

LUIZ ALBERTO HIROSHI HORITA

**SISTEMA PARA AVALIAÇÃO DE
PREENSÃO**

Trabalho de Conclusão de Curso apresentado à Escola de
Engenharia de São Carlos, da Universidade de São Paulo.
Curso de Engenharia Elétrica com ênfase em Eletrônica

ORIENTADOR: Alberto Cliquet Júnior

São Carlos

2015

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

H811s Horita, Luis Alberto Hiroshi
Sistema para Avaliação de Preenção / Luis Alberto
Hiroshi Horita; orientador Alberto Júnior Cliquet;
coorientador Renato Varoto. São Carlos, 2015.

Monografia (Graduação em Engenharia Elétrica com
ênfase em Eletrônica) -- Escola de Engenharia de São
Carlos da Universidade de São Paulo, 2015.

1. Tetraplegia. 2. Reabilitação. 3. Membros
superiores. 4. Preenção palmar. 5. Célula de carga. 6.
Arduino. I. Título.

FOLHA DE APROVAÇÃO

Nome: Luiz Alberto Hiroshi Horita

Título: "Sistema para avaliação de preensão"

Trabalho de Conclusão de Curso defendido e aprovado
em 27/11/2015,
com NOTA 10,0 (dez, zero), pela Comissão Julgadora:

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

Dr. Renato Varoto - (SEL/EESC/USP)

Mestre Renata Manzano Maria - (UNICAMP)

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Dr. José Carlos de Melo Vieira Júnior

Dedicatória

Dedico este trabalho ao meu pai, um grande exemplo de pessoa que desde sempre o vejo como um parâmetro a ser seguido no que diz respeito ao senso ético, ao gosto de fazer o bem ao próximo e à dedicação e amor à família;

À minha mãe, uma mulher extremamente batalhadora e forte, que em meu ponto de vista a enxergo como símbolo de superação, por ter passado a vida inteira tendo que trabalhar para ajudar a família, seja dentro ou fora de casa;

Ao meu irmão gêmeo, por sempre ter andado ao meu lado e sendo um “adversário” para diversos tipos de desafios, o que sempre me proporcionou motivação para estar sempre melhorando;

Ao meu irmão caçula, pela grande amizade, e mesmo sendo mais novo, através de seus atos, me ensinou a perder o medo de ser eu mesmo fora de casa;

Aos meus tios, tias e avós que me deram todo suporte quando precisei em todos os momentos de minha vida;

E a todos os meus vínculos de amizade no decorrer da minha vida de graduação, por todos os bons e maus momentos passados, que me fizeram crescer como pessoa, em especial às pessoas com quem tive maior afinidade e contato (Dani, Hideo (“Shoyu”), Caio, Eiji, Mayumi, Fernando, Hiromitsu, Grace, Caique, Rita, Cecília, Lucas (“Bigode”), Miki, Joice, Leonardo (“X”) e Heinz).

Agradecimentos

Agradeço ao professor Evandro, por mostrar o caminho para desenvolver a placa de aquisição no início do projeto; ao Rui, pelo auxílio na construção da carcaça do hardware; ao Leonardo ("X") e ao Verdini, pela grande ajuda no desenvolvimento da interface gráfica.

E dentro do ambiente de trabalho (LABCIBER), agradeço ao meu orientador, Cliquet, por me aceitar como orientado e me dar essa oportunidade de desenvolver este trabalho; e ao Dr. Renato Varoto, pelo apoio e assistência no decorrer do desenvolvimento desse projeto ao longo do ano.

"Uma pessoa bem sucedida é aquela que constrói uma base firme com os tijolos que outros jogam nele(a)."

David Brinkley

Resumo

HORITA, L. A. H. Sistema para Avaliação de Força de Prensão. Trabalho de Conclusão de Curso – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2015.

A medição de força de preensão palmar, antes utilizada apenas para análise clínico-funcional da mão e punhos de pacientes que sofrem de patologias e doenças em nervos, pode também ser utilizada para diagnosticar causas como a baixa densidade mineral óssea (DMO) em idosas, e até mesmo o estado nutricional; portanto, este trabalho teve como objetivo desenvolver um equipamento capaz de realizar medições da preensão palmar expondo-as em um gráfico em tempo real, podendo salvar arquivos de dados contendo as informações pessoais do paciente (nome, idade e sexo) e as medidas coletadas em cada sessão. Para alcançar o escopo traçado, o trabalho foi dividido em quatro partes principais, sendo estas a célula de carga, o circuito elétrico de tratamento do sinal analógico, a plataforma para aquisição de dados (Arduino) e a interface gráfica ao usuário (GUI), sendo o último desenvolvido em linguagem Python. Desenvolvidos todos os quatro estágios, ao final do projeto foram feitas sessões de teste de acuracidade do aparelho, aplicando-se forças constantes de 0, 50, 100, 150 e 200 Newtons, e simulando uma sessão aplicando-se uma força de preensão palmar real. Ao fim do trabalho, foi obtido um sucesso parcial, pois apesar do ótimo desempenho do circuito de tratamento do sinal analógico, da plataforma de aquisição e do software desenvolvido, o equipamento se mostrou exato apenas nas regiões de força próximas às intersecções da curva característica real e função de tendência encontrada do equipamento (em torno de 40N e 150N), isso ocorreu devido ao comportamento não perfeitamente linear do conjunto célula de carga utilizado.

Palavras chave: Tetraplegia, Reabilitação, Membros superiores, Prensão palmar, Célula de Carga, Arduino.

Abstract

HORITA, L. A. H. Sistema para Avaliação de Força de Prensão. Trabalho de Conclusão de Curso – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2015.

The measurement of hand-grip strength, before used just for clinical and functional analysis of the patient's hands and fists that suffers with pathologies and diseases of nerve, can also be used to diagnose causes like the low bone mineral density in elderly, and even the nutritional status; therefore, this work aims an equipment capable of performing measurements of the handgrip strength, showing it on a real time plotting graph, and capable of saving data files containing the patient's personal data (name, age and gender) and each point of measure collected in session. To achieve the scope, the work was divided into four main parts: load cell, electrical circuit of analog signal treatment, data acquisition platform (Arduino) and graphical user interface (GUI), the last one developed in Python program language. After developing all the four stages, in the end of the project some precision test sessions were done with the equipment, applying constant force and simulating a session applying real hand-grip strength. In the end of the work, a partial success was achieved, because although the great performance of the analog signal treatment circuit, of the acquisition platform and of the developed software, the equipment was just accurate in some intervals close to the intersection of real characteristic curve and the trend function of the equipment (around 40N and 150N), this happens because of the not perfectly linear behavior of the load cell used.

Keywords: Upper limbs, Rehabilitation, Quadriplegia, Handgrip, Load Cells, Arduino.

Lista de Figuras

<i>Figura 2-1 Posição padronizada para a realização dos testes de FPP [6].....</i>	3
<i>Figura 2-2 Aparelho comercial de exercício dos músculos das mãos, da marca Gripmaster.....</i>	5
<i>Figura 2-3 Representação do corpo da célula de carga na configuração viga dupla engastada [6].....</i>	5
<i>Figura 2-4 Curva de comportamento de um material [6]</i>	6
<i>Figura 2-5 Esquema para dimensionamento do corpo da célula de carga, onde o ponto no centro da figura é o local de aplicação da força [6].</i>	7
<i>Figura 2-6 As três peças principais da célula de carga já fabricadas [6].</i>	8
<i>Figura 2-7 Estrutura do extensômetro [6].....</i>	9
<i>Figura 2-8 Explicação do código padronizado, extensômetro Excel [6].....</i>	9
<i>Figura 2-9 Perfil do momento ao longo do corpo de prova sob força de preensão [6].</i>	10
<i>Figura 2-10 Posicionamento dos extensômetros sobre o corpo da célula de carga [6].</i>	10
<i>Figura 2-11 Extensômetros fixados ao corpo de prova da célula de carga [6].....</i>	11
<i>Figura 2-12 Célula de carga semi-finalizada [6].</i>	11
<i>Figura 2-13 Célula de carga finalizada [6].</i>	11
<i>Figura 2-14 Circuito Elétrico final representado por LTspice.</i>	13
<i>Figura 2-15 Ponte de Wheatstone</i>	14
<i>Figura 2-16 Teste de funcionamento do protótipo do circuito elétrico em protoboard..</i>	16
<i>Figura 2-17 Circuito elétrico final para amplificação, filtragem e correção de offset do sinal.</i>	17
<i>Figura 2-18 Posicionamento do dinamômetro sobre a célula de carga para a coleta das medidas.</i>	18
<i>Figura 2-19 Curva referente ao comportamento da primeira série de compressão da célula de carga com o circuito elétrico, e a linha de tendência final encontrada.....</i>	21
<i>Figura 2-20 Arduino NANO.....</i>	22
<i>Figura 2-21 Ambiente Integrado de Desenvolvimento do Arduino.....</i>	23
<i>Figura 2-22 Configuração para simulação de teste e desenvolvimento do firmware... </i>	24
<i>Figura 2-23 Geany IDE usado para programar em Python.</i>	25
<i>Figura 2-24 Diagrama de blocos da lógica da interface gráfica.</i>	26
<i>Figura 2-25 Teste de detecção e conexão automática da porta USB do equipamento.</i>	27
<i>Figura 2-26 Chart gerado pelo teste da função de plotagem do gráfico.</i>	27

<i>Figura 2-27 Arquivo Excel gerado no teste da função para salvar a sessão de medição.</i>	28
<i>Figura 2-28 Configuração do GUI desenvolvida.....</i>	29
<i>Figura 2-29 Interface Gráfica apóis o teste de funcionamento já com o hardware funcionando.</i>	30
<i>Figura 2-30 Janela exposta ao clicar no botão “Save”.</i>	31
<i>Figura 2-31 Arquivo Excel gerado e salvo no fim da sessão de medição da FPP.</i>	31
<i>Figura 3-1 Célula de carga com o cabo de conexão.</i>	32
<i>Figura 3-2 Vista frontal da caixa do hardware.</i>	33
<i>Figura 3-3 Parte interna da peça frontal da caixa do hardware.</i>	33
<i>Figura 3-4 Interior da caixa do hardware.</i>	34
<i>Figura 3-5 Vista traseira da caixa do hardware.</i>	34
<i>Figura 3-6 Hardware montado.</i>	35
<i>Figura 3-7 Baterias de 12V.</i>	35
<i>Figura 3-8 Teste de precisão com 0N aplicada à célula de carga.</i>	36
<i>Figura 3-9 Teste de precisão com 50N aplicada à célula de carga.</i>	37
<i>Figura 3-10 Teste de precisão com 100N aplicada à célula de carga.</i>	37
<i>Figura 3-11 Teste de precisão com 150N aplicada à célula de carga.</i>	38
<i>Figura 3-12 Teste de precisão com 200N aplicada à célula de carga.</i>	38
<i>Figura 3-13 Curva de histerese do comportamento do equipamento durante compressão e descompressão da célula de carga e linha de tendência.</i>	39

Lista de Tabelas

<i>Tabela 2-1 Medições referentes à primeira série do ato de compressão e descompressão da célula de carga, acoplada ao circuito elétrico de tratamento do sinal.</i>	19
<i>Tabela 2-2 Coeficiente angular e linear das linhas de tendência das curvas levantadas das cinco séries de medições da célula de carga acoplada ao circuito elétrico de tratamento do sinal.</i>	20

Lista de Siglas

FPP – Força de Prensão Palmar

DMO – Densidade Mineral Óssea

SATM – Sociedade Americana de Terapeutas de Mão

GUI – *Graphical User Interface*

LM7809 – Regulador de tensão

LM7909 – Regulador de tensão

CMR – *Common mode rejection*

R_G – Resistência de Ganho

Trimpot – *Trimmer potentiometer*

USB – *Universal Serial Bus*

EEPROM – *Electrically Erasable Programmable Read-Only Memory*

IDE – *Integrated Development Environment*

ADC – *Analog-to-Digital Converter*

PyQt4 – Biblioteca python para desenvolvimento de Interface Gráfica ao Usuário

LED – *Light Emitting Diode*

DIN – *Deutsches Institut für Normung*

SAE – *Society of Automotive Engineers*

Sumário

Dedicatória	I
Agradecimentos.....	III
Resumo	V
Abstract	VII
Lista de Figuras	IX
Lista de Tabelas	XI
Lista de Siglas	XII
Sumário	XIII
1. Introdução.....	1
1.1. Contexto da pesquisa.....	2
1.2. Métodos de Avaliação da FPP	2
1.3. Objetivos.....	3
2. Materiais e Métodos	4
2.1. Célula de Carga.....	4
2.1.1. Arquitetura.....	5
2.1.2. Dimensionamento.....	6
2.1.3. Montagem da célula de carga.....	8
2.2. Circuito Elétrico.....	12
2.2.1. Regulador de tensão	14
2.2.2. Extensômetros em Ponte de Wheatstone.....	14
2.2.3. Amplificador de Instrumentação	14
2.2.4. Filtro	15
2.2.5. Ajuste de Offset.....	15
2.2.6. Amplificador Inversor.....	15
2.2.7. Amplificador Não-Inversor	16
2.2.8. Montagem do Circuito Elétrico.....	16
2.2.9 Caracterização da célula de carga acoplada ao circuito elétrico.....	17

2.3. Arduino	21
2.3.1. Configuração do Arduino para coleta de dados	21
2.3.2. Instruções do Arduino.....	24
2.4. Python	25
2.4.1. Lógica da Interface Gráfica.....	25
2.4.2. Instruções do GUI.....	26
2.4.3. Teste do GUI	29
3. Resultados e discussão	32
3.1. Hardware	32
3.2. Caracterização do equipamento	36
Conclusão.....	41
Anexo	42
Anexo A – Tabela de padrões de Força de Prensão Palmar em adultos.....	42
Referências bibliográficas.....	43
Apêndices.....	44
Apêndice A – Instruções do Arduino.....	44
Apêndice B – Programação em Python – GUI.....	44
Apêndice C – Programação em Python – Classe para leitura de dados.....	49
Apêndice D – Programação em Python – Classe para salvar dados.....	50

1. Introdução

A medição de Força de Prensão Palmar (FPP) é bastante aplicada para a análise clínico-funcional da mão e punhos de pacientes com disfunções como patologias e doenças em nervos, que são a maioria dos casos dentre as possíveis causas de fraqueza nas mãos [6], porém essa prática pode ser de grande importância para avaliação quantitativa da melhora da força de preensão de um paciente com lesão medular durante o tratamento, considerando-se que até então não existem meios de testes funcionais para tetraplégicos [6].

Além disso, estudos recentes mostram que a força de preensão palmar pode ter correlações diretas com a Densidade Mineral Óssea (DMO) em idosas [2], e até mesmo com o estado nutricional, como foi observada em um experimento, no qual observou-se que a diminuição da força de preensão está relacionada com uma maior duração no internamento hospitalar, sendo que nesta situação, a porcentagem de perda da FPP é maior nas crianças desnutridas em comparação com as obesas [9].

Tendo isso em mente, é plausível dizer que é possível utilizar a análise do comportamento da FPP de um paciente como meio de diagnóstico para possíveis problemas de saúde, como as citadas anteriormente, além, é claro, de utilizar para avaliar a melhora da força em pacientes em tratamento de reabilitação.

Deve ser também levado em consideração que a força de preensão palmar possui correlação direta com a diferença de sexo e faixa etária, onde a média da força de preensão das pessoas do sexo masculino é nitidamente maior que as do sexo feminino; e no geral, há o aumento da força média na fase adolescente. O pico máximo encontra-se na fase adulta e a partir da fase de meia-idade ocorre o declínio da força de preensão em ambos os sexos [4]. Além disso, há também o fator da dominância lateral, isto é, a mão dominante tem maiores valores de força de preensão manual que a não dominante [5].

Logo, futuros estudos populacionais com amostras representativas atualizadas, serão necessários para avaliar a força da preensão em indivíduos saudáveis. A padronização desta força é um instrumento importante e que pode auxiliar no seguimento de pacientes com artropatias crônicas, miopatias congênitas e miopatias inflamatórias, entre as outras doenças correlacionadas à FPP [4].

1.1. Contexto da pesquisa

Não há um grande avanço no método de medição da força de preensão palmar. Até o momento, os mais utilizados são: dinamômetros hidráulicos (JAMAR), que são os mais citados em literaturas e considerados como padrão para o teste de FPP, com ajuste discreto da empunhadura contendo até cinco posições, adaptando-se à mão do paciente; os pneumáticos que medem a pressão, logo dependem da área onde a força é aplicada, podendo sofrer alterações de medida dependendo do tamanho da mão; e também os digitais que permitem a avaliação da curva força x tempo, o que proporciona outros parâmetros para análise da força, como por exemplo, tempo para atingir a força máxima, taxa de fadiga e área sob a curva, entre outros. [8]

Além disso, tem o trabalho de NUNES, “Equipamento para Medição de Força de Prensão Palmar” [6], o qual este trabalho retoma, utilizando a célula de carga nele desenvolvida.

1.2. Métodos de Avaliação da FPP

A utilização de um protocolo-padrão é importante para melhorar a precisão e a consistência do teste, uma vez que as divergências nos protocolos utilizados podem afetar a reproduzibilidade das medidas e a comparação dos valores absolutos com outros estudos [8].

Para isso, a posição adotada foi a padronizada para a realização dos testes de força, definida pela Sociedade Americana de Terapeutas de Mão (SATM) como sendo a melhor posição para medir a FPP. O paciente deve se sentar com o ombro aduzido, o cotovelo flexionado a 90°, o antebraço em posição neutra e o punho entre 0 e 30° de extensão e 0 a 15° de desvio ulnar [6], conforme mostrada na Figura 2-1.



Figura 2-1 Posição padronizada para a realização dos testes de FPP [6].

Em estudo realizado por Haidar, Kumar et al., constatou-se que tanto a média de três tentativas quanto o valor de uma única medida apresentaram alta consistência, sem diferença significativa entre os métodos [3]. Nos casos em que se realiza mais de uma tentativa, o intervalo de tempo de descanso entre elas influencia no desempenho da força, pois está diretamente relacionado com a fadiga muscular; considera-se prudente um descanso de no mínimo 1 minuto para neutralizar os efeitos da fadiga [8].

Com relação ao tempo de contração, existem poucos estudos que focam a influência do tempo de sustentação da força nos resultados [8]. O tempo de contração recomendado é de três segundos ou menos para que se evite um aumento considerável na pressão sanguínea e frequência cardíaca [7].

1.3. Objetivos

Este projeto tem como objetivo aplicar os conhecimentos de Engenharia Eletrônica para desenvolver um equipamento capaz de realizar medições da força de preensão palmar expondo-as em um gráfico em tempo real, podendo salvar arquivos de dados contendo informações pessoais do paciente como: nome, idade e sexo, e as medidas coletadas em cada sessão. Com isso, o dispositivo permite criar, no decorrer do tempo, um banco de dados com os arquivos gerados das sessões de medição já realizadas; logo, também será possível fazer uma padronização da FPP, separando

em categorias conforme a faixa etária e o sexo, tornando o estudo e a análise da força de preensão mais eficiente.

O trabalho concluído deve ser um equipamento de fácil manuseio e com uma Interface Gráfica simples e intuitiva.

2. Materiais e Métodos

O equipamento desenvolvido é constituído pelo *hardware* e *software*. Para o seu desenvolvimento, o trabalho foi dividido em 4 etapas:

- Estudo da célula de carga;
- Desenvolvimento do circuito elétrico, responsável pelo tratamento do sinal analógico;
- Projeto da placa de aquisição (Arduino), responsável pela conversão do sinal de analógico para digital;
- Implementação da interface gráfica, por onde o usuário pode conduzir o teste.

Destas quatro etapas, as três primeiras são as que compõem o *hardware*, sendo a última o *software*.

2.1. Célula de Carga

A célula de carga utilizada foi a mesma projetada no trabalho “Equipamento para Medição de Força de Prensão Palmar” [6], conforme dito anteriormente.

No desenvolvimento da célula de carga, foram levados em consideração alguns pontos importantes: por esta ser utilizada em hospitais e clínicas, sua manopla deve ser portátil, de fácil manuseio, possibilitando rápidas e práticas sessões de avaliação da força de preensão palmar.

A arquitetura da célula de carga foi baseada em um aparelho comercial para exercitar os músculos das mãos da marca *Gripmaster*, como mostra a Figura 2-2.



Figura 2-2 Aparelho comercial de exercício dos músculos das mãos, da marca Gripmaster.

2.1.1. Arquitetura

A arquitetura da célula de carga foi construída na configuração viga dupla engastada pela sua simplicidade e leveza, conforme mostra a Figura 2-3.

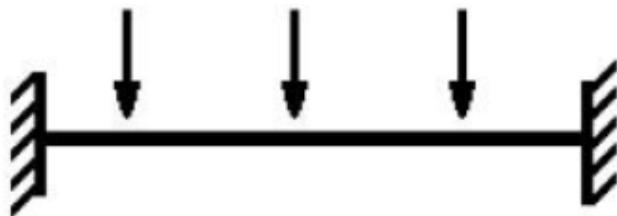


Figura 2-3 Representação do corpo da célula de carga na configuração viga dupla engastada [6].

A estrutura da célula de carga é composta de três principais peças: duas bases rígidas e fixas para servir de apoio à mão e para a fixação da terceira peça, o qual será o corpo de prova para receber a força de preensão palmar do paciente.

O material utilizado na sua construção foi o aço 1020, devido às suas características mecânicas [6]:

- Nome técnico: SAE 1020;
- Teor de Carbono (%): 0,20;
- Limite de elasticidade (Pa): 170 G;
- Limite de escoamento (Pa): 210 M;
- Limite de resistência à tração (Pa): 380 G;

- Alongamento (%): 25.

Para que o equipamento funcione corretamente, o material deve operar no regime elástico (vide Figura 2-4), ou seja, poder sofrer uma deformação pelo esforço aplicado e conseguir voltar à sua forma original, sem alteração de suas características físicas. Logo, a célula de carga foi dimensionada para que, dentro do intervalo de forças escolhidas, o material funcionasse na região onde a tensão é proporcional à deformação, de forma aproximadamente linear.

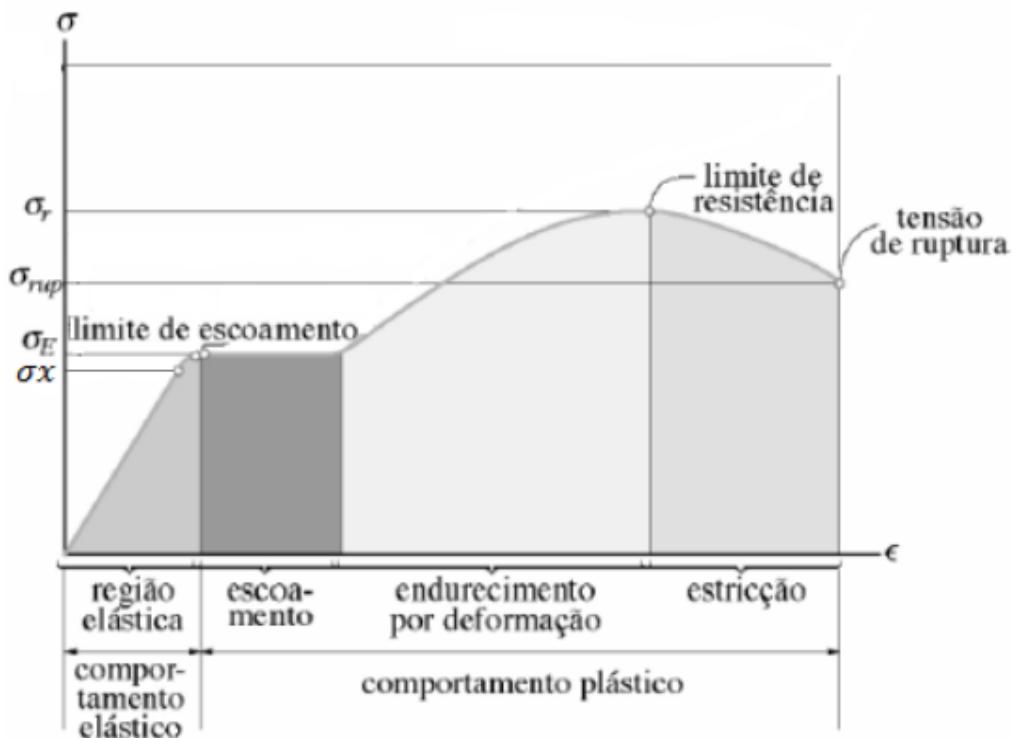


Figura 2-4 Curva de comportamento de um material [6].

2.1.2. Dimensionamento

Para o dimensionamento da célula de carga, foram adotadas quatro variáveis representando o comprimento (L), a largura (h), a altura (b) e a distância do ponto de aplicação da força e os extensômetros (x) (vide Figura 2-5).

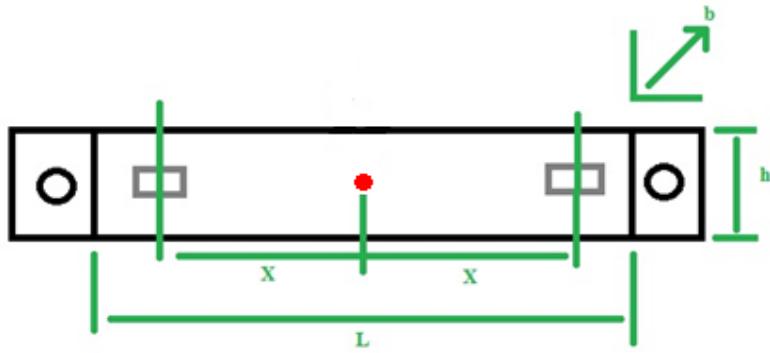


Figura 2-5 Esquema para dimensionamento do corpo da célula de carga, onde o ponto no centro da figura é o local de aplicação da força [6].

Para calcular a tensão de escoamento do material utilizado, chegou-se na equação (1) [6].

$$\sigma_x = \frac{P * L}{b * h^2} * \frac{3}{4} \quad (1)$$

Onde:

σ_x = a tensão do material;

P = força máxima aplicada no equipamento [N];

L = comprimento da célula de carga [m];

* observe que há uma margem para encaixe da peça;

b = altura da célula de carga [m];

h = largura da célula de carga [m];

Há algumas condições que foram levadas em consideração para realizar o dimensionamento da célula de carga:

- a tensão do material deve ser inferior à tensão limite, conforme equação (2), definida como a tensão de escoamento (vide Figura 2-4);
- o equipamento deve abranger todos os valores específicos de um adulto saudável (20 a 24 anos de idade), ou seja, de 0 a 120N, conforme a tabela no Anexo A, "Pinch Strength Norms for Adults" [6]. Mas para que o equipamento seja mais robusto, foi adotado 200N como sendo a força máxima;
- a manopla deve ter as dimensões aproximadas do tamanho padrão da mão de um adulto, sendo $L \leq 100\text{mm}$ e $h \approx 5\text{mm}$ [6].

$$\sigma_x < \sigma_{limite} \quad (2)$$

Onde:

σ_{limite} = a tensão limite, determinada como sendo limite de escoamento, que no caso é 210 MPa, conforme as propriedades do aço escolhido (SAE 1020).

No fim do dimensionamento, respeitando todas as condições, chegou-se nos valores:

- $L = 100 \text{ mm}$;
- $h = 20 \text{ mm}$;
- $b = 2,5 \text{ mm}$.

As peças que formam a célula de carga, fabricadas a partir do dimensionamento feito, são mostradas na Figura 2-6.

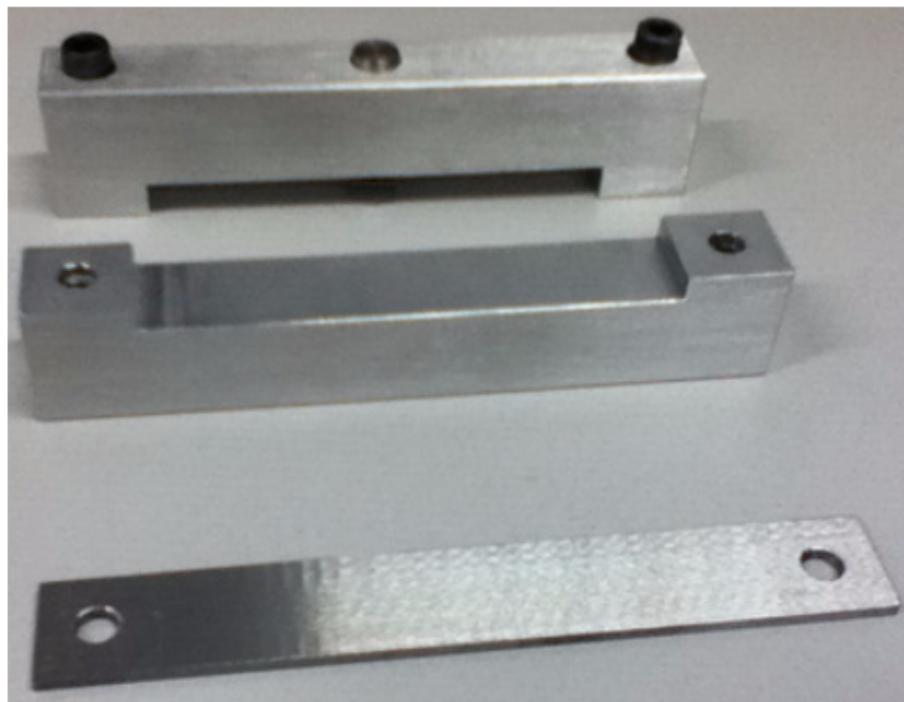


Figura 2-6 As três peças principais da célula de carga já fabricadas [6].

2.1.3. Montagem da célula de carga

Conforme já foi dito, o transdutor utilizado no corpo de prova neste trabalho foi o extensômetro (do inglês, “Strain-Gauge”), cuja estrutura pode ser observada na Figura 2-7.

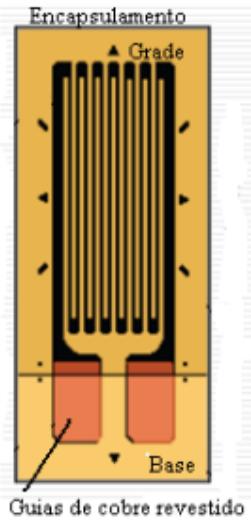


Figura 2-7 Estrutura do extensômetro [6].

Para entender como este componente funciona, imagine sua estrutura como um fio metálico de comprimento original “L” e resistência elétrica “R”; sabendo-se que a fórmula da resistência elétrica de um material é dada pela equação (3), com a deformação de seu corpo, haverá uma variação no comprimento do fio igual a “ $L+\Delta L$ ”, e portanto sua resistência também variará para “ $R+\Delta R$ ”.

$$R = \rho \times \frac{L}{A} \quad (3)$$

Os extensômetros utilizados no corpo de prova da célula de carga, são do modelo “PA-06-125AA-350-EN” do fabricante Excel Sensores, cujo código pode ser interpretado conforme a Figura 2-8.

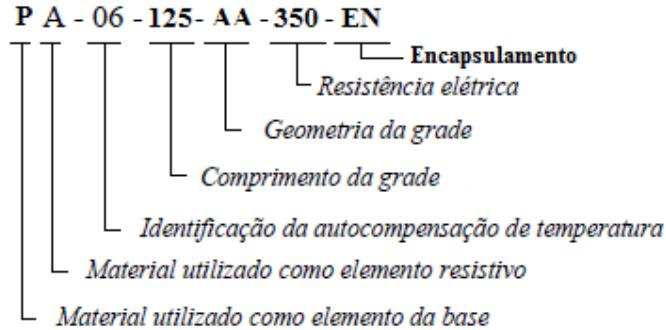


Figura 2-8 Explicação do código padronizado, extensômetro Excel [6].

Para definir a posição dos extensômetros sobre o corpo da célula de carga, foi calculado o momento ao longo da peça, considerando o ponto central como o local onde a força será aplicada, como mostra a Figura 2-9, onde “P” é a força máxima aplicada e “L” é o comprimento da peça.

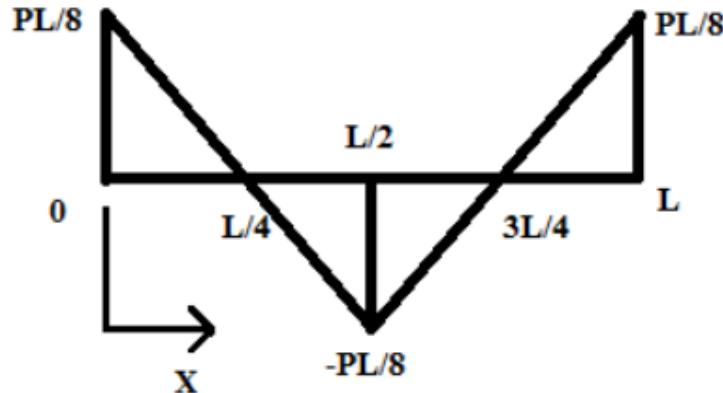


Figura 2-9 Perfil do momento ao longo do corpo de prova sob força de preensão [6].

Levando-se em conta a montagem e a segurança do equipamento, o melhor local onde a deflexão da peça atinge um valor mensurável foi definido como sendo a 1cm da origem do sistema, como mostra a Figura 2-10.

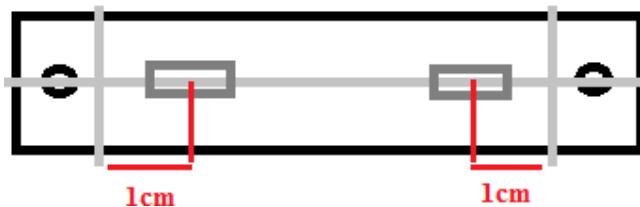


Figura 2-10 Posicionamento dos extensômetros sobre o corpo da célula de carga [6].

Por fim, os extensômetros foram fixados sobre a peça a ser deformada (Figura 2-11), as três peças que compõe a célula de carga forma montadas (Figura 2-12) e os acabamentos para melhor acomodação da mão sobre a manopla foram acrescentados (Figura 2-13).



Figura 2-11 Extensômetros fixados ao corpo de prova da célula de carga [6].

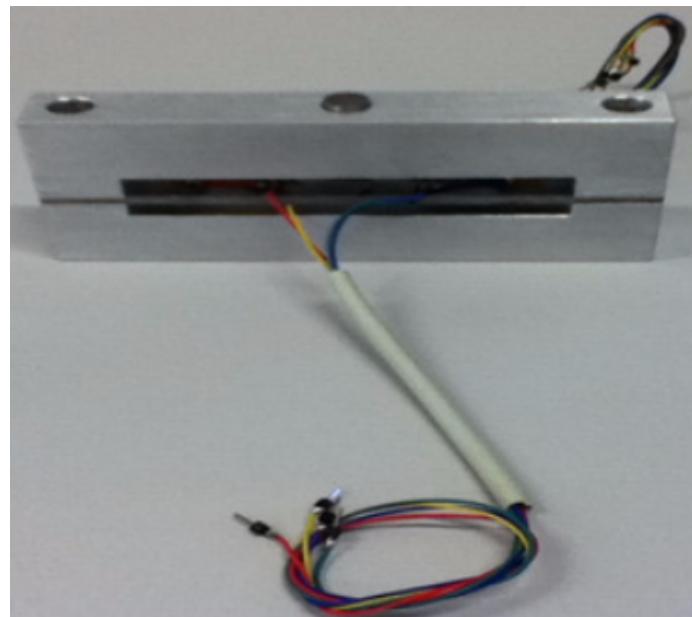


Figura 2-12 Célula de carga semi-finalizada [6].



Figura 2-13 Célula de carga finalizada [6].

2.2. Circuito Elétrico

Com a estrutura da célula de carga em mãos, o próximo passo do trabalho foi elaborar o circuito elétrico responsável por tornar possível a leitura analógica da Força de Prensão. Para tal, este circuito é constituído por:

- Regulador de tensão;
- Extensômetros conectados na configuração de Ponte de Wheatstone, colados sobre a barra metálica que sofrerá a deformação de forma a melhor captar o esforço da FPP;
- Amplificador de Instrumentação;
- Filtro;
- Circuito de ajuste de offset;
- Amplificador Inversor;
- Amplificador Não-Inversor.

Para reduzir a propensão de ruído observado ao se implementar a amplificação do sinal em apenas um estágio, fez-se necessário dividir este processo em três estágios: a amplificação de instrumentação, a amplificação inversora e a amplificação não-inversora. Além disso, considerando-se que a tensão máxima permitida pela placa de aquisição de dados utilizada é de 5V, os estágios de amplificação foram projetados para que o sinal atingisse no máximo a metade do valor limite permitido.

O circuito final pode ser observado na Figura 2-14, representado pelo software LTspice.

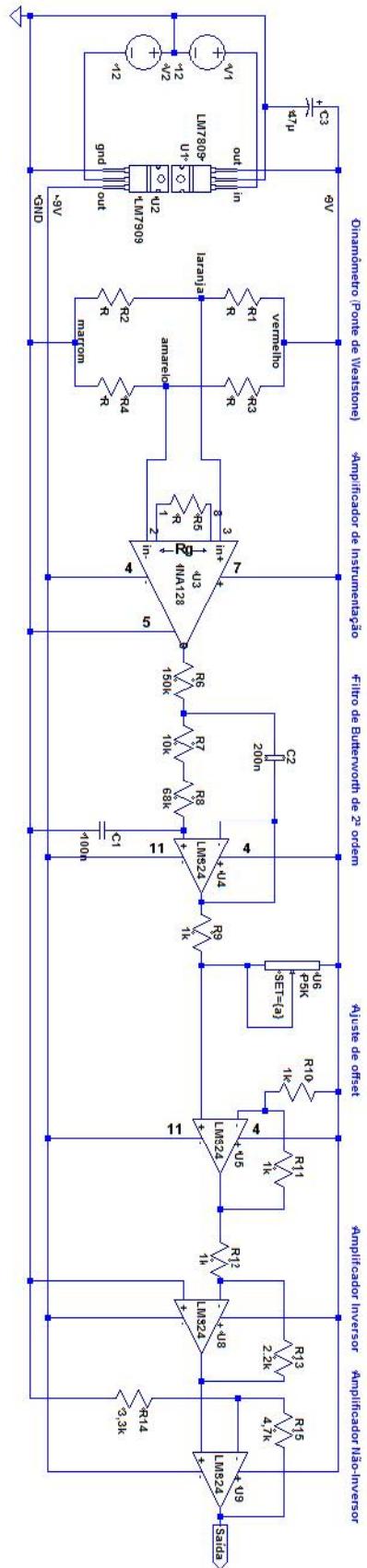


Figura 2-14 Circuito Elétrico final representado por LTspice.

2.2.1. Regulador de tensão

Primeiramente, para alimentar todo o circuito elétrico, foi feito um regulador de tensão a partir dos reguladores LM7809 e LM7909, para fixar as tensões em +9Volts e -9Volts a partir da alimentação por baterias de 12Volts. E para evitar a queda da tensão na linha +9Volts, foi conectado um capacitor eletrolítico de 47uF.

2.2.2. Extensômetros em Ponte de Wheatstone

Os extensômetros foram fixados sobre o corpo de prova, e conectados em configuração de ponte de Wheatstone (Figura 2-15), teoricamente, equilibrada; ou seja, alimentando-se a ponte com tensões contínuas de $\pm 9V$, na saída há uma tensão nula. Ao aplicar uma força sobre a célula de carga, a resistência dos extensômetros varia, desequilibrando a ponte de Wheatstone e causando uma pequena variação na tensão de saída, a qual será amplificada e tratada para torná-la mensurável para a análise da FPP.

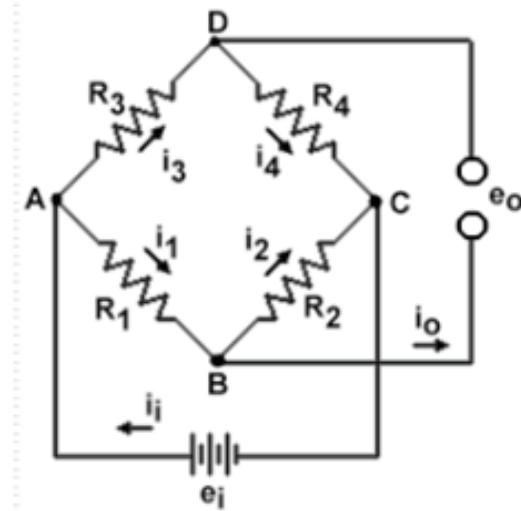


Figura 2-15 Ponte de Wheatstone

2.2.3. Amplificador de Instrumentação

Foi escolhido o amplificador de Instrumentação no primeiro estágio do circuito elétrico de tratamento do sinal por este agregar características como elevado CMR (*Common Mode Rejection*), elevada impedância nas entradas e baixo nível de offset.

O modelo usado foi o INA128, com resistência de ganho de 150Ω (R_G), fornecendo então um ganho de 334,34 V/V, segundo a equação (4) de ganho fornecida pelo *datasheet* do componente.

$$G = 1 + \frac{50k\Omega}{R_G} \quad (4)$$

2.2.4. Filtro

A implementação do filtro tem o importante papel de eliminar a existência de ruído causado por interferência externa. Levando em consideração o fato de que este sistema trabalhará com sinais com característica de corrente contínua, o filtro escolhido trabalhará com banda de passagem de baixa freqüência (10Hz).

Para isso, foi escolhido o filtro ativo de Butterworth de 2^a ordem por este não possuir *ripple*, ou ondulações na banda de passagem, e se aproximar a zero na banda rejeitada. Em sua montagem foi utilizado o componente LM 324.

2.2.5. Ajuste de Offset

Feito a pré-amplificação e a filtragem do sinal, foi feito um circuito de ajuste de *offset*, eliminando qualquer nível DC para que o valor do sinal seja zero enquanto não houver força aplicada à célula de carga. Para tal foi utilizado um amplificador LM324 em configuração como subtrator e ganho unitário; e para regular a tensão de *offset*, foi utilizado um *trimpot* de $5k\Omega$.

2.2.6. Amplificador Inversor

Devido à configuração construída da célula de carga, os pulsos do sinal gerado são negativos, porém a plataforma utilizada para a aquisição de dados (Arduino) não suporta tensões negativas em sua porta de entrada analógica, logo foi também adicionado um estágio de amplificador inversor. Para tal, foi também utilizado o LM324, um resistor de $1k\Omega$ e um de $2,2k\Omega$ em sua configuração de amplificador inversor com ganho de -2,2V/V.

2.2.7. Amplificador Não-Inversor

Por fim, foi feito um amplificador não-inversor como último estágio, utilizando-se do LM324, um resistor de $4,7\text{k}\Omega$ e um de $3,3\text{k}\Omega$, dando-lhe um ganho de $2,42\text{V/V}$. Para finalizar, foi colocado um diodo na saída do circuito para evitar qualquer pico negativo na porta de entrada do Arduino.

2.2.8. Montagem do Circuito Elétrico

O primeiro protótipo do Circuito Elétrico do projeto foi montado em um *protoboard*. Com o circuito montado foi realizado testes do tratamento do sinal, observando-se a saída com um osciloscópio, conforme mostra a Figura 2-16.

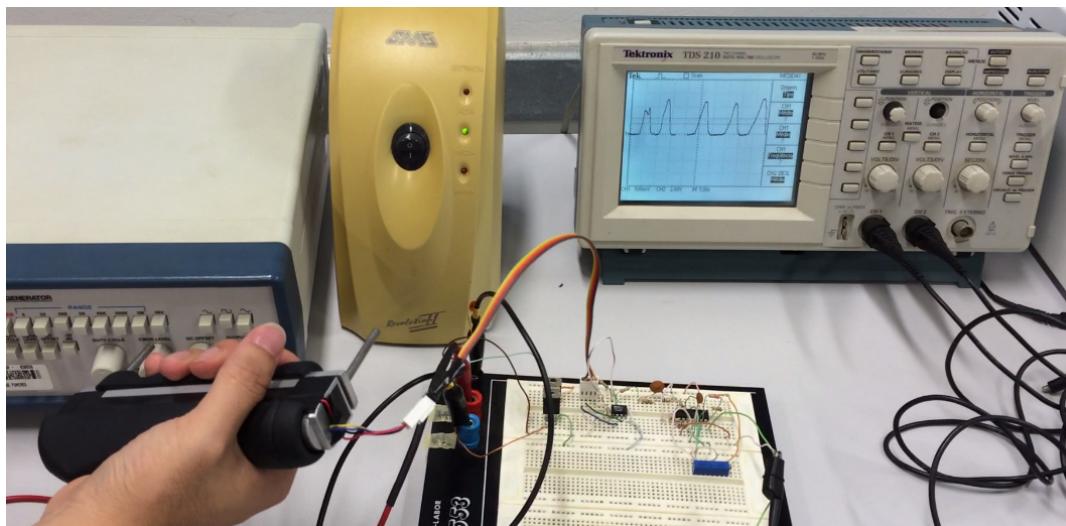


Figura 2-16 Teste de funcionamento do protótipo do circuito elétrico em protoboard.

Com o circuito funcionando dentro da condição exigida pela porta de entrada do Arduino, que é um sinal operando entre 0 e 5,0Volts, foi acondicionado o circuito na placa para montagem de protótipos, o qual foi fixada como parte do *hardware*, conforme mostra a Figura 2-17.

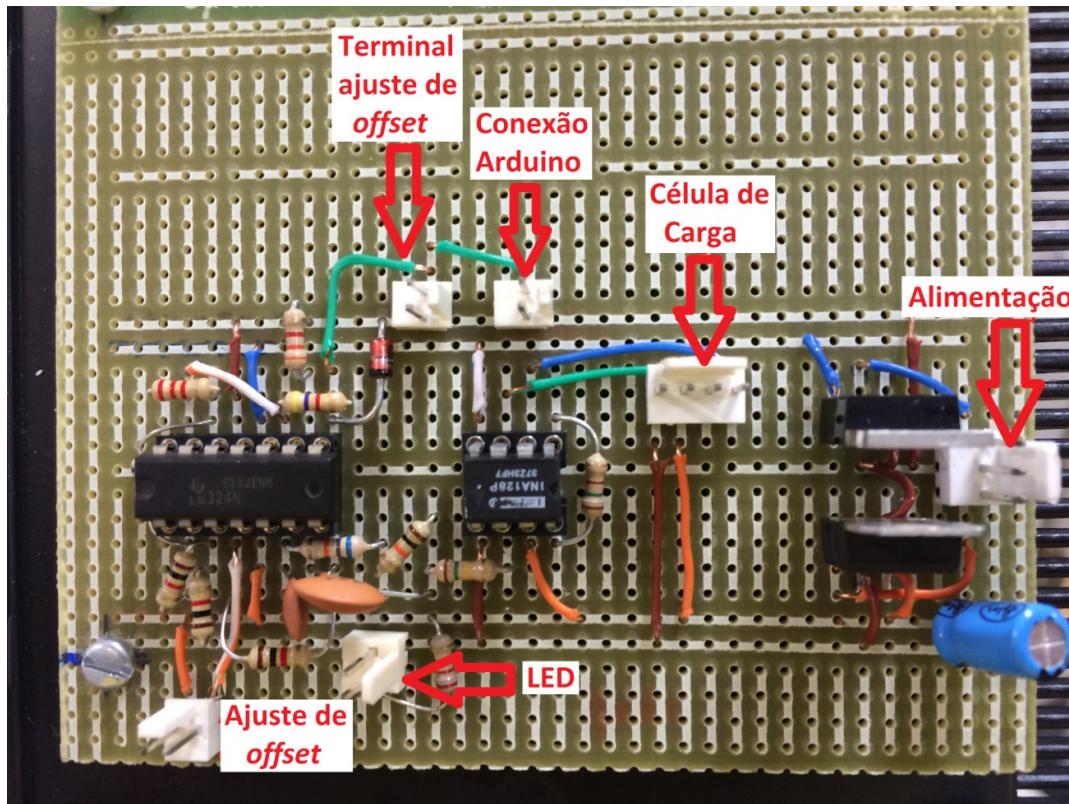


Figura 2-17 Circuito elétrico final para amplificação, filtragem e correção de offset do sinal.

2.2.9 Caracterização da célula de carga acoplada ao circuito elétrico

Para que seja possível converter o sinal em Volts para a unidade de força em Newtons, foi necessário levantar a curva característica da célula de carga juntamente com o circuito elétrico de amplificação. Para realizar esse levantamento, foi utilizado um dinamômetro (equipamento que aplica e indica a força exercida sobre a superfície desejada), conforme a Figura 2-18.



Figura 2-18 Posicionamento do dinamômetro sobre a célula de carga para a coleta das medidas.

Foram levantadas cinco séries independentes de medidas, tanto do ato de compressão, comprimindo a célula de carga sempre no sentido crescente de força aplicada, quanto de descompressão, descomprimindo a célula de carga no sentido decrescente de força aplicada. Uma das séries de compressão e descompressão pode ser observada na Tabela 2-1. Com as tabelas de dados estabelecidas, as linhas de tendência foram obtidas de cada curva (Tabela 2-2).

Tabela 2-1 Medições referentes à primeira série do ato de compressão e descompressão da célula de carga, acoplada ao circuito elétrico de tratamento do sinal.

Compressão		Descompressão	
Força [N]	Tensão [V]	Força [N]	Tensão [V]
0	0,005	0	-0,02
10	0,224	10,9	0,073
20,4	0,27	20,3	0,074
30,3	0,344	30	0,076
40,8	0,422	40,3	0,135
51,4	0,509	50,6	0,208
60,3	0,593	60,3	0,288
70,2	0,693	71,3	0,419
80,3	0,791	80,2	0,521
90,1	0,893	89,5	0,65
100,5	1,007	100,3	0,789
110,9	1,115	109,8	0,928
120,8	1,23	120,7	1,103
130,1	1,343	129,7	1,201
140,4	1,468	140,1	1,442
150,8	1,6	150	1,623
160,4	1,752	160,8	1,703
170,5	1,93	170,8	2,005
180,3	2,057	180	2,153
190,2	2,188	190,4	2,263
199,06	2,316	199,06	2,316

Por fim, fazendo-se a média aritmética, estimou-se a função linear que mais se aproxima do comportamento da célula de carga juntamente com o circuito de tratamento do sinal, como pode ser observado na Tabela 5-2.

Tabela 2-2 Coeficiente angular e linear das linhas de tendência das curvas levantadas das cinco séries de medições da célula de carga acoplada ao circuito elétrico de tratamento do sinal.

Curva de tendência				
	Equação	Coef. Angular	Coef. Linear	R ²
Compressão	1	91,386	0	0,9861
	2	93,709	0	0,9706
	3	93,69	0	0,9688
	4	92,329	0	0,981
	5	89,049	0	0,9861
Descompressão	1	93,301	0	0,8724
	2	92,486	0	0,89
	3	92,315	0	0,876
	4	92,836	0	0,8864
	5	89,427	0	0,9109
Média				
Final		92,0528	0	

Observe que em todos os casos o coeficiente linear se iguala a zero, isto acontece porque cada linha de tendência foi forçada a cruzar o eixo das ordenadas na origem, pois enquanto a célula de carga não sofrer força de preensão a saída deve ser zerada. Com isso, tem-se como sendo a equação linear característica do equipamento:

$$y = 92,0528 x \quad (5)$$

A curva referente ao comportamento da primeira série de compressão da célula de carga e a linha de tendência encontrada podem ser observadas na Figura 2-19.

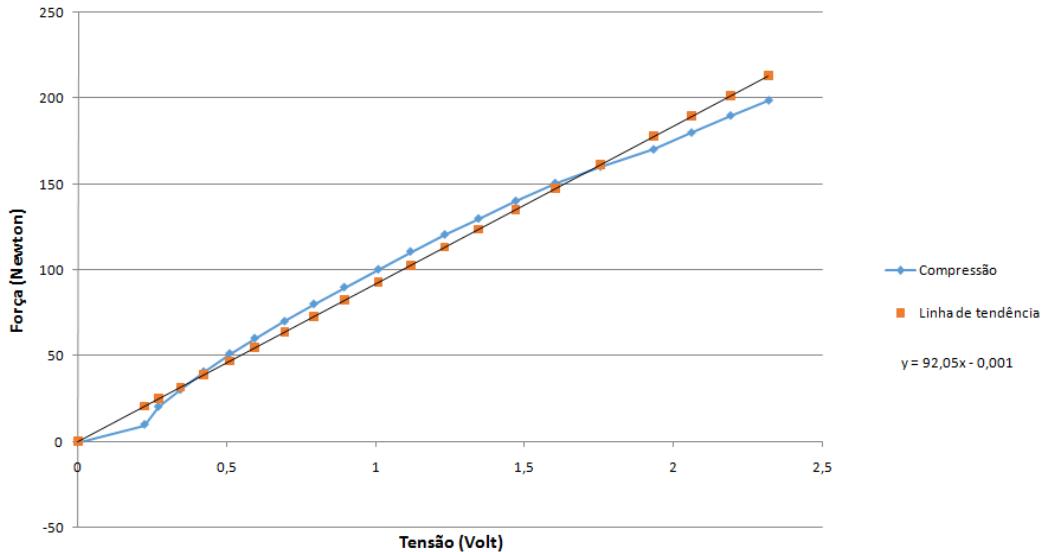


Figura 2-19 Curva referente ao comportamento da primeira série de compressão da célula de carga com o circuito elétrico, e a linha de tendência final encontrada.

2.3. Arduino

Arduino é uma plataforma baseada em uma simples placa microcontroladora, e uma área de trabalho para desenvolvimento de softwares para configurar segundo as necessidades desejadas do projeto em desenvolvimento.

O Arduino pode ser usado para desenvolver objetos interativos, leitura de entrada de uma variedade de chaves e sensores, e controle de luzes, motores e outras saídas físicas. Projetos com Arduino podem funcionar sozinhos, ou podem comunicar com softwares operando em um computador.

A linguagem de programação do Arduino tem origem em *Wiring*, e é essencialmente C/C++.

2.3.1. Configuração do Arduino para coleta de dados

Após o sinal analógico ser tratado pelo estágio de amplificação de instrumentação, o mesmo deve ser convertido para sinal digital de tal forma que seja possível ser enviado por uma conexão serial. Para a realização de tal tarefa foi escolhido o Arduino, uma plataforma baseada em uma placa microcontroladora ATmega328, tendo como características principais:

- Frequência de *clock* de 16MHz;
- 14 portas digitais de entrada ou saída;
- 6 entradas analógicas com Conversores Analógico/Digital de 10 bits cada;
- Memória *flash* de 32kB e EEPROM de 1kB;
- Conector USB utilizado para programação do microprocessador, comunicação serial e alimentação. (ARDUINO, 2015).

Para o desenvolvimento deste projeto, foi utilizado o Arduino modelo NANO (vide Figura 2-20), mas vale ressaltar que pode-se utilizar tanto o modelo NANO quanto o UNO do Arduino para operar este equipamento.

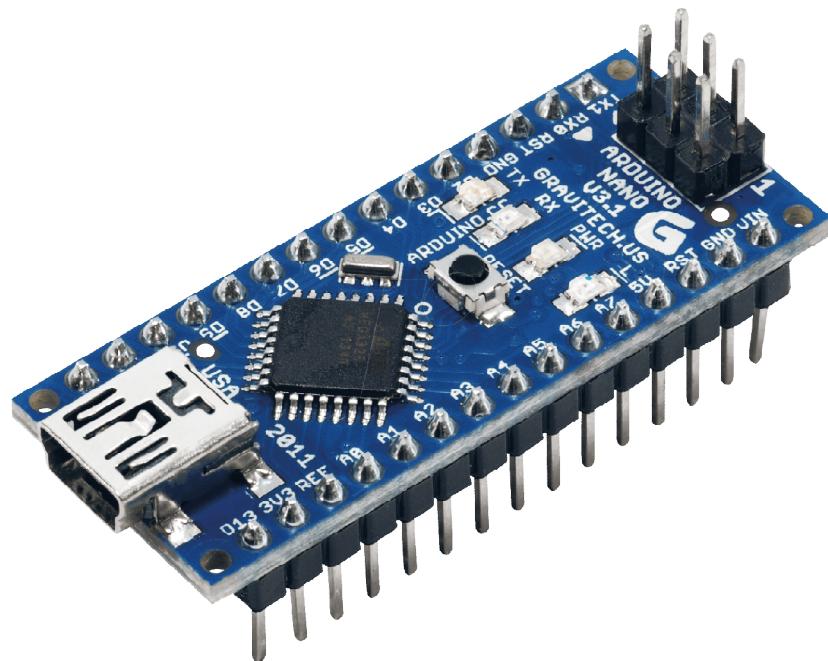


Figura 2-20 Arduino NANO

As vantagens de se trabalhar com o Arduino são as facilidades para acessar os seus periféricos, utilizadas na aquisição dos sinais deste trabalho; a comunicação via USB, já acoplada à placa, permitindo fácil e rápida programação do microcontrolador através de sua IDE (“*Integrated Development Environment*”, que significa Ambiente Integrado de Desenvolvimento), que pode ser encontrado no link de download de Software do Arduino (ARDUINO, 2015, *Download the Arduino Software*), mostrada na Figura 2-21; e a linguagem de programação feita em C/C++.

The screenshot shows the Arduino IDE interface. The title bar reads "ArduinoPythonPlotSaveDatah | Arduino 1.6.3". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for file operations like Open, Save, and Upload. The main area displays the following C++ code:

```
criado em 06/29/2015*/
int dado;           //Declara variavel inteiro 'dado'.
int FPP;           //Declara variavel inteiro 'FPP';
//FPP: Forca de Preensao Palmar.

void setup() {
  Serial.begin(9600);    //inicia a conexao serial.
}

void loop() {
  dado = analogRead(A0); //Atribui os valores lidos a variavel 'dado'.
  FPP = (dado*5.0/1023)*92.0528; //linha de tendencia com passagem na origem.
  Serial.print(FPP);        //Imprime os valores de FPP na comunicacao Serial.
  Serial.println();         //Imprime retorno (\n).

  delay(10);             //Da um espacamento de 10ms entre cada envio de dados.
}
```

The status bar at the bottom indicates "Arduino Nano, ATmega328 on COM3".

Figura 2-21 Ambiente Integrado de Desenvolvimento do Arduino.

Esse circuito recebe o sinal analógico como entrada em uma de suas portas de leitura analógica e o processa conforme o algoritmo nele programado, e por fim envia os dados por cabo serial USB a um computador.

Para facilitar o desenvolvimento dos algoritmos do projeto em um *protoboard*, de forma que não dependesse de todo o equipamento de laboratório (fonte de corrente contínua, o circuito de tratamento do sinal analógico e a célula de carga), utilizou-se um potenciômetro com os dois terminais extremos conectados ao terra e aos 5Volts de referência, e o terminal do cursor móvel conectado à porta de entrada analógica 0 (zero) do Arduino, conforme a Figura 2-22. Assim, é possível simular um sinal analógico na entrada do Arduino, variando a tensão entre zero e cinco volts.

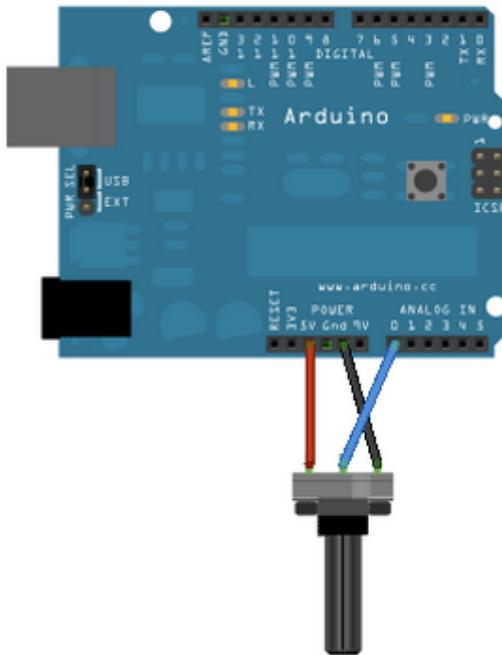


Figura 2-22 Configuração para simulação de teste e desenvolvimento do firmware.

2.3.2. Instruções do Arduino

A plataforma Arduino tem um circuito interno ADC (“*Analog to Digital Converter*”, que significa Conversor Analógico para Digital), que lê a variação de tensão na porta analógica e a converte em um índice entre 0 e 1023, pois a porta trabalha com 10 bits. Na linha de programação do Arduino, feita a leitura da entrada (valor atribuído à variável “*dado*”), converteu-se o índice para o valor da tensão lida, o qual foi substituído na variável “*x*” da equação (5) característica da célula de carga anteriormente levantada, obtendo-se então o valor convertido equivalente à força aplicada, conforme a equação (6).

$$FPP = \left(\frac{dado}{1023} \times 5 \right) \times 92,0528 \quad (6)$$

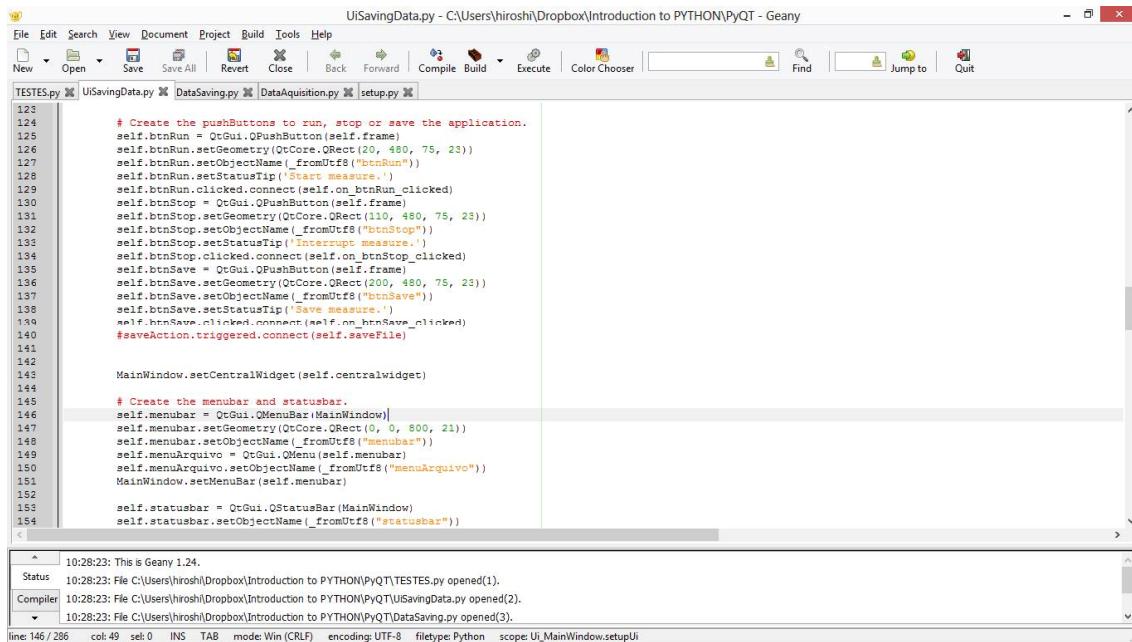
A instrução programada em C/C++ feita pode ser observada conforme o (Apêndice A – Instruções do Arduino).

2.4. Python

Para o desenvolvimento do Aplicativo do projeto, foi escolhida a linguagem de programação Python versão 2.7.9.

Python foi lançado por Guido Van Rossum em 1991, e atualmente é utilizado abertamente e gerenciado pela organização sem fins lucrativos *Python Software Foundation*; é uma linguagem orientada a objetos, funcional, interpretada e de alto nível. Uma de suas principais características é permitir a fácil leitura do código e exigir poucas linhas de código se comparado ao mesmo programa em outras linguagens.

Como ambiente de trabalho para desenvolver o aplicativo, foi utilizado o Geany IDE, um compilador que suporta diversos tipos de linguagem, encontrado em (Geany, 2015, *Download>>Releases*). O compilador pode ser observado na Figura 2-23.



The screenshot shows the Geany IDE interface with several tabs open: TESTES.py, UiSavingData.py, DataSaving.py, DataAquisition.py, and setup.py. The TESTES.py tab is active, displaying Python code for creating a GUI application using PyQt. The code includes definitions for push buttons (btnRun, btnStop, btnSave) and their corresponding slots (on_btnRun_clicked, on_btnStop_clicked, on_btnSave_clicked). It also sets up a menu bar and status bar. The status bar at the bottom shows log messages from the compiler, indicating files opened and actions taken. The code uses QtCore.QRect and QtGui.QPushButton for button geometry and creation.

```
123 # Create the pushButtons to run, stop or save the application.
124 self.btnExit = QtGui.QPushButton(self.frame)
125 self.btnExit.setGeometry(QtCore.QRect(20, 480, 75, 23))
126 self.btnExit.setObjectName(_fromUtf8("btnExit"))
127 self.btnExit.setStatusTip('Stop measure.')
128 self.btnExit.clicked.connect(self.on.btnExit_clicked)
129 self.btnExit = QtGui.QPushButton(self.frame)
130 self.btnExit.setGeometry(QtCore.QRect(110, 480, 75, 23))
131 self.btnExit.setObjectName(_fromUtf8("btnStop"))
132 self.btnExit.setStatusTip('Interrupt measure.')
133 self.btnExit.clicked.connect(self.on.btnExit_clicked)
134 self.btnExit = QtGui.QPushButton(self.frame)
135 self.btnExit.setGeometry(QtCore.QRect(200, 480, 75, 23))
136 self.btnExit.setObjectName(_fromUtf8("btnSave"))
137 self.btnSave.setStatusTip('Save measure.')
138 self.btnSave.clicked.connect(self.on.btnSave_clicked)
139 self.btnSave.clicked.connect(self.on.btnSave_clicked)
140 btnSaveAction.triggered.connect(self.btnSaveFile)
141
142 MainWindow.setCentralWidget(self.centralwidget)
143
144 # Create the menubar and statusbar.
145 self.menubar = QtGui.QMenuBar(MainWindow)
146 self.menubar.setGeometry(QtCore.QRect(0, 0, 800, 21))
147 self.menubar.setObjectName(_fromUtf8("menubar"))
148 self.menuArquivo = QtGui.QMenu(self.menubar)
149 self.menuArquivo.setObjectName(_fromUtf8("menuArquivo"))
150 Mainwindow.setMenuBar(self.menubar)
151
152 self.statusbar = QtGui.QStatusBar(MainWindow)
153 self.statusbar.setObjectName(_fromUtf8("statusbar"))

Status 10:28:23: This is Geany 1.24.
Status 10:28:23: File C:/Users/hiroshi/Dropbox/Introduction to PYTHON/PyQT/TESTES.py opened(1).
Compiler 10:28:23: File C:/Users/hiroshi/Dropbox/Introduction to PYTHON/PyQT/UiSavingData.py opened(2).
Compiler 10:28:23: File C:/Users/hiroshi/Dropbox/Introduction to PYTHON/PyQT/DataSaving.py opened(3).
line: 146 / 286 col: 49 sel: 0 INS TAB mode: Win (CRLF) encoding: UTF-8 filetype: Python scope: Ui_MainWindow.setupUi
```

Figura 2-23 Geany IDE usado para programar em Python.

2.4.1. Lógica da Interface Gráfica

Ao desenvolver a GUI (“*Graphical User Interface*”, que significa Interface Gráfica ao Usuário), a intenção foi deixar o mais simples possível para que o usuário possa utilizá-lo de maneira intuitiva.

Conforme mostra a Figura 2-24, ao iniciar o aplicativo, automaticamente, o programa deve se conectar ao equipamento plugado à entrada USB do computador;

feito isso o programa espera pelo comando do usuário. Ao pressionar o botão “Start”, o programa deve dar início à coleta de dados, plotando-os em tempo real no *chart* do aplicativo; se o botão “Stop” for pressionado durante a coleta de dados, o programa para a plotagem, caso contrário finalizará a coleta. Por fim, ao apertar o botão “Save”, abre-se a janela para salvar a medição realizada, tendo que preencher os dados do paciente corretamente, salvando-se a sessão de diagnóstico em um arquivo Excel (.xlsx).

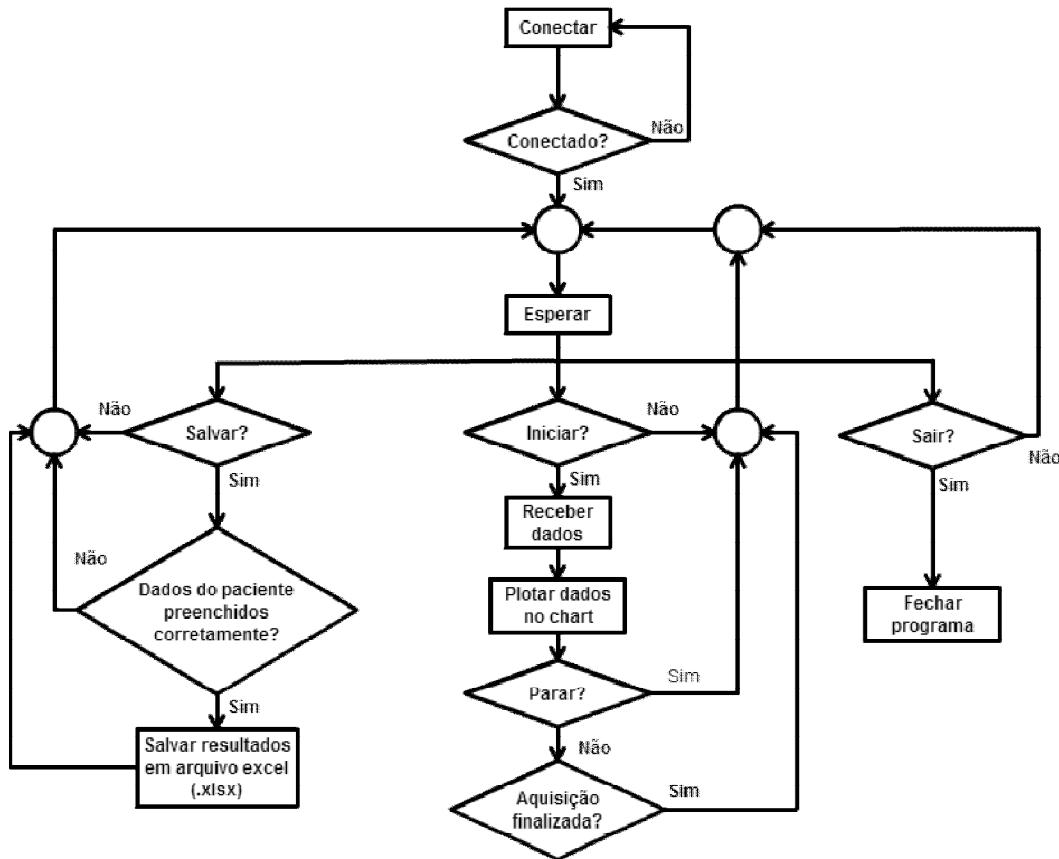


Figura 2-24 Diagrama de blocos da lógica da interface gráfica.

2.4.2. Instruções do GUI

Para desenvolver o aplicativo, primeiramente foram escritas e testadas, independentemente, as linhas de programa relativas às funções principais, que são:

- conectar automaticamente à porta USB do equipamento;
- iniciar a leitura de dados e desenhar o gráfico com taxa de amostragem de 100 pontos por segundo;
- parar ou finalizar a coleta de dados;

- salvar a sessão em um arquivo Excel (.xlsx).

Para testar a primeira função, ao se detectar a conexão USB, na janela do *command prompt* (Prompt de comando) imprimiu-se o nome da porta encontrada, como mostra a Figura 2-25.

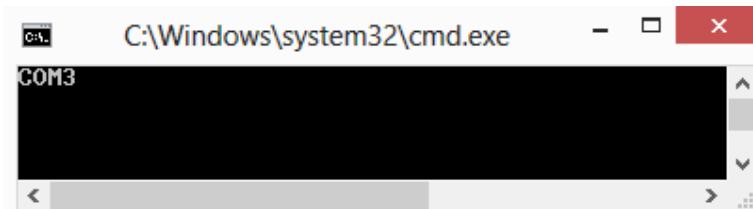


Figura 2-25 Teste de detecção e conexão automática da porta USB do equipamento.

No teste da impressão do gráfico em tempo real, utilizando a biblioteca *Matplotlib* do python, foi criado um *chart* só para plotagem, como pode ser observado na Figura 2-26.

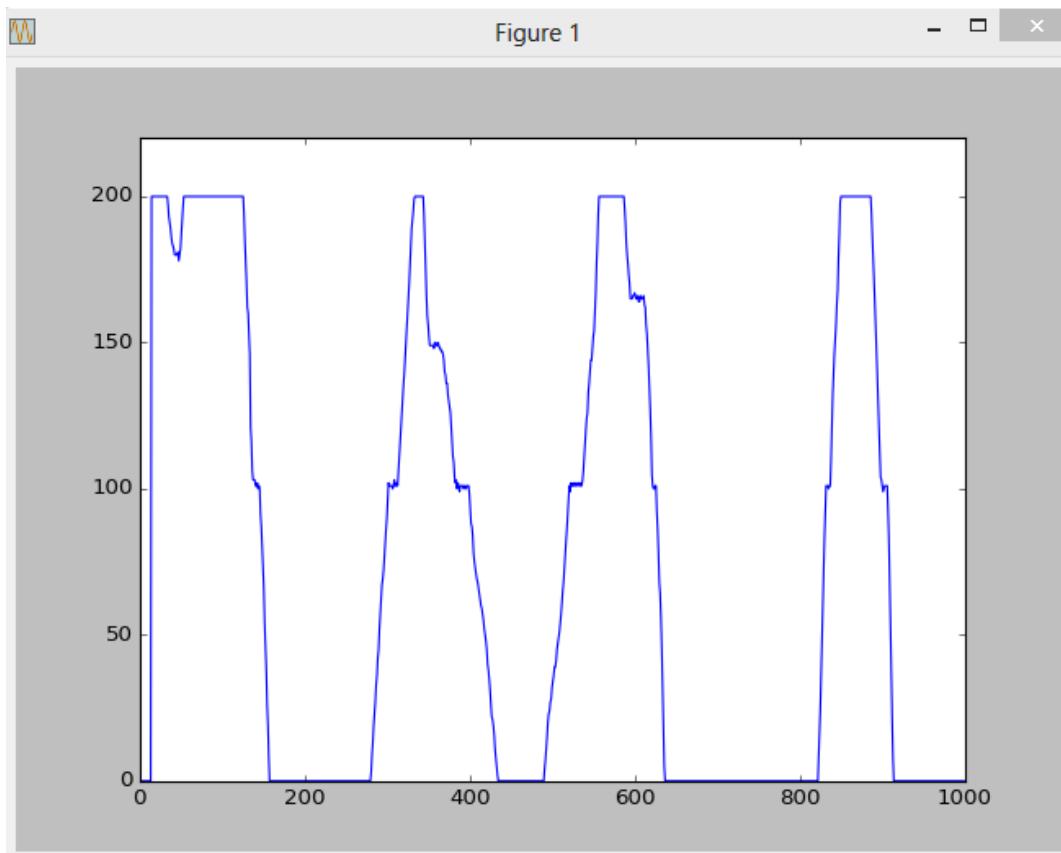


Figura 2-26 Chart gerado pelo teste da função de plotagem do gráfico.

Usando a mesma sessão de teste do gráfico em tempo real, testou-se a classe responsável por salvar os dados em arquivo Excel (.xlsx). O arquivo gerado deve conter as informações pessoais do paciente, uma coluna com o índice de cada ponto de dado e outra para o valor da força (em Newtons) coletado, e o gráfico gerado, como pode ser observado na Figura 2-27.

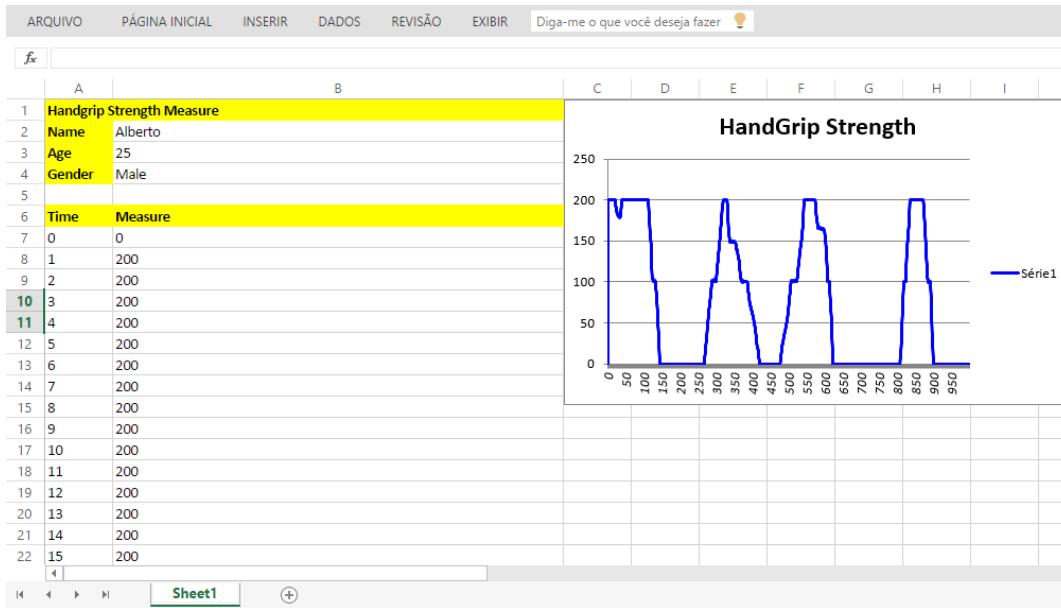


Figura 2-27 Arquivo Excel gerado no teste da função para salvar a sessão de medição.

Tendo todas as principais funções do aplicativo funcionando, usando a biblioteca PyQt4, integrhou-se todas em uma interface gráfica, na qual pode-se inserir o nome e a idade do(a) paciente por meio de duas *lineEdit* (linha de edição de texto), selecionar o gênero (masculino ou feminino) por meio de um *comboBox* (caixa de combinação), e acionar cada ação (“Start”, “Stop” e “Save”).

O aplicativo foi desenvolvido utilizando-se palavras na língua inglesa devido a ausência das acentuações do português na biblioteca padrão do python, o que gerava erros de interpretação pelo programa.

A configuração da interface gráfica do aplicativo desenvolvido pode ser observada na Figura 2-28.

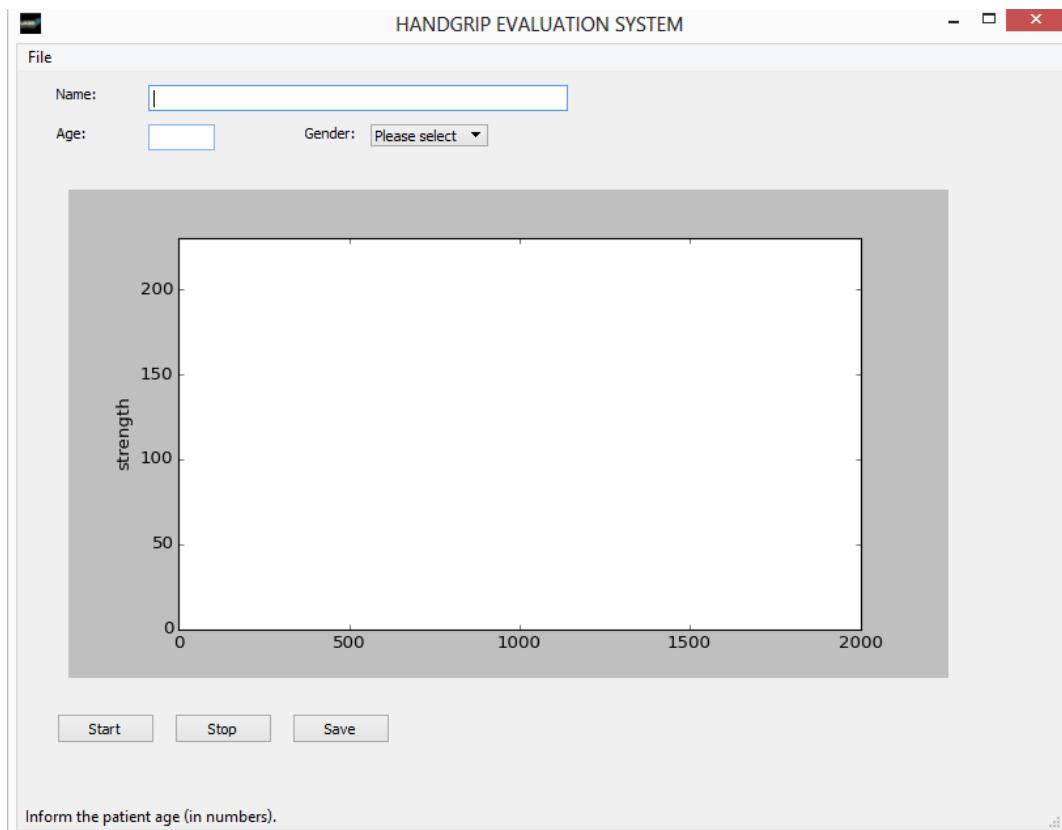


Figura 2-28 Configuração do GUI desenvolvida.

2.4.3. Teste do GUI

A última etapa do desenvolvimento da Interface Gráfica do projeto foi o teste de seu funcionamento. Simulou-se uma sessão de medição da força de preensão palmar, utilizando-se do equipamento completo.

Com o aplicativo já funcionando, foram preenchidos os espaços de dados do paciente, e foi feita a medição da FPP. O resultado da simulação pode ser observado na Figura 2-29.

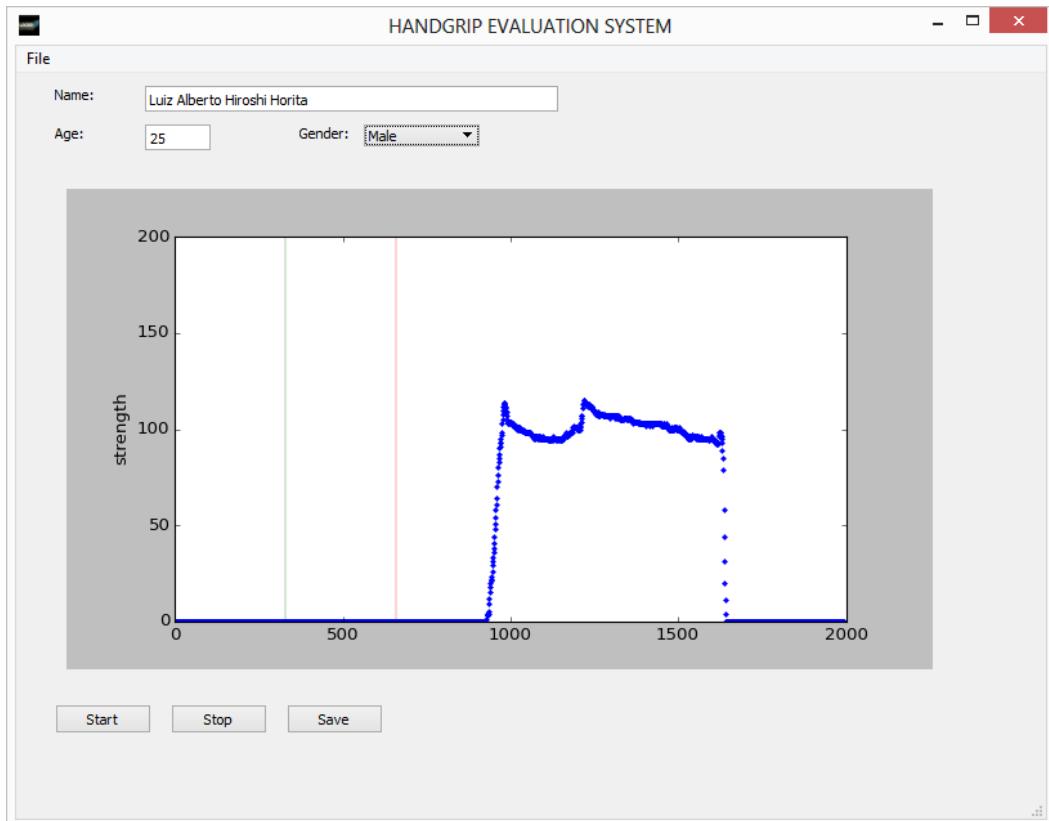


Figura 2-29 Interface Gráfica após o teste de funcionamento já com o hardware funcionando.

Terminado a coleta de dados da sessão, foi testada a função de salvar a medição em arquivo Excel. Ao pressionar o botão “Save”, a janela da função é aberta para nomear o arquivo e selecionar o diretório onde deve ser salvo, como mostra a Figura 2-30.

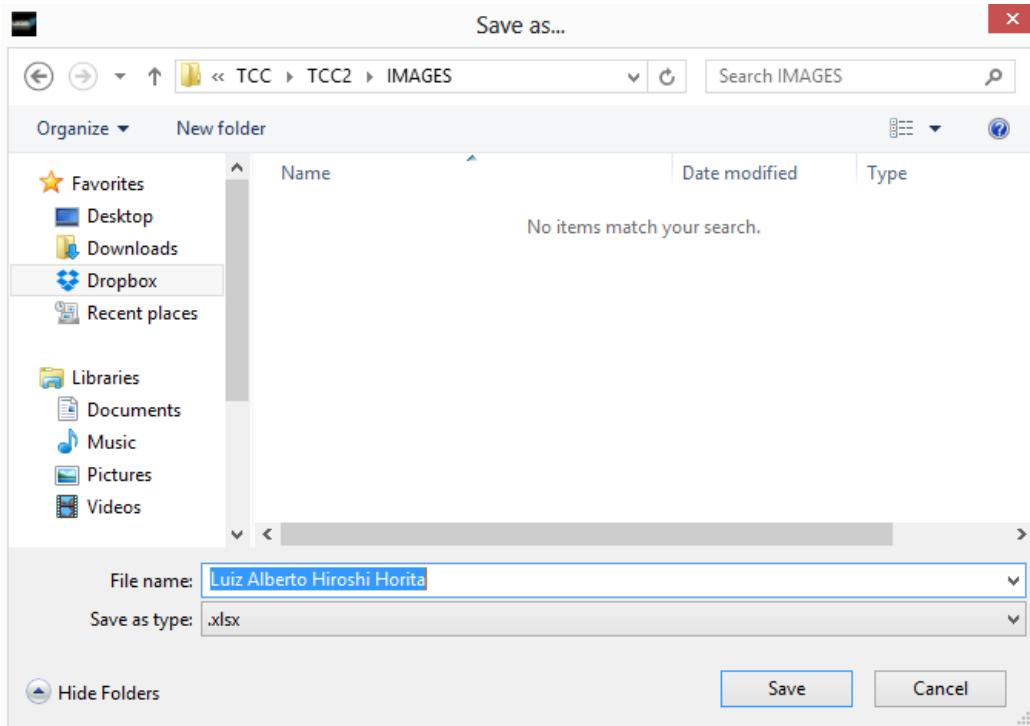


Figura 2-30 Janela exposta ao clicar no botão “Save”.

E finalmente, após salvar a sessão simulada, o arquivo gerado pode ser observado na Figura 2-31.

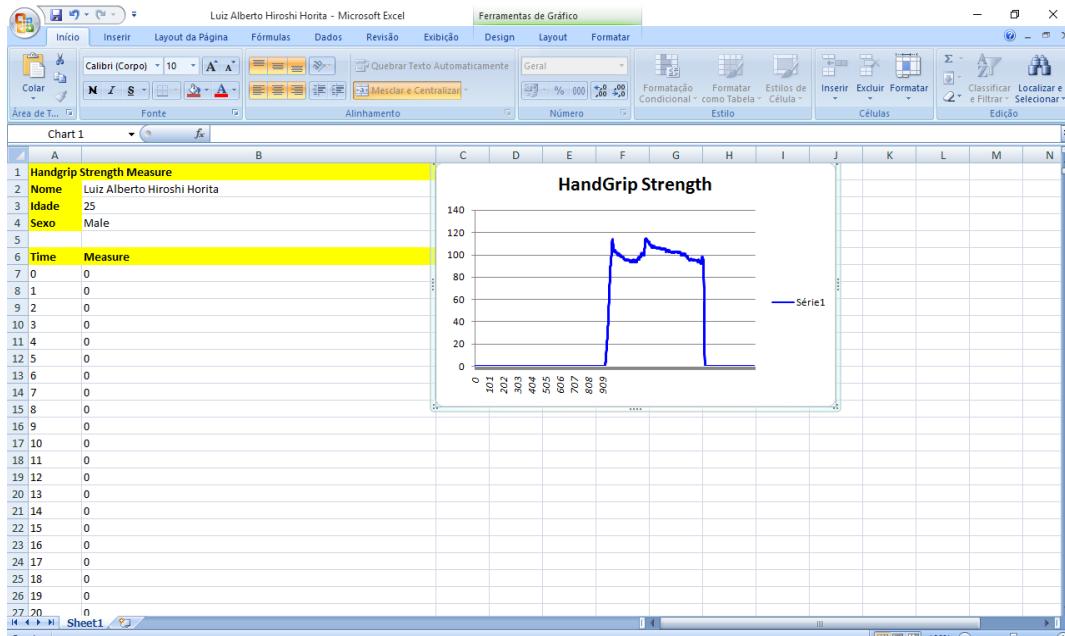


Figura 2-31 Arquivo Excel gerado e salvo no fim da sessão de medição da FPP.

3. Resultados e discussão

3.1. Hardware

Para a montagem do hardware, primeiramente foi feito um cabo com um terminal *molex* de um lado para se conectar à célula de carga, e um terminal *4 pin miniDIN*, para se conectar à caixa contendo o circuito elétrico (vide Figura 3-1).

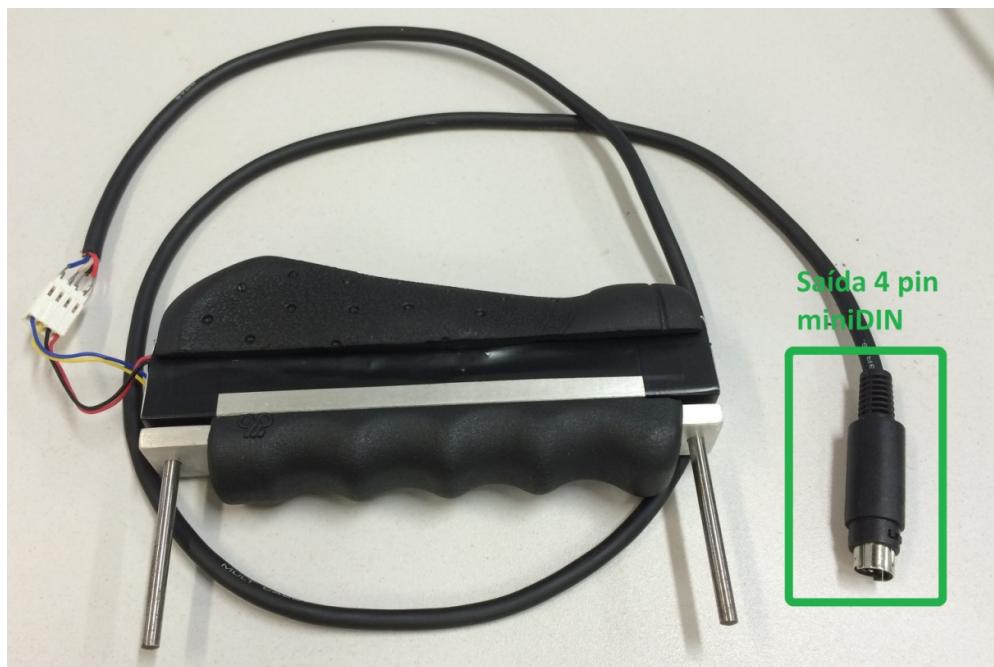


Figura 3-1 Célula de carga com o cabo de conexão.

Na parede frontal da caixa do hardware foi colocado um LED para indicar funcionamento do equipamento, o *trimpot* para ajuste de *offset*, o terminal *4 pin miniDIN* para conexão da célula de carga ao circuito elétrico e os terminais para monitoramento do ajuste de *offset*, conforme mostra as Figuras 3-2 e 3-3.



Figura 3-2 Vista frontal da caixa do hardware.

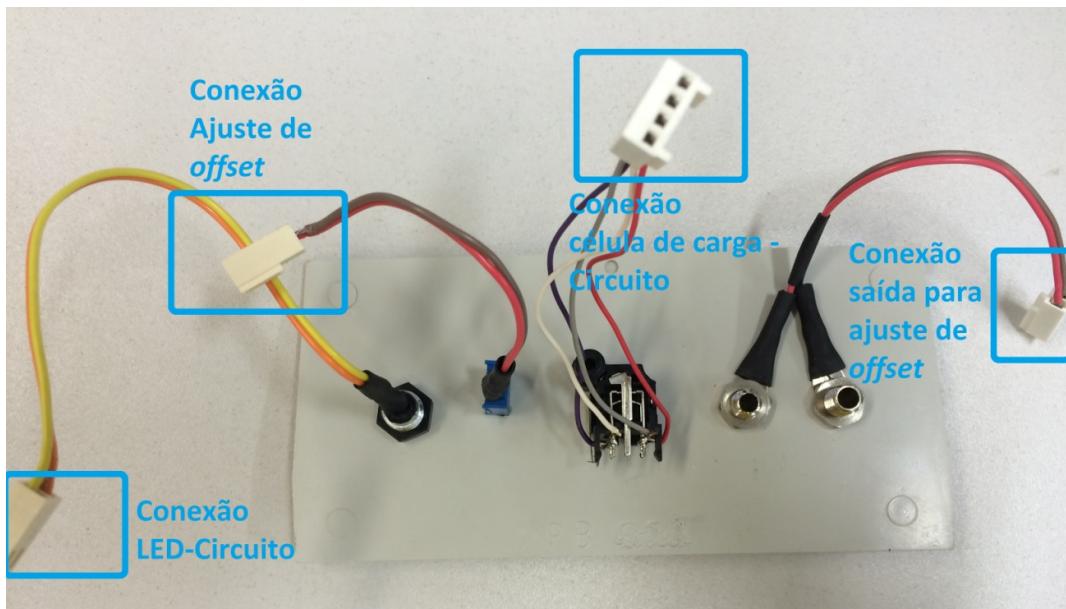


Figura 3-3 Parte interna da peça frontal da caixa do hardware.

No interior da caixa do hardware, o circuito elétrico de tratamento do sinal analógico foi fixado na parte inferior, e a placa Arduino posicionada ao lado, como mostra a Figura 3-4.

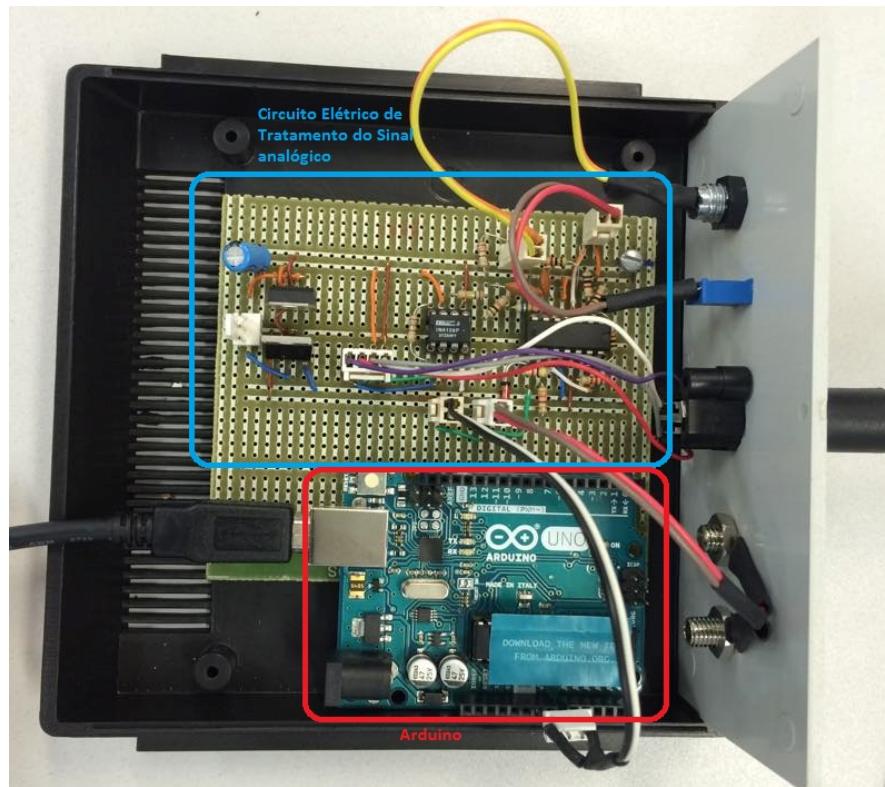


Figura 3-4 Interior da caixa do hardware.

Na parte traseira da caixa do hardware foram colocadas duas entradas para a alimentação do circuito, a chave para ligar e desligar o equipamento, e uma passagem para o cabo USB, conforme a Figura 3-5.



Figura 3-5 Vista traseira da caixa do hardware.

O configuração do hardware completo pode ser observada na Figura 3-6, e as baterias utilizadas para a alimentação do equipamento na Figura 3-7.

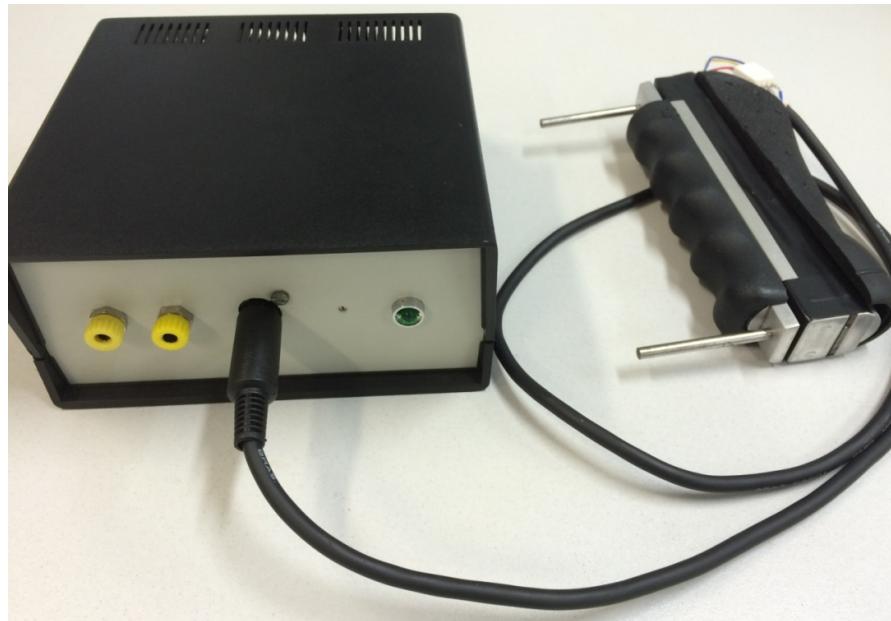


Figura 3-6 Hardware montado.



Figura 3-7 Baterias de 12V.

3.2. Caracterização do equipamento

Para a caracterização do equipamento, conectou-se o hardware ao computador e foram feitas cinco sessões de teste com aplicação de força constante de 0N, 50N, 100N, 150N e 200N sobre a célula de carga, utilizando-se um dinamômetro. Os testes podem ser observados nas Figuras 3-8, 3-9, 3-10, 3-11 e 3-12

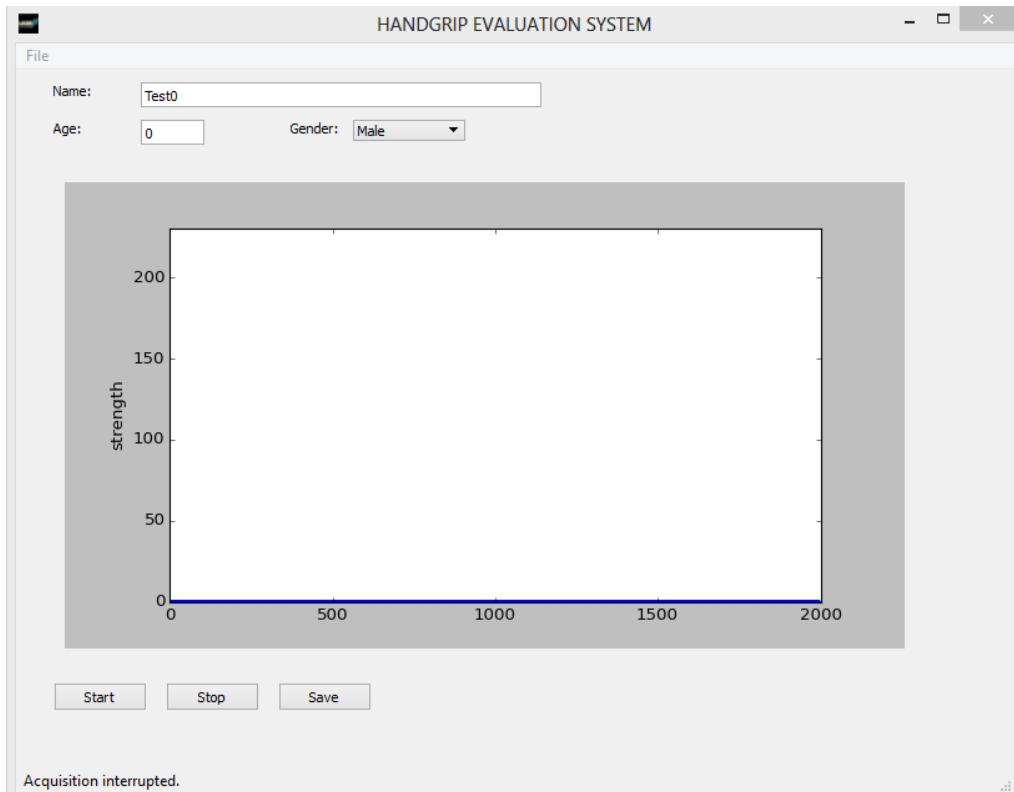


Figura 3-8 Teste de precisão com 0N aplicada à célula de carga.

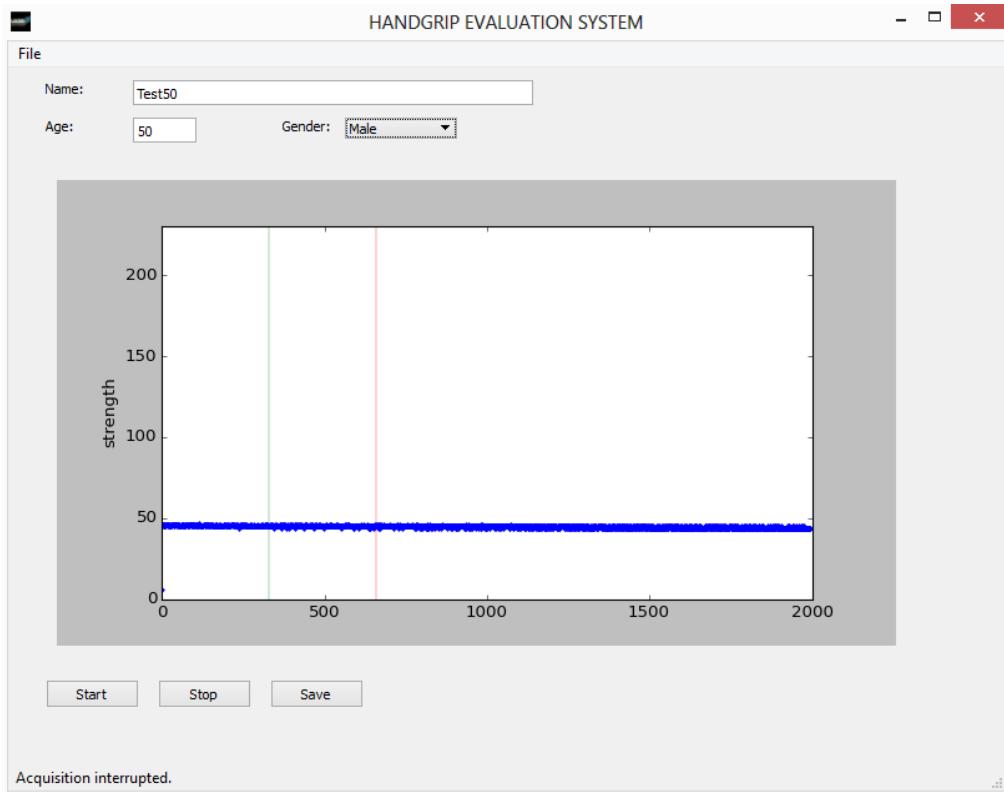


Figura 3-9 Teste de precisão com 50N aplicada à célula de carga.

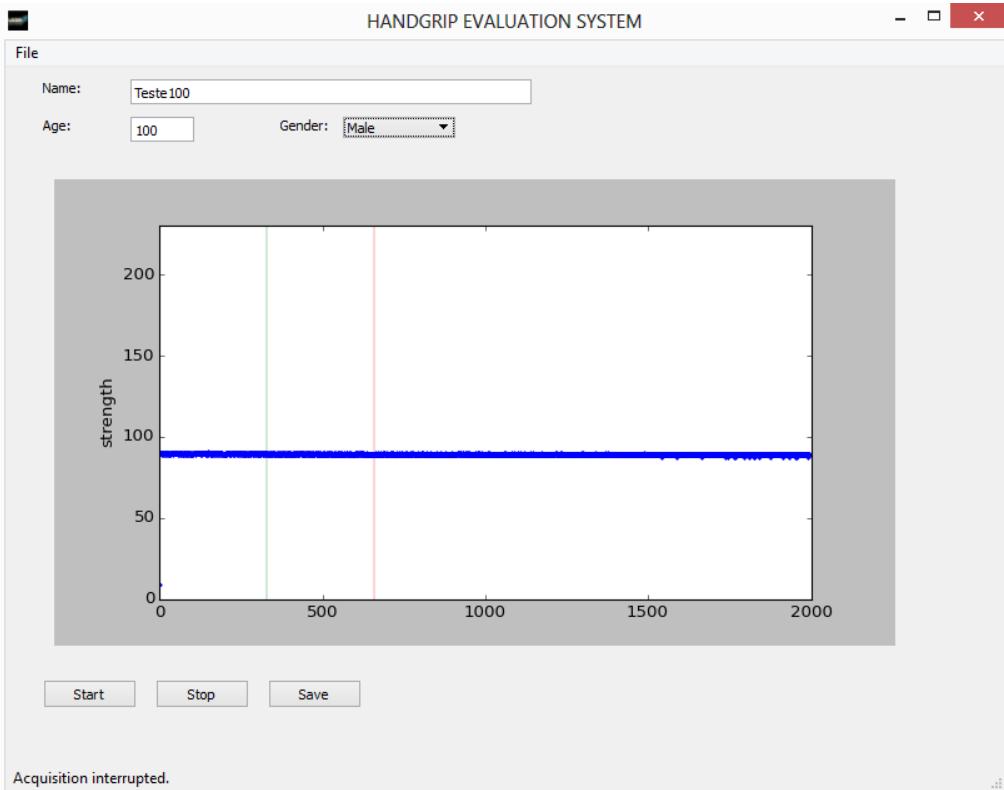


Figura 3-10 Teste de precisão com 100N aplicada à célula de carga.

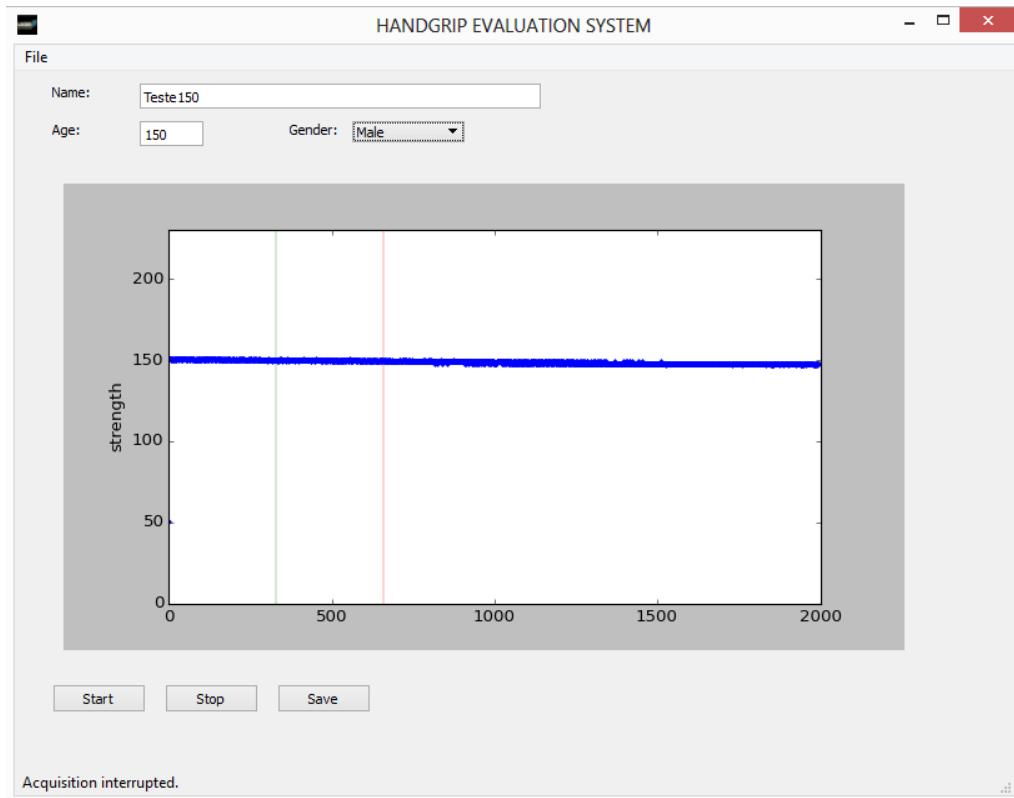


Figura 3-11 Teste de precisão com 150N aplicada à célula de carga.

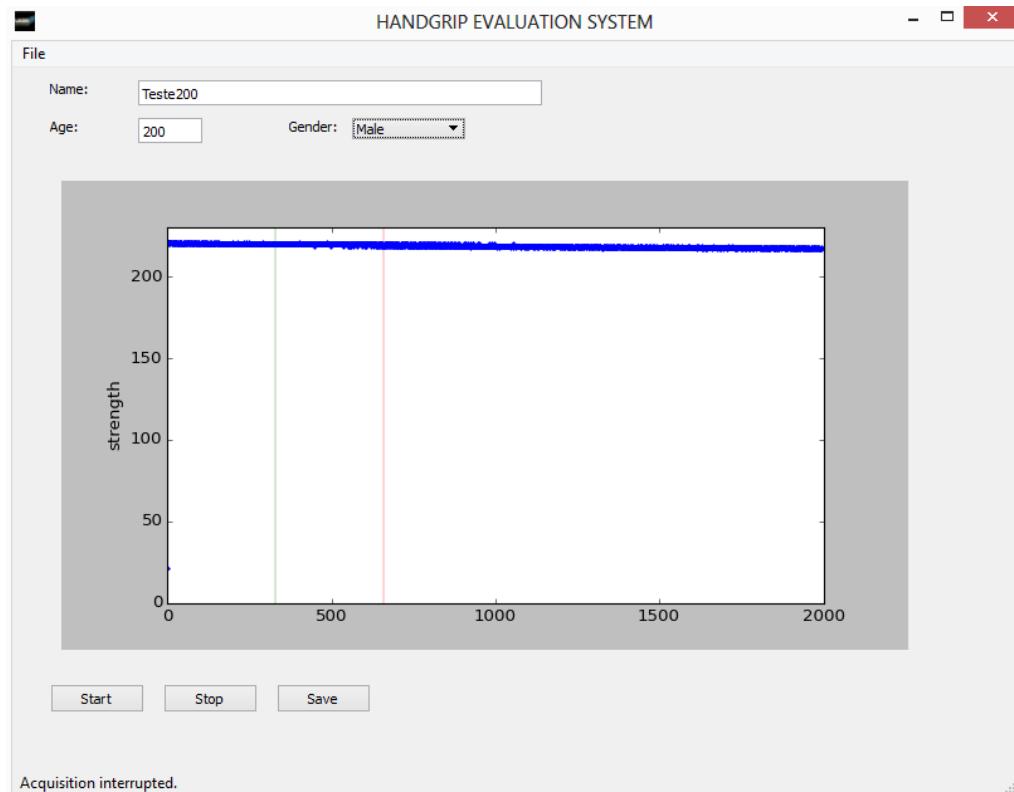


Figura 3-12 Teste de precisão com 200N aplicada à célula de carga.

Depois de realizadas as sessões de teste com força constante, a partir do arquivo Excel gerado de cada uma foram calculadas a média e o desvio padrão para a análise de precisão do equipamento. Os resultados levantados para cada sessão foram:

- 0N:	Média: 0;	Desvio Padrão: 0.
- 50N:	Média: 44,6886;	Desvio Padrão: 0,9196.
- 100N:	Média: 97,08246;	Desvio Padrão: 0,89233.
- 150N:	Média: 148,9165;	Desvio Padrão: 0,9917.
- 200N:	Média: 215,8755;	Desvio Padrão: 0,5423.

O comportamento observado nos testes realizados é condizente ao comportamento mostrado no gráfico levantado na Figura 2-19; a linha de tendência cruza com a curva de comportamento da célula de carga em dois pontos, próximos de 30N e 160N, sendo que entre esses pontos os valores da linha de tendência se mostram inferiores ao valores da curva de comportamento, e fora desse intervalo ocorre o oposto.

E ao desenhar as curvas relativas às medidas levantadas durante a compressão e a descompressão da célula de carga, percebe-se uma diferença de comportamento, criando uma imagem de curva de histerese, como pode ser observada na Figura 3-13.

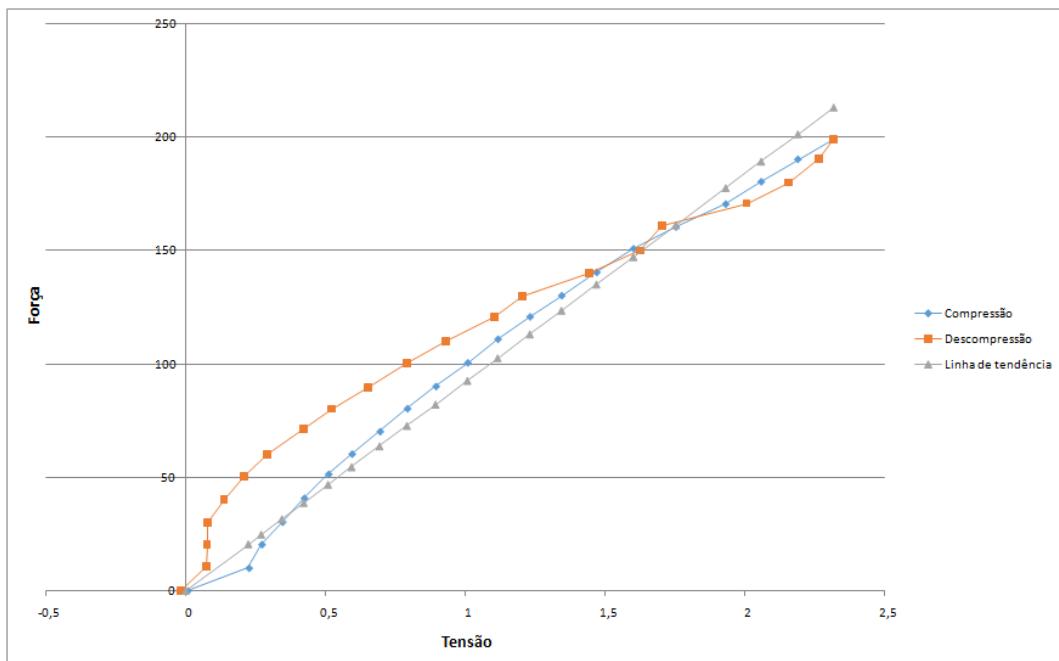


Figura 3-13 Curva de histerese do comportamento do equipamento durante compressão e descompressão da célula de carga e linha de tendência.

Isso ocorre pelo fato de o conjunto célula de carga não ter um comportamento perfeitamente linear.

Ao término deste trabalho, foram levantadas as seguintes características do sistema para avaliação da força de preensão palmar:

- Sensibilidade do equipamento: 10,86 mV/N;
- *Threshold* (valor mínimo indicado pelo equipamento): aumentando-se gradualmente a força aplicada sobre a célula de carga, observou-se que o menor valor de força pela qual o equipamento retornou uma resposta fora 1N;
- Faixa de valores de operação:
 - Força: 0 a 200N;
 - Tensão: 0 a 2,17V;
 - Níveis lógicos: 0 a 1023 (10bits);
- Geração de arquivo de dados: o sistema desenvolvido, ao executar o comando de salvar após a sessão de medição, um arquivo em formato Excel (.x/sx) é criado contendo os dados pessoais do paciente (nome, idade e sexo), cada ponto de medida coletado e um *chart* com o desenho do gráfico gerado; com isso, o equipamento tem a vantagem de criar um banco de dados para futura análise e estudo de comportamento da FPP.

Conclusão

É plausível dizer que este trabalho foi bem sucedido, embora apresente uma limitação. Como pôde ser observado, o equipamento se mostra exato apenas nas regiões de força próximas às intersecções da curva de comportamento da célula de carga com a curva da função de tendência estimada (em torno de 40N e 150N), devido ao comportamento não perfeitamente linear da célula de carga.

Por outro lado, o circuito de tratamento do sinal analógico teve um ótimo funcionamento, conseguindo amplificar, filtrar e corrigir o *offset*, com êxito, o sinal gerado na célula de carga.

A placa Arduino mostrou-se bastante eficiente e de fácil uso para a aquisição de dados e comunicação via cabo USB com o computador, o que possibilitou agilizar o desenvolvimento do projeto e dar maior tempo para o desenvolvimento da Interface Gráfica.

A maior parte deste projeto teve foco no desenvolvimento da Interface Gráfica, que obteve um resultado satisfatório, atendendo aos objetivos propostos, que eram a plotagem do gráfico em tempo real, espaço para entrada de dados pessoais do paciente e a capacidade de poder salvar as medições e os dados preenchidos em um único arquivo de dados, no formato Excel (.x/sx).

Comparando-se aos equipamentos que já existem, este tem as vantagens sobre o Jamar de poder expor em gráfico a medição feita e gerar um arquivo de dados, mas a desvantagem é que não possui ajustes da empunhadura (célula de carga); sobre o dinamômetro pneumático, este possui a vantagem de medir a força aplicada, e não a pressão, não dependendo do tamanho da mão do paciente para realizar a medição; e sobre o dinamômetro digital, este tem a vantagem de poder salvar o arquivo de dados do teste realizado.

Por fim, partindo deste trabalho, há oportunidades de desenvolvimento de novas versões deste equipamento, através de melhorias do mesmo (na construção da célula de carga, por exemplo), e inovações, como desenvolver um sistema *online*, no qual as clínicas que possuam esse equipamento possam salvar os arquivos de dados gerados na nuvem (*cloud*) do sistema e criar assim um banco de dados contendo maior espaço amostral para futuros estudos de comportamento da FPP para diagnósticos.

Anexo

Anexo A – Tabela de padrões de Força de Prensão Palmar em adultos.

Idade	Mão	Homem			Mulher		
		Média	DP	Intervalo	Média	DP	Intervalo
20-24	R	12.1	2.5	8.2-20.4	7.8	1.0	6.4-10.4
	L	11.7	2.6	6.8-19.1	7.4	1.3	5.0-10.9
25-29	R	11.8	2.0	8.6-15.9	8.0	1.5	5.9-13.2
	L	11.4	1.9	8.6-16.3	7.7	1.4	5.9-11.8
30-34	R	11.2	2.1	7.3-15.4	8.8	2.3	5.4-15.4
	L	11.5	2.6	6.8-16.8	8.2	2.2	5.4-14.5
35-39	R	11.9	1.9	8.6-16.3	7.9	1.9	5.9-13.2
	L	11.7	2.4	6.4-18.1	7.8	1.5	5.4-10.9
40-44	R	11.1	2.0	7.7-16.8	7.7	1.4	4.5-10.4
	L	11.2	2.2	6.8-16.8	7.5	1.6	4.5-11.3
45-49	R	10.9	1.5	8.6-15.0	8.1	1.4	5.4-12.2
	L	10.8	1.7	8.2-15.0	7.9	1.3	5.4-10.9
50-54	R	10.8	2.4	6.8-16.3	7.8	1.4	5.4-10.4
	L	10.9	2.6	7.3-16.3	7.4	1.3	5.4-10.0
55-59	R	10.8	2.2	7.3-15.4	7.3	1.4	5.0-11.8
	L	9.7	2.0	5.4-12.7	7.0	1.4	5.0-9.5
60-64	R	9.9	1.5	7.3-12.7	6.7	1.4	4.5-9.1
	L	9.6	1.5	6.8-12.2	6.5	1.2	4.5-9.1
65-69	R	9.7	1.4	6.8-11.3	6.4	1.4	3.6-9.1
	L	9.6	1.9	6.4-13.6	6.2	1.5	3.6-10.0
70-74	R	8.2	1.5	6.4-12.2	6.5	1.2	4.1-8.6
	L	8.5	1.5	5.9-12.2	6.4	0.9	4.5-7.7
75+	R	8.5	1.9	4.1-11.8	5.4	1.2	3.6-7.7
	L	8.3	1.7	4.5-11.8	5.2	1.2	2.7-7.3

Fonte: Equipamento para Medição de Força de Prensão Palmar, Universidade de São Paulo, 2012, Nunes; Bruno. Os dados são mostrados em unidade quilogramas (kg.).

Referências bibliográficas

- [1] Chris, D. (2015). *Hand Weakness (Weak Hand Grip) Causes and Normal Strength.* Retrieved 05 25, 2015, from HealthHype.com: www.healthhype.com/hand-weakness-weak-hand-grip-causes-and-normal-strength.html
- [2] COSTA, E. L., BASTOS Filho, P. S., MOURA, M. S., SOUSA, T. S., LEMOS, A., & PEDROSA, M. A. (2012, Julho). Efeitos de um programa de exercícios em grupo sobre a força de preensão manual em idosas com baixa massa óssea. *Arquivos Brasileiros de Endocrinologia & Metabologia*, 56 (5).
- [3] Haidar, S. G., Kumar, D., Bassi, R. S., & Deshmukh, S. C. (2004, Feb.). *Average versus maximum grip strength: which is more consistent?* Retrieved 11 04, 2015, from National Center for Biotechnology Information: www.ncbi.nlm.nih.gov/pubmed/14734079
- [4] MOURA, P. M. (2008). *ESTUDO DA FORÇA DE PREENSÃO PALMAR EM DIFERENTES FAIXAS ETÁRIAS DO DESENVOLVIMENTO HUMANO.* Universidade de Brasília, Faculdade de Ciências da Saúde, Brasília.
- [5] NASCIMENTO, M. F., BENASSI, R., CABOCLO, F. D., SALVADOR, A. C., & GONÇALVES, L. C. (2010). *Valores de referência de força de preensão manual em ambos os gêneros e diferentes grupos etários.* Centro Universitário Augusto Motta; Universidade Castelo Branco; Universidade Católica de Petrópolis; Laboratório de Bioquímica de Proteínas; UNIRIO., Rio de Janeiro, RJ.
- [6] NUNES, M. M., & BRUNO, N. B. (2012). *EQUIPAMENTO PARA MEDAÇÃO DE FORÇA DE PREENSÃO PALMAR.* Universidade de São Paulo, Escola de Engenharia de São Carlos, São Carlos.
- [7] RODRIGUES, H. L., OLIVEIRA, F. B., & COSTA, R. P. (2012). *MENSURAÇÃO DA FORÇA DE PREENSÃO PALMAR EM ATLETAS DE JIU-JITSU.* UEG, ESEFFEGO, Anápolis, GO.
- [8] SHIRATORI, A. P., IOP, R. R., BORGES Jr, N. G., DOMENECH, S. C., & GEVAERD, M. S. (2014). Protocolos de avaliação da força de preensão manual em indivíduos com artrite reumatóide: uma revisão sistemática. *Revista Brasileira de Reumatologia*. , 54 (2).
- [9] SILVA, C. C. (2013). *Força da preensão da mão como método de rastreio de desnutrição em crianças e adolescentes internadas.* Universidade do Porto, Faculdade de Ciências da Nutrição e Alimentação, Porto, Portugal.

Apêndices

Apêndice A – Instruções do Arduino

```
int dado;           // Declara a variável auxiliar para receber os dados.  
int FPP;          // Declara a variável que carregara o valor da FPP.  
  
void setup() {  
    Serial.begin(9600); // Inicia a conexão serial.  
}  
  
void loop() {  
    dado = analogRead(A0);      // dado recebe leitura da porta analógica A0.  
    FPP = (dado*5.0/1023)*92.0528; // FPP recebe o valor lido anteriormente  
                                //convertido para força de preensão.  
    Serial.print(FPP);         // Imprime valor da FPP na comunicação serial.  
    Serial.println();          // Imprime retorno (\n).  
    delay(10);                // Da um espaçamento de 10ms entre cada leitura.  
}
```

Apêndice B – Programação em Python – GUI

```
import time  
import matplotlib.pyplot as plt  
import numpy as np  
import xlsxwriter  
import serial  
import glob  
from matplotlib.backends.backend_qt4agg import FigureCanvasQTAgg  
from matplotlib.figure import Figure  
import sys  
from PyQt4 import QtGui, QtCore, uic  
from DataAquisition import Arduino  
from DataSaving import *  
  
try:  
    _fromUtf8 = QtCore.QString.fromUtf8  
except AttributeError:  
    def _fromUtf8(s):  
        return s  
  
try:  
    _encoding = QtGui.QApplication.UnicodeUTF8  
    def _translate(context, text, disambig):  
        return QtGui.QApplication.translate(context, text, disambig, _encoding)
```

```

except AttributeError:
    def _translate(context, text, disambig):
        return QtGui.QApplication.translate(context, text, disambig)

# Define o Matplot Widget para desenhar o gráfico.
class MatplotlibWidget(QtWidgets.QWidget):
    def __init__(self, parent=None):
        super(MatplotlibWidget, self).__init__(parent)

        self.figure = Figure()
        self.canvas = FigureCanvasQTAgg(self.figure)

        self.axis = self.figure.add_subplot(111)
        self.axis.set_xlabel('')
        self.axis.set_ylabel('strength')
        self.axis.set_xlim(0,2000)
        self.axis.set_ylim(0,230)
        self.axis.axvline(x = 330, linewidth=0.3, color='g')
        self.axis.axvline(x = 660, linewidth=0.3, color='r')

        self.layoutVertical = QtWidgets.QVBoxLayout(self)
        self.layoutVertical.addWidget(self.canvas)

    class Ui_MainWindow(object):
        def setupUi(self, MainWindow):
            self.Stop = True
            self.x = 0
            self.y = 0
            selfarduino = Arduino(serial_ports())
            self.timerDuino = QtCore.QTimer()
            self.timerDuino.timeout.connect(self.timerDuino_timeout)
            self.data = []
            self.index = []

            # Define o título e o tamanho padrão da janela principal.
            MainWindow.setObjectName(_fromUtf8("MainWindow"))
            MainWindow.resize(800, 600)
            self.icon = QtGui.QIcon()
            self.icon.addPixmap(QtGui.QPixmap(_fromUtf8("Capture2.png")),
                QtGui.QIcon.Normal, QtGui.QIcon.Off)
            MainWindow.setWindowIcon(self.icon) # Adiciona um ícone à janela principal.
            self.centralwidget = QtWidgets.QWidget(MainWindow)
            self.centralwidget.setObjectName(_fromUtf8("centralwidget"))

            # Cria um frame como o widget central do aplicativo.
            self.frame = QtGui.QFrame(self.centralwidget)
            self.frame.setGeometry(QtCore.QRect(10, 10, 741, 511))
            self.frame.setFrameShape(QtGui.QFrame.StyledPanel)
            self.frame.setFrameShadow(QtGui.QFrame.Raised)
            self.frame.setObjectName(_fromUtf8("frame"))

```

```

# Coloca um matplotlibWidget dentro do frame criado anteriormente.
self.matplotlibWidget = MatplotlibWidget(self.frame)
self.matplotlibWidget.setGeometry(QtCore.QRect(20, 71, 691, 391))
self.matplotlibWidget.setObjectName(_fromUtf8("matplotlibWidget"))

# Cria as linhas de edição.
self.nameEdit = QtGui.QLineEdit(self.frame)
self.nameEdit.setGeometry(QtCore.QRect(90, 0, 321, 20))
self.nameEdit.setText(_fromUtf8(""))
self.nameEdit.setObjectName(_fromUtf8("nameEdit"))
self.nameEdit.setStatusTip('Insert the complete name of the patient.')
self.ageEdit = QtGui.QLineEdit(self.frame)
self.ageEdit.setGeometry(QtCore.QRect(90, 30, 51, 20))
self.ageEdit.setObjectName(_fromUtf8("ageEdit"))
self.ageEdit.setStatusTip('Inform the patient age (in numbers).')

# Cria os labels de sinalização da janela.
self.label = QtGui.QLabel(self.frame)
self.label.setGeometry(QtCore.QRect(20, 0, 46, 13))
self.label.setObjectName(_fromUtf8("label"))
self.label_2 = QtGui.QLabel(self.frame)
self.label_2.setGeometry(QtCore.QRect(20, 30, 46, 13))
self.label_2.setObjectName(_fromUtf8("label_2"))
self.label_3 = QtGui.QLabel(self.frame)
self.label_3.setGeometry(QtCore.QRect(210, 30, 46, 13))
self.label_3.setObjectName(_fromUtf8("label_3"))

# Cria os comboBoxes para selecionar o gênero.
self.combo = QtGui.QComboBox(self.frame)
self.combo.addItem("Please select")
self.combo.addItem("Male")
self.combo.addItem("Female")
self.combo.setGeometry(QtCore.QRect(260, 30, 90, 17))
self.combo.activated[str].connect(self.onActivated)

# Cria os pushButtons para iniciar, parar e salvar.
self.btnRun = QtGui.QPushButton(self.frame)
self.btnRun.setGeometry(QtCore.QRect(20, 480, 75, 23))
self.btnRun.setObjectName(_fromUtf8("btnRun"))
self.btnRun.setStatusTip('Start measure.')
self.btnRun.clicked.connect(self.on_btnRun_clicked)
self.btnStop = QtGui.QPushButton(self.frame)
self.btnStop.setGeometry(QtCore.QRect(110, 480, 75, 23))
self.btnStop.setObjectName(_fromUtf8("btnStop"))
self.btnStop.setStatusTip('Interrupt measure.')
self.btnStop.clicked.connect(self.on_btnStop_clicked)
self.btnSave = QtGui.QPushButton(self.frame)
self.btnSave.setGeometry(QtCore.QRect(200, 480, 75, 23))
self.btnSave.setObjectName(_fromUtf8("btnSave"))
self.btnSave.setStatusTip('Save measure.')
self.btnSave.clicked.connect(self.on_btnSave_clicked)

```

```

MainWindow.setCentralWidget(self.centralwidget)

# Cria o menubar e a statusbar.
self.menubar = QtGui.QMenuBar(MainWindow)
self.menubar.setGeometry(QtCore.QRect(0, 0, 800, 21))
self.menubar.setObjectName(_fromUtf8("menubar"))
self.menuArquivo = QtGui.QMenu(self.menubar)
self.menuArquivo.setObjectName(_fromUtf8("menuArquivo"))
MainWindow.setMenuBar(self.menubar)

self.statusbar = QtGui.QStatusBar(MainWindow)
self.statusbar.setObjectName(_fromUtf8("statusbar"))
MainWindow.setStatusBar(self.statusbar)

self.actionSair = QtGui.QAction(MainWindow)
self.actionSair.setShortcutContext(QtCore.Qt.WindowShortcut)
self.actionSair.setObjectName(_fromUtf8("actionSair"))
self.menuArquivo.addAction(self.actionSair)
self.menubar.addAction(self.menuArquivo.menuAction())

self.retranslateUi(MainWindow)
QtCore.QMetaObject.connectSlotsByName(MainWindow)

def retranslateUi(self, MainWindow):
    MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow", None))
    self.label.setText(_translate("MainWindow", "Name:", None))
    self.label_2.setText(_translate("MainWindow", "Age:", None))
    self.label_3.setText(_translate("MainWindow", "Gender:", None))
    self.btnRun.setText(_translate("MainWindow", "Start", None))
    self.btnStop.setText(_translate("MainWindow", "Stop", None))
    self.btnSave.setText(_translate("MainWindow", "Save", None))
    self.menuArquivo.setTitle(_translate("MainWindow", "File", None))
    self.actionSair.setText(_translate("MainWindow", "Exit", None))
    self.actionSair.setShortcut(_translate("MainWindow", "Ctrl+Q", None))

def timerDuino_timeout(self):
    if self.x <= 2000 and self.Stop == False:
        self.index.append(self.x)
        self.x += 1
        self.data.append(self.arduino.GetData())
        if(self.x % 15 == 0):
            self.matplotlibWidget.axis.plot(self.index[-15:-1], self.data[-15:-1], 'b.')
            self.matplotlibWidget.canvas.draw()
    else:
        self.timerDuino.stop()
        self.statusbar.showMessage("Acquisition interrupted.")
        return
    self.statusbar.showMessage("End of acquisition.")

def onActivated(self, text): # Função que pega a palavra selecionada

```

```

try:
    self.word_gender = text # comboBox do gênero.
except AttributeError:
    self.word_gender = ""

@QtCore.pyqtSlot()
def on_btnRun_clicked(self):
    if self.Stop == False:
        return
    self.Stop = False
    self.x = 0
    self.y = 0
    self.index = []
    self.data = []
    self.samples = 0
    self.matplotlibWidget.axis.clear()
    self.matplotlibWidget.axis.set_xlabel("")
    self.matplotlibWidget.axis.set_ylabel('strength')
    self.matplotlibWidget.axis.set_xlim(0,2000)
    self.matplotlibWidget.axis.set_ylim(0,230)
    self.matplotlibWidget.axis.axvline(x = 330, linewidth=0.3, color='g')
    self.matplotlibWidget.axis.axvline(x = 660, linewidth=0.3, color='r')
    self.timerDuino.start(10)

@QtCore.pyqtSlot()
def on_btnStop_clicked(self):
    self.Stop = True
    self.statusbar.showMessage("Acquisition interrupted.")

@QtCore.pyqtSlot()
def on_btnSave_clicked(self):
    try:
        self.word_name = self.nameEdit.text()
        self.int_age = int(self.ageEdit.text())
        print self.nameEdit.text()
        print self.ageEdit.text()
        print self.word_gender
    except:
        self.statusbar.showMessage("Wrong parameters!")
    return

    self.fn = QtGui.QFileDialog.getSaveFileName(None, "Save as...", "", ".xlsx")
    self.title = self.fn
    print self.title
    self.excelFile = ExcelFile(str(self.title), str(self.word_name), self.int_age, str(self.word_gender), self.data, self.index)
    self.excelFile.SaveData()

    self.statusbar.showMessage("Saved %s.xlsx" % self.nameEdit.text(), msecs = 5000)

# Cria a janela principal chamando o objeto Ui_MainWindow.
class Main(QtGui.QMainWindow):

```

```

def __init__(self):
    QtGui.QMainWindow.__init__(self)
    self.ui = Ui_MainWindow()
    self.ui.setupUi(self)

# Encontra a porta serial.
def serial_ports():
    if sys.platform.startswith('win'):
        ports = ['COM%s' % (i + 1) for i in range(256)]
    else:
        raise EnvironmentError('Unsupported platform')

    result = []
    for port in ports:
        try:
            s = serial.Serial(port)
            s.close()
            result.append(port)
            porta = result[0]
        except (OSError, serial.SerialException):
            pass
    return porta

# Starts the application.
if __name__ == '__main__':
    print(serial_ports())
    app = QtGui.QApplication(sys.argv)
    window = Main()
    window.setWindowTitle('HANDGRIP EVALUATION SYSTEM')
    window.show()
    sys.exit(app.exec_())

```

Apêndice C – Programação em Python – Classe para leitura de dados

```

import serial
import random
import matplotlib.pyplot as plt
import numpy as np
import xlsxwriter

from matplotlib.backends.backend_qt4agg import FigureCanvasQTAgg
from matplotlib.figure import Figure
import sys
from PyQt4 import QtGui, QtCore, uic

class Arduino (object):
    def __init__(self, porta='COM3'):
        self.connected = False

```

```

self.ser = serial.Serial(porta, 9600) # ativa a conexao serial (certifique-se de que o baud rate esteja
sincronizado com o Arduino)

while not self.connected:
    self.serin = self.ser.read()
    self.connected = True

def GetData(self):
    try:
        self.data = self.ser.readline()
        self.value = int(self.data)
    except:
        self.value = 0
    return self.value

if __name__ == '__main__':
    arduino = Arduino()

    while (1):
        try:
            print arduino.GetData()
        except KeyboardInterrupt:
            break

```

Apêndice D – Programação em Python – Classe para salvar dados

```

import serial
import random
import matplotlib.pyplot as plt
import numpy as np
import xlsxwriter

from matplotlib.backends.backend_qt4agg import FigureCanvasQTAgg
from matplotlib.figure import Figure
import sys
from PyQt4 import QtGui, QtCore, uic

class ExcelFile (object):
    def __init__(self, title, name, age, gender, data, index):
        self.titulo = title
        self.nome = name
        self.idade = age
        self.sexo = gender
        self.dados = data
        self.indice = index

    def SaveData(self):
        # Cria um arquivo .xlsx chamado (self.titulo)

```

```

workbook = xlsxwriter.Workbook(self.titulo)
worksheet = workbook.add_worksheet()

# Adiciona formatos de caligrafia para as celulas do arquivo
bold = workbook.add_format({ # Formato em negrito e grifado em amarelo
    'bold': True,
    'align': 'left',
    'fg_color': 'yellow'})
data_format = workbook.add_format({ # Alinhamento dos dados a esquerda
    'align': 'left'})

# Formatacao do cabecalho do arquivo
worksheet.merge_range('A1:B1', 'Handgrip Strength Measure', bold) # Mescla as celulas A1 e B1
worksheet.write('A2', 'Nome', bold) # Indice da linha para o nome do paciente
worksheet.write('B2', self.nome, data_format) # Ex. de nome
worksheet.write('A3', 'Idade', bold) # Indice da linha para idade do paciente
worksheet.write('B3',self.idade, data_format) # Ex. de idade
worksheet.write('A4', 'Sexo', bold) # Indice da linha para sexo do paciente
worksheet.write('B4', self.sexo, data_format) # Ex. de genero sexual
worksheet.set_column('B:B', 60) # Aumenta o tamanho das celulas na coluna B do arquivo

# Adiciona os dados da medição ao arquivo excel
worksheet.write('A6', 'Time', bold) # Nomeia a coluna referente ao numero do ponto
worksheet.write('B6', 'Measure', bold) # Nomeia a coluna referente as medições
worksheet.write_column('B7', self.dados, data_format) # Adiciona os dados partindo da posição B7
worksheet.write_column('A7', self.indice, data_format) # Adiciona os índices dos pontos a partir de A7

# Cria um novo chart para a impressão do gráfico
chart = workbook.add_chart({'type': 'line'}) # Comando para criar um chart ao arquivo excel
chart.set_title({'name': 'HandGrip Strength'}) # Titula o gráfico de 'HandGrip Strength'

# Ajustes do eixo X do grafico
chart.set_x_axis({
    'name_font': {'size': 12, 'bold': True}, # Ajusta tamanho e formato da fonte de escrita
    'num_font': {'italic': True},
    'position_axis': 'on_tick'}) # Ajusta os pontos sobre as marcações do eixo

# Adiciona os pontos da serie sobre o chart
chart.add_series({
    'values': '=Sheet1!$B$7:$B$1006', # Seleciona os dados a serem colocados sobre o eixo Y
    'categories': '=Sheet1!$A$7:$A$2007', # Seleciona os dados a serem colocados sobre o eixo X
    'line': {'color': 'blue'}}) # Seleciona a cor do traco do grafico (Azul)

# Insere o chart ao worksheet do arquivo excel
worksheet.insert_chart('C1', chart)
workbook.close() # Fecha o arquivo.

if __name__ == '__main__':
    excelFile = ExcelFile()

```

