fato dos clientes MQTT não precisarem ficar solicitando e baixando recursos de que precisam[20]. Em vez disso, um servidor intermediário chamado *broker* se encarrega de enviar as informações desejadas aos clientes assim que elas forem sendo disponibilizadas, desde que os clientes façam suas subscrições nos tópicos destas informações. Portanto os clientes precisam manter suas conexões TCP abertas com o broker, ao contrário das conexões HTTP nas quais as conexões são abertas e então fechadas para cada par de requisição e resposta. Caso a conexão se perca, o broker MQTT armazena as mensagens e as envia aos clientes assim que eles voltarem a se conectar. Isso mostra que o MQTT é baseado em eventos e que sua comunicação é assíncrona por natureza.

Existem três conceitos principais que precisam estar presentes para que a troca de mensagens MQTT ocorra, além da própria mensagem: o tópico, o cliente e o servidor broker.

O tópico é uma cadeia de caracteres que define um escopo de mensagens para os quais os clientes podem publicar mensagens ou se subscreverem para receber as mensagens. Essa cadeia de caracteres pode ser simples ou ser parte de uma hierarquia maior de escopo de mensagens. Neste último caso os escopos devem ser separados por barras. Assim, uma estação meteorológica identificada pelo tópico ***tempo*** enviando medidas de temperatura e pressão precisa definir dois tópicos como, por exemplo, ***tempo/temperatura*** e ***tempo/pressao***.

Figura 7 – Comunicação entre clientes e broker MQTT.



Fonte: Pietikäinen[21] (2017).

O cliente MQTT representa qualquer dispositivo se comunicando com o broker MQTT por meio do protocolo MQTT[22]. Já o broker é um servidor responsável por receber as mensagens, filtrar os tópicos, gerenciar os clientes por meio de suas identificações, ou códigos ID, e enviar as mensagens a todos os subscritos[22]. Ele também é responsável por gerenciar as credenciais (usuário e senha) e uso de criptografia (SSL/TLS)[23] quando usados.

É importante notar, conforme ilustrado pela figura 7, que esse tipo de troca de mensagens baseado em tópicos desacopla os clientes uns dos outros de modo que os clientes conversam apenas com o servidor broker e não entre si.

No exemplo dos tópicos, havendo mais de uma estação meteorológica, uma possível solução para definir os tópicos de cada estação é remodelando-os para algo do tipo ***tempo/1/temperatura*** com o número designando a estação. Assim um cliente pode se subscrever em apenas um ou outro tópico ou então em todos. Caso queira a subscrição em todos os tópicos de todas as estações ele pode usar o caractere coringa #[24] que significa a subscrição dos tópicos de todos os escopos abaixo de sua hierarquia: ***tempo/#***. Se, por outro lado, o cliente desejar receber apenas as mensagens de pressão das estações, o caractere coringa +[24] é que deve ser usado: ***tempo/+/pressao***. Esse caractere significa a subscrição dos tópicos de todos os escopos da hierarquia na qual o caractere foi posicionado.

As mensagens são compostas pela carga útil, que contém a informação de fato, e o seu cabeçalho no qual ficam definidos seus parâmetros, tais como o tópico e o fator de *qualidade de serviço* (QoS) da mensagem. Esse fator pode variar entre os valores 0, 1 ou 2 e determina o nível de garantia de entrega e de recebimento das mensagens entre o cliente e o broker e não entre o publicador e o subscritor[23].

QoS igual a 0 ou, *no máximo uma vez*[25], é o nível no qual a mensagem é enviada ao publicador e apagada em seguida. Não há garantia e nem confirmação de entrega. Contudo, é o modo de envio mais rápido.

QoS igual a 1 ou, *pelo menos uma vez*[25], armazena a mensagem no publicador até que ele receba uma mensagem de confirmação de recebimento. Não há garantias de que a mensagem não vá ser recebida mais de uma vez caso o publicador envie mais de uma vez a mensagem.

QoS igual 2 ou, *exatamente uma vez*[25], garante que a mensagem será entregue apenas uma vez por meio do envio de confirmação de recebimento pelo recebedor e da confirmação de recebimento da confirmação pelo enviador. É o modo mais seguro, porém mais lento de todos[24].

2.4.3 Servidor de banco de dados PostgreSQL

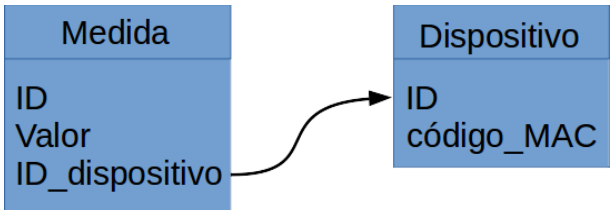
PostgreSQL é um banco de dados relacional de código aberto desenvolvido pelo PostgreSQL Global Development Group[26]. É multiplataforma, usa a linguagem padrão SQL como forma de consultar os dados mas também tem interfaces de programação em diferentes linguagens de programação, inclusive python.

Basicamente, um banco de dados relacional é um banco que representa os dados em modelos de tabelas - ou relações. Neste modelo, as linhas de uma tabela representam uma coleção de valores de um objeto ou entidade e as colunas, um determinado tipo de informação sobre o objeto, com seus campos armazenando os respectivos valores da informação.

As linhas de uma tabela são identificadas univocamente por um código de identificação chamado chave primária. Já o relacionamento entre os objetos de duas ou mais tabelas se dá por meio de chaves estrangeiras. Elas são simplesmente chaves secundárias presentes na tabela que armazena a relação e têm sua própria coluna nessa tabela. Os valores dessas chaves externas são os valores das chaves primárias nas tabelas que estão sendo relacionadas.

Assim, por exemplo, as informações de dispositivos Esp8266 e de medidas realizadas são armazenadas em duas tabelas, figura 8, uma para dispositivos e outra para medidas, com cada uma de suas linhas representando um objeto da tabela. O relacionamento entre elas, então, é feito incluindo uma coluna na tabela de medidas para armazenar o valor da chave primária do dispositivo correspondente da tabela de dispositivos.

Figura 8 – Relacionamento entre tabelas em banco de dados relacional.



Fonte: autoria própria (2017).

Isso permite realizar consultas mais abrangentes ao banco de dados ao utilizar informações de duas ou mais tabelas por meio de operações de *junção* de tabela, que utilizam as chaves estrangeiras para relacionar os dados entre as tabelas. Como exemplo, é possível citar a consulta por medidas contendo apenas aquelas realizadas por um dispositivo com determinado código MAC. Essa pesquisa precisa verificar ambas as tabelas de *Medida* e de *Dispositivo* e relacionar as informações de ambas por meio da chave estrangeira *ID_dispositivo*.

Uma vantagem do PostgreSQL sobre outros bancos de dados relacionais é que ele possui suporte ao armazenamento de dados no formato JSON[27], que é um formato de texto usado para descrever, armazenar e transmitir informações de objetos baseadas em coleções de pares chave/valor com esses valores podendo ser outros objetos, números, strings, valores booleanos e arrays.

Isso proporciona uma maior liberdade para modelar objetos que não são descritos estritamente em forma de tabelas, como é o caso dos valores das medidas desta aplicação web, por exemplo. Não se sabe à priori quantos pontos serão armazenados por medida e cada uma delas pode ter uma quantidade distinta de pontos, o que torna inviável definir uma coluna da tabela para cada ponto da medida. Uma solução mais adequada é definir na tabela de medidas um único campo JSON para armazenar as medidas, com este campo sendo um objeto contendo dois arrays, um para armazenar valores de força e outro para valores de tempo.

Com as características do banco de dados descritas, foram definidos para a aplicação web os seguintes modelos de dados de usuários, tabela 2, dispositivos, tabela 3, calibrações,

tabela 4, e medidas, tabela 5. O relacionamento entre os usuários e os dispositivos utilizados estão ilustrados pela figura 9. A figura 10 ilustra o relacionamento entre os dispositivos e suas calibrações. Já a figura 11 contém o relacionamento entre as medidas e os dispositivos e suas calibrações que foram utilizados na medição.

Tabela 2 – Tabela de usuários: users.

Tipo	Campo
uuid	id
string	username
string	password
string	firstname
string	lastname
string	email
boolean	admin
boolean	confirmed
timestamp	confirmed_on
timestamp	created_on
timestamp	updated_on
json	token
json	using_devices

Fonte: autoria própria (2017).

Tabela 3 – Tabela de dispositivos: devices.

Tipo	Campo
uuid	id
string	code
uuid	created_by
timestamp	created_on
uuid	updated_by
timestamp	updated_on

Fonte: autoria própria (2017).

Tabela 4 – Tabela de calibrações: calibrations.

Tipo	Campo
uuid	id
uuid	device_id
json	data
uuid	created_by
timestamp	created_on
uuid	updated_by
timestamp	updated_on

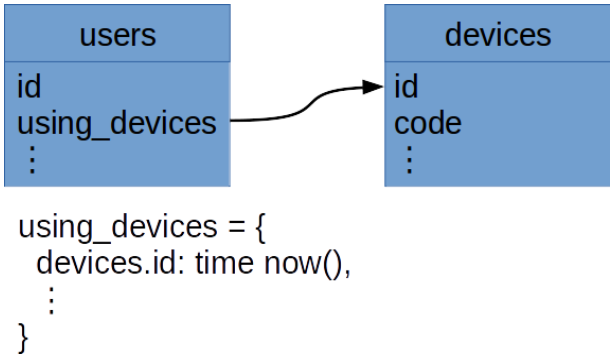
Fonte: autoria própria (2017).

Tabela 5 – Tabela de medidas: measures.

Tipo	Campo
uuid	id
uuid	device_id
uuid	calibration_id
json	data
uuid	created_by
timestamp	created_on
uuid	updated_by
timestamp	updated_on

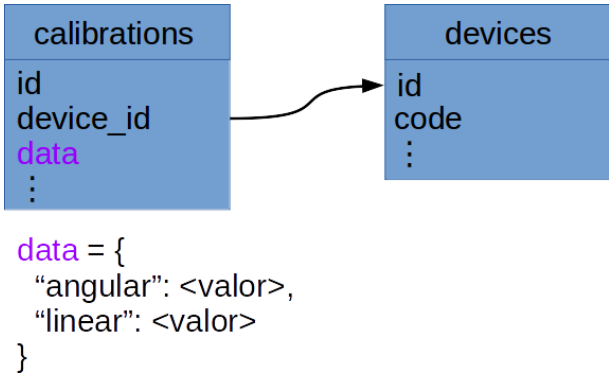
Fonte: autoria própria (2017).

Figura 9 – Relacionamento dos usuários com os dispositivos utilizáveis.



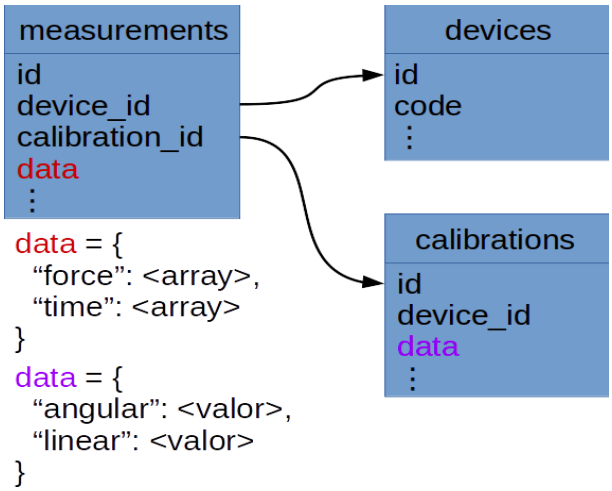
Fonte: autoria própria (2017).

Figura 10 – Relacionamento das calibrações com os dispositivos.



Fonte: autoria própria (2017).

Figura 11 – Relacionamento das medidas com as calibrações e os dispositivos.



Fonte: autoria própria (2017).

2.4.4 Biblioteca de aplicação Flask

Flask é uma biblioteca para desenvolvimento de aplicação web escrita em python e de código aberto.[28] Enxuta e de fácil aprendizagem, ela possui bastante documentação na internet e uma grande comunidade de usuários que desenvolvem bibliotecas auxiliares que estendem as funcionalidades das aplicações Flask.

Por meio das chamadas *routes*, ou rotas, essa biblioteca apresenta uma forma muito simples e intuitiva para disponibilizar os recursos de uma aplicação web. Eles são registrados em rotas de acesso que relacionam endereços URL dos recursos às funções em python responsáveis pelo processamento de tais recursos. Com esse conceito cada tela da aplicação e

cada operação de consulta ao banco de dados é um recurso e, portanto, tem uma rota associada.

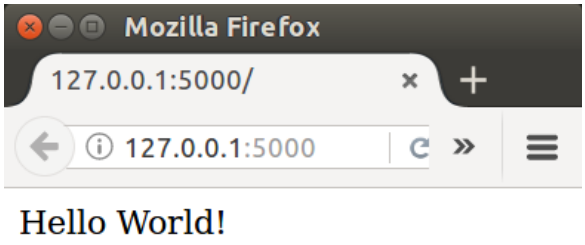
No clássico teste “olá mundo”, figura 12, para o caso de um servidor, por exemplo, a função *hello()* é atrelada à rota raiz "/" de forma simples e explícita. Assim, se o servidor da aplicação estiver sendo hospedado no endereço 127.0.0.1:5000, basta digitar 127.0.0.1:5000/ na barra de endereços do navegador para que o recurso *hello()* seja buscado no servidor da aplicação e retorne a frase "Hello World!" que então será impressa na tela do navegador, figura 13.

Figura 12 – Exemplo de um servidor Flask básico com seu sistema de roteamento.

```
1  from flask import Flask
2  app = Flask(__name__)
3
4  @app.route("/")
5  def hello():
6      return "Hello World!"
```

Fonte: autoria própria (2017).

Figura 13 – Resposta do servidor Flask ao acesso da rota “/”.

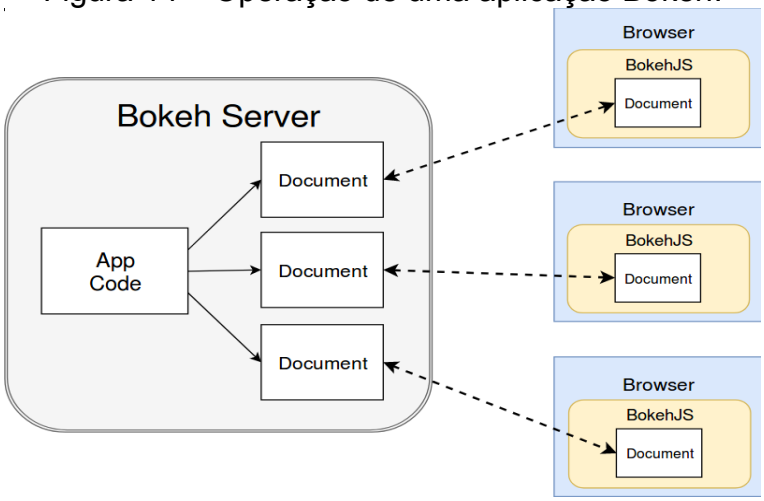


Fonte: autoria própria (2017).

2.4.5 Biblioteca gráfica Bokeh

Bokeh é uma biblioteca de código aberto para visualização gráfica e interativa, escrita em python e com a finalidade de ser utilizada por meio de navegadores de internet. Seu objetivo de projeto é proporcionar interatividade com alta performance tanto para casos com uma grande quantidade de dados quanto para streaming de dados.[29]

Figura 14 – Operação de uma aplicação Bokeh.



Fonte: CONTINUUM ANALYTICS (2017)[29].

O servidor Bokeh transforma o código da aplicação em um documento Bokeh que tem uma versão compilada para JavaScript e enviada ao navegador web onde ele é interpretado pela biblioteca JavaScript BokehJS[29], figura 14. Com isso esses documentos mantêm as informações da aplicação sincronizadas entre o servidor Bokeh e o navegador de internet.

Esse servidor Bokeh é uma modificação do servidor Tornado[30], que é um servidor web escrito em python e que utiliza uma biblioteca web de código assíncrono capaz de realizar comunicações com protocolo websocket. O protocolo websocket[31] é implementado, assim como o HTTP, sobre o protocolo TCP, mas, diferentemente desse, o protocolo websocket possui comunicação full-duplex, ou seja, possui dois canais de comunicação, com um para cada sentido da comunicação, o que permite que a comunicação nos canais aconteça simultaneamente. Ademais, o protocolo mantém o canal de comunicação aberto até que uma das partes a encerre. Com isso o protocolo websocket permite que haja uma grande redução no tamanho do cabeçalho a ser enviado, quando comparado ao HTTP, e também no tempo de transmissão dos dados, já que a conexão permanece aberta. Assim, o protocolo define regras de forma que o servidor envie dados continuamente ao navegador de internet sem que para isso o navegador necessite ficar fazendo solicitações de envio.

O servidor trabalhando com websocket, portanto, permite que a aplicação Bokeh seja capaz de transmitir um fluxo de dados em tempo real para um navegador de internet e isso é essencial para a criação da aplicação de medida de FPP.

Foram criadas três telas na aplicação do projeto utilizando a aplicação Bokeh. São as telas de calibração, de medição e de plotagem de medidas salvas.

2.5 SISTEMA MICROCONTROLADO

Este é o sistema responsável por realizar a conversão AD e o envio das medidas da célula de carga para a aplicação e também por executar e responder aos controles do usuário da aplicação web através de mensagens MQTT.

2.5.1 Firmware

A biblioteca MQTT em micropython só tem os níveis 0 e 1 de QoS e o uso de QoS igual a 1 introduz um tempo de espera que inviabiliza a aquisição e o envio das medidas em um fluxo de dados em tempo real. Além do mais, o armazenamento das medidas necessário ao QoS igual a 1 pode consumir toda a memória RAM do dispositivo e interromper o sistema. Portanto foi usado o nível de QoS 0 para as medidas. Embora exista a possibilidade de comprometimento na entrega das medidas com esse nível de QoS, essa possibilidade pode ser reduzida pela proximidade física entre o servidor broker e os clientes MQTT e pelo uso de um canal Wi-Fi com baixa interferência e sem competição por parte de dois ou mais roteadores.

Contudo, como é preciso manter um estado coeso de controle entre o módulo Esp8266 e a aplicação web, somente as mensagens de controle do instrumento foram configuradas para usar QoS igual a 1.

O firmware micropython foi desenvolvido para automaticamente executar os arquivos *boot.py* e *main.py*[32], nesta sequência, durante a inicialização do sistema e tem o intuito de permitir que eles sejam usados para preparar o sistema microcontrolado com as características iniciais necessárias para a execução da aplicação do usuário. A inicialização do sistema desta maneira, além de modularizar a aplicação e facilitar a manutenção de código, introduz um intervalo de tempo inicial que permite interromper a execução da aplicação principal do usuário caso ela contenha erros de codificação que levem o dispositivo a executar loops infinitos.

2.5.2 Aplicação

Devido aos dois arquivos *boot.py* e *main.py* serem esperados pelo firmware em suas versões de código fonte, eles não podem ser nem *congelados* nem pré-compilados para bytecode sob a pena de não serem reconhecidos pela inicialização. Portanto, o sistema

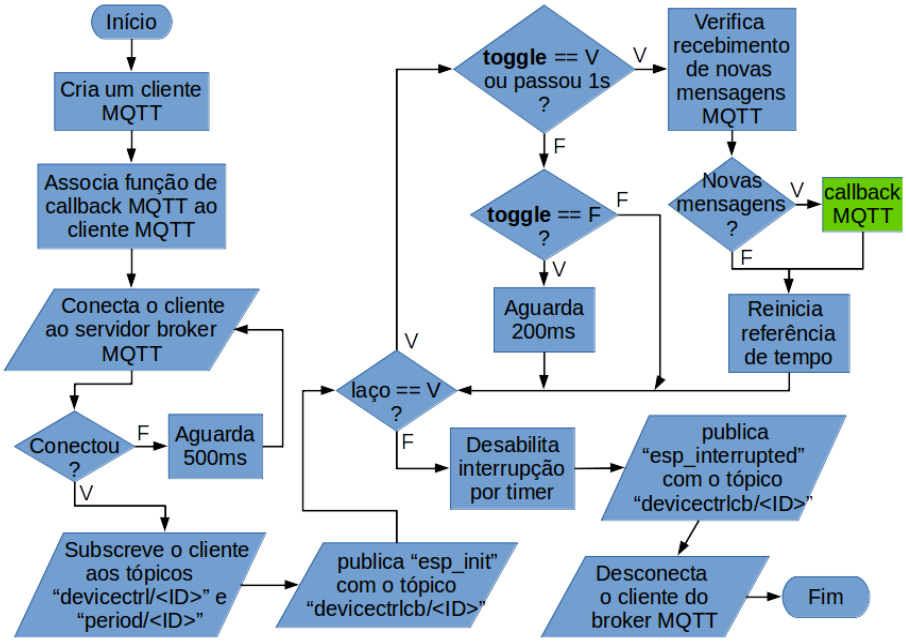
microcontrolado ficou dividido em três partes, correspondendo a três arquivos: a inicialização do sistema com a devida conexão do dispositivo à internet via Wi-Fi no arquivo *boot.py*, a chamada da aplicação principal no arquivo *main.py* e a execução da aplicação principal no arquivo *esp_app.mpy*.

A lógica de controle e instrumentação está contida no arquivo *esp_app.mpy*, que é inicializada pelo arquivo *main.py*, e só começa a ser executada depois que a inicialização obtiver sucesso em conectar o dispositivo à rede Wi-Fi, cuja configuração é feita no arquivo *boot.py*. A aplicação inicia sua operação, figura 15, associando rotinas de callback de temporização e de mensagens MQTT, que são recebidas e redirecionadas às suas respectivas interrupções. Depois cria uma conexão MQTT do dispositivo com o servidor broker para então entrar em um loop que fica verificando o recebimento de novas mensagens a cada 1s, quando o fluxo de leitura de medidas estiver ativo, ou a cada 200ms, quando não estiver ativo.

Existem seis variáveis globais na aplicação que a auxiliam no funcionamento do dispositivo:

- **tim**: objeto de configuração do temporizador.
- **toggle**: estado alternável do fluxo de dados de medidas.
- **web_ctrl_id**: identificação do usuário da aplicação web que enviou a mensagem.
- **dev_ctrl_id**: identificação do usuário da aplicação web que atualmente está controlando o dispositivo embarcado.
- **period**: valor em milissegundos do período no qual as medidas devem ser lidas.
- **measures**: array python usado como buffer circular que armazena sequencialmente os valores de medida de FPP e do instante de tempo no qual a medida foi lida, em microssegundos.

Figura 15 – Fluxograma da rotina principal da aplicação.



Fonte: autoria própria (2017).

São usados 5 tópicos MQTT para o controle, detalhados na tabela 6, sendo dois de subscrição e três de publicação. Eles têm um padrão contendo dois escopos, com o mais geral representando de fato qual o tipo da mensagem e o seguinte representando a identificação do dispositivo, o <ID>.

Tabela 6 – Tópicos MQTT da comunicação entre o Esp8266 e a aplicação web.

Tópico	Tipo	Significado
period/<ID>	subscrição	o usuário altera a frequência da medida
devicectrl/<ID>	subscrição	o usuário altera o estado de controle do módulo
devicectrlcb/<ID>	publicação	o módulo retorna seu estado de controle
measure/<ID>	publicação	o módulo envia as medidas em um fluxo de dados
calibration/<ID>	publicação	o módulo envia apenas 1 medida por vez

Fonte: autoria própria (2017).

O usuário da aplicação precisa se conectar ao dispositivos Esp8266 para poder controlá-lo. Isso é feito enviando ao dispositivo uma mensagem MQTT com o comando de conexão *conn* e o seu id de usuário para o tópico de controle devicectrl/<ID>. Com isso ele ganha a liberdade para alterar o período de leitura AD, em milissegundos, por meio do envio de uma mensagem para o tópico period/<ID>. Além de *conn*, também são aceitos outros comandos pelo tópico devicectrl/<ID> e são descritos na tabela 7 a seguir.

Tabela 7 – Relação de comandos aceitos pelo tópico de controle do dispositivo.

Comando	Significado
conn	conectar o usuário ao dispositivo
disc	desconectar o usuário do dispositivo
unsub	cancelar a subscrição do usuário aos tópicos do dispositivo e desconectá-lo
toggle	ativar/desativar o fluxo de dados de medição ao usuário
calibrate	enviar uma única medida de calibração ao usuário por comando

Fonte: autoria própria (2017).

Para cada mensagem de controle é respondida uma mensagem de confirmação de controle por meio do tópico devicectrlcb/<ID>, como pode ser visto na tabela 8. Apenas as duas últimas mensagens de confirmação não seguem esta regra, pois a iniciativa das operações associadas a elas parte do próprio dispositivo.

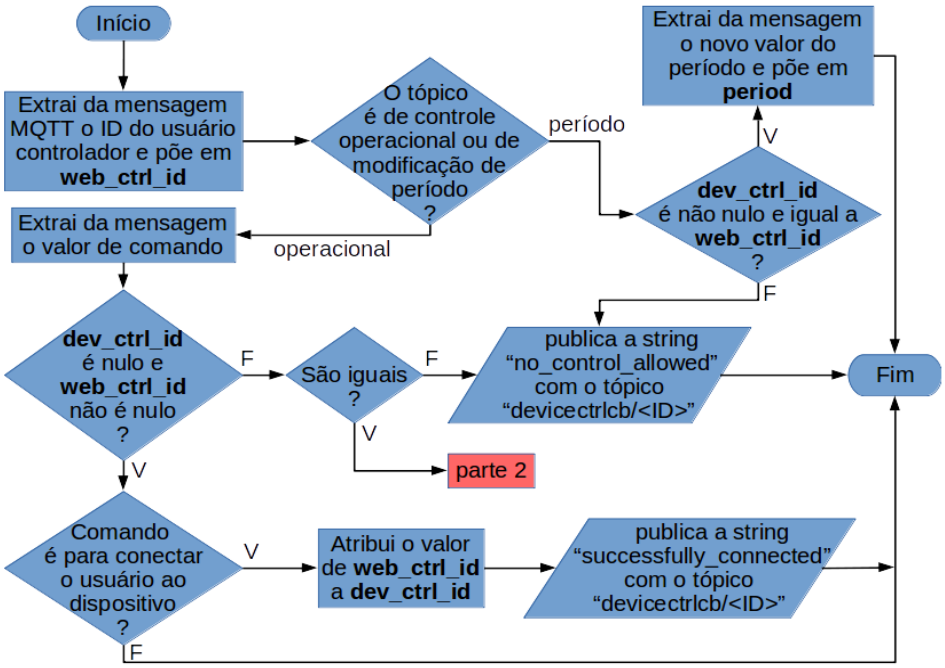
Tabela 8 – Relação de mensagens enviadas ao tópico de confirmação de controle.

Mensagem	Significado
successfully_connected	sucesso na conexão do usuário com o dispositivo
already_connected	alerta sobre o usuário já estar conectado ao dispositivo
successfully_disconnected	sucesso na desconexão do usuário ao dispositivo
ok_to_unsubscribe	o dispositivo está preparado para cancelamento da subscrição
no_control_allowed	alerta sobre o usuário não ser o controlador do dispositivo
esp_init	aviso sobre o dispositivo estar iniciando sua operação
esp_interrupted	aviso sobre o dispositivo estar interrompendo sua operação

Fonte: autoria própria (2017).

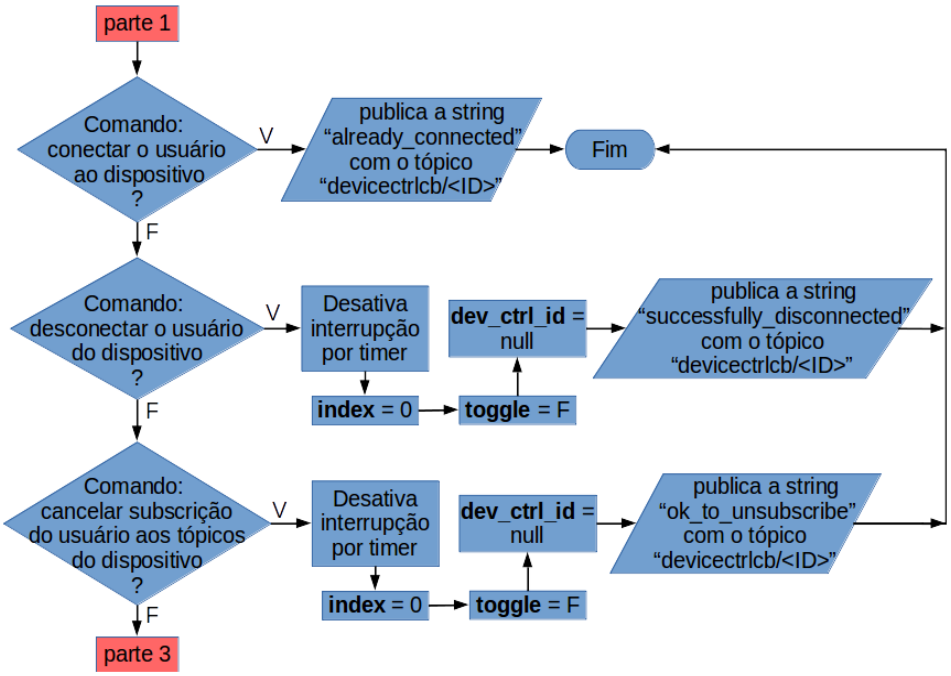
Todo o controle do sistema fica definido na função de callback das mensagens MQTT, figuras 16, 17 e 18, e se baseia no tipo de tópico e no par de chave e valor contido no corpo da mensagem recebida – em python a estrutura de chave e valor tem o nome de dicionário.

Figura 16 – Fluxograma da função de callback MQTT – parte 1.



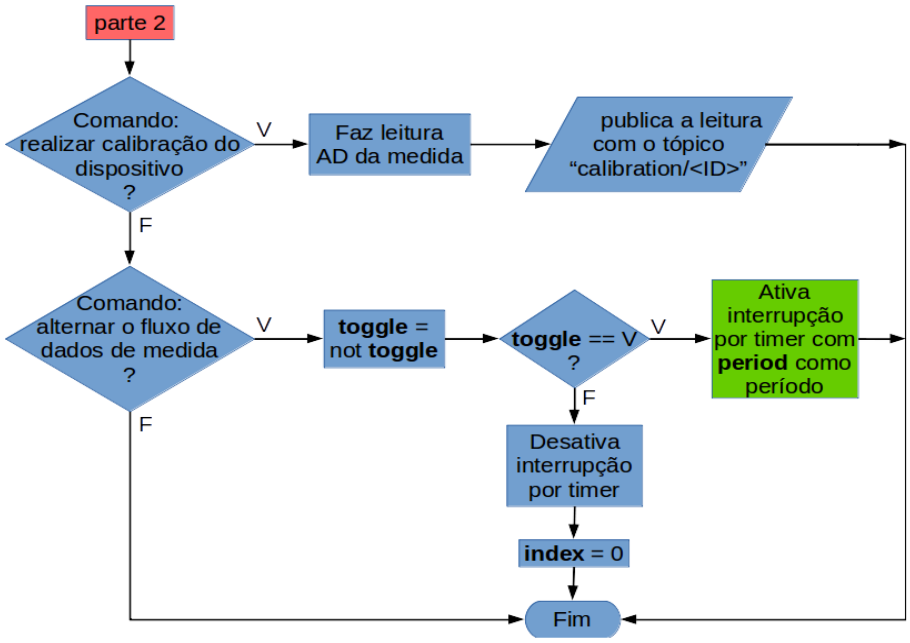
Fonte: autoria própria (2017).

Figura 17 – Fluxograma da função de callback MQTT – parte 2.



Fonte: autoria própria (2017).

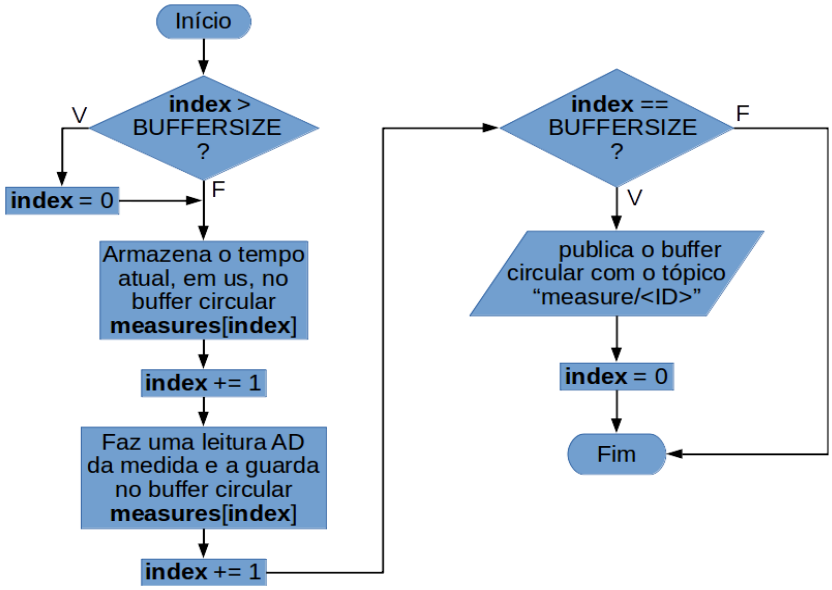
Figura 18 – Fluxograma da função de callback MQTT – parte 3.



Fonte: autoria própria (2017).

A rotina de callback de interrupção do temporizador, figura 19, é a responsável pela leitura das medidas e pelo envio destes dados à aplicação em lotes de 40 medidas. A cada chamada da interrupção, a função de callback armazena no buffer circular *measures* o valor do instante de tempo atual seguido pelo valor de leitura da medida. Quando o buffer armazena as 40 medidas AD ele as publica com o tópico measure/<ID>.

Figura 19 – Fluxograma da função de callback de interrupção do temporizador.



Fonte: autoria própria (2017).

2.6 CIRCUITO DE INSTRUMENTAÇÃO ELETRÔNICA

Este circuito é composto por seis blocos funcionais que representam:

- Alimentação com fonte de tensão simétrica.
- Célula de carga.
- Pré-amplificação de instrumentação.
- Filtro de ruídos.
- Subtrator para ajuste de offset.
- Amplificação de instrumentação.

Foram usados dois estágios de amplificação para que cada um deles possuísse baixo ganho a fim de minimizarem os efeitos não lineares da amplificação. Outra vantagem dessa abordagem é a possibilidade de serem introduzidos estágios intermediários entre esses estágios de amplificação para tratamento de ruídos e de offset. Assim o segundo estágio de amplificação deixa de amplificar a tensão de offset amplificada pelo primeiro estágio e pode aumentar o sinal da medida de modo linear e livre de ruídos.

2.6.1 Alimentação

O circuito eletrônico do dinamômetro foi pensado em razão do microcontrolador Esp8266 possuir como requisito tensão de alimentação contínua de 3,3V e tensão máxima de 1V para sua porta ADC. Como também são necessários no projeto amplificadores operacionais e de instrumentação, os circuitos integrados (CI) LM358 (operacional) e INA128 (de instrumentação) foram escolhidos por, além das suas características próprias de amplificadores, conseguirem amplificar sinais de até 1V operando com níveis de tensão de alimentação de ±3,3V.

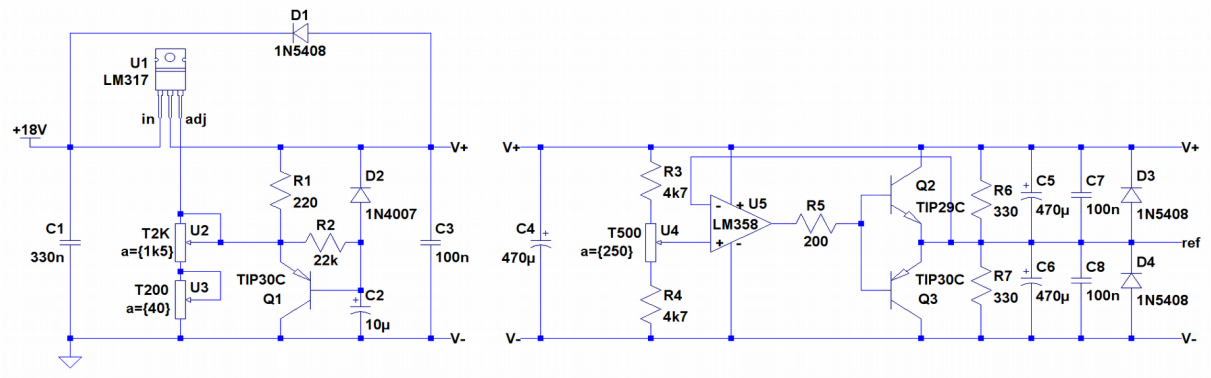
Por conta disso, o projeto pode reduzir sua **fonte de alimentação** de duas para apenas uma bateria de 12V, ou uma fonte de tomada de 12V, e configurar seu circuito de alimentação simétrica – **fonte simétrica** – para fornecer as tensões de ±3,3V.

Entretanto, embora uma fonte de alimentação de 12V baste para que sejam obtidas as tensões simétricas de alimentação em ±3,3V quando o microcontrolador Esp8266 é utilizado como módulo, por praticidade na prototipagem, o projeto foi desenvolvido com o kit de desenvolvimento do microcontrolador Esp8266, o que exigiu a elevação da tensão da fonte de

alimentação para 18V (duas fontes de tomada de 9V cada), pois o kit contém, na entrada da porta conversora de sinal analógico para digital, um circuito divisor de tensão que precisa ser compensado a fim de que o circuito de aquisição de sinal mantenha inalterado o projeto da tensão de saída entre 0V e 1V – essa compensação exige um nível de tensão da fonte simétrica de ±5V que não pode ser obtido quando a alimentação da fonte é de apenas 12V.

A modificação da alimentação de tensão de 12V para 18V e o circuito de compensação, contudo, são provisórios e utilizados para facilitar a prototipagem. Uma vez finalizada, o kit pode ser substituído pelo módulo Esp8266 e a alimentação retornar para 12V. As tensões da fonte simétrica em ±3,3V podem ser obtidas ajustando os trimpots U2, U3 e U4, ilustrados na figura 20 abaixo, desenhada com o auxílio do software LTSpice.

Figura 20 – Circuito eletrônico da fonte de alimentação.



Fonte: autoria própria, desenhado com software LTSpice (2017).

Os potenciômetros U2 e U3, no circuito da figura 20, determinam o nível de tensão de saída do regulador LM317 para 10V, com U3 introduzido para fazer um ajuste fino da tensão.

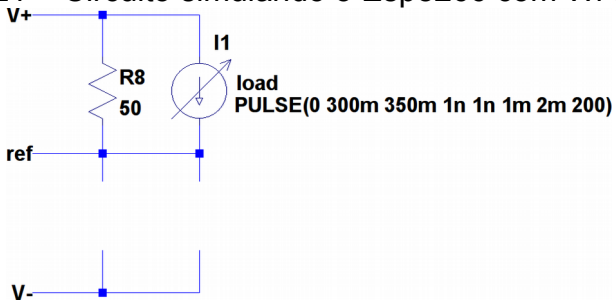
Os elementos Q1, R2 e C2 formam um subcircuito de proteção que reduzem o pico de corrente sobre o regulador LM317 quando o circuito da fonte simétrica é inicializado e ele é desativado depois de um tempo definido por R2 e C2, quando a tensão do capacitor C2 atinge o valor de tensão de saída do regulador.

Os capacitores C4, C5 e C6 servem para estabilizar a saída de tensão do circuito evitando que picos de corrente demandados pela carga alterem o valor de tensão fornecido. Os demais capacitores servem para filtrar sinais transientes ao prover um caminho de retorno deles à fonte de alimentação. Já os diodos adicionam uma proteção contra cargas que possam ser conectadas na saída do circuito da fonte simétrica.

Os resistores R3 e R4 definem o nível de tensão de referência de saída da fonte simétrica – o potenciômetro U4 é utilizado apenas para corrigir diferenças de valores entre esses dois resistores. Então o amplificador operacional (Amp Op) U5 compara a tensão de referência sobre U4 com a tensão sobre os emissores dos transistores Q2 e Q3 e realimenta essa diferença de tensão na base dos transistores. Como esse é um circuito em malha fechada, a diferença de tensão na saída do Amp Op produz uma tensão proporcional à tensão base-emissor (VBE) dos transistores, que é igual à tensão necessária para igualar as tensões de referência (VREF) e de U4. Assim VREF mantém simétricas as tensões V+ e V-, com o transistor Q2 atuando como fonte de corrente para cargas alimentadas por V- e o transistor Q3 como sorvedouro de corrente para cargas alimentadas por V+.

Nas especificações do módulo Esp8266 consta que ele tem 50Ω de impedância de carga, um consumo operacional de cerca de 80mA e picos de corrente de até 300mA quando está transmitindo dados via Wi-Fi. Com base nestas informações, foi feita uma simulação no LTSpice com uma carga resistiva de 50Ω em paralelo com uma carga ativa que consumia 200 pulsos de corrente de 300mA de pico, com período de 2ms e 50% de ciclo ativo, a fim de verificar como a fonte projetada responderia às demandas do microcontrolador, figura 21.

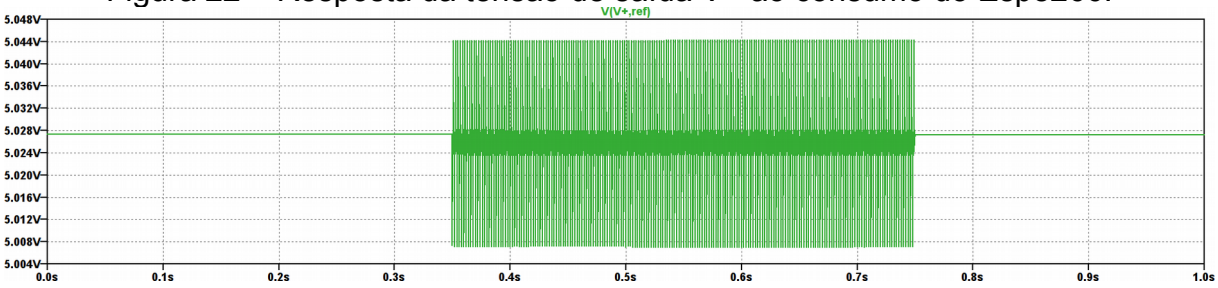
Figura 21 – Circuito simulando o Esp8266 com Wi-Fi ativo.



Fonte: autoria própria, desenhado com software LTSpice (2017).

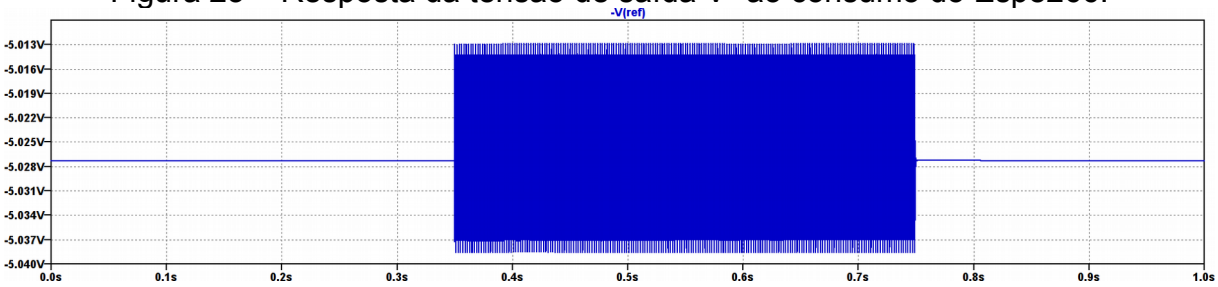
Como pode ser visto nas figuras 22, 23 e 24 a seguir, o módulo Esp8266 introduz um ruído de no máximo 20mV de pico (40mV pico a pico) às tensões de saída quando está com a conexão Wi-Fi ativa e não altera os níveis médios das tensões de saída.

Figura 22 – Resposta da tensão de saída V+ ao consumo do Esp8266.



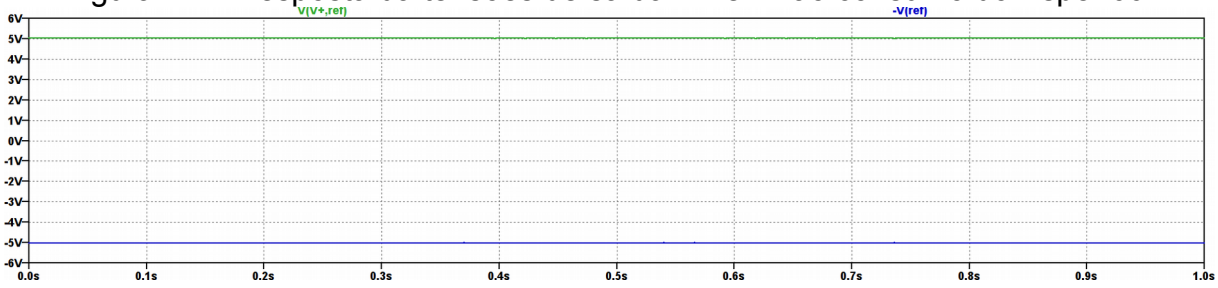
Fonte: autoria própria, desenhado com software LTSpice (2017).

Figura 23 – Resposta da tensão de saída V- ao consumo do Esp8266.



Fonte: autoria própria, desenhado com software LTSpice (2017).

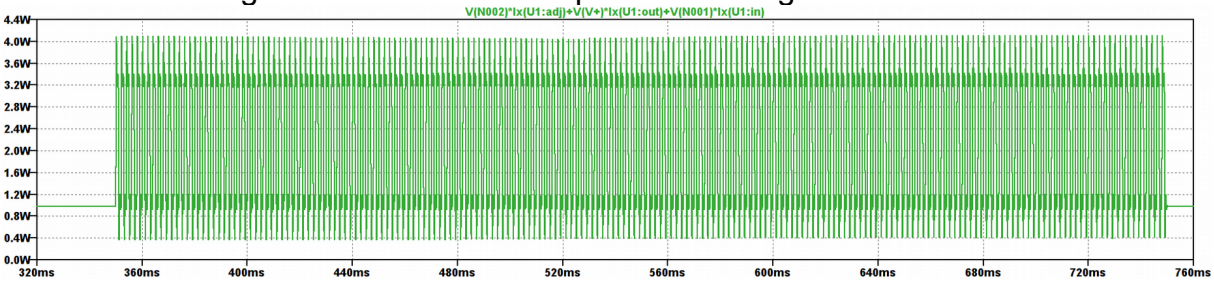
Figura 24 – Resposta da tensões de saída V+ e V- ao consumo do Esp8266.



Fonte: autoria própria, desenhado com software LTSpice (2017).

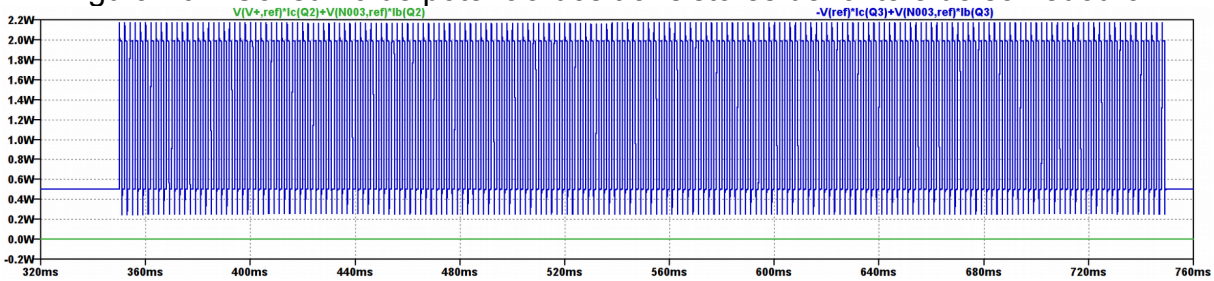
Nas figuras 25 e 26 são mostrados os consumos de potência do regulador LM317 e dos transistores Q2 e Q3 responsáveis por estabilizar a tensão VREF. Durante as comunicações Wi-Fi do Esp8266, o consumo médio de potência foi de 2W no regulador e de 1,2W nos transistores. Conforme seus datasheets, estes elementos conseguem dissipar tais quantidades de potência sem a necessidade de radiadores de calor em temperatura ambiente de 25 ºC.

Figura 25 – Consumo de potência do regulador LM317.



Fonte: autoria própria, desenhado com software LTSpice (2017).

Figura 26 – Consumo de potência dos transistores de fonte e de sorvedouro.



Verde: transistor Q2 (sorvedouro)

Azul: transistor Q3 (fonte)

Fonte: autoria própria, desenhado com software LTSpice (2017).

2.6.2 Célula de carga

Os dinamômetros usados nos projetos de (NUNES; BRUNO, 2012) e (HORITA, 2015) são do tipo digital e capazes de traçar curvas de força pelo tempo. Ambos usam o mesmo aparato sensor composto por uma célula de carga, quatro extensômetros e uma manopla encaixada a ela. Esse aparato também será usado neste projeto.

Segundo (NUNES; BRUNO, 2012), o aparato foi desenvolvido com o objetivo de ser leve, portátil e de uso fácil para ser operado em hospitais e clínicas.

Sua arquitetura apresenta duas bases rígidas e fixas que servem como apoio para a mão e fixação de uma terceira peça de pouca espessura e deformável o suficiente para servir como corpo de prova para a força de preensão palmar a ser medida.

A deformação dessa peça precisa ocorrer dentro do regime elástico-linear de seu material constituinte sob a pena de ele ficar deformado permanentemente. Nesse regime o material se deforma linearmente com a tensão aplicada a ele. Dessa forma, o aço 1020 foi escolhido e, à partir de suas especificações, foi dimensionado para que suporte deformações por tensões de até 200N, equivalentes a 20kgf aproximadamente. Esse valor de tensão foi

determinado considerando os dados tabelados em Nunes[7] (2012, p. 16 apud MATHIOWETZ, 1985, p. 69-72)[9], mas a tabela em questão diz respeito à valores de força de pinça entre os dedos polegar e indicador e não de FPP. Como, porém, a célula foi construída para medir a força de pacientes debilitados fisicamente e considerando que 20kgf abrange cerca de metade da FPP de um adulto saudável, como visto na introdução deste trabalho, o intervalo de tensão de 0kgf a 20kgf é razoável.

Adicionalmente a estes requisitos, a peça de prova ainda precisa ter dimensões compatíveis com a anatomia da mão humana.

As peças da célula de carga foram fabricadas com essas especificações e estão ilustradas nas duas figuras a seguir. A peça de prova da célula de carga é a peça presente na parte inferior da figura 27 e suas dimensões são de 100mm de comprimento por 20mm de largura e 2,5mm de espessura. As demais peças da figura formam a base para fixação da peça de prova da célula de carga, como pode ser visto na figura 28.

Figura 27 – Peças da célula de carga.

Fonte: Nunes e Bruno (2012)[7].

Figura 28 – Célula de carga montada.

Fonte: Nunes e Bruno (2012)[7].

Para converter a deformação mecânica da peça em valores de tensão elétrica, os extensômetros foram colados à peça de prova da célula de carga a uma distância de 10mm das

extremidades efetivas da peça de prova ao longo de seu comprimento, com essas extremidades efetivas sendo consideradas a partir da distância na qual o furo da peça esteja equidistante delas e das extremidades físicas da peça ao longo de seu comprimento.

Os elementos transdutores usados têm a capacidade de alterar seus valores de resistência elétrica em função de suas deformações mecânicas de modo que produzam uma diferença de tensão elétrica entre seus estados relaxado e tracionado ou relaxado e comprimido, que é linearmente proporcional à taxa de deformação. Adicionalmente a essas características, o projeto teve como requisitos[7] a escolha de extensômetros com capacidade de medir pequenas deformações, que possuíssem dimensões reduzidas e compatíveis com as dimensões da célula de carga fabricada e que apresentassem constante de calibração estável, que fossem flexíveis e de baixo custo. Por isso foram escolhidos os extensômetros **PA-06-125AA-350-EN** do fabricante Excel Sensores.[7]

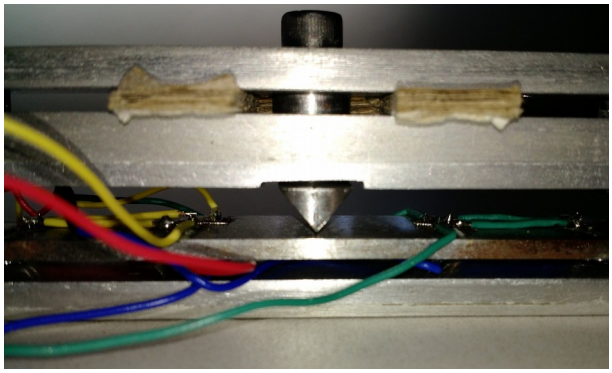
Esses extensômetros têm resistência elétrica nominal de 350Ω e foram posicionados no corpo de prova da célula de carga "onde a deflexão na peça atinja um valor mensurável levando em conta a montagem e segurança do equipamento" (NUNES; BRUNO, 2012, p. 24). Nestas condições, variações de tensões mecânicas de até 200N aplicadas à célula de carga produzem variações das resistências dos extensômetros de até 2,88Ω.[7]

Estão ilustradas nas figuras 29-32 a seguir os estágios de montagem da célula de carga com a colagem dos extensômetros no corpo de prova, a fiação soldada nestes transdutores, a célula finalizada e já com a manopla adicionada para que a célula de carga tenha um encaixe adequado nas mãos das pessoas.

Figura 29 – Célula de carga com extensômetros colados.

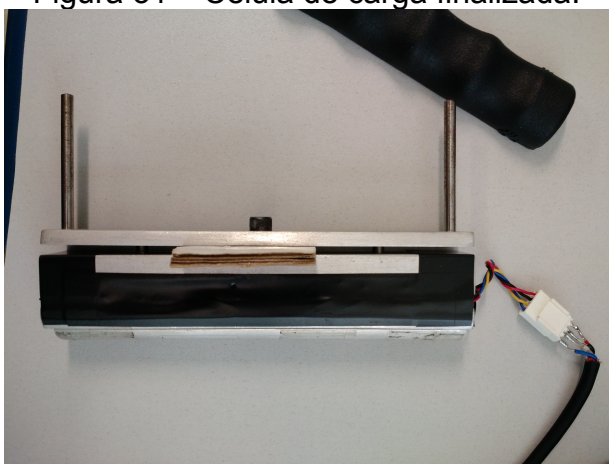
Fonte: Nunes e Bruno (2012)[7].

Figura 30 – Célula de carga com fiação soldada.



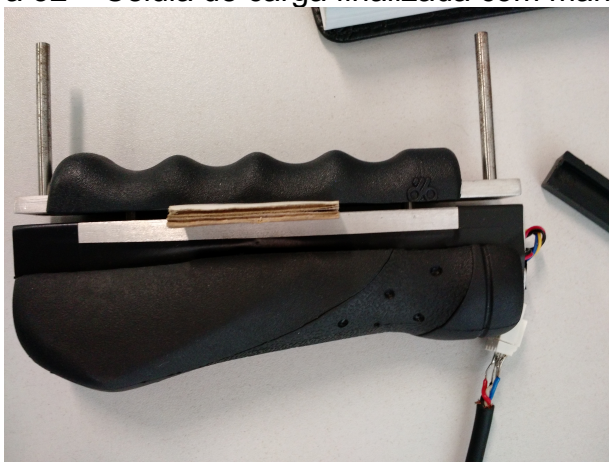
Fonte: autoria própria (2017).

Figura 31 – Célula de carga finalizada.



Fonte: autoria própria (2017).

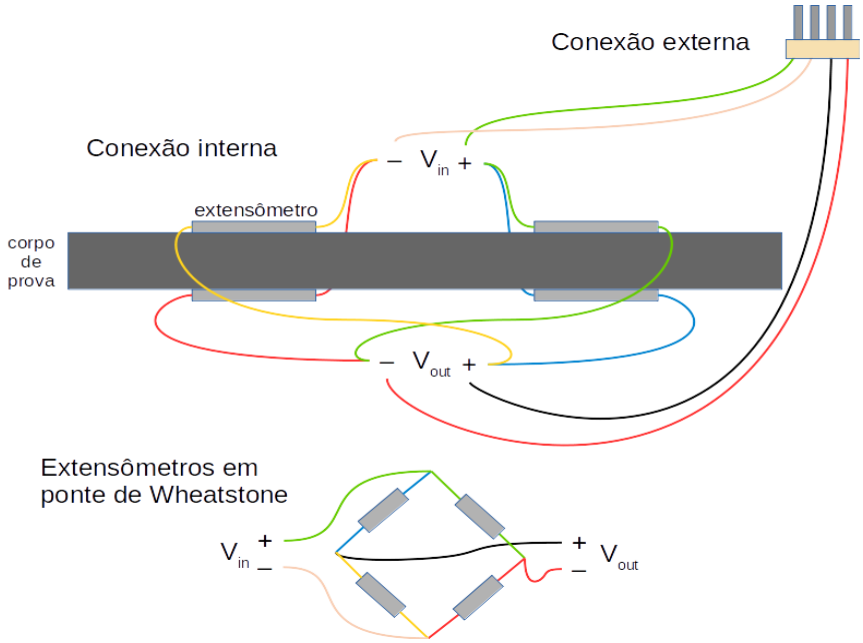
Figura 32 – Célula de carga finalizada com manopla.



Fonte: autoria própria (2017).

A leitura da FPP sobre a célula de carga é feita configurando os extensômetros da célula em ponte de Wheatstone completa, na qual todos os quatro elementos da ponte são variáveis, com dois terminais usados para alimentação da ponte e dois para a leitura da ponte, como pode ser visto na figura 33.

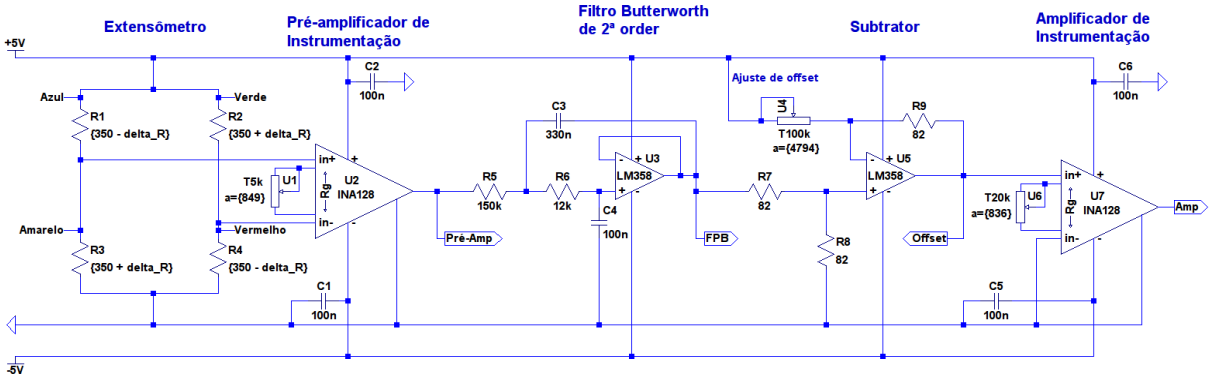
Figura 33 – Conexões dos extensômetros em ponte de Wheatstone completa.



Fonte: autoria própria (2017).

No circuito de aquisição de sinal projetado com o LTSpice, figura 34, a ponte é desenhada com cada resistor representando um extensômetro e com as indicações de cores correspondendo às cores dos fios soldados nos respectivos extensômetros, como ilustradas nas figuras 30 e 33, parte inferior.

Figura 34 – Circuito eletrônico de aquisição.



Fonte: autoria própria, desenhado com software LTSpice (2017).

Externamente, porém, os códigos de cores da fiação são distintos e representam os terminais da ponte de Wheatstone. Assim, devido a uma inversão de sentido ocorrida no momento da soldagem dos conectores, os fios verde e branco – representado pela cor rosa pêssego na figura 33 – correspondem à V+ e VREF e os fios preto e vermelho correspondem às polaridades positiva e negativa da tensão de saída da ponte, conforme pode ser observado pela ilustração inferior da figura 33.

2.6.3 Filtro de ruído

O filtro de ruído aplicado foi um passa baixas ativo do tipo Butterworth de 2ª ordem, como nos projetos anteriores, empregando o Amp Op LM358 e configurando-o para apresentar banda de passagem de 10Hz. Foi decidido posicioná-lo logo após o pré-amplificador de instrumentação para eliminar possíveis ruídos aumentados pelo pré-amplificador e assim evitar que exerçam influência no ajuste de offset no próximo estágio.

2.6.4 Calibração de offset

Por haver pequenas variações de valores de resistência elétrica entre os extensômetros, devido aos processos de fabricação e de montagem de cada um deles sobre a célula de carga não ser exatamente igual – durante a colagem um extensômetro pode ter sido esticado e outro não – as diferenças entre eles acabam sendo transmitidas para a ponte de Wheatstone como um sinal de tensão mecânica, quando de fato é um sinal de offset que é amplificado pelo estágio de pré-amplificação. Isso precisa ser corrigido para que haja uma correspondência entre valores de 0N de força aplicada e de 0V medido. Para essa tarefa, o Amp Op LM358 agora foi usado em uma configuração subtratora com ganho unitário e regulagem de offset por meio de um potenciômetro.

2.6.5 Amplificadores de instrumentação

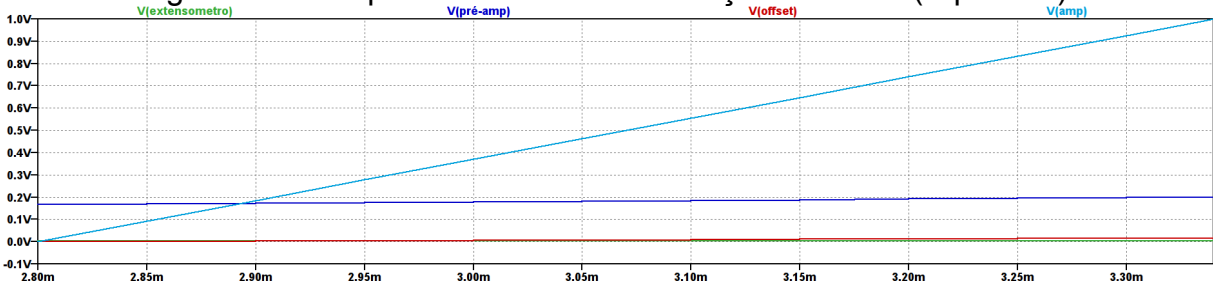
Como dito anteriormente, usar dois amplificadores de instrumentação tem o objetivo de dividir o ganho de amplificação entre eles de forma que os ganhos individuais sejam baixos e assim realizem a amplificação do sinal medido de forma linear.

Com o auxílio de um dinamômetro de calibração, foi constatado que variações de 0N a 200N produziram valores de tensão de 2,80mV a 3,34mV. Assim, os Amp Op usados nos estágios de pré-amplificação e de amplificação foram INA128 com o primeiro estágio tendo o trimpot U1 ajustado para uma resistência de ganho de 849Ω, correspondendo a um ganho de 59,88V/V, e o segundo estágio tendo o trimpot U6 ajustado para uma resistência de 836Ω, configurando um ganho de 60,82V/V. Tais valores foram assim definidos para compensar o estágio de controle de offset, que remove quase 200mV do sinal vindo do estágio de pré-amplificação, como pode ser visto nas figuras 35 e 36.

Com isso as variações de tensões mecânicas entre 0N e 200N são transformadas em variações elétricas entre 0V e 1V, que então serão enviadas para o sistema microcontrolado realizar a conversão AD.

As figuras 35 e 36 a seguir ilustram a resposta de tensão elétrica do circuito de aquisição da figura 34 em função da tensão mecânica dada através da variação da resistência ôhmica dos extensômetros. Como pode ser visto, o circuito possui uma linearidade satisfatória.

Figura 35 – Resposta do circuito às variações de FPP (V por mV).



Verde: tensão de saída da célula de carga

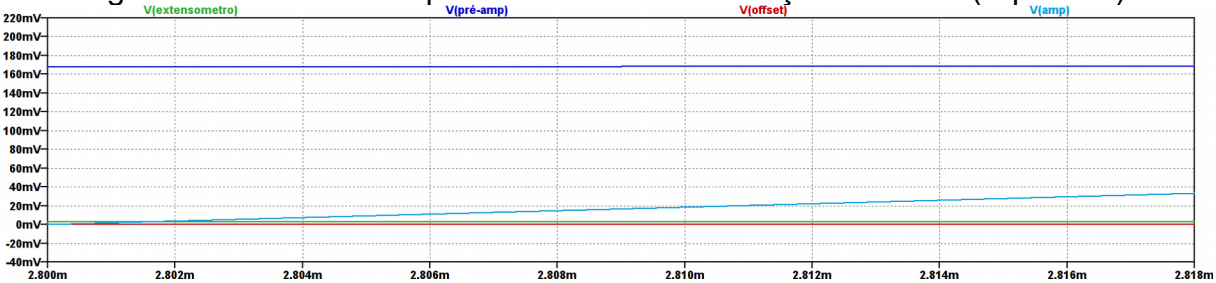
Azul escuro: tensão de saída do estágio de pré-amplificação

Vermelho: tensão de saída do estágio de correção de offset

Azul claro: tensão de saída final

Fonte: autoria própria, desenhado com software LTSpice (2017).

Figura 36 – Início da resposta do circuito às variações de FPP (V por mV).

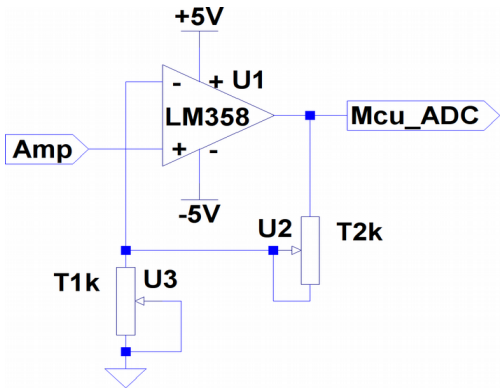


Fonte: autoria própria, desenhado com software LTSpice (2017).

2.6.6 Circuitos auxiliares

Como mencionado anteriormente, o circuito divisor de tensão na porta AD do kit de desenvolvimento NodeMcu Esp8266 criou a necessidade de se introduzir um circuito para compensação de ganho do sinal de saída do circuito de aquisição. Ele é um circuito de ganho 3, figura 37, e sua entrada não inversora *Amp* deve ser conectada à saída *Amp* do circuito de aquisição de medida, figura 34.

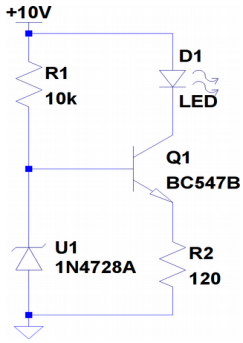
Figura 37 – Circuito de compensação de ganho para o kit NodeMcu.



Fonte: autoria própria, desenhado com software LTSpice (2017).

É prática comum usar LEDs para saber se o circuito está ligado ou não. Como o circuito da fonte simétrica permite que as tensões de entrada e de saída sejam ajustáveis, o circuito driver de LED, figura 38, foi conectado à saída do estágio de regulação do circuito da fonte simétrica, figura 20, entre os capacitores C3 e C4.

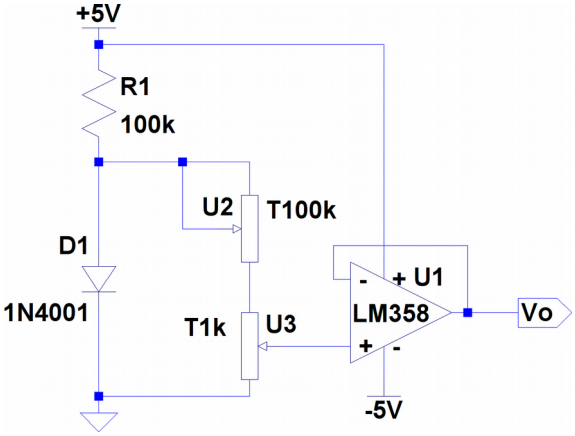
Figura 38 – Circuito driver para o LED indicador de funcionamento.



Fonte: autoria própria, desenhado com software LTSpice (2017).

O circuito da figura 39 gera sinais de tensão em milivolts em corrente contínua e foi construído para substituir a célula de carga nos testes do circuito de aquisição representado pela figura 34.

Figura 39 – Circuito da fonte de tensão em milivolts.

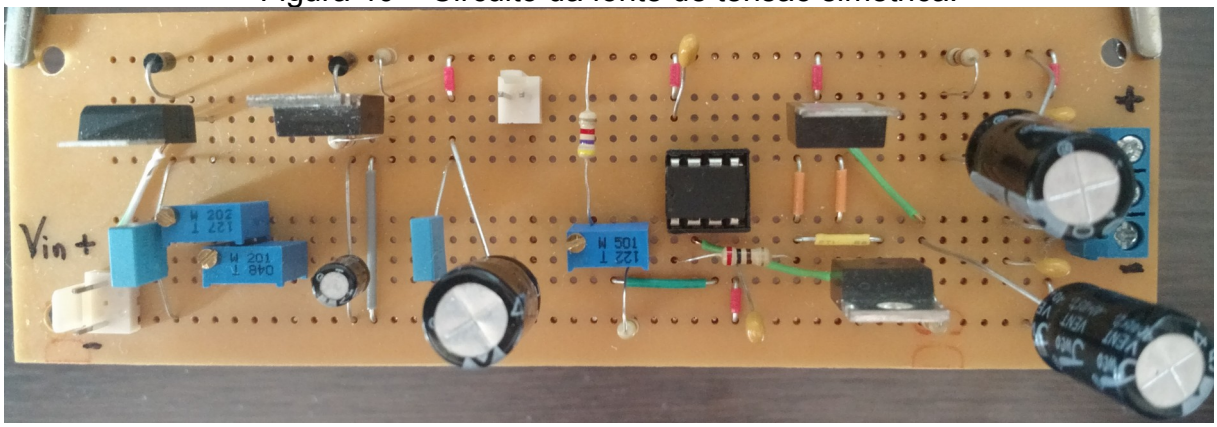


Fonte: autoria própria, desenhado com software LTSpice (2017).

2.6.7 Montagem

A fonte de tensão simétrica projetada pelo circuito da figura 20 foi montada em uma placa padrão PP-01 de 145mm por 45mm, figura 40. Foram utilizados conectores molex tanto para a entrada de alimentação da fonte quanto para a saída de alimentação do circuito driver do LED indicador de funcionamento do instrumento. Quanto à saída de alimentação para o circuito de aquisição de sinal da célula de carga, foi utilizado um conector multipolar com parafusos.

Figura 40 – Circuito da fonte de tensão simétrica.

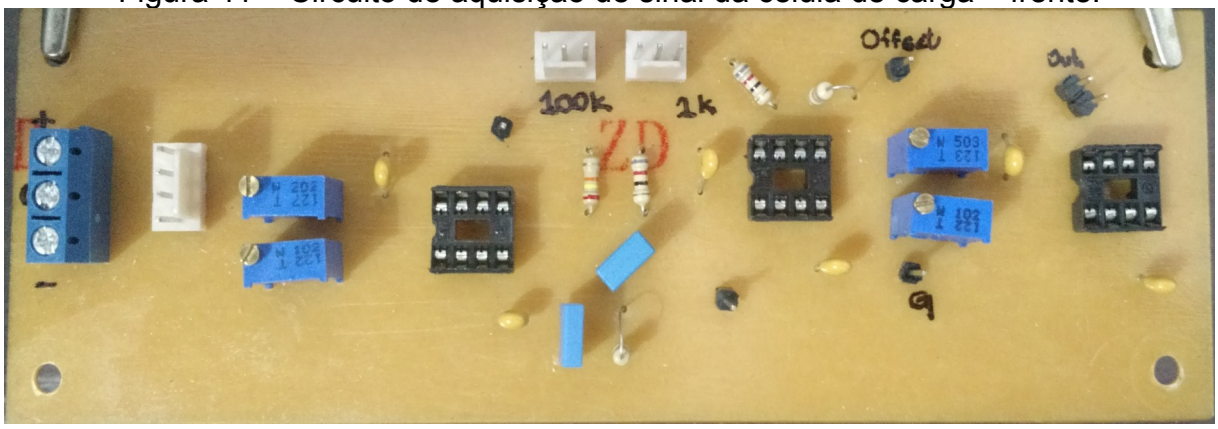


Fonte: autoria própria (2017).

O circuito de aquisição de sinal da célula de carga, figura 34, foi montado sobre uma placa de circuito impresso, figura 41, confeccionada especificamente para ele, uma vez que isso permitia organizar melhor os componentes em uma área de 150mm por 5mm, de dimensão semelhante à da fonte de tensão simétrica, além de possibilitar reservar uma extensa área de cobre da placa, figura 42, para criar uma malha de tensão de referência consistente a fim de minimizar ruídos durante a amplificação de sinal.

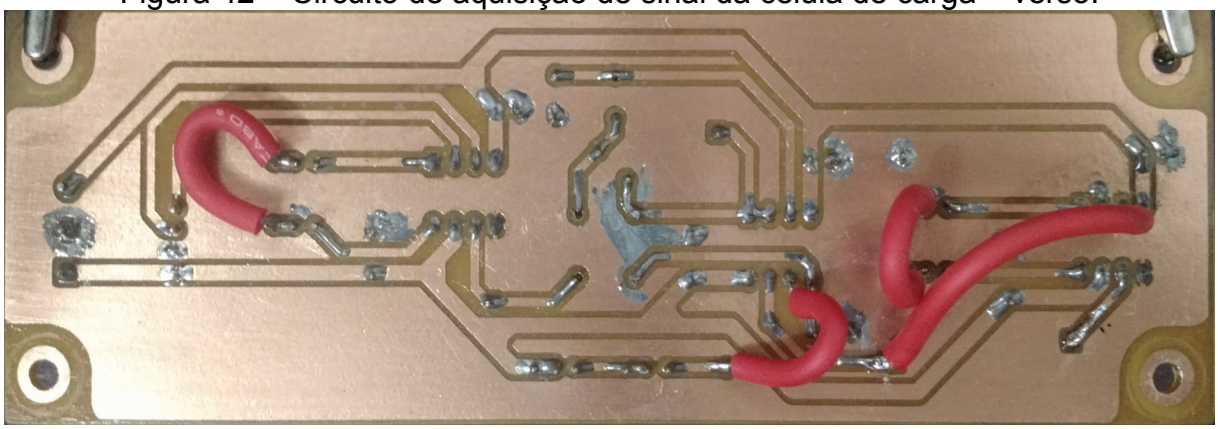
O conector multipolar também foi utilizado para ligar a alimentação do circuito de aquisição, figura 41. Já as conexões com a célula de carga e com os trimpots de ajuste de offset utilizaram os conectores molex porque os posicionamentos desses conectores definem as polaridades com que devem ser conectados os respectivos cabos de conexão. Além deles, também foram introduzidos conectores dupont macho nas saídas dos estágios do circuito para facilitar suas respectivas leituras de tensão durante o processo de prototipagem.

Figura 41 – Circuito de aquisição de sinal da célula de carga – frente.



Fonte: autoria própria (2017).

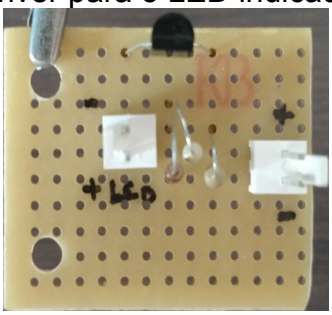
Figura 42 – Circuito de aquisição de sinal da célula de carga – verso.



Fonte: autoria própria (2017).

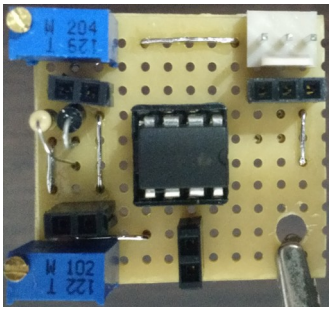
Os circuitos auxiliares de driver de LED, figura 38, e da fonte de tensão contínua em milivolts, figura 39, foram construídos em placas padrão – figuras 43 e 44, respectivamente – e montados com foco em minimizar suas áreas.

Figura 43 – Circuito driver para o LED indicador de funcionamento.



Fonte: autoria própria (2017).

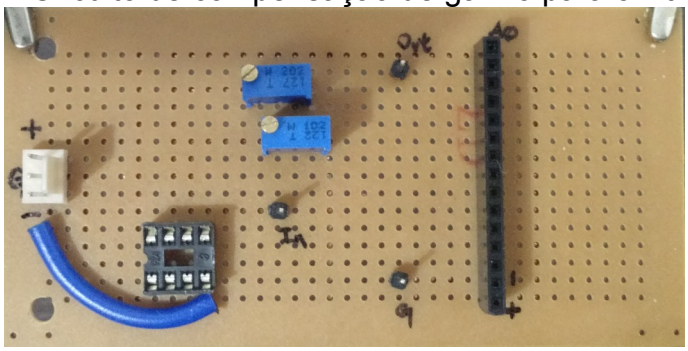
Figura 44 – Circuito da fonte de tensão em milivolts.



Fonte: autoria própria (2017).

O circuito de compensação de ganho do kit NodeMcu Esp8266, figura 37, também foi confeccionado em placa padrão, figura 45. O kit Esp8266 deve ser conectado ao barramento de pinos do circuito de compensação com seus pinos coincidindo com os pinos marcados na placa do circuito – *A0* é o pino da porta ADC.

Figura 45 – Circuito de compensação de ganho para o kit NodeMcu.



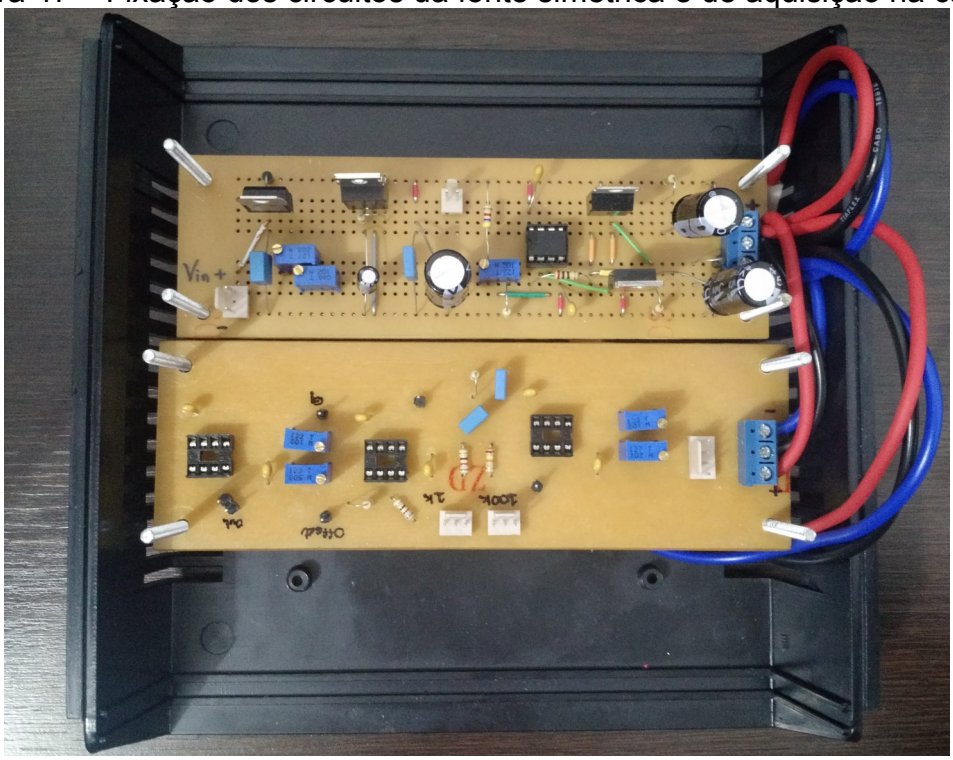
Fonte: autoria própria (2017).

Os circuitos do instrumento foram embutidos em uma caixa patola PB-209, figura 46, e fixados por parafusos com 50mm de comprimento, figuras 47 e 48.

Figura 46 – Instrumento montado.

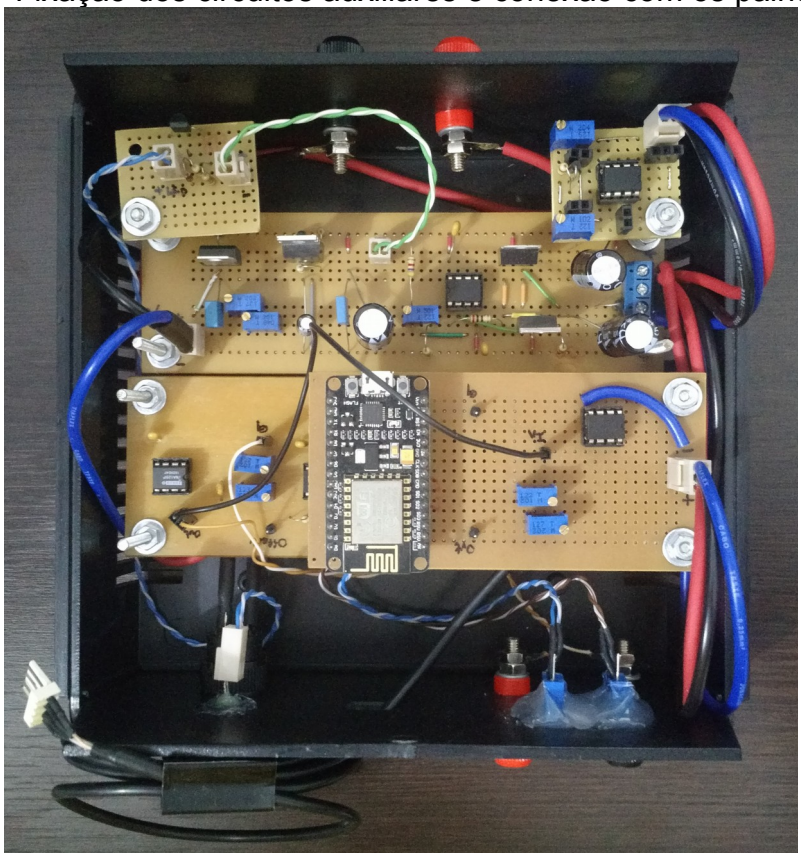
Fonte: autoria própria (2017).

Figura 47 – Fixação dos circuitos da fonte simétrica e de aquisição na caixa.



Fonte: autoria própria (2017).

Figura 48 – Fixação dos circuitos auxiliares e conexão com os painéis laterais.



Fonte: autoria própria (2017).

No painel frontal da caixa, figura 49, foram colocados o LED indicador de funcionamento do instrumento, um botão para ligar e desligar, os trimpots para o ajuste de offset e terminais banana para a leitura da saída do circuito de aquisição de sinal a fim de se fazer o ajuste de offset.

Figura 49 – Painel frontal da caixa – lado externo.

Fonte: autoria própria (2017).

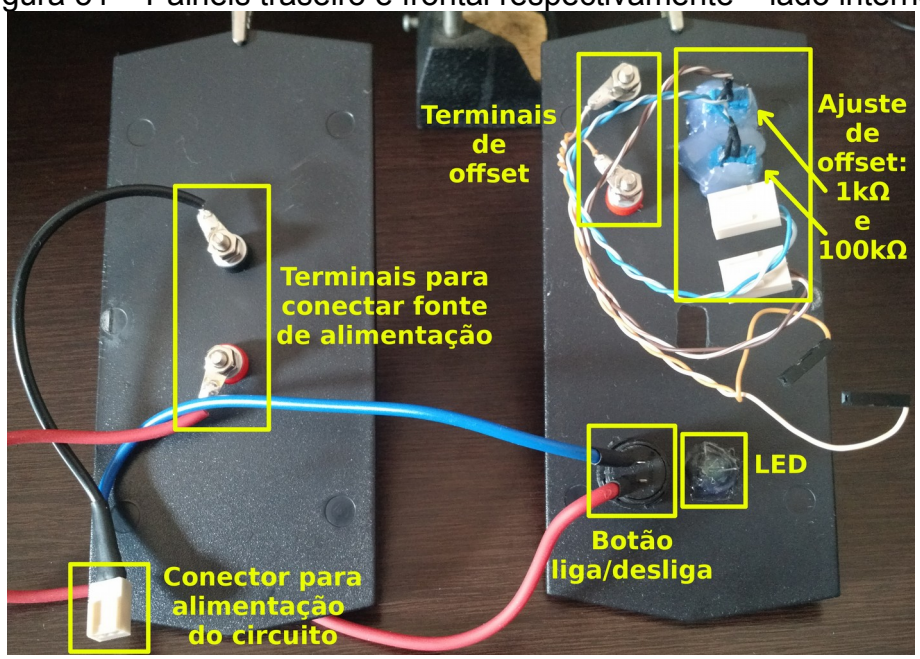
No painel traseiro, figura 50, foram colocados apenas os terminais banana para a alimentação do instrumento.

Figura 50 – Painel traseiro da caixa – lado externo.

Fonte: autoria própria (2017).

A figura 51 a seguir ilustra as partes internas dos dois painéis laterais com os seus componentes fixados.

Figura 51 – Painéis traseiro e frontal respectivamente – lado interno.



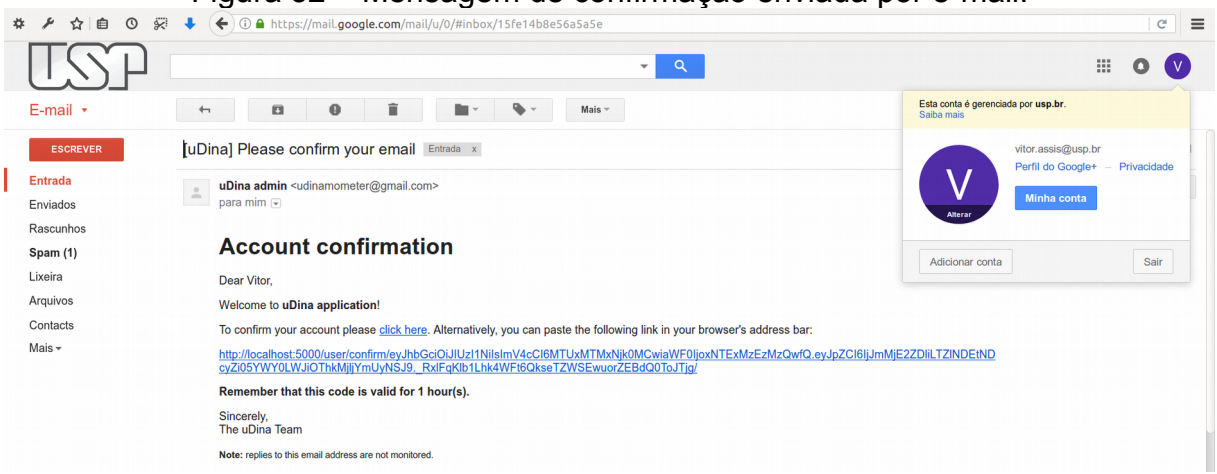
Fonte: autoria própria (2017).

3 RESULTADOS

3.1 APLICAÇÃO WEB

A aplicação contém um controle de usuários básico, com sistema de login, de reset de senha, de registro de novos usuários, de envio de e-mail de confirmação e de alteração de e-mail e de senha de usuário. Quando um novo usuário é registrado, o reset de senha de um usuário cadastrado é solicitado ou o usuário troca seu endereço de e-mail, uma mensagem de confirmação, figura 52, é enviada ao endereço de e-mail deste usuário com um link que contém um token de validação. Esse token é verificado pela aplicação apenas quando o link enviado ao e-mail é ativado e somente após sua validação é que o registro de novo usuário ou as modificações de senha ou de e-mail são efetivadas na aplicação.

Figura 52 – Mensagem de confirmação enviada por e-mail.



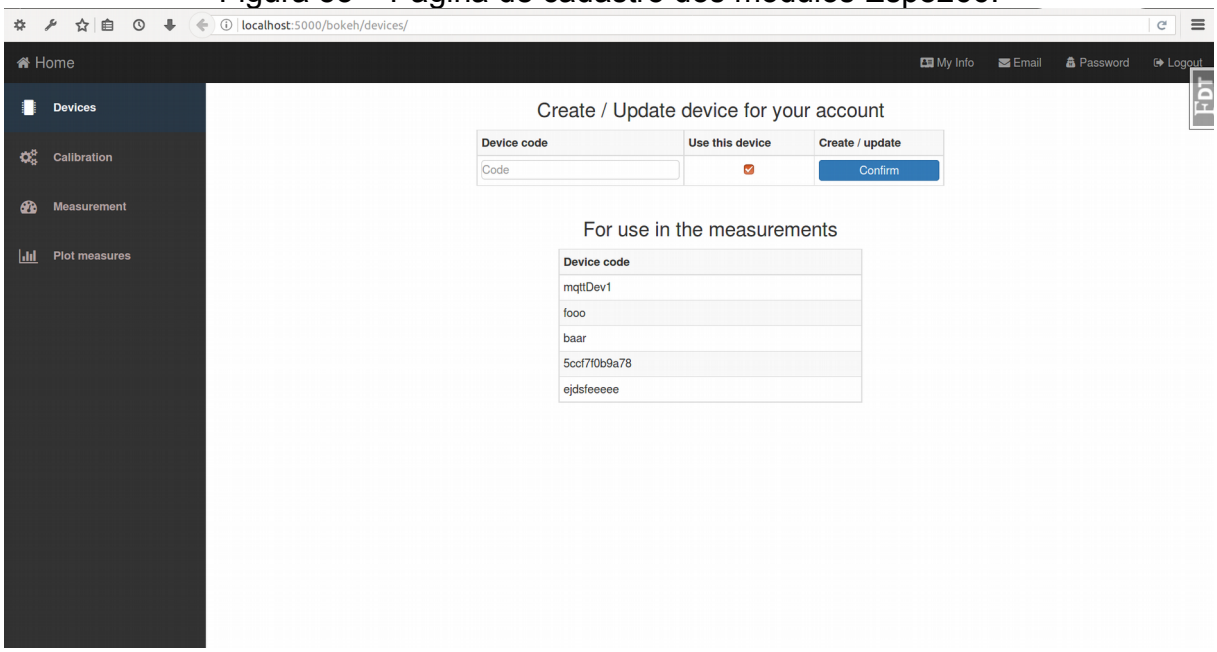
Fonte: autoria própria (2017).

Uma vez cadastrado e autenticado, o usuário da aplicação web é redirecionado para a tela *Devices*, figura 53, que faz o controle dos dinamômetros por meio dos seus dispositivos Esp8266. Os instrumentos são identificados por seus microcontroladores, que são cadastrados no sistema usando como códigos de identificação seus endereços MAC. Isso porque cada endereço MAC é único e identifica o hardware da interface de rede de um dispositivo.

O cadastro é feito com ou sem a caixa de seleção *use this device* marcada. Essa seleção indica se o dispositivo deve ser posto como opção de dispositivos de medida nas telas de calibração e de medição. A quantidade de dispositivos é limitada a 5 e sempre serão

mostrados nas telas de calibração e medição os últimos 5 dispositivos adicionados pelo usuário que foram marcados com a caixa de seleção. Quando marcados, eles são mostrados na tabela de dispositivos *For use in the measurements*.

Figura 53 – Página de cadastro dos módulos Esp8266.

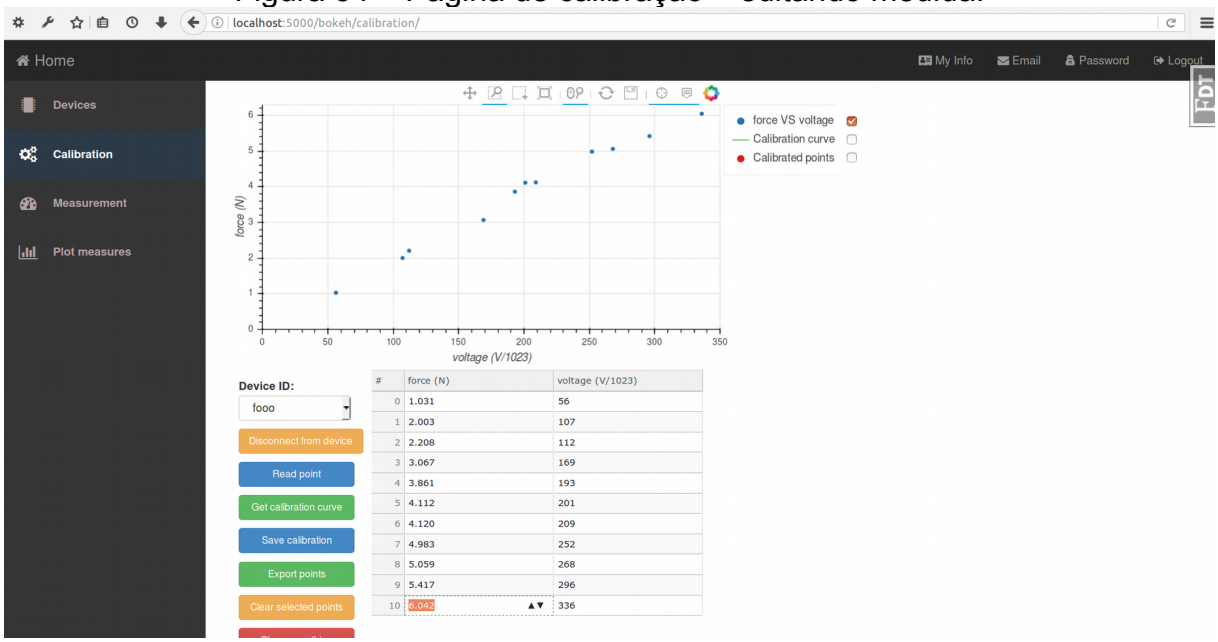


Fonte: autoria própria (2017).

O próximo passo é navegar até a tela de calibração *Calibration*, figura 54. Após selecionar o dispositivo em *Device ID*, deve-se conectar a aplicação ao dispositivo clicando no botão *Connect to device*. Quando a conexão for estabelecida a cor do botão ficará amarela e seu nome será alterado para *Disconnect from device*. A partir de então, deve-se realizar leituras de medidas ponto a ponto por meio do botão *Read point*. A cada clique no botão uma nova linha será preenchida na tabela contendo o valor de tensão digitalizada correspondente à tensão da FPP amostrada pelo conversor AD do instrumento de medida. O valor da força, em newtons, sempre será nulo e deve ser alterado manualmente com o valor indicado pelo dinamômetro de calibração. Deve-se usar um ponto como separador decimal.

Os dados das medidas presentes nas figuras dos resultados da aplicação web, figuras 53-60, não são dados reais de medida. Eles foram gerados por uma função de teste que produz valores de medidas aleatórias para simular a leitura da célula de carga.

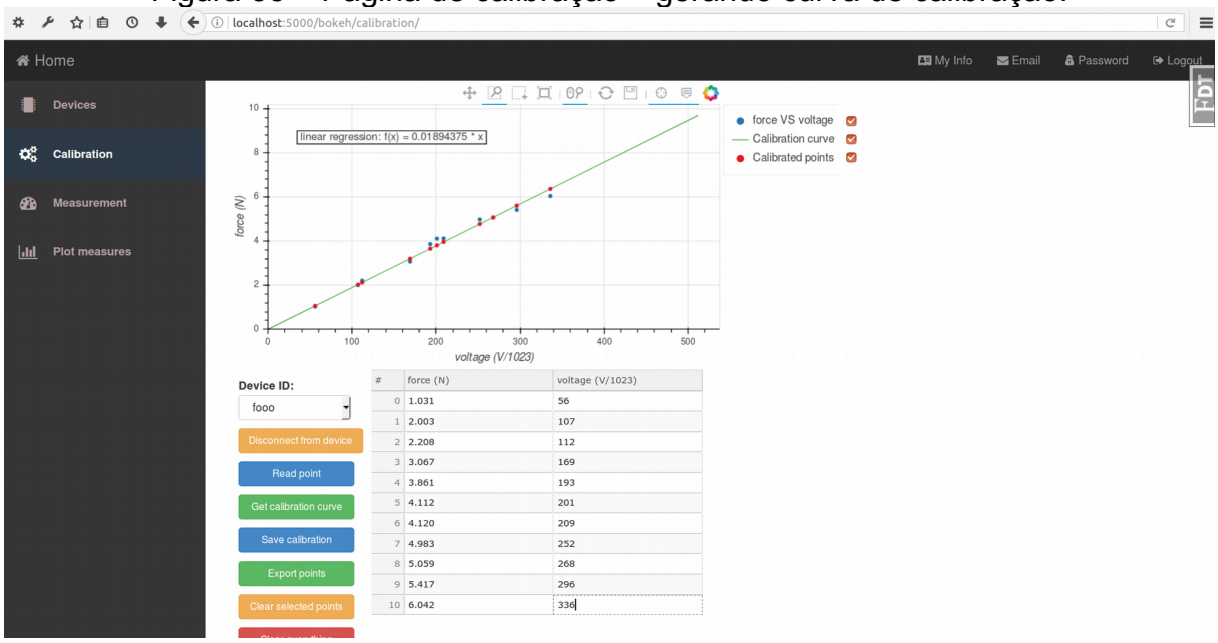
Figura 54 – Página de calibração – editando medida.



Fonte: autoria própria (2017).

Quando a quantidade de pontos amostrados for considerada suficiente, a curva de calibração deve ser gerada por meio do botão *Get callibration curve*, figura 55. O clique no botão vai disparar um processo que calcula a regressão linear dos pontos e plota sua curva no gráfico juntamente com os respectivos pontos calibrados.

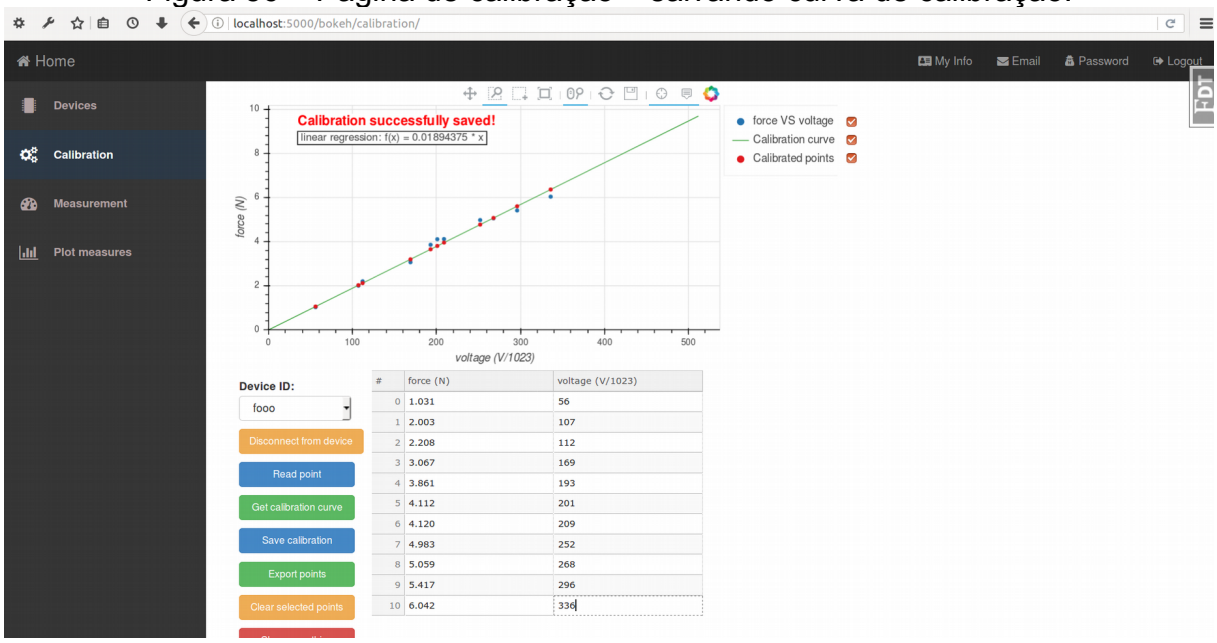
Figura 55 – Página de calibração – gerando curva de calibração.



Fonte: autoria própria (2017).

Caso esteja satisfatória, a calibração deve ser salva no sistema por meio do botão *Save calibration* para que ela possa ser usada pelo processo de leitura das medidas na tela *Measurement*. Ao final deste processo, uma mensagem de confirmação de salvamento é impressa no gráfico de calibração, figura 56.

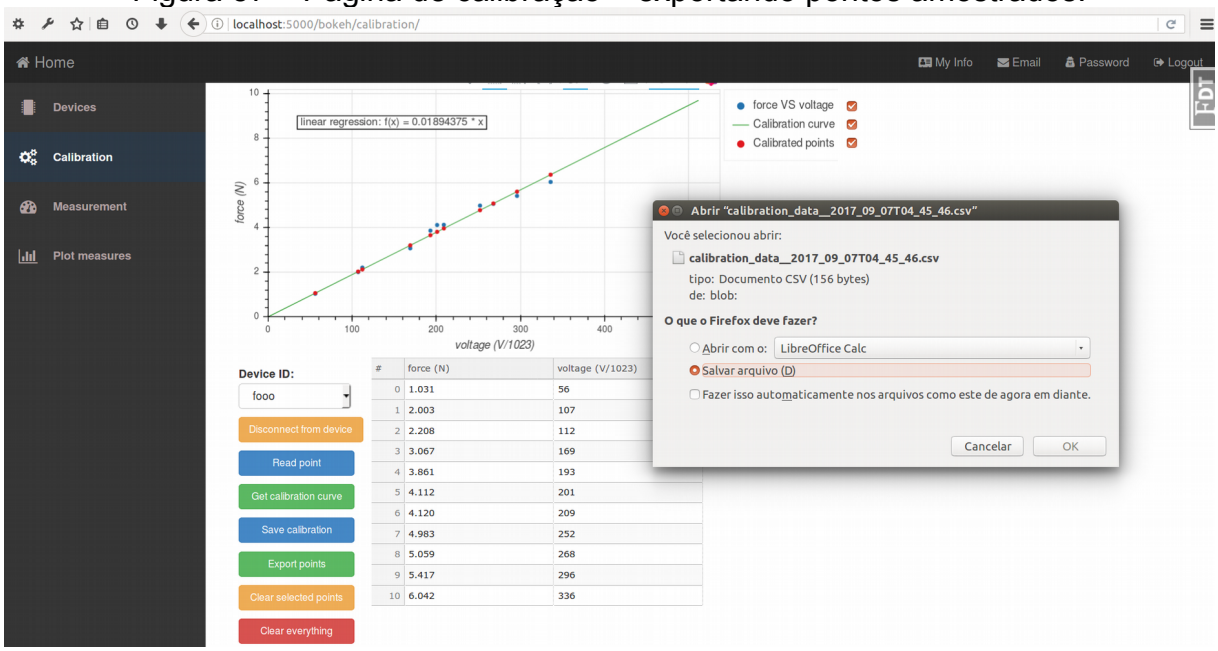
Figura 56 – Página de calibração – salvando curva de calibração.



Fonte: autoria própria (2017).

O sistema salva no banco de dados apenas a equação de calibração. Os pontos amostrados são descartados quando o usuário sair da tela, trocar de dispositivo ou quando eles forem apagados total ou parcialmente pelos botões *Clear everything* e *Clear selected points*, respectivamente. Portanto, caso seja necessário analisar os pontos amostrados para a calibração, os mesmos devem ser exportados pelo botão *Export points*, que vai gerar uma planilha no formato CSV com a tabela de pontos da tela, figura 57. Por ora, não foi implementado um recurso para importar para o sistema de calibração os pontos amostrados provenientes de uma planilha CSV.

Figura 57 – Página de calibração – exportando pontos amostrados.



Fonte: autoria própria (2017).

Com a calibração do instrumento salva, ele já pode ser usado para realizar a leitura de medidas por meio da tela *Measurement*, figura 58, com a conexão ao dispositivo feita de maneira idêntica à da tela de calibração.

A leitura do fluxo de dados no gráfico *Streaming data*, gráfico superior da figura 58, é iniciada e pausada pelo botão *Start streaming* que altera seu nome para *Stop streaming* quando o fluxo de dados estiver ativo. A janela de fluxo de dados armazena 6000 pontos, que equivalem a 30 segundos de leitura quando sua frequência for de 200Hz. Isso permite que a leitura seja feita continuamente sem a preocupação de o navegador de internet consumir eventualmente toda a memória do computador. Opcionalmente, o botão *Reset stream* pode ser usado para apagar os pontos do fluxo de dados armazenados pelo navegador.

A frequência de leitura das medidas é determinada pela escolha do respectivo período de amostragem, que pode ser alterado movendo horizontalmente a barra deslizante *Period* e cujo valor correspondente em frequência é atualizado no título do gráfico *Streaming data*. Os valores dos períodos estão em milissegundos e variam entre 5ms e 50ms com resolução de 1ms, representando uma variação de frequência entre 20Hz e 200Hz. Por padrão a frequência de inicialização foi definida em 200Hz.

Para selecionar a partir do fluxo de dados apenas os pontos relevantes à análise – a critério do usuário – deve-se, no gráfico de fluxo de dados, clicar e arrastar o mouse a fim de formar uma janela de seleção. Quando o mouse for solto, os dados selecionados serão

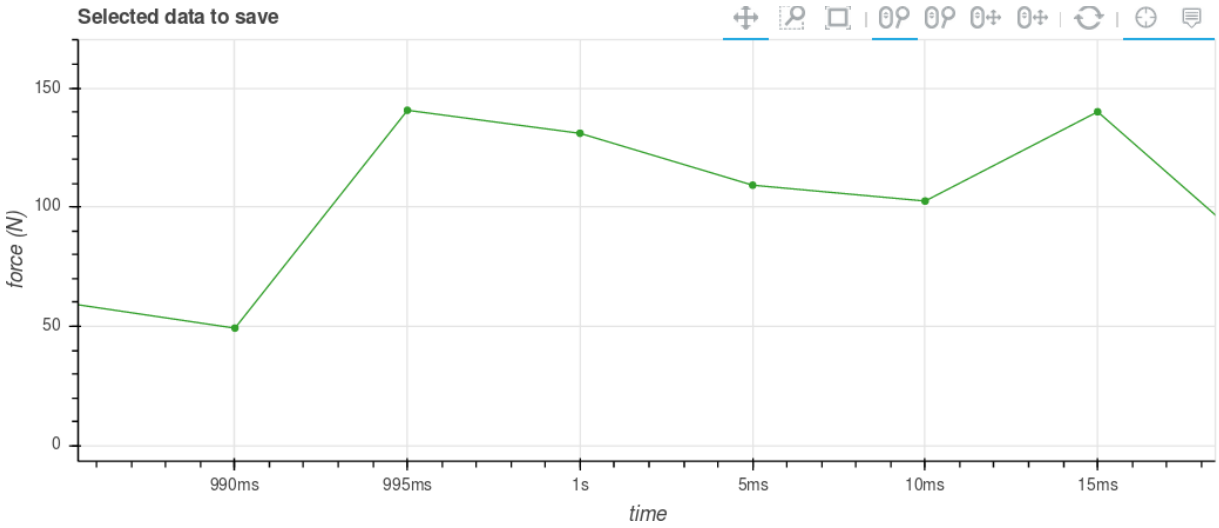
desenhados no gráfico de seleção *Selected data to save*, gráfico inferior da figura 58. Sendo satisfatórios, eles podem ser salvos no banco de dados pelo botão *Save selection* ou exportados para uma planilha no formato CSV pelo botão *Export selection* de maneira semelhante às funções presentes na tela *Calibration*. Mais uma vez, não foi implementado um recurso para importar para o sistema de medição os pontos amostrados a partir de uma planilha CSV.

Figura 58 – Página de medidas – realizando leitura e salvando trecho relevante.

Fonte: autoria própria (2017).

A respeito dos gráficos das medidas, as escalas de tempo variam conforme a ampliação aplicada a eles. O objetivo disso é reduzir a poluição visual da escala mostrando apenas a informação da escala relevante ao nível da ampliação. Entretanto, caso o ponto tenha valor nulo em uma dada escala, será usado o valor de escala de seu nível superior. Assim, a ampliação do gráfico até a escala de dezenas de milissegundos em torno de 1s, por exemplo, mostrará este ponto com valor de 1s, enquanto seus pontos vizinhos serão mostrados com seus valores em escalas de dezenas de milissegundos relativos a este valor de 1s, como pode ser observado na figura 59.

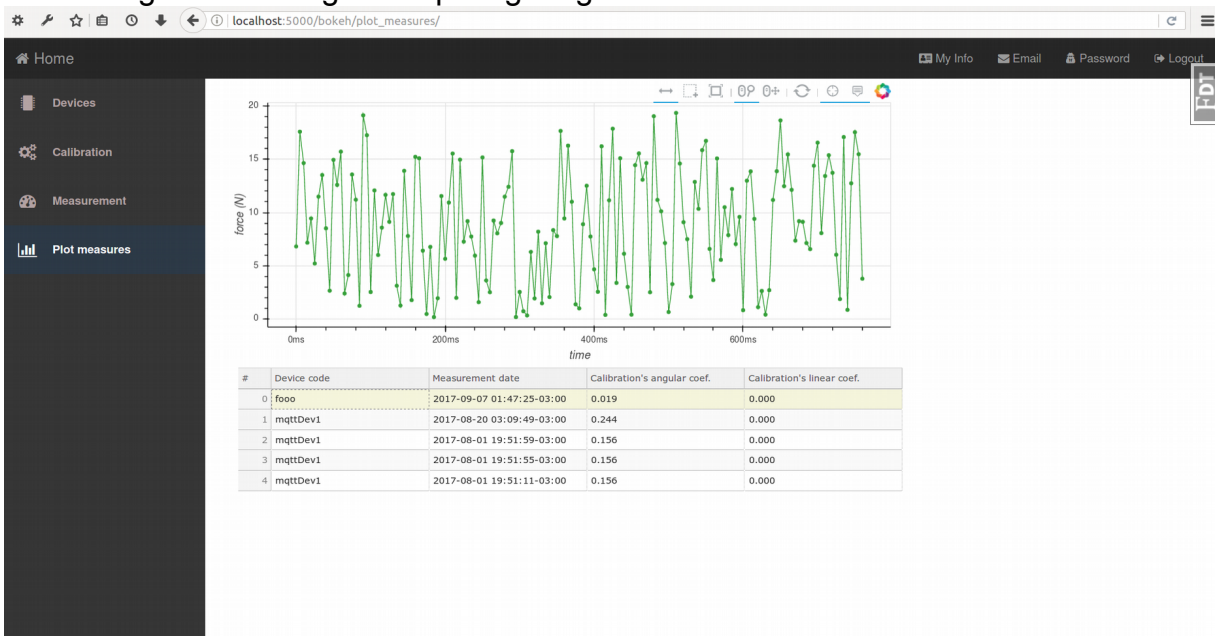
Figura 59 – Escala relativa de tempo.



Fonte: autoria própria (2017).

Por fim, a tela *Plot measures*, figura 60, disponibiliza para visualização, listadas em uma tabela, as últimas 5 medidas salvas pelo usuário na aplicação web. Basta clicar sobre sua respectiva linha na tabela para que a medida seja plotada. Apenas as últimas 5 medidas são mostradas porque não foi possível desenvolver, a tempo da entrega do projeto, um sistema de paginação de dados que operasse internamente à aplicação gráfica Bokeh para que se mantivesse a interatividade entre a plotagem gráfica e o tabelamento dos dados das medidas.

Figura 60 – Página de plotagem gráfica das últimas 5 medidas salvas.



Fonte: autoria própria (2017).

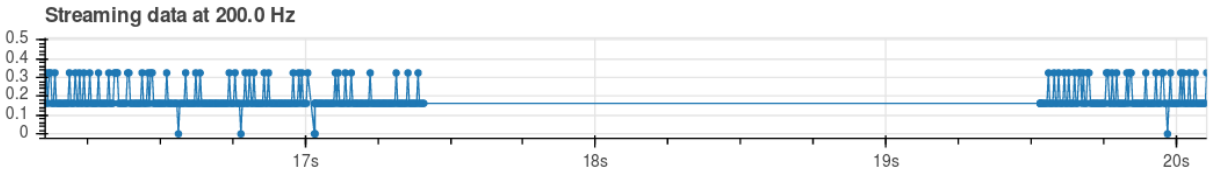
3.2 SISTEMA MICROCONTROLADO

O teste do sistema microcontrolado foi realizado somente com o kit Esp8266 e a aplicação web, com a porta AD do kit sem conexão com qualquer fonte de sinal. A curva de calibração utilizada foi inventada apenas para a realização do teste de comunicação e controle do dispositivo e, por isso, os valores das medidas em si não têm correspondência com medidas reais.

O sistema de controle e comunicação entre a aplicação web e o dispositivo Esp8266 obteve êxito em suas operações, realizando tarefas como conectar-se à aplicação web, associação da operação corrente com o usuário logado na aplicação, seleção de frequência de amostragem AD e controle de leitura e envio das medidas à aplicação.

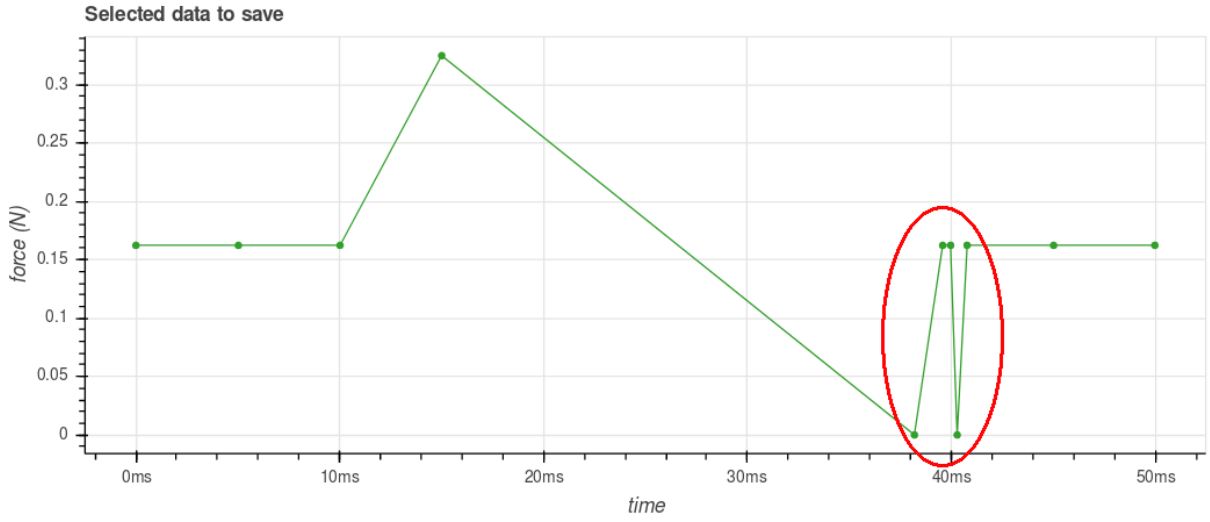
Entretanto, não foi possível enviar as mensagens MQTT com a constância necessária à operação do instrumento. Ora sua conexão com o servidor Mosquitto se perdia e as mensagens se acumulavam até ocuparem toda a capacidade de memória do microcontrolador, gerando erro de exceção e interrompendo seu sistema, ora o tempo de envio da mensagem ultrapassava o intervalo de tempo do período de amostragem da medida, figura 61. Devido a este último caso, quando o fluxo de dados voltava a ser enviado, algumas medidas AD foram realizadas e enviadas com período de amostragem inferior ao período definido na aplicação web, figura 62.

Figura 61 – Erro: período sem dados enviados superior ao período de leitura AD.



Fonte: autoria própria (2017).

Figura 62 – Erro: dados enviados incompatíveis com o período de amostragem.



Fonte: autoria própria (2017).

Em testes de depuração de código da aplicação do microcontrolador, foi constatado que o sistema opera com folga de memória RAM. Por isso, é possível que a origem desse problema esteja associada à maneira como o firmware do Esp8266 gerencia seus recursos internos ao utilizar o protocolo Wi-Fi ou ainda tenha a ver com a implementação e operação da biblioteca MQTT do dispositivo. É necessário, portanto, que se faça uma investigação mais detalhada sobre essas questões levantadas em um trabalho futuro.

3.3 CIRCUITO DE INSTRUMENTAÇÃO ELETRÔNICA

Com o auxílio de um dinamômetro de precisão a célula de carga foi submetida a tensões entre 0N e 200N, sendo alimentada pelos 5V da fonte simétrica do projeto. Os valores de tensão elétrica gerados pela célula foram então medidos e registrados na tabela 9 a seguir (os valores correspondentes a 0N e 200N foram utilizados como referência para o cálculo dos valores dos resistores de ganho dos amplificadores de instrumentação do circuito de aquisição da figura 34).

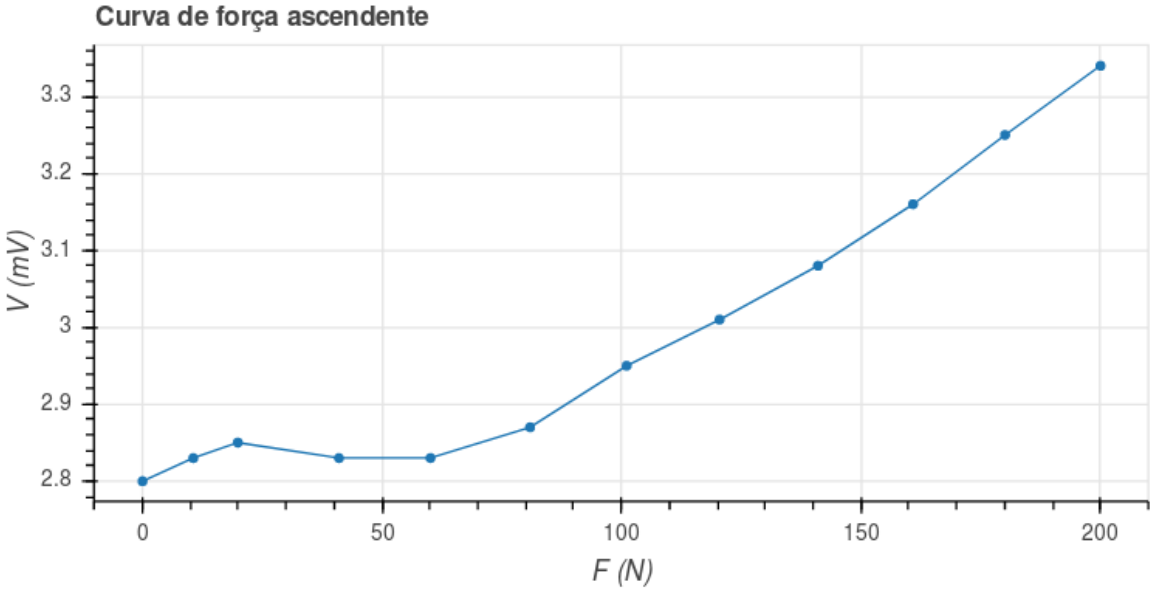
O mesmo teste foi feito alimentando a célula de carga com 5V a partir de uma fonte de bancada e o resultado encontrado foi análogo ao do encontrado com o teste usando a fonte simétrica do projeto.

Tabela 9 – Resposta da célula de carga à aplicação de força ascendente.

F (N)	V (mV)
0,0	2,80
10,6	2,83
19,9	2,85
41,0	2,83
60,1	2,83
80,9	2,87
101,1	2,95
120,5	3,01
141,0	3,08
160,9	3,16
180,1	3,25
200,0	3,34

Fonte: autoria própria (2017).

Figura 63 – Resposta da célula de carga à aplicação de força ascendente.



Fonte: autoria própria (2017).

A figura 63 representa graficamente os dados da tabela 9 e mostra que não foi detectada uma variação significativa de tensão do sinal para variações ascendentes de força aplicada entre 0N e 60N – 0kgf e 6kgf, aproximadamente. Esse intervalo de indefinição

abrange cerca de um terço da escala do dinamômetro e compreende justamente a região da escala na qual se supõe serem medidas as FPP de pacientes bastante debilitados.

Não foi possível investigar a origem do problema na célula de carga a tempo da entrega deste trabalho. Pode ter acontecido que um ou mais extensômetros tenham sido avariados por uso indevido, ou ainda tenham sido parcialmente descolados da célula de carga devido ao tempo, ao calor ou à qualidade da cola utilizada na fixação.

Em testes utilizando o circuito auxiliar para gerar tensões em milivolts, figura 39, no lugar da célula de carga, a amplificação destes valores de tensão ocorreram na faixa de valores projetados de 0V a 1V e de maneira linear, mas a tensão de saída apresentou flutuações da ordem de ±100mV que inviabilizaram tabelar a relação entre o sinal da fonte de milivolts – sinal da célula de carga – e a tensão de saída do circuito de aquisição. O mesmo nível de flutuação de tensão também foi observado quando o circuito de aquisição foi testado com a célula de carga.

A não linearidade foi maior ainda para variações descendentes, com os níveis de tensão de saída variando de 1V a 0V para variações de força de 200N a 100N, chegando a até -200mV para força em torno de 30N e então voltando a 0V quando a força retornava para 0N.

Em testes posteriores, foi constatado que os resistores do circuito de ajuste de offset com conexão com a fonte de alimentação simétrica, no circuito de aquisição de dados da figura 34, forneciam um caminho para propagar os ruídos da fonte simétrica para o sinal da célula de carga a ser amplificado. Assim, com o filtro de ruídos posicionado antes do circuito de offset e com o ganho de instrumentação em cerca de 60V/V, as pequenas flutuações da fonte simétrica produziram resultados insustentáveis no sinal de saída do circuito de aquisição. Portanto, no circuito da figura 34, o estágio de ajuste de offset deve ser posicionado após o estágio pré-amplificação e antes do estágio de filtragem de ruídos. Não houve tempo hábil, contudo, para confeccionar um outro circuito e validar essa correção, de forma que isso deve ser feito em trabalhos futuros.

3.4 TESTE INTEGRADO

Devido à não linearidade da resposta da célula de carga à variações de força e à flutuação de tensão de saída do circuito de aquisição, não foi possível realizar um teste integrado entre o circuito eletrônico e os sistemas da aplicação web e microcontrolado.

3.5 DISCUSSÃO

Embora a aplicação web tenha atingido os objetivos propostos por este trabalho, os problemas encontrados no sistema microcontrolado, na célula de carga e no circuito de aquisição impossibilitaram que o projeto fosse validado em sua totalidade.

É necessário que se faça uma investigação mais detalhada sobre essas questões levantadas para que possam ser corrigidas, se possível, ou para que se constate a inviabilidade da proposta do projeto.

4 CONCLUSÃO

Dentre as três frentes do projeto – sistema da aplicação web, sistema microcontrolado e circuito eletrônico – apenas o sistema da aplicação web obteve êxito quanto ao esperado pelo projeto. Conseguiu entregar a execução de tarefas de cadastramento de usuários e de instrumentos (por meio de seus módulos Esp8266), de calibração de instrumentos por software e de recepção e plotagem dos dados das medidas, além de realizar comandos de controle como conexão e desconexão da aplicação com o instrumento, alteração de frequência de amostragem e de possibilitar salvar medidas em um banco de dados.

Quanto ao circuito eletrônico, ele apresentou níveis de flutuação de tensão que são inaceitáveis (±100mV) e, embora tenha seus estágios de ganho adequados para amplificar as tensões dos sinais fornecidos pela célula de carga para valores entre 0V e 1V, como projetado, obteve leituras da célula de carga não-lineares, pondo em cheque a viabilidade do instrumento como um todo. A montagem dos extensômetros na célula de carga e leiaute do circuito de aquisição precisam ser investigados e posteriormente corrigidos.

Por último, apesar de o sistema microcontrolado ter conseguido realizar tarefas como conexão à internet e ao broker mosquitto, responder aos comandos de controle enviados por meio da aplicação web e de realizar medidas AD conforme as frequências de amostragens determinadas pela aplicação web, ele não conseguiu operar de maneira contínua como o previsto. Esporadicamente ele parava de enviar os dados por intervalos de tempo variados e esse comportamento inviabiliza o seu uso no instrumento, que precisa operar em tempo real. É necessário que se faça uma investigação aprofundada sobre a operação do firmware do dispositivo Esp8266 com relação ao gerenciamento de recursos internos do microcontrolador e de seu controle sobre a comunicação Wi-Fi, bem como a implementação e operação da biblioteca MQTT, em micropython, que foi utilizada pelo sistema para identificar e corrigir, se possível, a causa do problema que impede o envio constante de dados do dispositivo para a aplicação web.

De maneira geral, o projeto cumpriu o seu objetivo de promover a utilização dos conhecimentos adquiridos ao longo do curso de graduação e, ainda que parcialmente, atendeu o objetivo de recepcionar e armazenar as leituras de FPP em um servidor para então disponibilizá-las graficamente por meio da aplicação web. Contudo, não obteve sucesso ao entregar um instrumento operacional para medições de força de preensão palmar.

4.1 DESAFIOS FUTUROS

Os problemas encontrados no trabalho atual ficam como desafio a serem solucionados em trabalhos futuros:

- célula de carga: comportamento não-linear.
- circuito de aquisição: sinal de saída com flutuação de até ±100mV.
- Esp8266: mensagens MQTT não são enviadas de modo constante.
- aplicação web: sistema de paginação das tabelas contidas nas telas com aplicação Bokeh.

Além dos problemas a serem resolvidos, a interface da aplicação web pode ser aprimorada para conter:

- um modelo de dados e um sistema de cadastro para os pacientes avaliados nas medições.
- um modelo de dados e um sistema de cadastro para registrar os parâmetros adotados nas medidas de forma com que elas possam posteriormente ser agrupadas por protocolo de medição.
- uma seção administrativa para gerenciar os usuários do sistema, seus pacientes e as medidas realizadas.
- uma ou mais telas mostrando os valores de parâmetros de FPP, por paciente, tais como força média, mínima e máxima, sustentação de força, fadiga, curva de evolução histórica do paciente, etc.

Quanto a questões internas do sistema, podem ser feitos:

- uso de certificados digitais de autenticidade tanto para os servidores quanto para os dispositivos Esp8266 utilizados a fim de permitir usar criptografia nas comunicações para garantir a segurança e a integridade dos dados dos pacientes.
- hospedagem do sistema da aplicação web por meio de um serviço de nuvem ou de IP fixo.
- substituição do servidor Flask por um servidor HTTP próprio para uso em produção.
- uso de um servidor proxy web, como o Nginx, para realizar balanceamento de carga de dados, manipular arquivos estáticos, indexar arquivos, etc.

- adição de um sistema de gerenciamento automatizado de processos para gerenciar os serviços da aplicação, como o Supervisor.

REFERÊNCIAS

1 SHIRATORI, A. P.; IOP, R. R.; BORGES JR., N. G.; DOMENECH, S. C.; GEVAERD, M. S. Protocolos de avaliação da força de preensão manual em indivíduos com artrite reumatoide: uma revisão sistemática. Revista Brasileira de Reumatologia, [S.l.], v. 54, n. 2, p. 140-147, 2014.

2 HELLIWELL, P.; HOWE, A.; WRIGHT, V. Functional assessment of the hand: reproducibility, acceptability, and utility of a new system for measuring strength. Annals of the Rheumatic Diseases, [S.l.], v. 46, p. 203-208, 1987.

3 MOREIRA, D.; ALVAREZ, R. R. A. Avaliação da força de preensão palmar com o uso do dinamômetro Jamar® em pacientes portadores de hanseníase atendidos em nível ambulatorial no Distrito Federal. Hansenologia Internationalis, [S.l.], v. 27, n. 2, p. 61-69, 2002.

4 DIAS; J. A., OVANDO; A. C., KÜLKAMP; W. K., BORGES JR.; N. G. Força de preensão palmar: métodos de avaliação e fatores que influenciam a medida. A Revista Brasileira de Cineantropometria e Desempenho Humano, [S.l.], v. 12, n. 3, p. 209-216, 2010.

5 FIGUEIREDO, I. M. et al. Teste de força de preensão utilizando o dinamômetro Jamar®. Revista Acta Fisiátrica, [S.l.], v. 14, n. 2, p. 104-110, 2007.

6 KAMIMURA, T.; IKUTA, Y. Evaluation of grip strength with a sustained maximal isometric contraction for 6 and 10 seconds. Journal of Rehabilitation Medicine, [S.l.], v. 33, p. 225-229, 2001.

7 NUNES M. M.; BRUNO, N. B. Equipamento para Medição de Força de Preensão Palmar. 2012. 54 f. Trabalho de Conclusão de Curso (Engenharia Elétrica com ênfase em Eletrônica) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2012.

8 HORITA, L. A. H. Sistema para avaliação de preensão. 2015. 52 f. Trabalho de Conclusão de Curso (Engenharia Elétrica com ênfase em Eletrônica) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2015.

9 MATHIOWETZ, V. et. al. Grip & Pinch Strenght: Normative Data for Adults. Arch Phys Med Rehab, [S.l.], v. 66, p. 69-72, 1985.

10 MATHIOWETZ, V. Effects of Three Trials on Grip and Pinch Strength Measurements. J Hand Ther, [S.l.], v. 3, p. 195-198, 1990.

11 CAPORRINO, F. A. et al. Estudo populacional da forca de preensão palmar com dinamômetro Jamar. Rev Bras Ortop, [S.l.], v. 33, n. 2, p. 150-154, 1998.

12 NATHAN, D. E.; JOHNSON, M. J.; MCGUIRE, J. R. Design and validation of low-cost assistive glove for hand assessment and therapy during activity of daily living-focused robotic stroke therapy. Journal of Rehabilitation Research & Development, [S.l.], v. 46, n. 5, p. 587-602, 2009.

13 GIT PROJECT, 2017. Disponível em: <https://git-scm.com/>. Acesso em: 31 outubro 2017.

14 GITHUB, 2017. Disponível em: <https://github.com/>. Acesso em: 31 outubro 2017.

15 GEARBEST, [S.l.], [20--?]. Disponível em: <http://www.gearbest.com/transmitters-receivers-module/pp_227651.html>. Acesso em: 31 maio 2017.

16 PYTHON SOFTWARE FOUNDATION, [S.l.], 2017. Disponível em: <https://www.python.org/doc/essays/blurb/>. Acesso em: 31 maio 2017.

17 GEORGE ROBOTICS LIMITED, [Cambridge], 2017. Disponível em: <https://micropython.org/>. Acesso em: 31 maio 2017.

18 LIGHT, R. A., Mosquitto: server and client implementation of the MQTT protocol. The Journal of Open Source Software, v. 2, n. 13, maio 2017. doi:10.21105/joss.00265.

19 ECLIPSE, 2017. Disponível em: <http://www.eclipse.org/paho/>. Acesso em: 31 maio 2017.

20 HIVEMQ, [2015] Disponível em: <http://www.hivemq.com/blog/how-to-get-started-with-mqtt>. Acesso em: 31 maio 2017.

21 PIETIKÄINEN, S. PAGE FAULT BLOG, 2 mar 2017. Disponível em: <https://pagefault.blog/2017/03/02/using-local-mqtt-broker-for-cloud-and-interprocess-communication/>. Acesso em: 31 maio 2017.

22 HIVEMQ, [2015] Disponível em: <http://www.hivemq.com/blog/mqtt-essentials-part-3-client-broker-connection-establishment>. Acesso em: 31 maio 2017.

23 BARROS, M. MQTT - Protocolos para IoT. Embarcados, [S.l.], 15 jun. 2015. Disponível em: <https://www.embarcados.com.br/mqtt-protocolos-para-iot/>. Acesso em: 31 maio 2017.

24 MOSQUITTO PROJECT, 2017. Disponível em: <https://mosquitto.org/man/mqtt-7.html>. Acesso em: 31 maio 2017.

25 ECLIPSE, 2017. Disponível em: <https://www.eclipse.org/paho/files/mqttdoc/MQTTClient/html/qos.html>. Acesso em: 31 maio 2017.

26 POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2017. Disponível em: <https://www.postgresql.org>. Acesso em: 31 maio 2017.

27 POSTGRESQL TUTORIAL TEAM, 2017. Disponível em: <http://www.postgresqltutorial.com/postgresql-json/>. Acesso em: 31 maio 2017.

28 FLASK, 2017. Disponível em: <http://flask.pocoo.org/>. Acesso em: 31 maio 2017.

29 CONTINUUM ANALYTICS, 2017. Disponível em: <http://bokeh.pydata.org/en/latest/>. Acesso em: 31 maio 2017.

30 TORNADO TEAM, 2017. Disponível em: <http://www.tornadoweb.org>. Acesso em: 31 maio 2017.

31 IETF TOOLS, 2017. Disponível em: <https://tools.ietf.org/html/rfc6455>. Acesso em: 31 maio 2017.

32 GEORGE, D. P., SOKOLOVSKY, P. et al, [Cambridge], 2017. Disponível em: <http://docs.micropython.org/en/v1.9.1/esp8266/esp8266/general.html?highlight=file%20system#boot-process>. Acesso em: 31 maio 2017.