



Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Projeto Integrador de Engenharias 2

EasyKey: Chaveiro Automatizado

Brasília, DF

2017

EasyKey

EasyKey: Chaveiro Automatizado

Projeto de uma máquina CNC para copiar chaves feitas por alunos do curso de Projeto Integrador de Engenharias 2 da Universidade de Brasília

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Profº Alex Reis, Profº Paulo Meirelles, Profº Randher Viana, Profº
Sebastien Rondineau

Brasília, DF

2017

Integrantes

Integrante	Matrícula	Engenharia
Felipe Caixeta Pereira	13/0063541	Aeroespacial
João Paulo Lopes Ferreira	12/0014351	Automotiva
Laércio Silva De Sousa Júnior	13/0119105	Software
Luciano Henrique Nunes de Almeida	11/0063392	Software
Luis Henrique Nunes Guimarães	13/0122971	Software
Rodrigo Bonifácio de Medeiros	13/0132586	Eletrônica
Renato Cesar da Silva Agnello	12/0053896	Eletrônica
Samuel Angelo Dantas Rocha	14/0056980	Energia
Tayná Rodrigues Andrade	13/0017728	Energia
Victor Luis Quintarelli Bertolino	13/0136646	Energia
Vinícius Borges Brum	12/0023598	Automotiva
Vinícius Lisboa do Nascimento	11/0143256	Eletrônica

Lista de ilustrações

Figura 1 – EAP do projeto EasyKey. Fonte: Autores	13
Figura 2 – Organograma da equipe. Fonte: Autores	15
Figura 3 – Desenvolvimento do Subsistema do Projeto Estrutural. Fonte: Autores . . .	21
Figura 4 – Proposta de desenho estrutural inicial. Fonte: Autores	22
Figura 5 – Estrutura externa básica e base para sistema de movimentação. Fonte: Au- tores	22
Figura 6 – Sistema de trilhos e movimentação. Fonte: Autores	23
Figura 7 – Estrutura desenvolvida para armazenar demais subsistemas. Fonte: Autores	23
Figura 8 – Conjunto de aquisição de imagens. Fonte: Autores	24
Figura 9 – Desenho da estrutura básica completa. Fonte: Autores	25
Figura 10 – Simulação para Modal de Corpo Livre. Fonte: Autores.	25
Figura 11 – Frequências naturais para os 12 primeiros modos de vibração para Análise Modal de Corpo Livre. Fonte: Autores	25
Figura 12 – Simulação Estática Estrutural. Fonte: Autores	26
Figura 13 – Deformação para análise Estática. Fonte: Autores	26
Figura 14 – Estrutura externa após soldagem das barras externas da estrutura. Fonte: Autores	27
Figura 15 – Construção da base e fixação dos trilhos para movimentação. Fonte: Autores	27
Figura 16 – Fixação da base de trilhos na estrutura principal. Fonte: Autores	28
Figura 17 – Fixação dos motores de passo na estrutura. Fonte: Autores	28
Figura 18 – Fixação do moto esmeril na estrutura. Fonte: Autores	28
Figura 19 – Estrutura para aquisição de imagens fabricada por Impressora 3D. Fonte: Autores	29
Figura 20 – Carcaça com tampa aberta. Fonte: Autores	30
Figura 21 – Tampa. Fonte: Autores	30
Figura 22 – Suporte para fixação da chave virgem. Fonte: Autores	31
Figura 23 – Fonte 12V/20A. Fonte: Autores	32
Figura 24 – Fonte produzida. Fonte: Autores	32
Figura 25 – Motor acoplado aos eixos. Fonte: Autores	33
Figura 26 – Dimensões do motor. Fonte: (MITSUMI, 2017)	34
Figura 27 – Relação Torque x Velocidade. Fonte: (MITSUMI, 2017)	35
Figura 28 – Moto Esmeril de Bancada Lynus MAC-50, Fonte: Manual Lynus MAC-50 .	35
Figura 29 – Associação dos CIs L297 e L298 - Fonte: Datasheet L297	37
Figura 30 – Diagrama de blocos do driver - Fonte: Cartens, et. al., 2015	37
Figura 31 – Ponte H interna do L298 - Fonte: Cartens, et. al., 2015	38
Figura 32 – Simulação do Driver no Software Proteus - Fonte: Autores	39

Figura 33 – Layout PCB do <i>Driver</i> de Comando e Controle - Fonte: Autores	39
Figura 34 – Layout PCB do <i>Driver</i> de Comando e Controle - Fonte: Autores	40
Figura 35 – Layout PCB do <i>Driver</i> de Comando e Controle - Fonte: Autores	40
Figura 36 – Layout 3D PCB do <i>Driver</i> de Comando e Controle - Fonte: Autores	41
Figura 37 – <i>Driver</i> de Comando e Controle - Fonte: Autores	41
Figura 38 – Regulador de Velocidade - Fonte: Newton Braga	42
Figura 39 – Circuito Dimmer e Relé de Acionamento - Fonte: Autores	43
Figura 40 – Fluxo da aplicação e comunicação com outros subsistemas. Fonte: Autores	44
Figura 41 – Diagrama de sequência do EasyKey. Fonte: Autores	46
Figura 42 – Imagem tirada da PiCamera através da aplicação. Fonte: Autores	47
Figura 43 – Exemplo de reconhecimento do modelo da chave com template definido. Fonte: Autores	48
Figura 44 – Etapas do tratamento da imagem para aquisição de código G. Fonte: Autores	49
Figura 45 – Exemplo de pontos de referência para medida de escala. Fonte: Autores	50
Figura 46 – Exemplo de tela utilizando materialize. Fonte: Autores	51
Figura 47 – Tela de pagamento atual apenas com PayPal. Fonte: Autores	51
Figura 48 – Esquemático do circuito do sensor infravermelho. Fonte: Autores	53
Figura 49 – Sensor Infravermelho. Transmissor (azul) e receptor (preto). Fonte: Autores	53
Figura 50 – Modelos de chaves de fim de curso. Fonte: (MACHADO, 2012)	54
Figura 51 – Teste de movimentação dos eixos utilizando furadeira. Fonte: Autores	55
Figura 52 – Teste nos drivers. Fonte: Autores	56
Figura 53 – Teste do circuito Dimmer. Fonte: Autores	57
Figura 54 – Teste do sensor infravermelho. Fonte: Autores	57
Figura 55 – Imagem da simulação do código G no software CutViewer. Fonte: Autores	58
Figura 56 – Finalização da estrutura externa. Fonte: Autores	59

Lista de tabelas

Tabela 1 – Papéis dos membros	14
Tabela 2 – Responsabilidade das equipes	14
Tabela 3 – Orçamento de custo de mão de obra do projeto	16
Tabela 4 – Orçamento de custo dos materiais utilizados no projeto	17
Tabela 5 – Cronograma geral do projeto	18
Tabela 6 – Principais Materiais utilizados	26
Tabela 7 – Especificações do motor de passo	34
Tabela 8 – Especificações Técnicas Fresadora - Fonte: Manual Moto Esmeril Lynus MAC-50	36
Tabela 9 – Componetes Eletrônicos PCB	41
Tabela 10 – Funcionamento dos estados no MSP - Fonte: Autores	52

Sumário

1	INTRODUÇÃO	11
	Introdução	11
1.1	Contextualização	11
1.2	Objetivo Geral	11
1.3	Objetivos Específico	11
2	FASES DE CONCEPÇÃO	13
2.1	Estrutura Analítica do Projeto (EAP)	13
2.2	Descrição do Projeto	13
2.3	Propósito do projeto	13
2.4	Recursos Humanos	14
2.5	Restrições e Riscos	15
2.5.1	Restrições	15
2.5.2	Riscos	15
2.6	Estimativa de Custos	16
2.7	Cronogramas de atividades	18
2.8	Plano de comunicação	19
2.8.1	Reuniões Presenciais	19
2.8.2	Acompanhamento Externo	19
3	PRODUTO	21
3.1	Estrutura	21
3.1.1	Design Inicial	22
3.1.2	Aprimoramento/Refinamento	22
3.1.3	Simulações	24
3.1.4	Aquisição de Materiais	26
3.1.5	Construção	27
3.1.6	Acabamento	29
3.2	Sistema de Alimentação	31
3.2.1	Fonte de Alimentação	31
3.3	Motores	33
3.3.1	Motores de Passo	33
3.3.2	Máquina Fresadora	34
3.4	Sistema de Comando e Controle	36
3.4.1	Driver de Comando e Controle	36

3.4.2	Circuito Dimmer	42
3.5	Sistema de Ventilação	43
3.6	Painel de Comando	43
3.7	Sistema de Proteção	44
3.8	O Software	44
3.8.1	Fluxo da aplicação	44
3.8.2	Tecnologias utilizadas	44
3.8.3	Repositório	45
3.8.4	Licença	45
3.8.5	Diagrama de sequência	45
3.8.6	Funcionalidades desenvolvidas	45
3.8.6.1	Captura da imagem	45
3.8.6.2	Código de captura da imagem	46
3.8.6.3	Reconhecimento do modelo da chave	47
3.8.6.4	Tratamento de imagem	47
3.8.6.5	Conversão da imagem para código G	49
3.8.6.6	Interface com o usuário	50
3.8.6.7	Sistema de venda	50
3.8.6.8	Testes desenvolvidos	51
3.8.6.9	Comunicação com o MSP	51
3.9	Sistema Embarcado e Sensoriamento	52
3.9.1	Sistema Embarcado	52
3.9.1.1	MSP430 LaunchPad	52
3.9.1.2	Comunicação Serial	52
3.9.2	Sensores	52
3.9.2.1	Sensor Infravermelho	53
3.9.2.2	Chave de fim de curso	54
4	PLANO DE TESTES	55
4.1	Objetos de Teste	55
4.1.1	Estrutura	55
4.1.2	Sistema de Alimentação	56
4.1.3	Motores, Drivers de Comando e Controle e Circuito Dimmer	56
4.1.4	Sensoriamento	57
4.1.5	Sistemas de Software	58
5	INTEGRAÇÃO DOS SUBSISTEMAS	59
5.1	Integração dos subsistemas	59
6	CONSIDERAÇÕES FINAIS	61

7	AUTOAVALIAÇÃO	63
	REFERÊNCIAS	65
	APÊNDICES	67
	APÊNDICE A – DESENHOS TÉCNICOS ESTRUTURAIS	69

1 Introdução

1.1 Contextualização

O mercado de chaveiros no Brasil é abrangente, dinâmico e necessita de pessoal qualificado e apresenta demanda constante. A cópia de chaves é tarefa essencial e bastante solicitada à esses profissionais.

Atualmente, não são encontradas no Brasil máquinas de cópia automatizada de chaves simples. As copiadoras existentes em sua maioria são semi-automatizadas e necessitam da interferência manual de um operador durante o processo, passível de falhas ou erros de procedimento.

Este trabalho possibilitará atualização tecnológica do procedimento de cópia de chaves simples à baixo custo, evitando a interferência manual no equipamento. Ressalta-se que as chaves simples serão referência por serem bastante utilizadas no dia-a-dia, presentes em fechaduras de dispositivos comuns encontrados no Brasil.

1.2 Objetivo Geral

Construção de máquina CNC para realização de cópias de chaves de cadeados populares da marca PADO, de forma automatizada a partir do escaneamento chave original.

1.3 Objetivos Específico

Atingir o objetivo geral a partir da integração de subsistemas aplicados a cada engenharia da Universidade de Brasília Campus Gama - Engenharia Aeroespacial, Engenharia Automotiva, Engenharia de Energia, Engenharia de Software e Engenharia Eletrônica, sendo eles:

- Construção da estrutura física do equipamento;
- Funcionamento do equipamento por meio de motores, drivers, e outros sistemas;
- Alimentação do equipamento;
- Implementação de interface para gerenciamento da operação;
- Implementação de processamento de imagens;
- Implementação do sistema eletrônico.

2 Fases de concepção

2.1 Estrutura Analítica do Projeto (EAP)

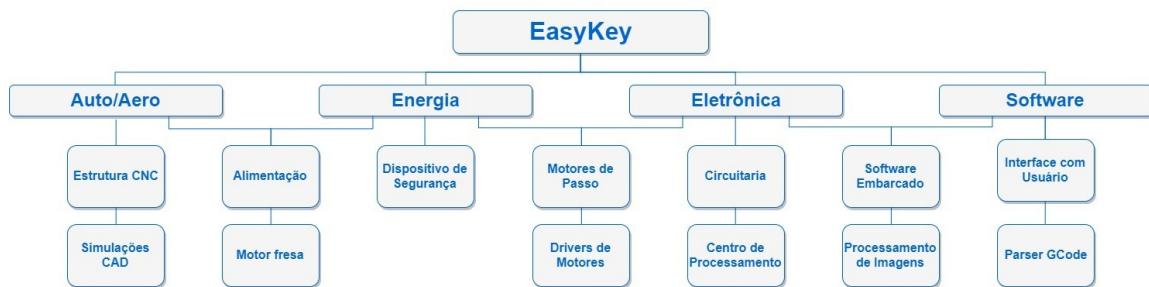


Figura 1 – EAP do projeto EasyKey. Fonte: Autores

2.2 Descrição do Projeto

O projeto EasyKey visa digitalizar, automatizar e otimizar o processo de cópia de chaves, é um projeto desenvolvido a partir da interdisciplinaridade de todas as engenharias do Campus Gama da UnB - Aeroespacial, Automotiva, Eletrônica, Energia e Software. Sua descrição engloba a leitura dos segredos de uma chave que será copiada em chave virgem por uma CNC - Controle Numérico Computadorizado - que é um dispositivo eletrônico e mecânico altamente preciso no qual é capaz de movimentar vários eixos, sendo que neste projeto serão dois eixos.

Sua operação será fácil, em que a chave é colocada no dispositivo de leitura, escaneada, e após confirmação via software. O cliente poderá colocar uma chave virgem que será copiada na CNC, e resultará na cópia da chave desejada.

2.3 Propósito do projeto

A EasyKey trata-se de um projeto de máquina copiadora de chaves automática, ou seja, uma máquina que corta uma chave virgem a partir de outra chave e por meio de um equipamento computadorizado. Este equipamento tem como finalidade a melhoria no processo de cópia de chaves e pode ser utilizado por chaveiros que querem agregar tecnologia ao seu negócio, visto que a maioria das copiadoras são mecânicas.

A máquina a ser desenvolvida é um dos módulos que pode ser acoplado a uma “vending machine”, ou seja, posteriormente pode ser usada para construção de uma máquina de vendas autônoma.

2.4 Recursos Humanos

Foi definido o escopo do projeto e a equipe foi dividida em quatro subequipes, cada uma com atividades a serem desenvolvidas na área de atuação de sua engenharia, e na última fase do projeto, os produtos desenvolvidos pelas subequipes irão se integrar para finalizar a construção do equipamento.

Para melhor gerenciamento dos processos ao longo do projeto, foram designados dois gerentes gerais e um subgerente por subárea, totalizando seis gerentes, sendo eles, subgerente de estrutura, subgerente de energia, subgerente de software e subgerente de eletrônica. Os demais membros, compuseram suas equipes, para cumprir as metas pré-estabelecidas neste projeto. As tabelas 8 e 2 e a figura 2.4 apresentam os papéis e responsabilidades de cada membro dentro da estrutura organizacional do projeto.

Tabela 1 – Papéis dos membros

Membro	Papel
Felipe Caixeta Pereira	Equipe de Estrutura
João Paulo Lopes Ferreira	Equipe de Estrutura
Laércio Silva De Sousa Júnior	Gerente do Projeto e Equipe de Software
Luciano Henrique Nunes de Almeida	Gerente de Software e Equipe de Software
Luis Henrique Nunes Guimarães	Equipe de Software
Rodrigo Bonifácio de Medeiros	Gerente de Eletrônica e Equipe de Eletrônica
Renato Cesar da Silva Agnello	Equipe de Eletrônica
Samuel Angelo Dantas Rocha	Gerente de Energia e Equipe de Energia
Tayná Rodrigues Andrade	Gerente do Projeto e Equipe de Energia
Victor Luis Quintarelli Bertolino	Equipe de Energia
Vinícius Borges Brum	Gerente de Estrutura e Equipe de Estrutura
Vinícius Lisboa do Nascimento	Equipe de Eletrônica

Tabela 2 – Responsabilidade das equipes

Equipe	Responsabilidade
Estrutura	Projetar e construir a estrutura da máquina e garantir a fixação dos elementos e movimentação dos componentes.
Energia	Dimensionar e instalar os sistemas de alimentação, dispositivos de segurança e acionamento da máquina.
Eletrônica	Dimensionamento e instalação do sistema de processamento de imagem, sensores que garantem o controle e a comunicação entre os componentes.
Software	Projetar a arquitetura do software de interpretação e desenvolvimento garantindo o processamento de imagem, precisão de corte, bem como a qualidade e velocidade e desenvolver o sistema web e servidor.

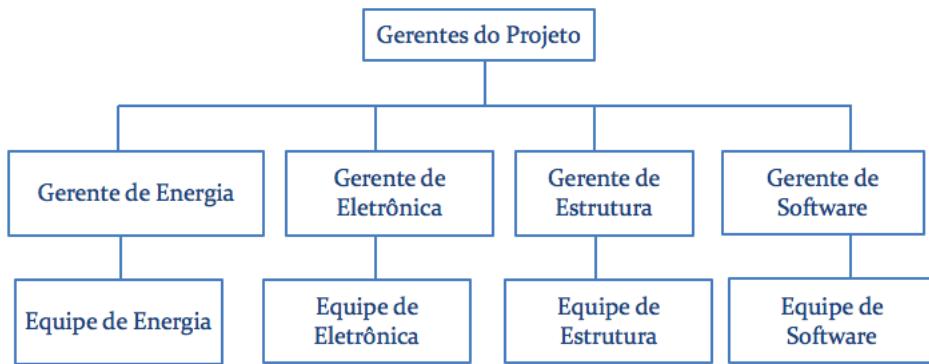


Figura 2 – Organograma da equipe. Fonte: Autores

2.5 Restrições e Riscos

2.5.1 Restrições

As restrições identificadas para a realização do projeto são:

- EasyKey funcionará apenas com chaves de cadeado da marca PADO;
- Tempo para execução do projeto;
- Quantidade de membros das equipes.

2.5.2 Riscos

Os principais riscos para o desenvolvimento do projeto EasyKey são listados abaixo:

- Redução da quantidade de membros da equipe do projeto, caso algum integrante abandone a disciplina;
- Problemas de saúde com algum membro da equipe, reduzindo a produtividade;
- Problemas financeiros com algum membro da equipe, reduzindo a verba para aquisição de insumos;
- Falhas em equipamentos utilizados para auxiliar o desenvolvimento do projeto (Laptops, motores, drivers, ferramentas, etc);
- Atraso na disponibilização de recursos necessários aos envolvidos do projeto;
- Qualidade dos produtos/subprodutos não responder às expectativa do cliente;
- Atrasos em processos chave do projeto;

- Possíveis defeitos encontrados durante a homologação do produto/subproduto pelo cliente;
- Desentendimento entre membros da equipe;

2.6 Estimativa de Custos

Os custos do projeto são definidos a partir dos custos de mão de obra e dos custos de materiais necessários para a confecção do produto. O projeto será realizado por alunos de graduação da Universidade de Brasília, da disciplina Projeto Integrador 2. O custo estabelecido pela equipe é de R\$16,80 por hora trabalhada por cada integrante, esse valor foi estimado a partir do Relatório do Decanato de Gestão de Pessoas da Universidade de Brasília, quanto ao custo relacionado a cada estudante para UnB. A estimativa de custo relacionada a mão de obra, é apresentada na Tabela 3:

Tabela 3 – Orçamento de custo de mão de obra do projeto

Integrantes	Total de horas por integrante	Valor da hora	Custo da mão de obra
12	150	R\$16,80	R\$30.240,00

Os custos relacionados aos materiais que serão utilizados durante a realização do projeto foram levantados por cada subequipe de acordo com a necessidade das quais para a execução do projeto com a qualidade desejada, esses valores são apresentados na tabela 4:

Tabela 4 – Orçamento de custo dos materiais utilizados no projeto

Material	Quantidade	Preço	Preço total
Arame para solda - 1 Kg	1	R\$23,90	R\$23,90
Barra de Aço 6 m x 20 mm x 20 mm	1	R\$23,70	R\$23,70
Botão de emergência tipo cogumelo com trava	1	R\$25,00	R\$25,00
Chapa de Aço 3 m x 1,20 m de 1,5 mm	1	R\$180,00	R\$180,00
Chave geral com tranca	1	R\$15,00	R\$15,00
Chave táctil	3	R\$7,00	R\$21,00
CI L297 (controle)	2	R\$60,00	R\$60,00
CI L298 (acionamento da bobina)	2	R\$60,00	R\$60,00
Conecotor macho/fêmea	2	R\$15,00	R\$30,00
Contatores	1	R\$50,00	R\$50,00
Cooler e dissipador	4	R\$30,00	R\$120,00
Dip Switch	2	R\$15,00	R\$30,00
Disjuntor	2	R\$10,00	R\$20,00
Display touch capacitivo 7"	1	R\$200,00	R\$200,00
Fonte 12V	1	R\$60,00	R\$60,00
Fonte de 5V	1	R\$40,00	R\$40,00
Fonte do motor (até 46V)	1	R\$100,00	R\$100,00
Fresa	1	R\$0,01	R\$0,01
Fusível	4	R\$3,00	R\$12,00
Módulo Câmera Raspberry Pi 5 Mega pixel	1	R\$70,00	R\$70,00
Motores de passo	2	R\$170,00	R\$340,00
Parafusos Diversos	1	R\$50,00	R\$50,00
Pares de Trilho Gaveta	4	R\$60,00	R\$240,00
Placa de Acrílico 1 m x 0,5 m de 0,2 mm	1	R\$38,50	R\$38,50
Placa de Circuito Impresso	1	R\$30,00	R\$30,00
Prototipagem	1	R\$100,00	R\$100,00
Raspberry Pi 3B+	1	R\$200,00	R\$200,00
Regulador de tensão (LM2596)	2	R\$10,00	R\$20,00
Relê de acionamento	2	R\$30,00	R\$60,00
Soquete	1	R\$15,00	R\$15,00
Tarugo	1	R\$40,00	R\$40,00
Trimpot	2	R\$4,00	R\$8,00
Valor Total			R\$2.282,11

O custo total orçado para o projeto, é de R\$32522,11, somando-se os custos relacionados a mão de obra e estimativa dos custos para aquisição dos materiais. Deste valor, R\$2.282,11 são referentes ao orçamento de materiais utilizados para a construção da máquina. Foi gasto R\$1935,11 na compra de materiais e equipamentos para confecção da máquina, totalizando o custo de equipamentos, materiais e mão de obra no valor de R\$32.175,11.

2.7 Cronogramas de atividades

A disciplina Projeto Integrador 2 possui 3 momentos em que os pontos de controle do projeto são apresentados e avaliados de acordo com cada fase concluída. As datas de ponto de controle foram estabelecidas pelos professores orientadores, sendo elas:

- Ponto de Controle 1 – 01/09 e 06/09

✓ Fase 1 e 2

- Ponto de Controle 2 - 08/11 e 10/11

✓ Fase 3

- Ponto de Controle 3 – 01/12 e 06/12

✓ Fase 4

A fim de satisfazer os prazos definidos, cada subequipe estabeleceu cronogramas para satisfazer as demandas dos pontos de controle e manter o bom desempenho do projeto. O cronograma geral é apresentado na tabela 5:

Tabela 5 – Cronograma geral do projeto

ID	Descrição	Duração	Início	Fim	Equipe
1	Realizar testes unitários	6d	26/10/2017	02/11/2017	Todas equipes
2	Testes com todas subáreas	6d	01/11/2017	08/11/2017	Todas equipes
3	Finalização do Relatório - Ponto de Controle 2	3d	01/11/2017	03/11/2017	Todas equipes
4	Elaboração da apresentação - Ponto de Controle 2	4d	03/11/2017	08/11/2017	Todas equipes
5	Apresentação 2	3d	08/11/2017	10/11/2017	Todas equipes
6	Rotina de testes de subsistemas	6d	09/11/2017	16/11/2017	Energia
7	Integração com as subáreas	6d	09/11/2017	16/11/2017	Energia
8	Teste do projeto integrado	6d	09/11/2017	16/11/2017	Todas equipes
9	Elaboração de manual	5d	17/11/2017	23/11/2017	Todas equipes
10	Homologação do produto final	5d	17/11/2017	23/11/2017	Todas equipes
11	Elaboração documentação ponto de controle 3	5d	24/11/2017	30/11/2017	Todas equipes
12	Conectar subáreas	11d	28/09/2017	12/10/2017	Eletrônica
13	Adaptações estruturais	11d	16/10/2017	30/10/2017	Eletrônica

14	Finalizar relatório 3	4d	15/11/2017	20/11/2017	Todas equipes
15	Conectar subáreas	11d	29/09/2017	13/10/2017	Estrutura
16	Adaptações estruturais	11d	16/10/2017	30/10/2017	Estrutura
17	Finalizar relatório 3	4d	15/11/2017	20/11/2017	Todas equipes
18	Interface de venda em tela touch	6d	10/11/2017	17/11/2017	Software
19	Integrar engenharias	6d	17/11/2017	24/11/2017	Todas equipes
20	Testes com todas subáreas	6d	24/11/2017	01/12/2017	Todas equipes
21	Finalização do Relatório - Ponto de Controle 3	6d	24/11/2017	01/12/2017	Todas equipes
22	Elaboração da apresentação - Ponto de Controle 3	3d	29/11/2017	01/12/2017	Todas equipes
23	Apresentação 3	6d	01/12/2017	08/12/2017	Todas equipes

2.8 Plano de comunicação

Este tópico tem como objetivo mostrar como os membros do projeto se organizaram para seguir a evolução do projeto, de acordo com o planejamento do projeto. Os encontros entre os membros do grupo tem como objetivo sanar as dúvidas que surgem com o desenvolvimento do projeto, além de ter uma grande importância para alinhar as ideias de cada área. Sendo assim, a comunicação foi dividida em reuniões presenciais e acompanhamento externo as quais estão citadas nas subseções abaixo:

2.8.1 Reuniões Presenciais

Todas as nossas reuniões presenciais ocorrerão nas quartas-feiras de 16h às 17h50min e nas sextas-feiras de 14h às 17h50m. Estes horários são referentes aos horários definidos pelo curso de Projeto Integrador 2. Essas reuniões têm como objetivo a definição das próximas atividades, replanejamento de prazos, alinhamento da equipe tanto em geral quanto específicos em cada área.

2.8.2 Acompanhamento Externo

Como os membros da equipe devem estar atualizados, colaborando e participando durante o projeto, foi determinada a necessidade da utilização das seguintes ferramentas:

- Whatsapp/Telegram: ferramenta utilizada para discussões, avisos e novas ideias sobre o projeto. Foi criado um grupo com todos os integrantes da equipe. Google Drive: ferramenta responsável em manter o armazenamento de todo o material utilizado

pela equipe, são eles: observações, anotações, rascunhos e planilhas produzidas pela equipe. Desta maneira, foi criada uma pasta e compartilhada entre os integrantes.

- Overleaf: ferramenta usada para a construção dos relatórios que serão entregues, estes relatórios são feitos em LaTeX. Desta forma, foi criado um projeto na mesma e disponibilizado aos integrantes.
- BitBucket: ferramenta utilizada para o versionamento de códigos que serão gerados pela equipe de software. Nesta ferramenta foi criado um repositório chamado easykey para os membros desenvolverem.
- Trello: ferramenta que é responsável pela organização das tarefas, das estipulações de prazos de cada tarefa e atribuição de atividades para os integrantes. Foi criado um quadro de backlog para cada área além dos quadros que mostram o status da atividade(Fazendo, Em revisão, Feito).
- Telefonia: ferramenta que será utilizada apenas em casos extremos, quando por algum motivo outros meios de comunicação falharem.

3 Produto

Neste tópico será detalhado o desenvolvimento do projeto e da fabricação da EasyKey. Para facilitar a compreensão, este foi dividido em subseções, com ênfase específica nas áreas da mecânica (estrutura da máquina), energia e eletrônica (alimentação, motores, drivers de acionamento e potência, sistema de ventilação, proteção e painel de comando, sensores), e desenvolvimento de sistemas (computador, software de controle, processamento de imagem, e interface máquina-usuário). Cada subseção aborda o produto relacionado à subequipe correspondente.

3.1 Estrutura

O desenvolvimento dos trabalhos da equipe de estruturas estão sendo executados de acordo com Planejamento Geral do Projeto *EasyKey* e simplificado conforme figura 3.

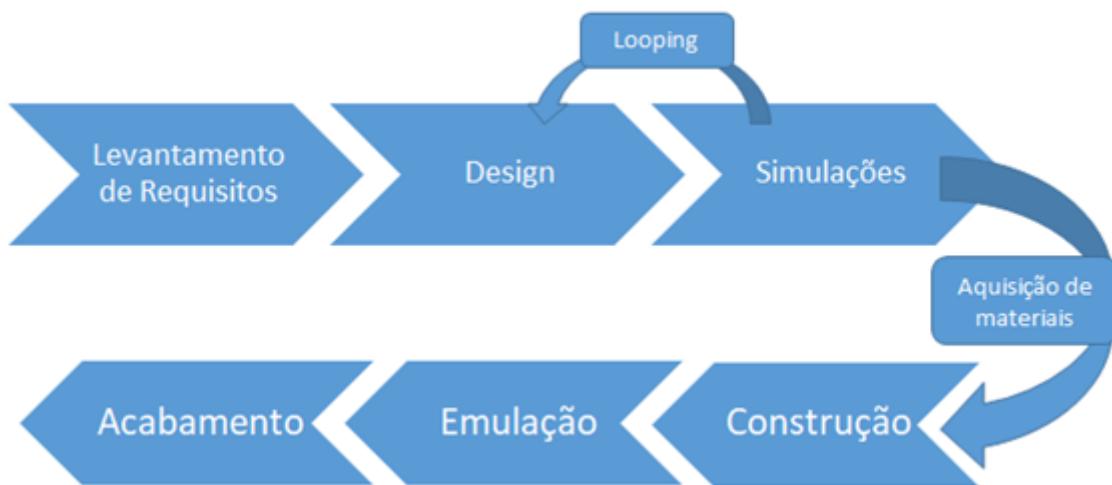


Figura 3 – Desenvolvimento do Subsistema do Projeto Estrutural. Fonte: Autores

Na etapa de levantamento de requisitos ocorreu a identificação das necessidades do produto. Durante a etapa de concepção e design foi realizada o projeto em *CAD* ([SHIGLEY, 2011](#)), que em seguida foi submetido à etapa de simulação, onde ocorreu a verificação do projeto mediante aos requisitos técnicos iniciais ([HIBBELER, 1999](#)). Nesse último estágio ocorreu a realimentação da etapa anterior, visto que foram necessários modificações e correções de projeto, sendo aqui inserido um ponto de *looping*. Em continuidade, houve a aquisição de materiais, com posterior início da construção da estrutura do protótipo e realização dos testes de emulação. Os trabalhos atuais encontram-se entre as etapas de emulação e acabamento.

3.1.1 Design Inicial

Para que os requisitos determinados fossem atendidos, a equipe convergiu ao layout inicial mostrado na Figura 4. A estrutura planejada consiste na união de barras de Aço SAE 1020 de perfil quadrado de 20x20 mm com 1,5 mm de espessura.

Algumas dificuldades surgiram no desenvolvimento, por exemplo, a complexidade de fabricação da estrutura proposta e a alocação dos componentes essenciais, bem como a integração com os outros sistemas. Desta forma, foram necessárias mudanças e aprimoramento no projeto inicial.

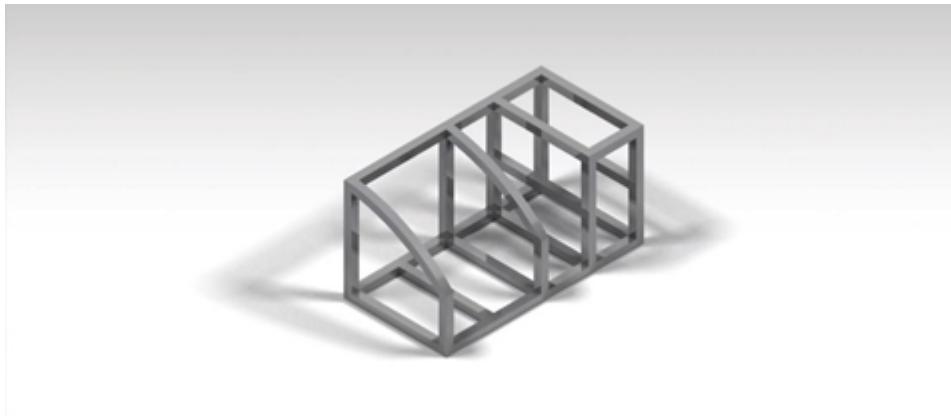


Figura 4 – Proposta de desenho estrutural inicial. Fonte: Autores

3.1.2 Aprimoramento/Refinamento

A equipe continuou buscando um layout cuja fabricação pudesse ser executada de maneira mais rápida e eficiente, determinando um novo layout conforme a Figura 5. A estrutura foi definida em formato retangular com 545 mm x 445 mm x 150 mm.

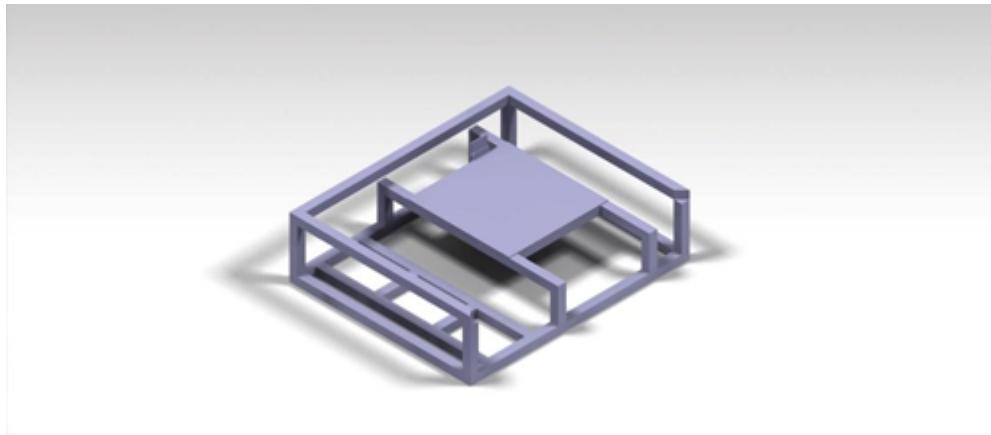


Figura 5 – Estrutura externa básica e base para sistema de movimentação. Fonte: Autores

Esse novo layout permitiu que os objetivos da estrutura fossem alcançados com facilidade, melhorando a integração entre os sistemas essenciais ao projeto, permitindo ainda

a incorporação de novos módulos e evolução/aprimoramento do produto, de acordo com a necessidade de acréscimo de componentes.

Na Figura 6 é possível visualizar o sistema de trilhos definidos pela equipe, bem como as estruturas primária e secundária de movimentação, que em sua montagem final permitem a movimentação da chave virgem até o disco de fresamento em dois eixos.

O sistema de trilhos é composto por corrediças telescópicas devido a facilidade de movimentação, acoplamento, integração e baixo custo de aquisição/manutenção. O acoplamento e movimentação do sistema será controlado por dois eixos que deverão ser acionados por motores de passo.

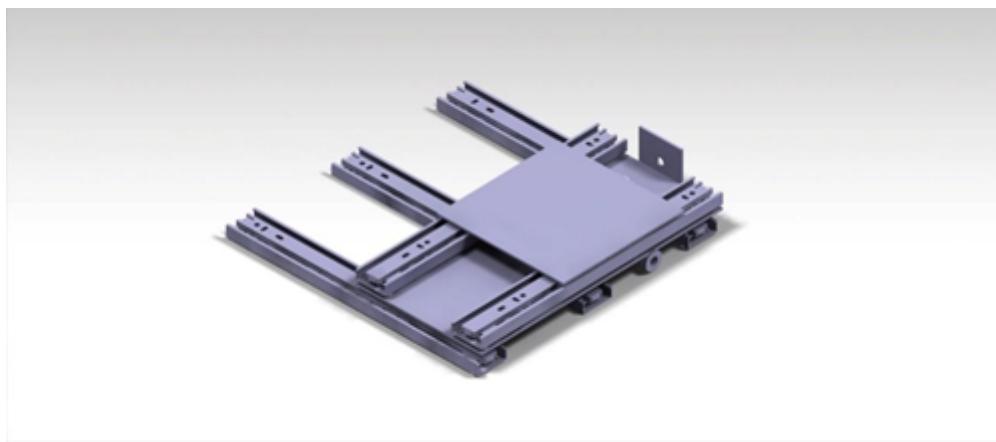


Figura 6 – Sistema de trilhos e movimentação. Fonte: Autores

A estrutura base foi desenvolvida para permitir a adição de subsistemas em módulos e na Figura 7 é possível visualizar o módulo desenvolvido para acoplamento do sistema de aquisição de imagens, bem como a alocação dos componentes de alimentação e controle eletrônico do protótipo.

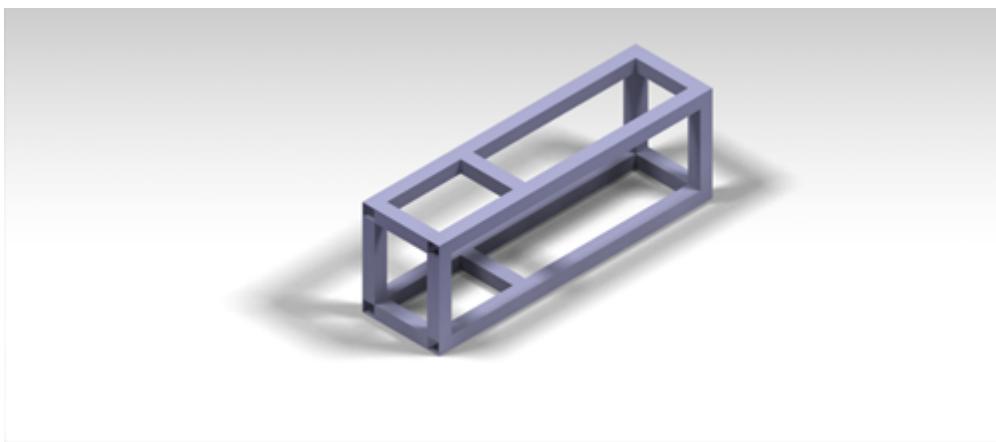


Figura 7 – Estrutura desenvolvida para armazenar demais subsistemas. Fonte: Autores

A estrutura do sistema de aquisição de imagens foi desenvolvido para abrigar a placa *Raspberry Pi* e a câmera no topo. Na base tem-se uma placa de acrílico branco leitoso e

uma unidade de luzes do tipo *LED*. As chapas estruturais para fixação dos componentes de imagem utilizam a técnica de impressão 3D denominada *Fused Deposition Modeling* (FDM) e são unidas entre si por tirantes em barras de aço. O mesmo tirante será usado para a fixação na estrutura em sua montagem final.

O material escolhido para impressão foi o polímero poliácido láctico (PLA) e sua escolha reúne bons requisitos estruturais, baixo preço, acessibilidade, confiabilidade e fácil fabricação ([ARNOLD, 2012](#)).

A chave a ser copiada deverá ser posicionada acima da placa de acrílico, que ao ser iluminada permitirá a captura do contorno do segredo da chave pela câmera conforme Figura 8.

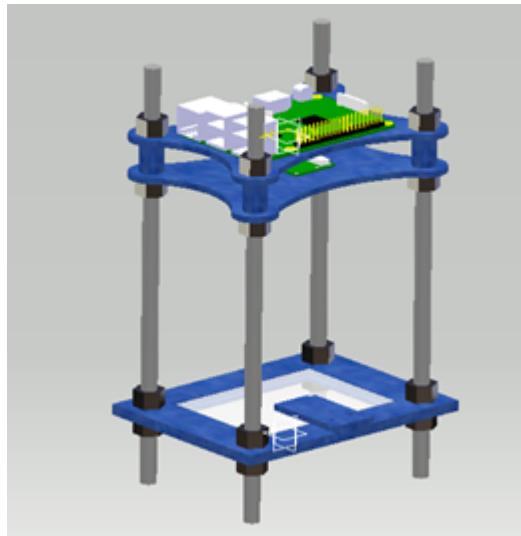


Figura 8 – Conjunto de aquisição de imagens. Fonte: Autores

Durante o processo de desenvolvimento da estrutura principal, a equipe iniciou a etapa de desenvolvimento dos elementos de conexão, união dos motores de passo com os fusos e de outros elementos estruturais necessários para integração com outros módulos, como a fixação de tela para interação do usuário e máquina. Na Figura 9 pode ser visualizado o desenho final da estrutura básica do produto desenvolvido e aprovado para a etapa de construção.

Os desenhos técnicos dos componentes estruturais estão disponíveis no Apêndice A.

3.1.3 Simulações

Essa etapa tem como objetivo utilizar o auxílio computacional para otimizar o dimensionamento da estrutura, de modo a atingir os objetivos especificados com a menor utilização de material e menor peso final. Foram realizados simulações no software *Ansys®*, análise modal de corpo livre e uma análise estrutural estática. Os resultados são mostrados

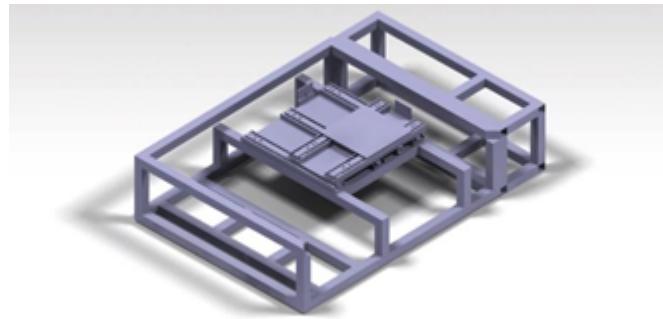


Figura 9 – Desenho da estrutura básica completa. Fonte: Autores

nas Figuras 10 e 11. Documentação de apoio do [ANSYS](#) serviu de suporte para execução das atividades.

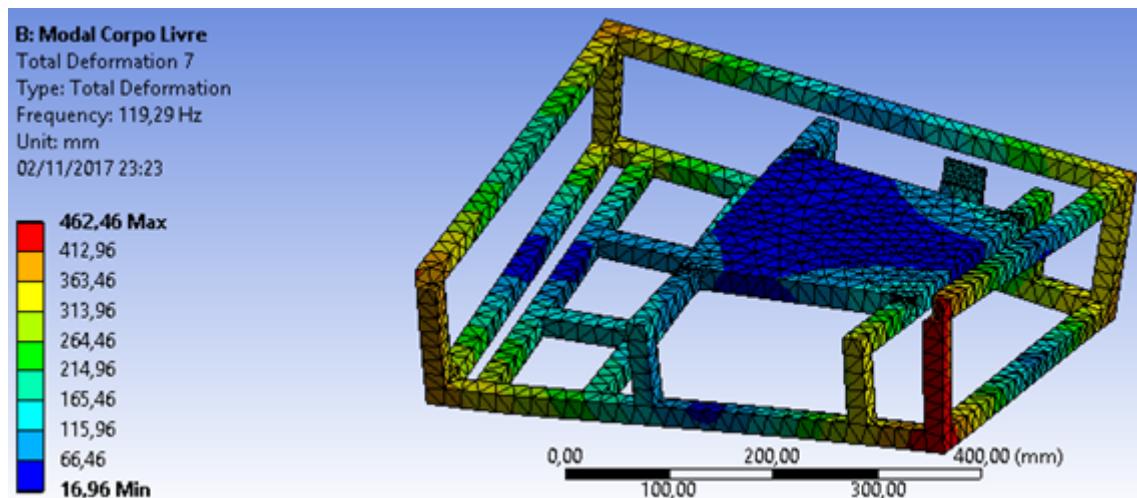


Figura 10 – Simulação para Modal de Corpo Livre. Fonte: Autores.

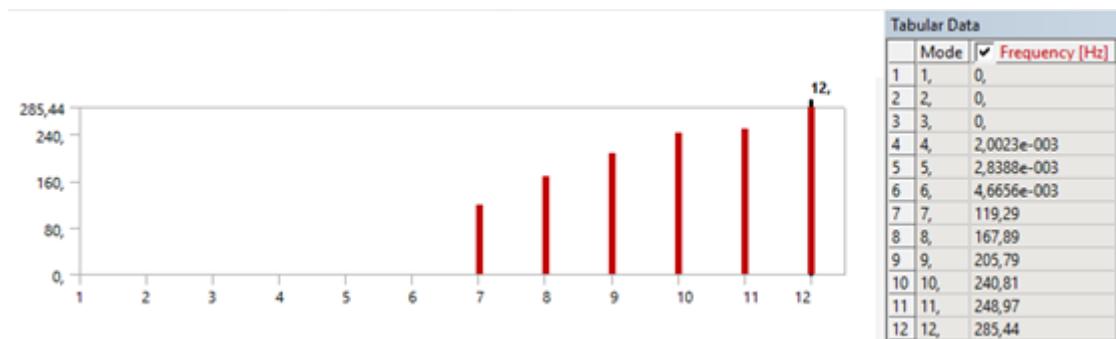


Figura 11 – Frequências naturais para os 12 primeiros modos de vibração para Análise Modal de Corpo Livre. Fonte: Autores

A análise modal é aplicada para obtenção dos modos de vibração e das frequências naturais da estrutura ([SINGIRESU, 1999](#)). Observamos que a geometria do sistema não apre-

senta erros, pois a resposta da análise nos apresenta os seis primeiros modos de vibração próximos de zero e os outros modos com frequência natural muito elevada.

A análise estrutural realizada apresenta as deformações existentes na estrutura devido aos carregamentos nela impostos. Nesse caso do projeto tem-se o peso dos elementos que são acoplados na estrutura (motores e demais componentes). Para apresentação da resposta, a estrutura foi engastada onde esta é apoiada e o peso dos elementos foi distribuído conforme suas fixações. As respostas são apresentadas nas Figuras 12 e 13.

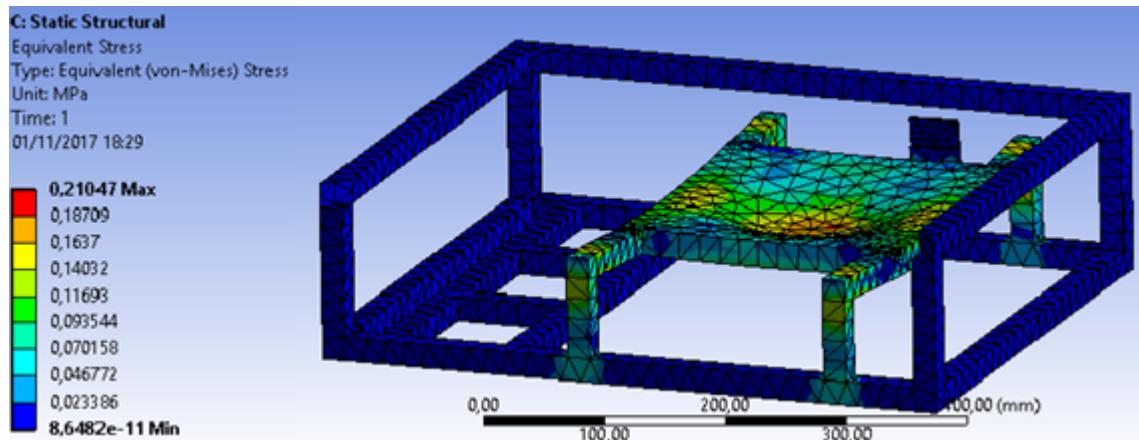


Figura 12 – Simulação Estática Estrutural. Fonte: Autores

	Time [s]	<input checked="" type="checkbox"/> Minimum [mm]	<input checked="" type="checkbox"/> Maximum [mm]
1	1,	0,	2,2032e-003

Figura 13 – Deformação para análise Estática. Fonte: Autores

3.1.4 Aquisição de Materiais

A escolha dos materiais para a construção do protótipo baseou-se no custo previsto para o projeto de financiamento próprio, na usinabilidade e possibilidade de construção em espaço disponibilizado pela universidade. Os principais materiais utilizados nessa fase estão listados na Tabela 6.

Tabela 6 – Principais Materiais utilizados

Item	Especificação	Quantidade
Barra Quadrada	20x20 mm de 1,2mm - Aço 1020	2x 6m
Corrediça Telescópica	JOMARCA® 30188	5
Chapa Retangulares	250x250 mm de 2mm – Aço 1020	2
Parafusos diversos	Diâmetro - 4mm	18
Porcas diversas	Diâmetro - 4mm	10

3.1.5 Construção

O processo de fabricação da estrutura se iniciou com o corte dos componentes nas medidas previamente estabelecidos no software. Em seguida, os componentes já cortados foram unidos utilizando-se o processo de soldagem por *MIG/MAG* (Metal Inert Gas / Metal Active Gas), como pode ser visualizado nas Figuras 14, 15 e 16.



Figura 14 – Estrutura externa apó soldagem das barras externas da estrutura. Fonte: Autores



Figura 15 – Construção da base e fixação dos trilhos para movimentação. Fonte: Autores

Em seguida, foram produzidos os suportes dos motores de passo, responsáveis pela rotação dos eixos de movimentação, e os mesmos foram posicionados na estrutura. O passo seguinte foi a fixação dos trilhos na base, por meio de porcas soldadas na estrutura.

A partir da fixação dos trilhos na base, foram desenvolvidas as plataformas seguintes, bem como a fixação do moto esmeril que será utilizado como fresadora. Com todos os trilhos e plataformas em seus devidos lugares, foram posicionados os motores de passo e a partir deles, foram posicionados os fusos para a movimentação conforme as Figuras 17 e 18 .

Tendo finalizada a parte essencial à estrutura, foi desenvolvida a fixação do dispositivo de aquisição de imagem, bem como a armação necessária para o acoplamento dos componentes de alimentação.



Figura 16 – Fixação da base de trilhos na estrutura principal. Fonte: Autores



Figura 17 – Fixação dos motores de passo na estrutura. Fonte: Autores

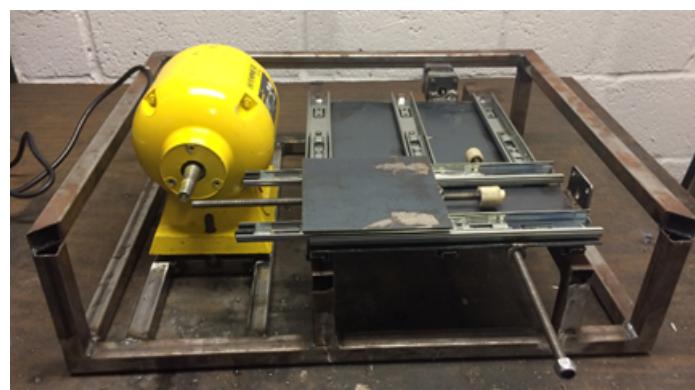


Figura 18 – Fixação do moto esmeril na estrutura. Fonte: Autores



Figura 19 – Estrutura para aquisição de imagens fabricada por Impressora 3D. Fonte: Autores

3.1.6 Acabamento

A etapa de acabamento tem por objetivo finalizar toda a subseção de estruturas. Após ser averiguado o funcionamento pleno dos eixos e trilhos de movimentação - parte essencial para o funcionamento da máquina, a estrutura foi toda pintada. Em seguida, foi desenvolvido um invólucro fabricado em chapa galvanizada com 550 mm x 470 mm x 220 mm de espessura variável, para efeito de proteção ao usuário. Esse invólucro precisa ainda ser móvel de maneira a permitir a manutenção da máquina, garantindo acesso aos trilhos e eixos e também aos componentes mais sensíveis de energia e eletrônica. Este requisito foi resolvido com a fabricação de uma tampa com dobradiça.



Figura 20 – Carcaça com tampa aberta. Fonte: Autores

Na tampa desse invólucro foram adicionadas ainda duas seções de acrílico de 250 mm x 150mm com 3mm de espessura cada, uma para visualização do processo de usinagem e outra para acesso à fixação da chave virgem.



Figura 21 – Tampa. Fonte: Autores

A fixação da chave a ser usinada foi desenvolvida para ser agregada à mesa superior de movimentação, fabricada na impressora 3D, ela apresenta rigidez o suficiente para a função e sua leveza se torna uma característica importante para não sobrecarregar os motores de movimentação.



Figura 22 – Suporte para fixação da chave virgem. Fonte: Autores

Para encerrar, foi instalada borracha trefilada em todos as arestas cortantes, tanto da estrutura principal quanto nos módulos, visando mais uma vez a segurança do usuário.

3.2 Sistema de Alimentação

3.2.1 Fonte de Alimentação

A alimentação da máquina EasyKey foi projetada para atender às diversas tensões exigidas pelos componentes do sistema. A máquina foi conectada à rede por meio de conexão devidamente aterrada, e foi confeccionado barramento para distribuição das tensões.

A máquina é conectada a rede de 220V, alimentando o Moto Esmeril, e a fonte industrial utilizada para conversão da corrente AC em corrente DC, necessária para alimentar os componentes eletrônicos. A fonte dimensionada para alimentação do lado DC do sistema tem configuração de 12V de tensão e 20A de corrente. Para os componentes que necessitam de 5V de tensão, utilizou-se regulador de tensão para ajustar alimentação ao valor necessário.

O levantamento de carga dos componentes eletrônicos do sistema é apresentado abaixo:

- ✓ +3,3 V: alimentação de processadores, reduzindo o consumo de energia;
- ✓ +5 V: *Raspberry PI*, MSP430, driver de comando e controle;
- ✓ +12 V: alimentação dos *coolers* e alimentação dos motores de passo.



Figura 23 – Fonte 12V/20A. Fonte: Autores

A fonte bivolt produzida pela equipe consiste numa fonte de alta potência, com capacidade de até 10 A. Por meio de um mini transformador 220V/12V 10 A, a fonte foi abaixada e retificada para corrente contínua por meio de 3 pontes retificadoras RS7407. Foram utilizadas 3 pontes porque cada uma fornece uma capacidade de 4 A. O circuito auxiliar consiste em 3 capacitores, um de 6800 μ F, um de 100 μ F, e um de 470 μ F. Esses capacitores servem para filtrar a tensão. Foi utilizado dois transistores PNP TIP 127 para aumentar a potência da fonte. Cada transistor aumenta a capacidade da fonte em 5 A, e por isso foi utilizado dois. O diodo 1N4007 e os fusíveis atuam na proteção do circuito. O regulador 7805 abaixa a tensão 12Vdc para 5Vdc.

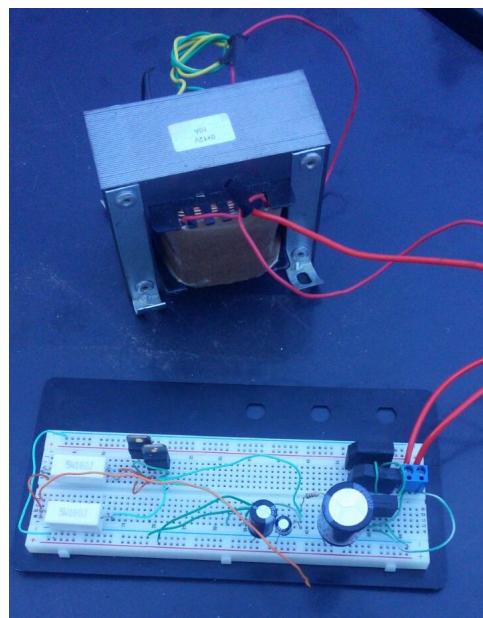


Figura 24 – Fonte produzida. Fonte: Autores

3.3 Motores

3.3.1 Motores de Passo

Para a movimentação dos eixos superiores e inferiores da CNC serão utilizados dois motores de passo. Os motores de passo são motores de corrente contínua e proporcionam a movimentação do rotor em incrementos angulares discretos, controlados por impulsos elétricos sequenciais aplicados nas bobinas do estator. Passo é um incremento angular gerado por um impulso, e o valor em graus deste deslocamento é relacionado a forma de acionamento).



Figura 25 – Motor acoplado aos eixos. Fonte: Autores

A sequência dos motores se dá por dois tipos de passo. O passo completo proporciona um deslocamento de $1,8^\circ$, que acontece sempre quando se magnetiza dois eletroímãs do estator por vez. O meio passo proporciona metade do deslocamento angular, $0,9^\circ$, e neste tipo de sequência ocorre a magnetização apenas de uma bobina, ou um grupo de bobinas, entre um passo e outro (PAZOS, 2002). Os motores de passo podem ser classificados quanto ao tipo, e quanto a estrutura. Quanto ao tipo, há duas configurações: unipolar e bipolar. Em relação a estrutura, eles são divididos em motores de relutância variável, motores de ímã permanente, ou híbridos (CONDIT RESTON; JONES, 2004). Os motores utilizados são bipolares e híbridos. Essa configuração possui apenas um enrolamento por fase, e seu rotor é multi-dentado, além de haver um ímã permanente em seu eixo. O uso desse tipo de motor é devido a sua alta precisão, e uma boa relação do torque com o deslocamento angular.

A escolha dos motores para o projeto foi dada visando menores custos para o projeto. Inicialmente, os dois motores testados para serem utilizados eram ambos NEMA 17, pois

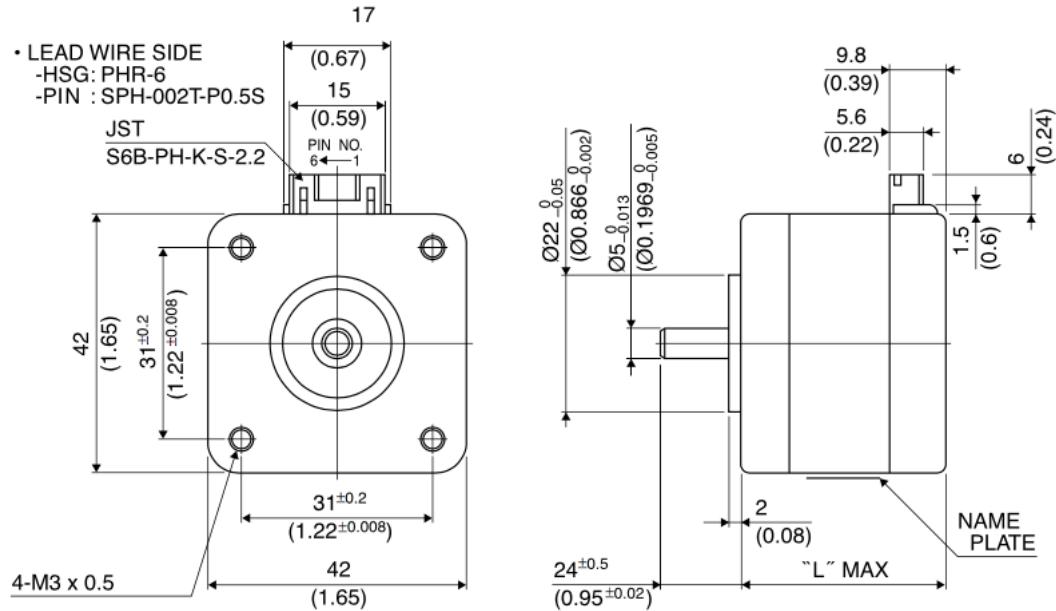


Figura 26 – Dimensões do motor. Fonte: ([MITSUMI, 2017](#))

haviam sido doados. Durante a integração, notou-se que um dos motores não tinha torque suficiente para movimentar o eixo inferior devido aos materiais utilizados na estrutura. Para solucionar, um dos motores foi substituído por um NEMA 23. As especificações dos motores, bem como o gráfico de torque e velocidade do NEMA 17, são demonstrados abaixo e foram retirados do catálogo da ([MITSUMI, 2017](#)) e da ([AKIYAMA, 2017](#)), empresas fabricantes dos motores.

Tabela 7 – Especificações do motor de passo

Modelo	17PM-K142B	AK23-10F8FN
Ângulo de Passo (graus)	1.8	1.8
Tipo do Motor	BI-POLAR	BI-POLAR
Corrente (A)	1.0	1.0
Resistência (ohm)	4.6	2.6
Torque Estático (mNm)	380	980,6
Indutância (mH)	12.9	12.0
Corrente	1.7A	1.4A
Inercia do Rotor ($g \cdot cm^2$)	62	300
Massa (g)	300	1000

3.3.2 Máquina Fresadora

Motor elétrico é toda máquina designada à transformação de energia elétrica em energia mecânica. Para este trabalho, utilizaremos o motor de indução para a movimentação do disco de corte, constituindo a fresadora que fará o corte da chave virgem. O escolhido foi o Moto Esmeril de Bancada Lynus MAC-50, apresentado na imagem a seguir, pois atende

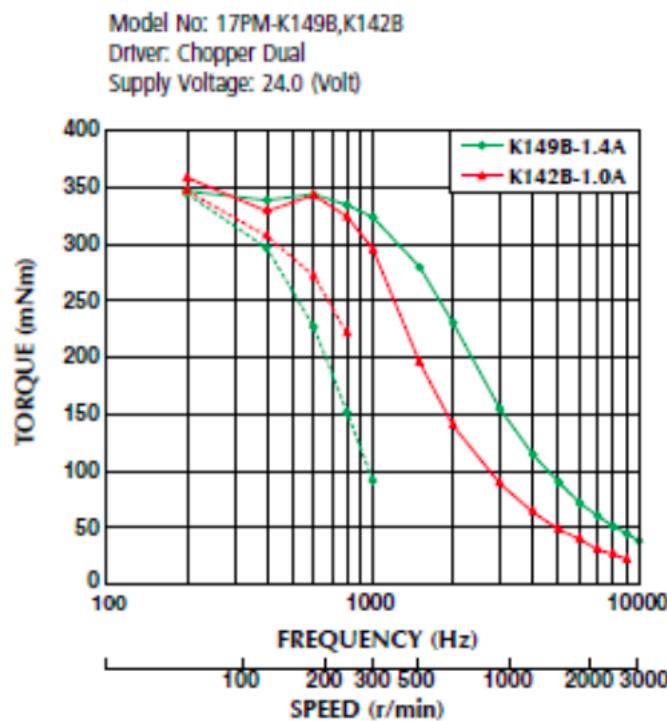


Figura 27 – Relação Torque x Velocidade. Fonte: ([MITSUMI, 2017](#))

às necessidades do projeto, por ser o tipo de motor utilizado em chaveiros convencionais e de melhor custo benefício comparado às outras marcas e modelos.



Figura 28 – Moto Esmeril de Bancada Lynus MAC-50, Fonte: Manual Lynus MAC-50

A partir das especificações do motor, é possível definir o conjugado, que é a medida do esforço necessário para girar um eixo, dada por:

Para o motor livre de carga,

$$C (N.m) = \frac{9555.P (kW)}{n (rpm)}$$

$$C (N.m) = \frac{9555.0,36}{3580} = 0,96$$

Para o motor com carga,

$$C (N.m) = \frac{9555.0,36}{3092} = 1,11$$

O Moto Esmeril de Bancada Lynus MAC-50 possui as seguintes especificações:

Tabela 8 – Especificações Técnicas Fresadora - Fonte: Manual Moto Esmeril Lynus MAC-50

Tensão de alimentação	CA: 127/220V AC (manual)
Diâmetro de Rebolo	6"
Frequência	60Hz
Rotação livre	3580 rpm
Rotação com carga	3092 rpm
Potência	200W
Potência máxima	360W
Corrente	1.75A
Temperatura de trabalho	80 C
Dimensões (A x L x P)	235 x 340 x 185 mm
Peso	5,6 Kg

3.4 Sistema de Comando e Controle

3.4.1 Driver de Comando e Controle

O controle dos motores de passo utilizados na EasyKey, para movimentação independente dos eixos, é realizado por um hardware composto por uma parte de potência e outra de controle, o *driver* de comando e controle. Para a construção dos *drivers*, realizou-se pesquisas em referenciais bibliográficos, em busca de trabalhos relacionados ao projeto desenvolvido, a fim de encontrar circuitos já utilizados para controle de motores de passo em máquinas CNC.

A partir dos resultados encontrados, optou-se por um circuito eletrônico que associa dois componentes eletrônicos fabricados pela *STMicroelectronics*®, os circuitos integrados (CI) L297 e L298, conforme o circuito apresentado nas figuras 29 e 30 .

Na associação entre os dois circuitos integrados, o L297 é responsável pelo controle da corrente nos motores, sendo o tipo de controle por *PWM chopper*, seu oscilador capacitor-

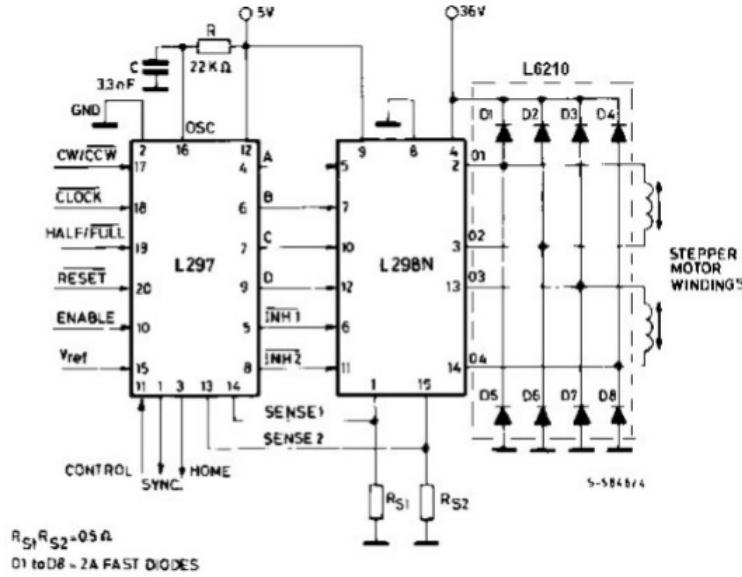
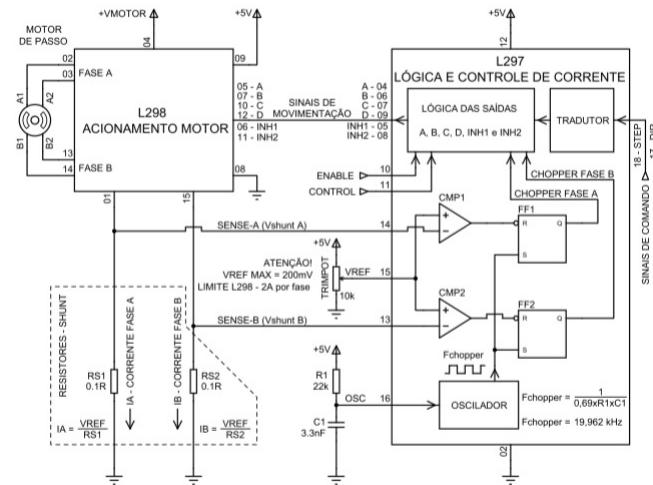
Figura 29 – Associação dos CIs L297 e L298 - Fonte: *Datasheet L297*

Figura 30 – Diagrama de blocos do driver - Fonte: Cartens, et. al., 2015

resistor interno define a frequência de operação, sendo a frequência definida por $1/0.69RC$, o valor de $R=22k$ e $C=3.3nF$, sendo $F_{chopper}=19.96kHz$, frequência recomendada pelo *datasheet* do L297. A leitura da corrente é feita pelos resistores *Shunt* conectados nas portas **SENSA** e **SENSB** do L297 e L298 e a queda da tensão é diretamente proporcional a corrente em cada bobina, logo $I_{motor}=2.V_{shunt}$, pois os resistores possuem valor de 0.5Ohms. O controle é feito pelos amplificadores operacionais internos que trabalham como comparadores e *Flip-flops RS* que é setado de acordo com o *PWM chopper*, portanto um *trimpot* que faz um divisor de corrente com o resistor de 10k fará o controle da corrente indicando a tensão na porta **VREF**, como os motores trabalham com 1A, essa tensão deve ser de 0.5V. O método de controle de corrente permite alimentar o motor de passo com uma tensão muito superior a tensão nominal, mantendo a corrente igual ou inferior a tensão nominal. Este aumento da

tensão faz com que a corrente nominal nas bobinas seja atingida em menor tempo, pelo fato da equação do indutor ser dependente da tensão e variação da corrente. O funcionamento do controle é bem simples, quando $V_{shunt} < V_{ref}$, a corrente na fase é menor que a corrente de ajuste e eles são alimentados normalmente pela fonte do motor, quando $V_{shunt} > V_{ref}$ a corrente na fase é maior que a corrente de ajuste, assim o controle entra em ação, a alimentação é interrompida e acionada novamente apenas quando receber o próximo pulso do oscilador e quando V_{ref} for menor que V_{shunt} . A porta **CONTROL** permite definir o controle com desenergização lenta ou rápida, sendo modificada por uma chave alavanca, e a melhor opção depende de cada motor, a porta **HOME** acende o LED quando a lógica das fases dos motores for reiniciada. O L297 tem como entrada o **CLOCK** que receberá O PWM do MSP430, **ENABLE** indica se é (5V) ou não (0V) para acionar o motor, comando que vem do microcontrolador, **HALF/FULL**, sendo nível lógico alto para meio passo do motor e nível lógico baixo para passo completo, essa porta é controlada por uma chave alavanca. O **RESET** que reinicia a lógica de ativação dos motores é ativa em nível baixo por um **PUSH BUTTON**. **CW/CCW** define o sentido da rotação 5V para horário e 0V para anti-horário e é controlada pelo MSP430. As saída do L297 são composta pelas portas **A,B,C e D** ativas por nível lógico alto e **INHA e INHB** ativas por nível lógico baixo, essas saídas são conectadas ao L298 formado por duas pontes H independentes de oito transistores de potência e portas lógicas como mostra a figura 31, essas saídas servem para ativar e definir a lógica da rotação dos motores, as saídas **OUT1, OUT2, OUT3 e OUT4** do L298 são conectadas às fases do motor de passo para seu devido funcionamento. Oito diodos de potência de transição rápida do tipo *Schottky* servem para proteger o circuito da força contra eletromotriz das fases dos motores. ([MICROELETRONICS, 2001](#))

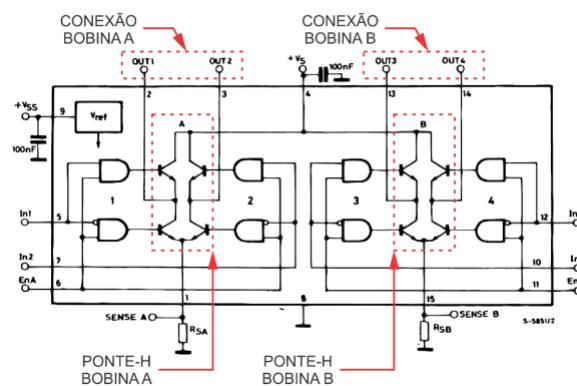
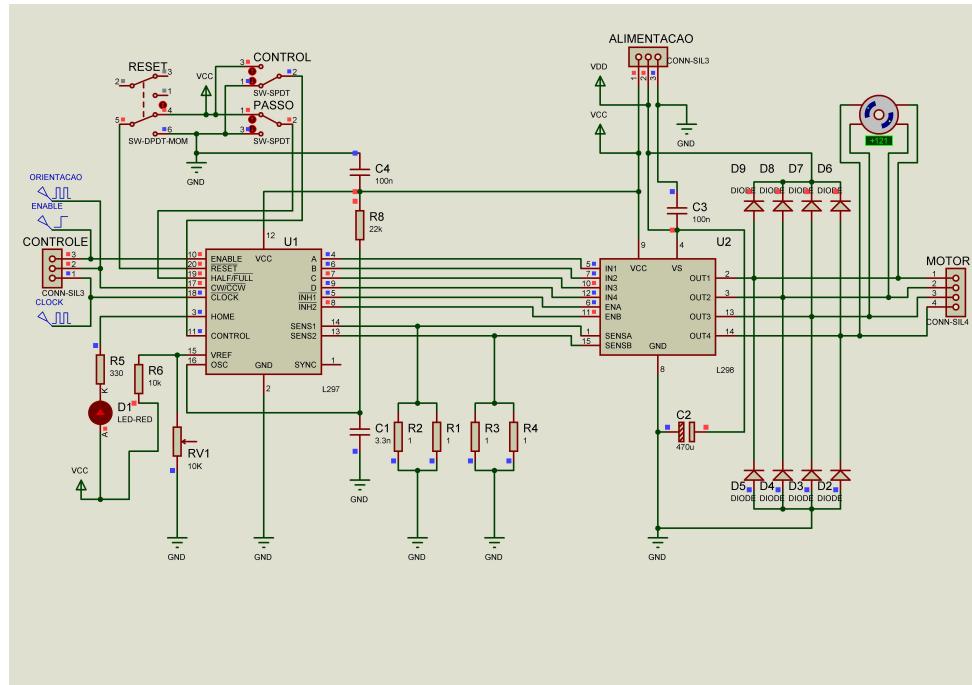
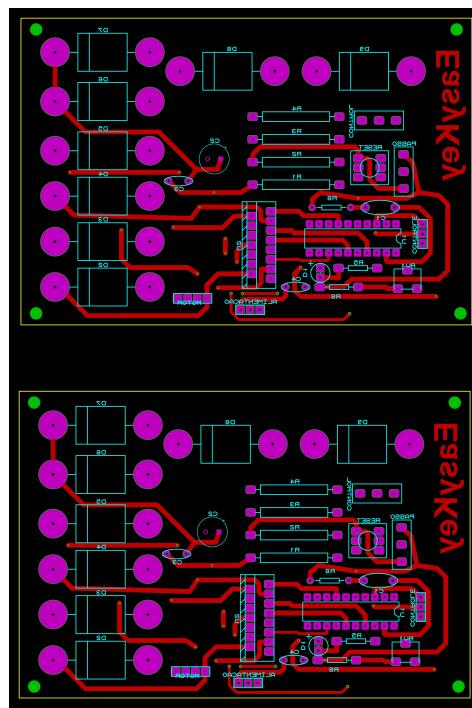


Figura 31 – Ponte H interna do L298 - Fonte: Cartens, et. al., 2015

Inicialmente, foi simulado o circuito escolhido no software *Proteus*, para validá-lo. Após a obtenção dos resultados na simulação, desenhou-se o *layout* da PCI (Placa de circuito impresso) no *Proteus*, em seguida o circuito foi confeccionado em placa de circuito impresso dupla face, as linhas vermelhas indicam trilhas de cobre na parte superior da PCI e as azuis indicam a parte inferior.

Figura 32 – Simulação do *Driver* no Software Proteus - Fonte: AutoresFigura 33 – Layout PCB do *Driver* de Comando e Controle - Fonte: Autores

Para a confecção da placa, foram utilizados outros componentes eletrônicos para montagem do circuito, e estes são listados na tabela 9:

Conforme apresentado na seção 3.1 Estrutura, a máquina foi projetada a partir da movimentação de dois eixos, utilizando-se dois motores de passo para a movimentação. Para a decodificação dos sinais transmitidos pela *RaspberryPy* e *MSP430*, construiu-se dois

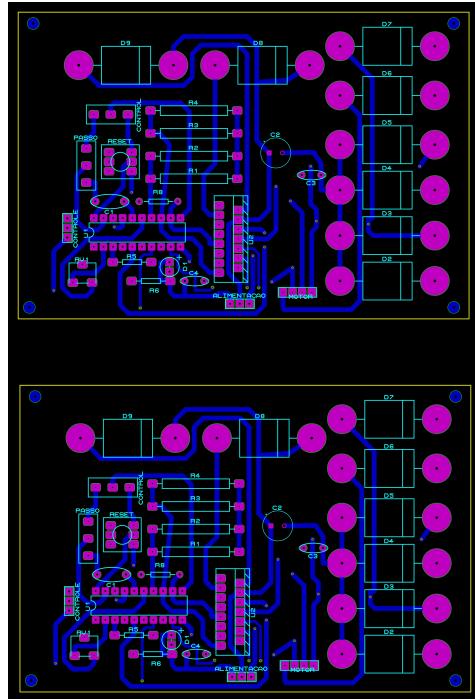


Figura 34 – Layout PCB do *Driver de Comando e Controle* - Fonte: Autores

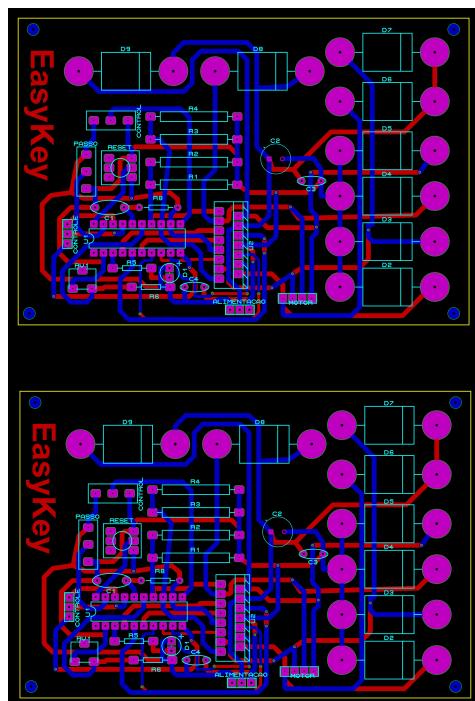


Figura 35 – Layout PCB do *Driver de Comando e Controle* - Fonte: Autores

drivers idênticos em placas de circuito impresso, uma pra cada motor de passo, que pode ser visualizado na figura 37.

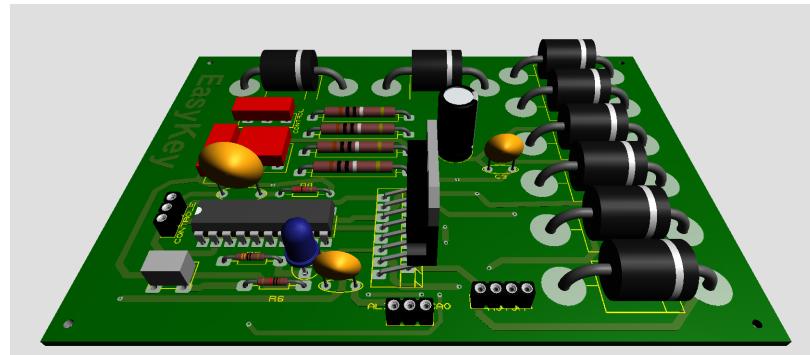


Figura 36 – Layout 3D PCB do *Driver* de Comando e Controle - Fonte: Autores

Tabela 9 – Componentes Eletrônicos PCB

Componente	Unidade	Função
Capacitor 3,3 nF	1	Capacitor para o oscilador interno do L297
Capacitor 470 μ F	1	Capacitor de acoplamento do L298
Capacitor 100 nF	2	Capacitor de acoplamento do L297
Chave alavanca 3 portas	2	Controle da energização do circuito, e do passo
Diodos Schottky	8	Protege os CIs da corrente induzida dos motores
Push Button	1	Função Reset
Resistor 1 Ohm	4	Resistor Shunt para medição de corrente
1 LED	1	Indicador de inicialização da lógica dos motores
Resistor 330 Ohms	1	Limita a corrente do LED
Resistor 22k Ohms	1	Resistor para o oscilador interno do L297
Trimpot 10k Ohms	1	Divisor de tensão para o TRIMPOT

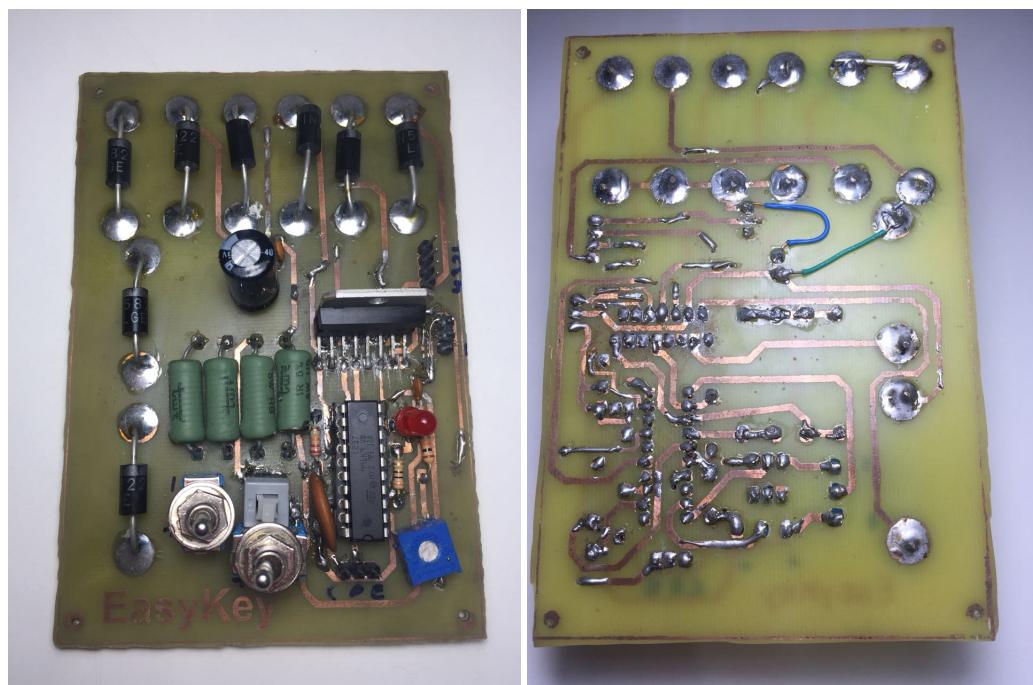


Figura 37 – *Driver* de Comando e Controle - Fonte: Autores

3.4.2 Circuito Dimmer

Em processos de usinagem, um parâmetro importante para o acabamento da peça a ser usinada é a velocidade de corte. Um dos parâmetros importantes para o acabamento é a velocidade de fresamento, que varia de acordo com cada material, e do diâmetro da ferramenta de corte. Para isso, faz-se necessário regular a velocidade da máquina de corte. Para isso, criou-se um circuito de controle de potência da carga, caracterizado como circuito *dimmer*. Esse circuito de eletrônica de potência permite variar a potência entregue a uma carga, e consequentemente a velocidade de rotação do eixo, utilizando um TRIAC, que através de uma corrente aplicada em sua porta, permite controlar a corrente de saída para a carga, atuando na variação de potência por meio do controle de fase, ou seja, variando o ângulo de disparo e controlando a parcela da energia da rede que pode ser transferida ao motor. Além disso, utiliza-se um DIAC que dispara no início ou fim dos semiciclos da tensão alternada da rede de energia (BRAGA, 2017). O circuito DIMMER foi simulado no software *PROTEUS*. O esquema é demonstrado na figura 38.

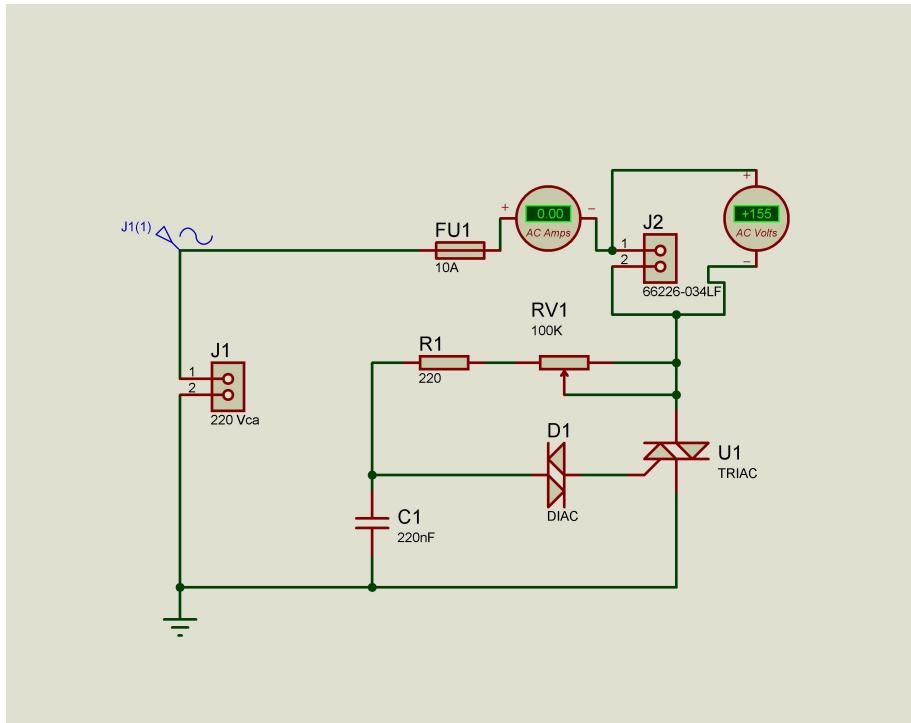


Figura 38 – Regulador de Velocidade - Fonte: Newton Braga

Juntamente implementado com o circuito *dimmer*, foi projetado circuito de acionamento do moto esmeril via relé, a fim de ter o controle da alimentação por meio da MSP, para ligar o motor apenas no momento de corte da chave. A placa confeccionada é apresentada na figura abaixo.

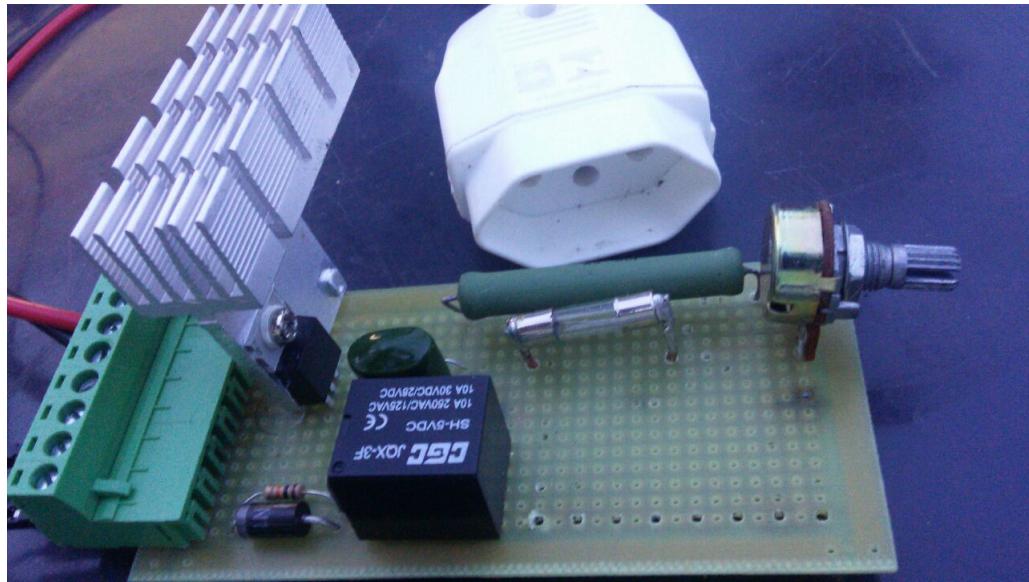


Figura 39 – Circuito Dimmer e Relé de Acionamento - Fonte: Autores

3.5 Sistema de Ventilação

A energização dos drivers de comando, e do circuito dimmer, faz com que alguns componentes do circuito aqueçam, atingindo temperaturas que podem danificar o sistema, para prevenir estas ocorrências e evitar reparos nas PCBs, como a troca destes, instalou-se equipamentos para ventilar os circuitos. Este sistema é constituído por três *coolers* e três dissipadores instalados nos *drivers* de comando e controle dos motores de passo, e no circuito dimmer. O sistema de ventilação do painel de comando, foi projetado com dois *coolers* que ventilam o sistema, e um *cooler* que retira o ar quente da caixa, como um exaustor. Os dissipadores foram colocados nos CIs L298 dos *drivers* de comando e controle, e no triac do circuito dimmer.

3.6 Painel de Comando

Os equipamentos utilizados para comando, controle e alimentação da máquina, foram acomodados em um painel de comando, a fim de facilitar o controle e a manutenção do sistema. Neste painel, foram acomodados os dois *drivers* de comando e controle, o circuito *dimmer*, e a fonte de alimentação. Todo o cabeamento de alimentação do sistema e a conexão do moto esmeril com o circuito *dimmer*, foi instalado pelo lado esquerdo do painel de comando. O cabeamento voltado para a parte de controle por meio do MSP, conexões com os motores de passo, e sensores foram encaminhados pela lateral direita deste painel.

A disposição dos componentes dentro do painel, foi realizada mediante acomodação dos *drivers* próximos as entradas de ar, promovidas pelos *coolers* utilizados para ventilação. A fonte, os barramentos e o circuito *dimmer* foram acomodados de maneira a reduzir a

quantidade de fios dentro do painel, a fim promover clareza para manutenção do mesmo.

3.7 Sistema de Proteção

Por medidas de segurança, toda a parte elétrica e eletrônica será mantida no painel de comando. Como descrito anteriormente, o painel acomoda os drivers de comando e controle, circuito *dimmer* e a fonte de alimentação. A máquina foi devidamente aterrada, e foi utilizado um disjuntor de 16A para evitar danos às placas em caso de sobrecarga do sistema, além de um fusível de 2,755A para proteger a carga monofásica.

3.8 O Software

3.8.1 Fluxo da aplicação

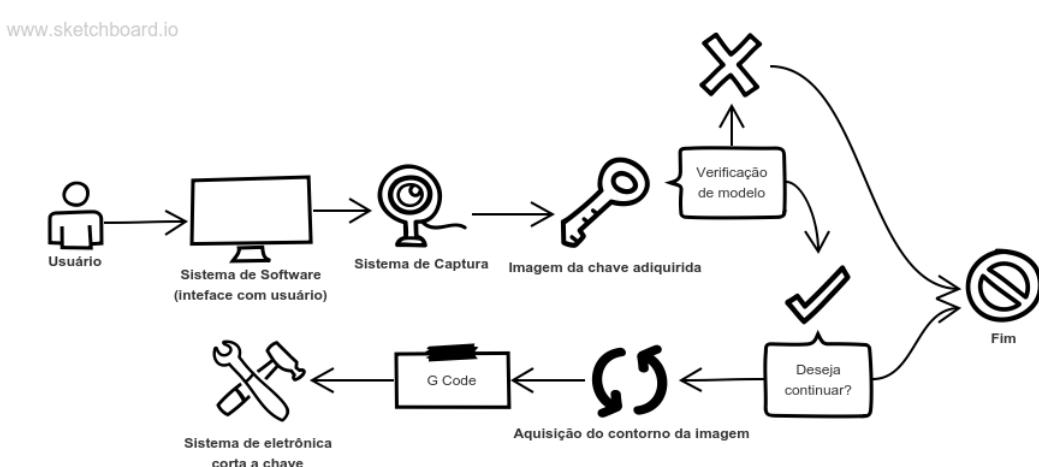


Figura 40 – Fluxo da aplicação e comunicação com outros subsistemas. Fonte: Autores

3.8.2 Tecnologias utilizadas

- **Python 3.5** - linguagem de programação de alto nível, multiparadigma, interpretada e de tipagem dinâmica e forte;
- **Django Framework 1.11** - framework para desenvolvimento web, escrito em Python, open source, que utiliza o padrão model-template-view. (MTV), utiliza por padrão banco de dados Sqlite3;
- **OpenCV 3.3** - A *Open Source Computer Vision Library* (OpenCV) é uma *lib open source* e multiplataforma que possui módulos de processamento de imagem e vídeo.
- **Código G** - O *Código G* é uma linguagem a ser interpretada por uma máquina *Controle Numérico Computadorizado* (CNC).

- **Docker e Docker-Compose** - Docker foi usado para criar um ambiente de desenvolvimento virtualizado e padronizado.
- **Travis** - A integração contínua, integrada diretamente com o repositório, realiza a build e executa os testes. Caso algo algum erro aconteça o Travis envia uma notificação aos desenvolvedores.

3.8.3 Repositório

O repositório pode ser encontrado no seguinte [link](#). Para este projeto, se tornou necessário a criação de uma organização, denominada *EasyKey-PI2-2017-2* no *github* para tornar possível a criação de 2 repositórios, onde um é voltado para o sistema *Django* e o outro é voltado para o código que roda dentro da placa **MSP**.

3.8.4 Licença

Escolhemos para o projeto a licença **MIT**, que é uma licença permissiva. Onde qualquer pessoa que tiver acesso ao código desenvolvido no repositório pode usar, copiar, modificar, mesclar, publicar, distribuir ou vender cópias do *software*, as únicas condições impostas para tanto são apenas manter o aviso de *copyright* e uma cópia da licença em todas as cópias do software.

3.8.5 Diagrama de sequência

Com a realização do sistema, foi visto que a melhor maneira de representá-lo pela sub-equipe de software de forma mais técnica foi através do diagrama de sequência (Figura: [41](#)), onde é possível observar o fluxo normal da aplicação para o corte de uma chave, além das entidades que participam do sistema:

3.8.6 Funcionalidades desenvolvidas

3.8.6.1 Captura da imagem

A imagem que está sendo utilizada nos testes é estática e foi capturada utilizando um aparelho celular. Contudo, foram feitos testes para se obter a imagem através da aplicação pela *Raspberry Pi* e utilizando a biblioteca em *Python* chamada *PiCamera*. Estes testes se mostraram bastante satisfatórios, mesmo em condições de iluminação precárias, como mostrado na Figura [42](#).

A imagem capturada deve conter pelo menos 2 elementos: a chave completa, com seu segredo voltada para o lado esquerdo; e dois pontos a uma distância de 1 centímetro entre eles.

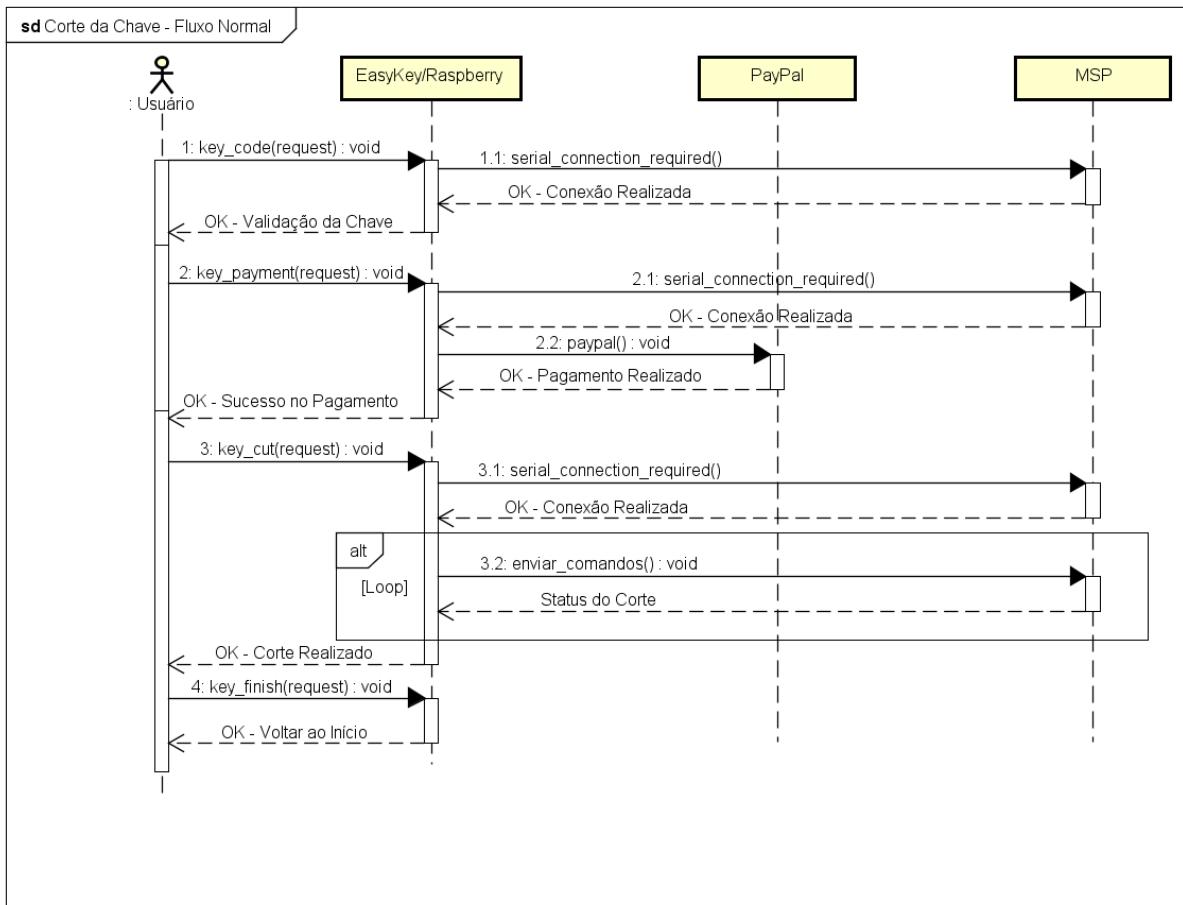


Figura 41 – Diagrama de sequência do EasyKey. Fonte: Autores

A chave deve estar completa para se obter o contorno do segredo, assim como deve conter a sua cabeça, que será utilizada no reconhecimento do modelo de chave (Subseção 3.8.6.3). Seu segredo deve estar voltado para o lado esquerdo para facilitar a aquisição do código G por conta da disposição dos *pixels* no *array* que forma a imagem (Subseção ??).

Os dois pontos servem para se obter a escala de referência e para calcularmos a distância entre cada um dos *pixels* da imagem. Essa escala é usada no cálculo dos pontos cartesianos do código G.

3.8.6.2 Código de captura da imagem

Utilizamos também a estrutura feita em uma impressora 3D para aquisição de imagens projetada pelo subsistema de estrutura, na qual possui uma base para fixação tanto para a *Raspberry Pi* quanto para a *PiCamera* que fará com que as imagens fiquem sempre a mesma altura. O foco pode ser ajustado através das barras rosadas e através da utilização de lentes.

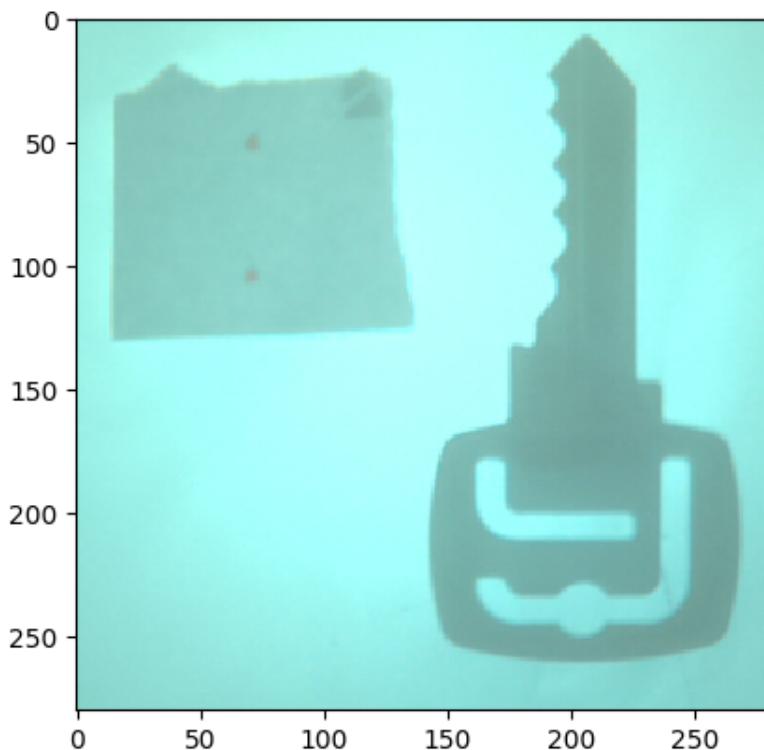


Figura 42 – Imagem tirada da PiCamera através da aplicação. Fonte: Autores

3.8.6.3 Reconhecimento do modelo da chave

Para realizar o desenvolvimento desta funcionalidade utilizou-se o método *Template Matching* do *OpenCV*, que possibilita a partir de uma determinada imagem de modelo, verificar a correspondência de outra imagem com o modelo determinado, caso a imagem seja correspondente, o método retorna um *array* de *arrays* contendo a localização dessa correspondência, caso contrário é retornado um *array* sem informações. No nosso sistema validamos o modelo da chave a partir desse *array*, onde verificamos se há alguma informação ou não, para permitir a continuação do sistema para a próxima etapa que é o tratamento de imagem (Subseção 3.8.6.4).

3.8.6.4 Tratamento de imagem

Após a aquisição da imagem pela *PiCamera*, o reconhecimento correto do modelo da chave e o aceite do usuário para prosseguir com a cópia da chave, a aplicação irá fazer alguns tratamentos na imagem para que seja possível fazer a leitura do segredo.

O primeiro tratamento feito é transformar a imagem em escala de cinza. Como não há necessidade de reconhecer nenhuma outra cor, isso irá diminuir as dimensões do *array* da imagem de três para duas dimensões. Em seguida são feitos dois recortes na imagem em duas áreas conhecidas, que são os locais onde estarão o segredo da chave e os pontos de referência de escala. Ambos os cortes são feitos usando técnicas de *slicing* de *arrays* da

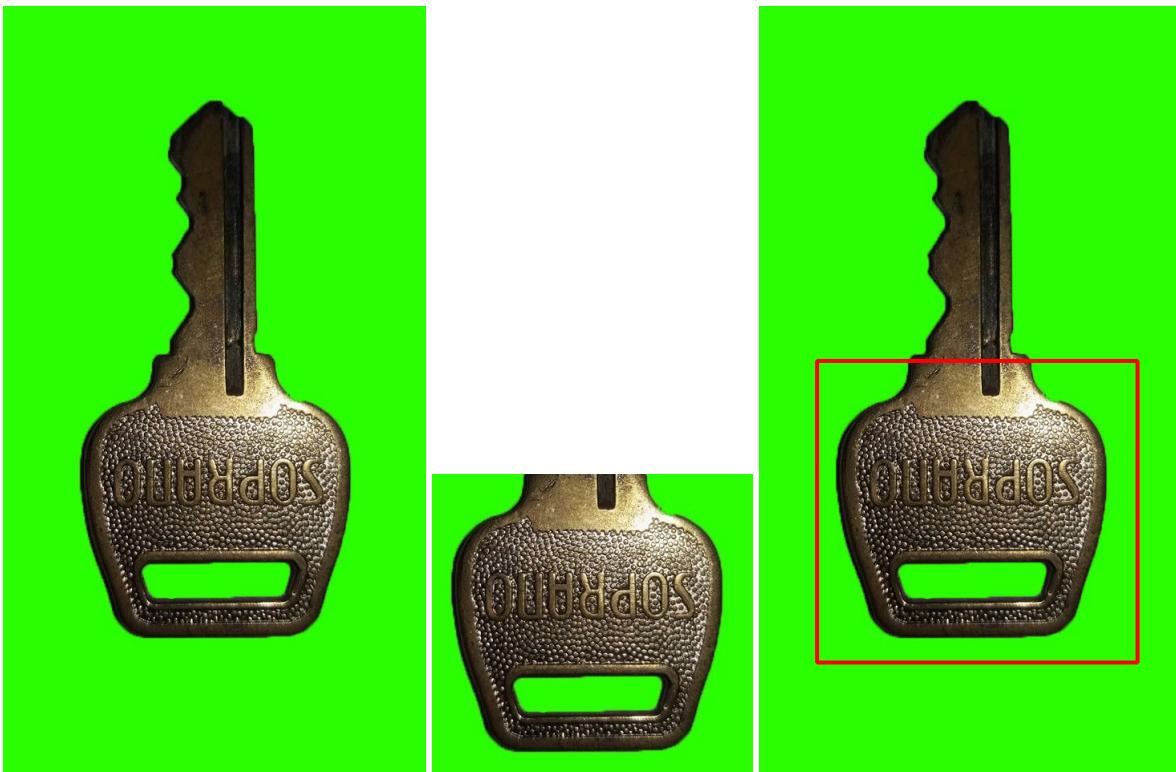


Figura 43 – Exemplo de reconhecimento do modelo da chave com template definido. Fonte: Autores

linguagem *Python*.

Ambas as imagens são borradadas utilizando a função de *cv2.GaussianBlur* para remover possíveis ruídos ([OPENCV, 2015](#)). Após suavizar a imagem usando o *Gaussian Blur*, usamos a função *cv2.threshold* para encontrar os limites binários da imagem ([OPENCV, 2017](#)). Todos os *pixels* que estiver entre os limites inferiores passados para esta função serão coloridos para a cor preta. Todos os *pixels* entre os limites superiores informados serão coloridos para branco.

A figura [44](#) mostra as etapas descritas na ordem de execução, contudo as cores não estão sendo mostradas corretamente. Isso se deve ao fato do *OpenCV* utilizar o sequenciamento de cores BGR, enquanto a biblioteca *Matplotlib*, usada somente para salvar esta figura, usa o sistema RGB.

Ao final do tratamento, teremos então dois *NumPy arrays* de duas dimensões, do tipo *uint8* e que possui apenas *pixels* com valores 0 para preto e 255 para branco. Essas imagens binárias estarão então com o contorno do segredo da chave e com os pontos de referência da escala.

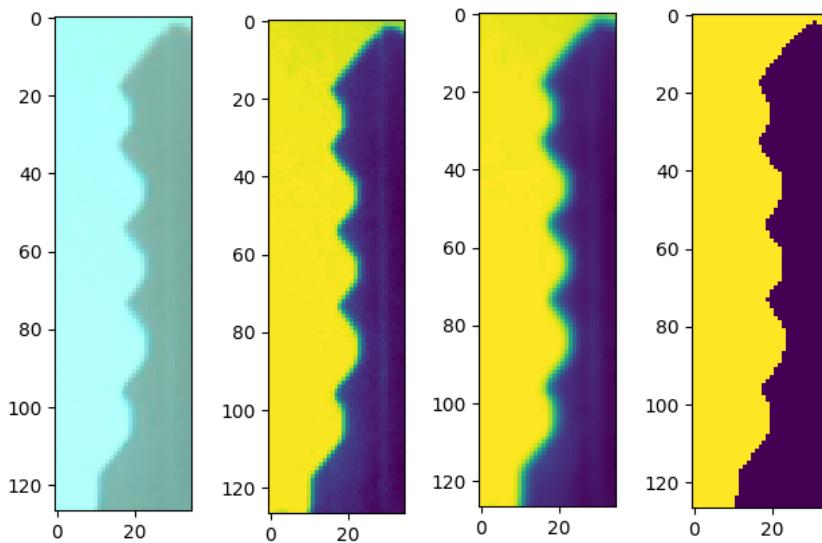


Figura 44 – Etapas do tratamento da imagem para aquisição de código G. Fonte: Autores

3.8.6.5 Conversão da imagem para código G

No desenvolvimento do código para conversão da imagem em código G, foram feitos vários testes, com várias bibliotecas, inclusive de outras linguagens, como *Node.js* e *C++*, além de ferramentas de linha de comando para edição de imagens. Para alguns testes foram feitas conversões das imagens em formato *.jpg* para imagens vetorizadas. Porém, em muitas dessas conversões se perdia qualidade no traçado da chave, dessa forma, desconsideramos a conversão para imagem vetorizada. As bibliotecas de conversão de imagem para código G em 2 dimensões encontradas não tiveram um aproveitamento satisfatório quanto a qualidade do código G esperado, onde algumas bibliotecas não tinha uma opção para definir escala, e apenas apresentavam alguns comandos em código G nos quais não iríamos utilizar, entre outros pequenos problemas. Dessa forma, decidimos realizar a conversão da imagem já pré-processada (Subseção 3.8.6.4), em código G na própria linguagem *Python*. Definimos então que usariamos apenas 3 códigos, que são:

- **G0** - aproximação rápida e sempre será a nossa primeira linha de código. Ela faz uma aproximação rápida da chave para uma posição X e Y bem próxima da fresa
- **G1** - interpolação linear irá mover a chave para uma posição X e Y e fará com que a fresa comece a desbastar a chave virgem.
- **M2** - finaliza o processo de usinagem da chave, desliga o motor da fresa e move a chave para sua posição inicial, próxima do usuário.

Para a criação do código G, utilizamos as imagens binárias resultantes do tratamentos da imagem (Subseção 3.8.6.4).

A primeira tarefa é a de estabelecer uma escala. Para isso, usaremos a imagem que contém os dois pontos separados por um centímetro. A proporção de centímetros por *pixels* é encontrada a partir de uma iteração em toda a matriz da imagem. Encontra-se a posição do primeiro pixel de cor preta do primeiro ponto da imagem e em seguida a posição do último pixel, também de cor preta, do segundo ponto da imagem. A diferença entre o último pixel e o primeiro nos mostra a distância em *pixels* que um centímetro possui.

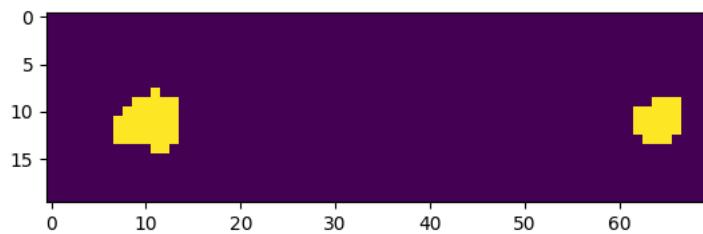


Figura 45 – Exemplo de pontos de referência para medida de escala. Fonte: Autores

Em seguida, percorre-se *pixel* por *pixel* a imagem do segredo da chave para encontrar sua ponta, que possui valor do *pixel* igual a 0. Este ponto (x, y) da matriz da imagem é guardado como referência de ponto zero. Em seguida, dentro de dois laços de repetição aninhados, a aplicação percorre toda a matriz, da esquerda para a direita, até encontrar um pixel preto. Para cada pixel encontrado, o programa irá calcular a diferença entre o ponto zero e o valor (x, y) atual. Esses valores são multiplicados pela escala que foi encontrada na tarefa anterior, que é dada em cm/*pixel* e são escritos em um arquivo chamado *gcode.nc*. Após percorrer toda a imagem, adicionamos o código M2 na última linha do arquivo.

3.8.6.6 Interface com o usuário

A interface com usuário foi feita com o *framework* Django Framework 1.11 em Python 3.5. É um sistema web, porém o usuário não irá utilizá-lo remotamente, apenas pela interface *touch screen* provida pelo subsistema de eletrônica. O navegador não estará visível para o usuário, onde todas as barras de ferramentas e endereços estarão ocultas. No intuito de tornar a interface mais amigável com usuário, foi utilizado um *framework* de CSS chamado Materialize que se baseia no *Material Design* da Google.

3.8.6.7 Sistema de venda

O sistema de venda, que está no modelo de um produto idealizado (Subseção ??), começou a ser feito. Com o desenvolvimento dessa funcionalidade foi acrescentado uma nova etapa na utilização da máquina, que é o módulo de pagamento, este módulo, está sendo sendo desenvolvido em Python 3.5 utilizando a API de pagamentos do *PayPal*. Posteriormente, pretendemos adicionar outros meios de pagamentos a esse módulo.



Figura 46 – Exemplo de tela utilizando materialize. Fonte: Autores



Figura 47 – Tela de pagamento atual apenas com PayPal. Fonte: Autores

3.8.6.8 Testes desenvolvidos

Foram desenvolvidos o total de seis testes unitários, cada teste foi pensado em testar separadamente cada módulo da classe Chave, esta classe é onde há os métodos das principais funcionalidades do sistema.

3.8.6.9 Comunicação com o MSP

Foi desenvolvido o código no sistema *Django* que interage com o código desenvolvido para o *MSP*. Onde o código *python* inicia a comunicação serial com o *MSP*, e envia para o mesmo linha por linha do código *G* gerado através da funcionalidade 3.8.6.5, já no código no *MSP* desenvolvemos uma máquina de estados, denominados: ESPERA, LEITURA, ENVIO e TERMINO. A tabela 10 simplifica o que cada estado quer dizer.

Tabela 10 – Funcionamento dos estados no MSP - Fonte: Autores

Estado	Especificação
ESPERA	Neste estado o MSP fica em espera para começar uma comunicação com o sistema Django, o sistema Django envia o caracter 's' e então o estado é mudado para LEITURA.
LEITURA	Neste estado o sistema <i>Django</i> envia uma string para o MSP contendo uma linha do código G seguido pelo caracter 'f' onde esse caracter significar que foi enviado uma instrução para a movimentação e então o estado é mudado para ENVIO.
ENVIO	Neste estado o código <i>MSP</i> verifica que tipo de instrução de movimentação foi recebida, caso seja uma instrução correta o MSP envia um <i>feedback</i> para o sistema <i>Django</i> com o caracter 'c', em caso de erro o <i>feedback</i> enviado é o caracter 'e', ao verificar que houve um erro o sistema <i>Django</i> reenvia a linha da instrução e então quando chegar ao fim do código G através do comando 'M2', o estado é mudado para TERMINO.
TERMINO	Neste estado o MSP movimenta os motores para o ponto 0 e envia um feedback para o sistema <i>Django</i> contendo o caracter 'q' que representa que o fim da operação e então o estado é mudado para ESPERA.

3.9 Sistema Embarcado e Sensoriamento

3.9.1 Sistema Embarcado

O Projeto EasyKey conta com um sistema embarcado. Sistemas Embarcados são sistemas microprocessados dedicados a uma função específica.

3.9.1.1 MSP430 LaunchPad

Foi necessária a inclusão de um microcontrolador, pois a Raspberry teria problemas com o tempo real para o envio de comandos para os drivers. O Microcontrolador escolhido é capaz de comandar dois motores em paralelo e operar na faixa de 3,3V. O MSP430 LaunchPad foi escolhido por estes fatores e pontos a mais. O LaunchPad possui conversor AD/DA, que será usado pelos sensores, e trabalha com linguagem C.

3.9.1.2 Comunicação Serial

Para a comunicação entre a Raspberry Pi 3 e o MSP, foi usado cabo USB conectados em ambos e o protocolo de comunicação serial com interface UART, em linguagem C. Essa conexão já proporciona alimentação ao LaunchPad.

3.9.2 Sensores

Para um melhor desempenho do projeto, foi utilizado um sensor infravermelho e sensores de fim de curso. Cada um deles atua em uma área diferente, sendo elas segurança,

verificação e controle.

3.9.2.1 Sensor Infravermelho

O sensor infravermelho é composto por dois componentes, sendo eles um LED infravermelho (transmissor) e um fototransistor (receptor). Ele foi colocado na estrutura de tal forma a verificar a posição da tampa (aberta ou fechada). Caso a tampa esteja aberta, o receptor receberá um sinal e fará com que a fresa não possa ser acionada. E se a tampa estiver fechada, o sinal será interrompido, fazendo com que seja possível ligar a fresa. Foi feito o esquemático do circuito do sensor infravermelho no *software* Proteus, como pode ser visto na Figura 48

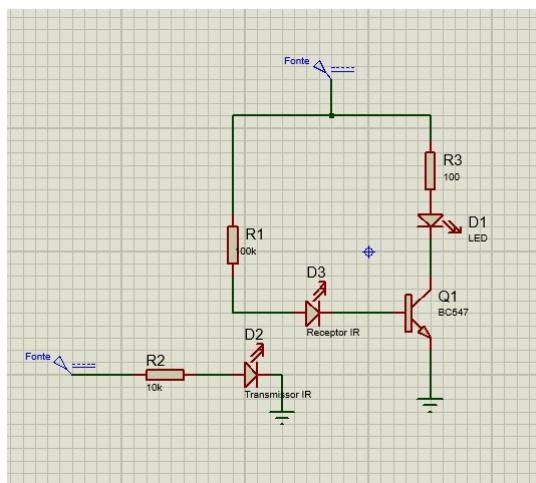


Figura 48 – Esquemático do circuito do sensor infravermelho. Fonte: Autores

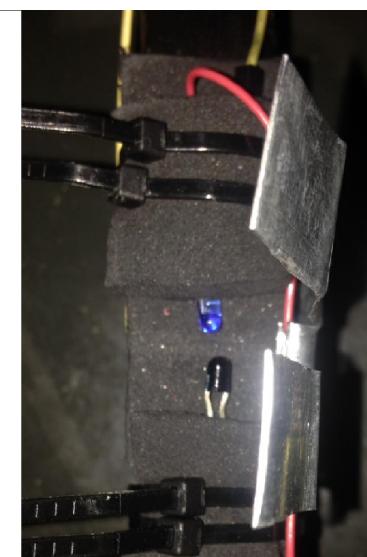


Figura 49 – Sensor Infravermelho. Transmissor (azul) e receptor (preto). Fonte: Autores

3.9.2.2 Chave de fim de curso

A chave de fim de curso serve para limitar o movimento dos motores de passo. Isso ajuda na preservação do motor e da estrutura, pois assim os motores não irão aplicar força desnecessária na parte da estrutura quando eles retornarem para a posição inicial quando a máquina terminar uma cópia de chave solicitada. São dispositivos eletromecânicos e funcionam como interruptor de comando elétricos, ou seja, quando houver a pressão de fim de curso ele emite um sinal ao MSP430 para a tomada de decisão, a figura 50 mostra o componente que é utilizado .



Figura 50 – Modelos de chaves de fim de curso. Fonte: ([MACHADO, 2012](#))

4 Plano de Testes

Este plano diz respeito as atividades realizadas nas rotinas de testes estabelecidas para os subsistemas antes da integração da máquina. O planejamento dos testes, e os riscos foram apresentados no ponto de controle anterior visando a revisão e a verificação de qualquer possível erro, além da validação de todos os subsistemas que estavam em funcionando antes da integração dos subsistemas. Assim, para os eventos inesperados, buscou-se medidas corretivas e mitigadoras para não acarretar prejuízos a integração.

4.1 Objetos de Teste

4.1.1 Estrutura

Um dos procedimentos de teste e verificação estruturais foram as inspeções visuais e de medições, utilizando paquímetro e réguas, verificando a concordância da geometria conforme desenhos técnicos e simulações. Foi realizado ainda a verificação de encaixe de componentes de outros subsistemas, como por exemplo a coincidência de furos de fixação.

O sistema de trilhos foi limpo e lubrificado antes de serem testados. Em seguida, foi utilizado uma furadeira para emular as movimentações dos eixos conforme a Figura 51, verificando o alinhamento entre eles, garantindo o correto funcionamento.



Figura 51 – Teste de movimentação dos eixos utilizando furadeira. Fonte: Autores

4.1.2 Sistema de Alimentação

Foram verificadas as tensões e correntes de saída da fonte ATX, projetada para alimentação do subsistema. No entanto, nos testes para completa verificação durante o processo de integração dos subsistemas, verificou-se que a fonte projetada não supria a demanda de corrente exigida pela máquina. Para corrigir a falha, foi dimensionada nova fonte com maiores valores de tensão e corrente do que a anterior, passando-se a utilizar fonte industrial de 12V e 20A.

4.1.3 Motores, Drivers de Comando e Controle e Circuito Dimmer

Os motores de passo foram testados com drivers convencionais que acompanhavam as especificações destes, juntamente com uma fonte de bancada. Foi verificado que os motores estavam funcionando e em boa condição. Posteriormente, os mesmos motores foram usados para testar os drivers de comando fabricados, também em bancada. Os drivers operaram como o esperado, funcionando em todos os modos de sequência (meio passo e passo completo) e acionamento (reset e control). No entanto, durante o período de integração constatou-se que o motor Nema 17, dimensionado para as condições iniciais do projeto não deslocava a mesa com a precisão e velocidades necessárias para o corte. Por isso optou-se pela troca pelo motor de passo Nema 23, que atendeu as necessidades do projeto. Estes drivers já foram acoplados a estrutura e testados novamente.

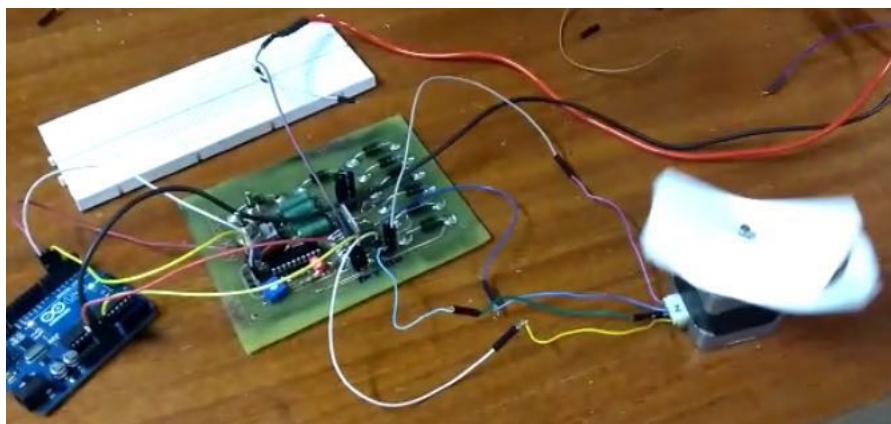


Figura 52 – Teste nos drivers. Fonte: Autores

A máquina fresadora foi testada anteriormente para análise da velocidade de corte, e a velocidade nominal da máquina não prejudicou o fresamento da chave. Entretanto, durante a integração da máquina, a velocidade precisou ser regulada, através do circuito *dimmer* para evitar danos a estrutura da máquina, devido ao excesso de velocidade para fresa-mento da chave. O dimmer foi fixado à estrutura e ligado ao MSP-430 para o acionamento do relé e motor consequentemente. Tudo foi testado.



Figura 53 – Teste do circuito Dimmer. Fonte: Autores

4.1.4 Sensoriamento

Foram feitos diferentes testes para o sensor piezoelétrico e o sensor infravermelho. Para o sensor infravermelho, foi feito o teste prático, onde foi montado o circuito e testado manualmente, como pode ser visto na Figura 54.

O microcontrolador utilizado para testar o sensor infravermelho foi o MSP430, que foi o microcontrolador escolhido para controlar os sensores na versão final do projeto. É utilizada a linguagem C para sua programação e ele possui conversores AD/DA, garantindo fácil portabilidade do uso dos sensores de um para o outro. Foram feitos mais teste, com os sensores já posicionados na estrutura, visando a interrupção da fabricação da chave.

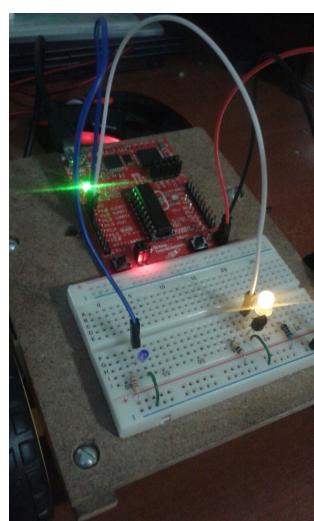


Figura 54 – Teste do sensor infravermelho. Fonte: Autores

4.1.5 Sistemas de Software

Os testes com sistema de software foram feitos durante o desenvolvimento das outras sub-equipes, dessa forma, foi necessário simular algumas estruturas, como a estrutura de tirar as fotos da chave. Foi necessário vários ajustes para que a estrutura improvisada funcionasse de acordo com o esperado, com isso foi realizada a captura de imagem (Subseção 3.8.6.1). Após termos pronto o tratamento de imagem (Subseção 3.8.6.4) e reconhecimento do modelo da chave (Subseção 3.8.6.3), foi utilizada a estrutura improvisada para realização dos testes, visto que a sub-equipe de estrutura não havia entregue a estrutura para realização da captura de imagens.

Após a sub-equipe de software ter concluído o módulo de conversão de imagem para código G (Subseção ??). Realizamos testes no código G com simulações em softwares que simulam o funcionamento de uma CNC. Um dos softwares utilizados para a simulação do código G foi o CutViewer.

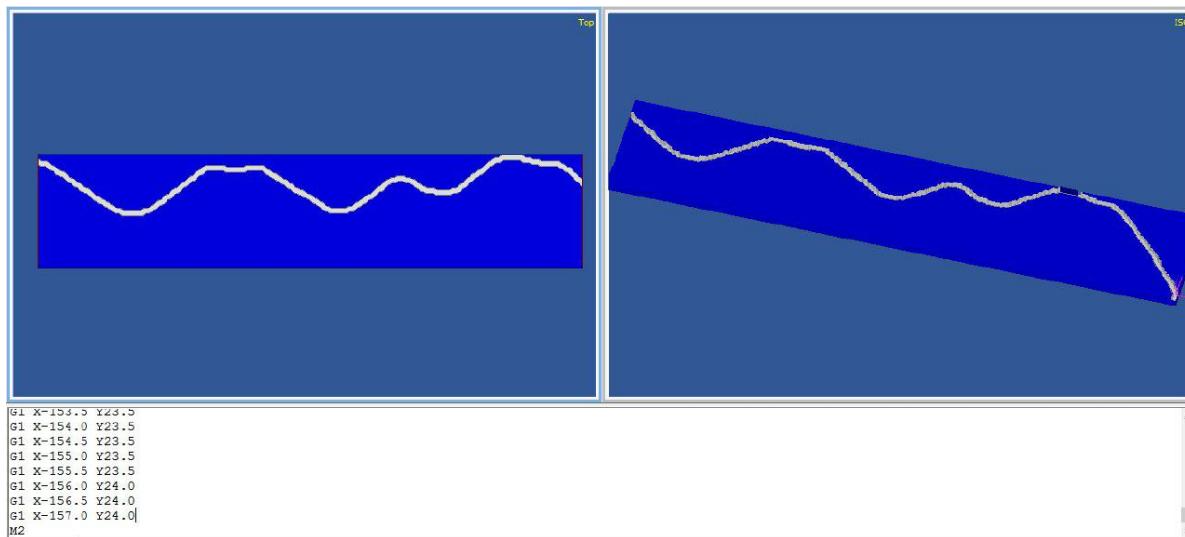


Figura 55 – Imagem da simulação do código G no software CutViewer. Fonte: Autores

5 Integração dos Subsistemas

5.1 Integração dos subsistemas

O plano de integração aborda as etapas para a integração dos subsistemas, assim como as correções e acabamento da máquina como um todo.

A estrutura construída foi testada e aprovada pelas subequipes quanto a preparação e disposição para agregar os demais subsistemas.

Durante a integração, foram abertas perfurações entre a estrutura principal e os módulos, para a passagem das conexões entre os sistemas e alimentadores.

Além disso, foi necessária a troca do suporte do motor de passo do eixo inferior pois a mesa apresentou peso maior do que o esperado, comprometendo a movimentação.

Para finalizar, os eixos foram revisados para garantir o alinhamento do sistema, os trilhos foram novamente lubrificados e a conexão entre o eixo e o motor de passo da mesa superior foi refeita, de modo a garantir a máxima transmissão de torque.

É importante ressaltar que a concepção e construção estrutural foi realizada para que outros subsistemas possam ser adicionados, conectados e testados sem interferência entre si.



Figura 56 – Finalização da estrutura externa. Fonte: Autores

Foram realizados os testes nos motores DC e AC, e ambos apresentaram bom funcionamento. A partir disto, o acoplamento dos motores na estrutura e a estabilização dos componentes na caixa de comando foram feitos para possibilitar a calibração dos motores com os sistemas de software e embarcados. Paralelamente, foram fixados à estrutura os sen-

sores de fim de curso e infravermelho.

O MSP430 LaunchPada foi adicionado para atuar com os sensores e motores. Primeiramente os sensores serão testados no MSP, e calibrados. Posterior a isto, a Raspberry Pi 3 será ligada serialmente com o Launchpad. Eles se comunicarão por interface UART e essa comunicação será testada. Todos os dados de comando serão transmitidos da Raspberry para o MSP430 executá-los. Após os testes e a confirmação de funcionamento, a Raspberry e o Microcontrolador também serão fixados a estrutura

Os drivers de comando e controle foram testados novamente e fixados no painel de comando, juntamente com as fontes de alimentação, dimmer, rele e dispositivos de resfriamento de forma em que todos ficaram estabilizados e isolados. Além do acoplamento dos circuitos mencionados, foi acoplado um disjuntor à máquina e fora do painel de controle para que houvesse uma maior facilidade de acesso, proporcionando mais segurança.

Para a integração da imagem em código G com a execução dos comandos foi necessário um interpretador de Gcode. Esse interpretador foi feito especificamente para a Easy Key, visando ler e gerar comandos para os motores. Baseado em bibliotecas em linguagem C. Ele usa dois tipos básicos, G0 e G1. O G0 trata da parte de posicionamento, portanto ele é mais rápido. O G1 trata do corte em si, logo ele é mais lento que os códigos do tipo G0. Ambos trabalham com os 3 eixos (X, Y e Z) mas apenas com movimentos em linha reta e diagonal, os mesmos que existem no código G da chave. Como foi utilizado o MSP para comandar os motores de passo, o interpretador foi feito em linguagem C.

Com o interpretador criado, foram feitos ajustes em seu código para calibra-lo em acordo com o fuso usado e os demais aspectos da estrutura (sensores de fim de curso) e do funcionamento do EasyKey.

As integrações de software se relacionam principalmente com a sub-equipe de estrutura, para tirar fotos, e com a sub-equipe de engenharia eletrônica que é a responsável pelo desenvolvimento do interpretador de código G.

Com a estrutura de captura de fotos da chave pronta, é necessário fazer os ajustes para que a foto da chave seja sempre capturada da mesma forma, ou seja, a chave tem que estar sempre na mesma posição. Além de replicar os testes feitos na estrutura improvisada. Os testes a serem realizados com a engenharia eletrônica são relacionados ao interpretador, no qual a sub-equipe de eletrônica ainda não concluiu. As sub-equipas de software e eletrônica precisam estar juntas na integração do interpretador principalmente por conta da comunicação dos módulos de software com o interpretador e o controlador da máquina CNC.

Após todas essas etapas concluídas, e a integração final de todos os sub-sistemas, os testes nas chaves foram iniciados.

6 Considerações Finais

As máquinas copiadoras de chaves são uma realidade em países desenvolvidos, porém no Brasil essa tecnologia não é disseminada ainda. O desenvolvimento da máquina EasyKey irá modificar a dinâmica da prestação de serviços de chaves, e facilitará o trabalho tanto para o cliente, quanto para o empresário, por se tratar de um sistema autônomo.

A equipe se preocupou com a pesquisa de mercado, visando entregar uma máquina que fosse compatível com as ferramentas utilizadas pelos profissionais da área. O uso de ambientes virtuais para simulação da máquina, desde a estrutura até a integração computador-máquina e máquina-usuário, foi importante para o planejamento da estratégia de testes, e validar o funcionamento de todos os sistemas, além de ter contribuído para redução dos custos.

Terá início agora a fase de maior comprometimento do projeto, a integração dos subsistemas. Espera-se que cada fase siga o cronograma pré estabelecido e que na presença de qualquer problema, as medidas corretivas sejam tomadas buscando entregar o produto da melhor forma.

O produto final idealizado é uma máquina que se torne uma *vending machine*, na qual todo o serviço de cópia de chaves de cadeados da marca e tamanho já especificados seja feito de forma totalmente autônoma, ou seja, que não precise de outra pessoa a não ser o próprio usuário que deseja realizar a cópia da chave. Seria adicionado ao fluxo de utilização do usuário duas etapas, uma na qual ele realiza o pagamento das cópias que ele irá fazer e outra que o cliente pega a chave fornecida pela máquina e posiciona onde ela deverá ser cortada, para a realização da cópia.

7 Autoavaliação

Felipe Caixeta: Design inicial, Modelagem de componentes, Auxílio aos Processos de Fabricação, Compra de materiais, Simulações, Documentação e Relatório.

João Paulo: Design inicial, Modelagem de componentes, Fabricação e usinagem, Interpretação de simulações e Relatório. relatório.

Laércio Silva: Captura da Imagem da Chave, Reconhecimento do Modelo da Chave, Tratamento de Imagem, Conversão da Imagem Para Código G, Interface Com o Usuário, Relatório, Confecção das PCBs.

Luciano Henrique: Captura da Imagem da Chave, Reconhecimento do Modelo da Chave, Tratamento de Imagem, Conversão da Imagem Para Código G, Interface Com o Usuário, Sistema de Venda, Comunicação com o MSP e Relatório.

Luis Henrique: Captura da Imagem da Chave, Reconhecimento do Modelo da Chave, Tratamento de Imagem, Conversão da Imagem Para Código G, Interface Com o Usuário, Sistema de Venda, Comunicação com o MSP, Testes Unitários e Relatório.

Renato Agnello: Definição, simulação e teste dos circuitos dos sensores, Fabricação da PCI e teste dos drivers dos motores de passo, Teste do circuito regulador de velocidade, Relatório.

Rodrigo Bonifácio: Definição do sensores e simulações, Projeto do layout dos drivers, fabricação da PCI dos drivers, teste dos drivers e motores,relatório.

Samuel Dantas: Simulação do Driver e do Regulador de Velocidade, Fabricação dos Drivers, Teste dos Drivers e Motores, Fonte de Tensão, Sistema de Resfriamento, Sistema de Proteção do lado AC, Relatório.

Tayná Andrade: Confecção do circuito do dimmer, teste dos drivers, estabilidade do painel de controle, confecção do painel de controle e relatório.

Victor Bertolino: Testes dos drivers, teste motores de passo, sistema de ventilação, testes e confecção do circuito dimmer, montagem painel de comando e relatório.

Vinícius Borges: Design inicial, Modelagem de componentes, Fabricação e usinagem, Simulações, Interpretação de simulações, Desenhos Técnicos e Relatório.

Vinícius Lisboa: Integração de Drives, Motores e MSP-430, Simulação e Layout dos Drivers e do Regulador de Velocidade, Fabricação da PCI dos Drivers, Teste dos Drivers e Motores, Relatório.

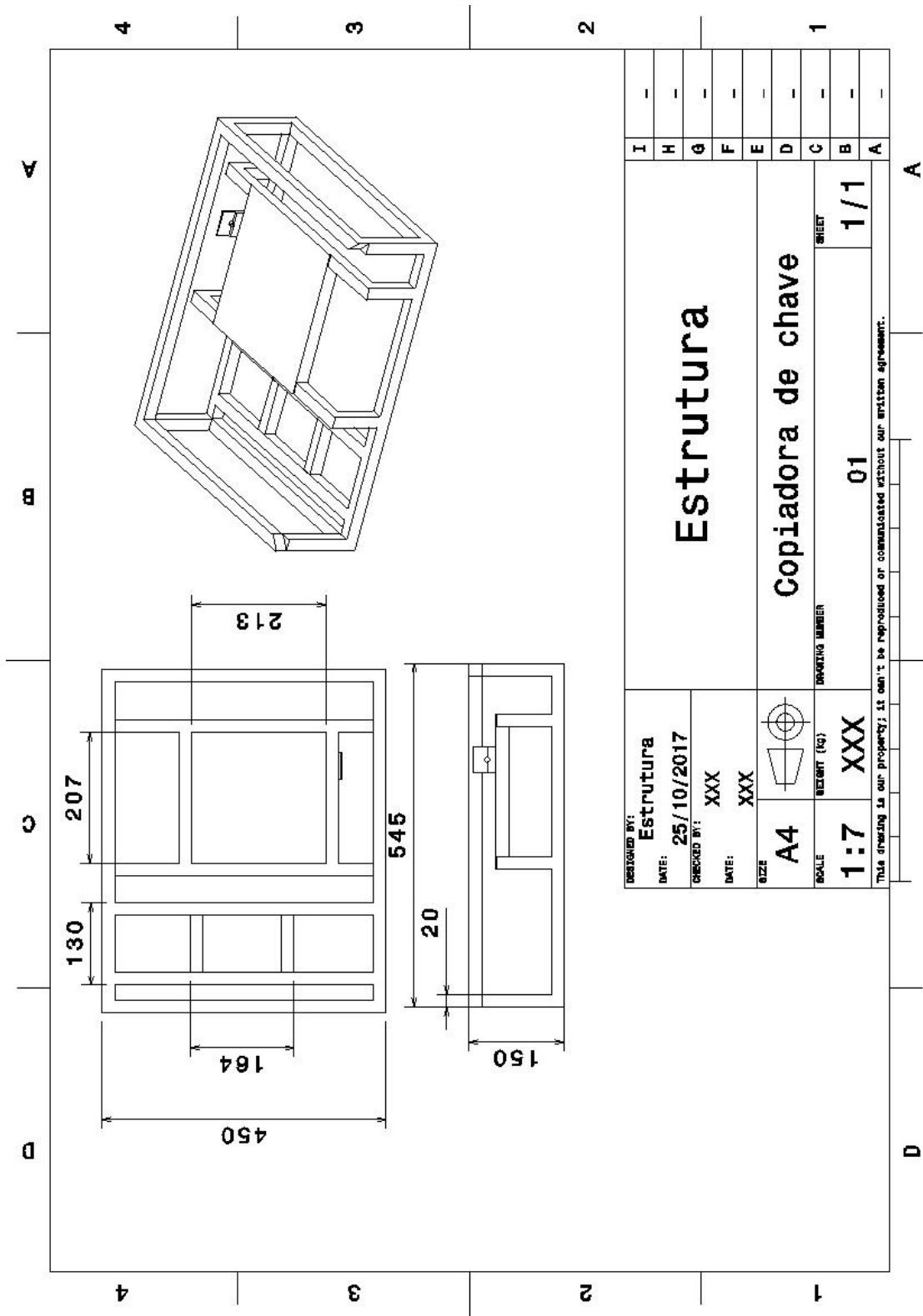
Referências

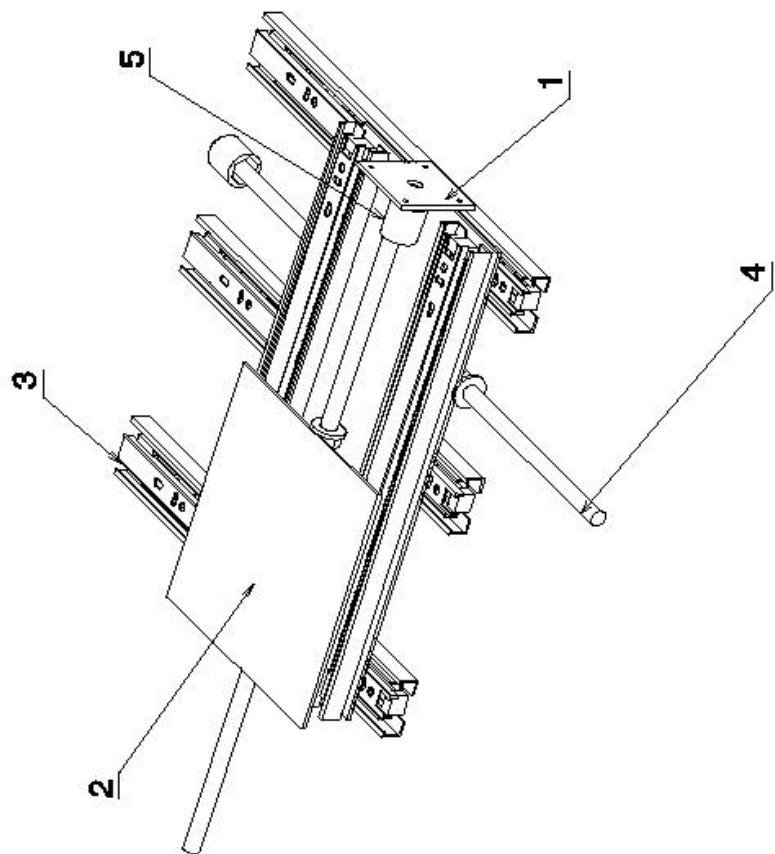
- AKIYAMA, S. e. A. I. *DATASHEET PARA MOTOR DE PASSO MODELO AK23/10F8FN1.8.* 2017. Citado na página 34.
- ANSYS, I. *Manual do software ANSYS 18.0.* 2017. Citado na página 25.
- ARNOLD, S. M. e. a. *Material Selection for Aerospace Systems - NASA Technical Reports, 13-14.* 2012. Disponível em: <<https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20120001798.pdf>>. Acesso em: 20 Setembro 2017. Citado na página 24.
- BRAGA, N. *Controle de velocidade para motores.* 2017. Disponível em: <<http://newtoncbraga.com.br/index.php/artigos/54-dicas/1392-art023.html>>. Acesso em: 15 Outubro 2017. Citado na página 42.
- CONDIT RESTON; JONES, D. W. Stepping motors fundamentals. 2004. Citado na página 33.
- HIBBELER, R. C. *Mecânica Estática.* 8a edição. ed. Rio de Janeiro: Ltc Editora, 1999. Citado na página 21.
- MACHADO, P. R. da S. *Sensor de Pressão.* 2012. Disponível em: <<http://www.ebah.com.br/content/ABAAAA5MsAG/sensores-pressao>>. Acesso em: 12 Setembro. 2017. Citado 2 vezes nas páginas 4 e 54.
- MICROELECTRONICS, I. S. *Stepper Motors Controllers.* 2001. Disponível em: <<http://www.st.com/content/ccc/resource/technical/document/datasheet/f9/35/6e/3f/48/18/48/51/CD00000063.pdf/files/CD00000063.pdf/jcr:content/translations/en.CD00000063.pdf>>. Acesso em: 17 Agosto 2017. Citado na página 38.
- MITSUMI, I. M. *Manual dos motores NEMA 17PM-K.* 2017. Disponível em: <<http://www.eminebea.com/en/product/rotary/steppingmotor/hybrid/standard/data/17pm-k.pdf>>. Acesso em: 03 Outubro 2017. Citado 3 vezes nas páginas 3, 34 e 35.
- OPENCV. *Smoothing Images.* 2015. Disponível em: <https://docs.opencv.org/3.1.0/d4/d13/tutorial_py_filtering.html>. Acesso em: 03 novembro. 2017. Citado na página 48.
- OPENCV. *Image Thresholding.* 2017. Disponível em: <https://docs.opencv.org/trunk/d7/d4d/tutorial_py_thresholding.html>. Acesso em: 03 novembro. 2017. Citado na página 48.
- PAZOS, F. Automacao de sistemas e robotica. Rio de Janeiro Brazil, v. 1 Edicao, 2002. Citado na página 33.
- SHIGLEY, J. E. *Projeto de Engenharia Mecânica.* 8a edição. ed. Porto Alegre: Amgh Editora, 2011. Citado na página 21.
- SINGIRESU, S. R. *Mechanical Vibrations.* Massachusetts: Addison Wesley Publisinhg Company, 1999. Acesso em: 21 Setembro 2017. Citado na página 25.

Apêndices

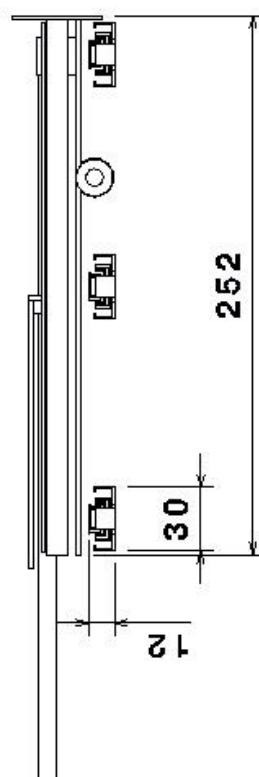
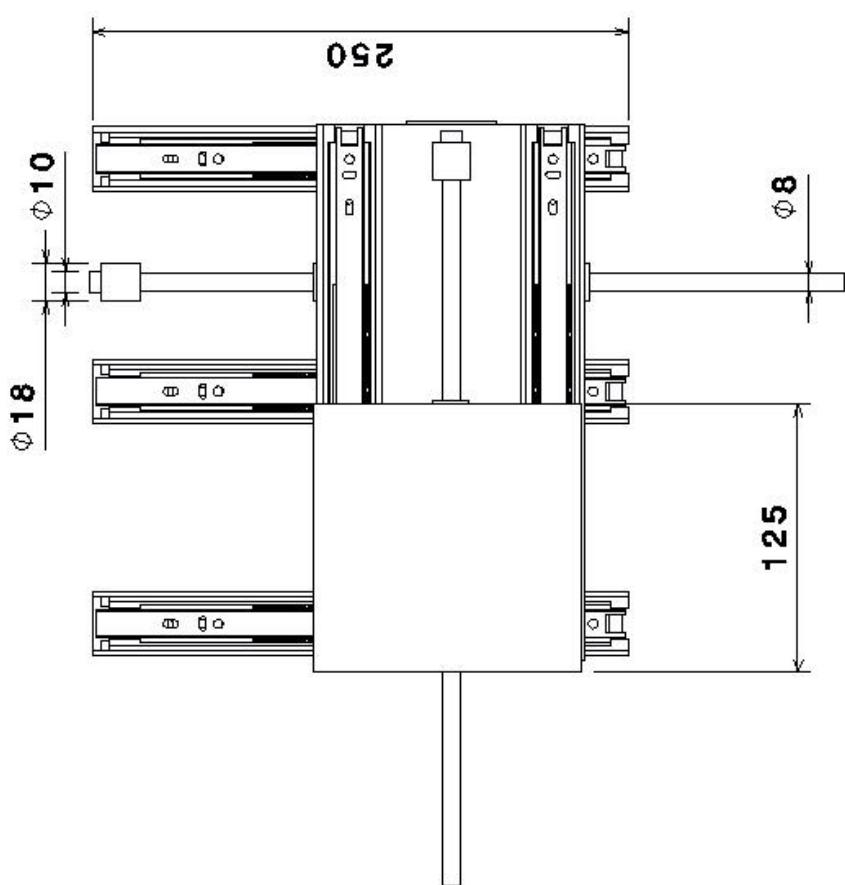
APÊNDICE A – Desenhos Técnicos

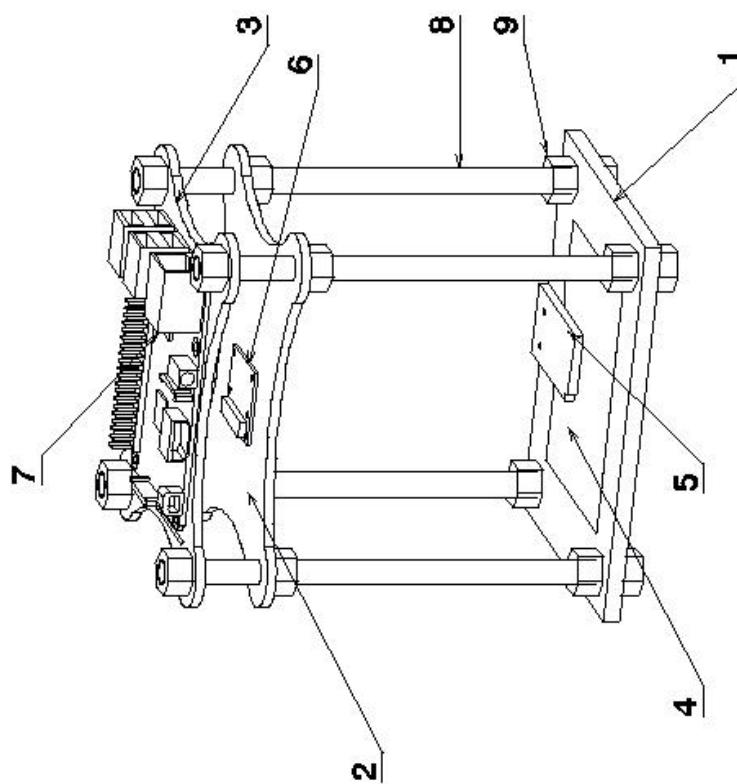
Estruturais





Number	Quantity	Part
1	1	Mesa 1
2	1	Mesa 2
3	5	Trilho
4	2	Fuso
5	2	Acoplamento fuso motor





Number	Quantity	Part
1	1	Base inferior
2	1	Base para câmera
3	1	Base para Raspberry
4	1	Acrílico
5	1	Apoio para chave
6	1	Câmera
7	1	Raspberry
8	4	hastes - barra roscada
9	16	Porca

