



Universidade de Brasília - UnB  
Faculdade UnB Gama - FGA  
Projeto Integrador Engenharia 2

## **Autochopp - Máquina Autônoma de Compra e Serventia de Chopp**

Projeto Autochopp

Orientadores: Alex Reis, Paulo Meirelles, Rhander Viana e  
Sebastièn Rondineau

Brasília, DF  
Agosto de 2017





Clóves Júnior, Edson Gomes, Felipe Côrrea, Filipe Ribeiro, Gabriel Henrique, Gabriela Volpato, Guilherme Matias, Ithallo Junior, Jéssica Brito, Lucas Raposo, Oziel da Silva, Phelipe Wener, Victor Henrique.

## **Autochopp - Máquina Autônoma de Compra e Serventia de Chopp**

Relatório técnico referente à disciplina de Projeto Integrador 2, reunindo os cursos de Engenharias presentes no Campus Gama, da Universidade de Brasília.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Alex Reis, Paulo Meirelles, Rhander Viana e Sébastien  
Rondineau

Brasília, DF  
Agosto de 2017

# **Lista de ilustrações**

Figura 1 – Simulação do sistema de Refrigeração. Fonte: Própria. . . . .	11
Figura 2 – Processo de Montagem do Sistema de Refrigeração. Fonte: Própria. . . . .	12
Figura 3 – Moldagem do chirller. Fonte: Própria. . . . .	13
Figura 4 – Desenho esquemático do chirller. Fonte: Própria. . . . .	13
Figura 5 – Vista em corte da mangueira com o tubo de alumínio inserido. Fonte: Própria. . . . .	13
Figura 6 – Tê adaptado para entrada e saída de chopp. Fonte: Própria. . . . .	14
Figura 7 – Sistema aberto e alocação dos dispositivos. Fonte: Própria. . . . .	15
Figura 8 – Nipel de gás. Fonte: Própria. . . . .	17
Figura 9 – Tubos Desconectados. Fonte: Própria. . . . .	18
Figura 10 – Processo de Flangeamento. Fonte: Própria. . . . .	19
Figura 11 – Recarga do sistema com gás R22. Fonte: Própria. . . . .	20
Figura 12 – Máquina de estados que representa o funcionamento do subsistema de tiragem de chopp. . . . .	24
Figura 13 – Sensor de temperatura DS18B20. Fonte: Própria. . . . .	25
Figura 14 – Esquemático do sensor de temperatura. Imagem da internet. . . . .	25
Figura 15 – Sistema de acionamento de motor de passo. . . . .	27
Figura 16 – Solenoide utilizado. . . . .	27
Figura 17 – Vista Explodida. Fonte: Própria. . . . .	31
Figura 18 – Frame Principal. Fonte: Própria. . . . .	32
Figura 19 – Deformação Frame. Fonte: Própria. . . . .	33
Figura 20 – Deformação Local. Fonte: Própria. . . . .	34
Figura 21 – Fator de Segurança. Fonte: Própria. . . . .	34
Figura 22 – Tensão Local. Fonte: Própria. . . . .	35
Figura 23 – Tensão. Fonte: Própria. . . . .	35
Figura 24 – Deformação da chapa após redimensionamento. Fonte: Própria. . . . .	36
Figura 25 – Perfis Cortados. Fonte: Própria. . . . .	37
Figura 26 – Esmerilhadeira usada para acabamento dos cortes. Fonte: Própria. . . . .	37
Figura 27 – Guilhotina usada no corte de chapas. Fonte: Própria. . . . .	38
Figura 28 – Maquina para soldagem MIG. Fonte: Própria. . . . .	38
Figura 29 – Solda realizada na estrutura. Fonte: Própria. . . . .	39
Figura 30 – Frame principal após o término da soldagem. Fonte: Própria. . . . .	39
Figura 31 – Mecanismo de inclinação do chopp. Fonte: Própria. . . . .	40
Figura 32 – Modelo de Domínio. Fonte: Própria. . . . .	42
Figura 33 – Diagrama de Classes - Models. Fonte: Própria. . . . .	43
Figura 34 – Diagrama de Classes - Controllers. Fonte: Própria. . . . .	44

Figura 35 – Cobertura de Testes. Fonte: Própria.	45
Figura 36 – Estrutura de diretórios do Ionic 3.	49
Figura 37 – Telas do Aplicativo. Fonte: Própria.	50
Figura 38 – Tela do Administrador. Fonte: Própria.	50
Figura 39 – Diagrama de classes do sistema de validação de compras. Fonte: Própria.	52
Figura 40 – Cronograma para a fase de integração de subsistemas. Fonte: Própria.	55

## **Lista de tabelas**

Tabela 1 – Testes Iniciais do Sistema de Refrigeração	16
---	----

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>7</b>
<b>2</b>	<b>ENERGIA</b>	<b>9</b>
<b>2.1</b>	<b>Casos de Teste</b>	<b>9</b>
2.1.1	NoBreak	9
2.1.1.1	Retificador	9
2.1.1.2	Inversor	9
<b>2.2</b>	<b>Sistema de Refrigeração</b>	<b>9</b>
2.2.0.1	Simulação do sistema de Refrigeração	11
2.2.1	Montagem e Construção do Sistema de Refrigeração	12
2.2.1.1	Descrição das etapas	12
2.2.1.1.1	Manufaturação do Chirller	12
2.2.1.1.2	Desmontagem do Sistema	14
2.2.1.1.3	Montagem do Sistema	15
<b>2.3</b>	<b>Sistema de Proteção</b>	<b>16</b>
<b>3</b>	<b>ELETRÔNICA</b>	<b>21</b>
<b>3.1</b>	<b>Requisitos do Projeto</b>	<b>21</b>
<b>3.2</b>	<b>Casos de teste</b>	<b>22</b>
3.2.1	Controle de Temperatura	22
3.2.2	Abertura do Reservatório de Copos	22
3.2.3	Controle de Saída de Chopp	23
<b>3.3</b>	<b>Funcionamento</b>	<b>23</b>
<b>3.4</b>	<b>Solução adotada</b>	<b>24</b>
3.4.1	Controle de temperatura	24
3.4.2	Controle de saída de chopp	25
3.4.2.1	Presença do copo	25
3.4.2.2	Estimar volume no copo	25
3.4.2.3	Garantir o colarinho escolhido	26
3.4.3	Atuação dos motores	26
3.4.4	Abertura do reservatório de copos	27
<b>3.5</b>	<b>Relatório de teste</b>	<b>28</b>
3.5.1	Relatório Controle de Temperatura	28
3.5.1.1	Introdução	28
3.5.1.2	Materiais	28
3.5.1.3	Procedimento Experimental	28

3.5.1.4	Resultados e Discussão . . . . .	28
3.5.1.5	Conclusão . . . . .	28
3.5.2	<b>Relatório Abertura do Compartimento de Copos . . . . .</b>	29
3.5.2.1	Introdução . . . . .	29
3.5.2.2	Materiais . . . . .	29
3.5.2.3	Procedimento Experimental . . . . .	29
3.5.2.4	Conclusão . . . . .	29
3.5.3	<b>Relatório Saída de Chopp . . . . .</b>	29
3.5.3.1	Introdução . . . . .	30
3.5.3.2	Materiais . . . . .	30
3.5.3.3	Procedimento Experimental . . . . .	30
3.5.3.4	Conclusão . . . . .	30
<b>4</b>	<b>ESTRUTURA . . . . .</b>	<b>31</b>
<b>4.1</b>	<b>Levantamento dos esforços a serem levantados . . . . .</b>	<b>31</b>
<b>4.2</b>	<b>Simulação . . . . .</b>	<b>33</b>
<b>4.3</b>	<b>Fabricação . . . . .</b>	<b>36</b>
<b>5</b>	<b>SOFTWARE . . . . .</b>	<b>41</b>
<b>5.1</b>	<b>WebService . . . . .</b>	<b>41</b>
5.1.1	Requisitos do Projeto . . . . .	41
5.1.1.1	Cadastrar usuário . . . . .	41
5.1.1.2	Autenticar usuário . . . . .	41
5.1.1.3	Realizar Pagamento . . . . .	41
5.1.1.4	Gerar Qrcode . . . . .	41
5.1.1.5	Validar qrcode . . . . .	41
5.1.1.6	Mostrar chopp . . . . .	42
5.1.1.7	Salvar dados dos sensores . . . . .	42
5.1.1.8	Mostrar dados dos sensores . . . . .	42
5.1.2	Projeto . . . . .	42
5.1.3	Solução Adotada . . . . .	43
5.1.3.1	Arquitetura . . . . .	43
5.1.3.1.1	Models . . . . .	43
5.1.3.1.2	Controllers . . . . .	44
5.1.4	Casos de Teste . . . . .	45
5.1.5	Gerência e Configuração . . . . .	46
<b>5.2</b>	<b>Aplicativo Mobile . . . . .</b>	<b>46</b>
5.2.1	Requisitos do Projeto . . . . .	46
5.2.1.1	Comprar Chopp . . . . .	46
5.2.1.2	Cadastrar Usuário . . . . .	46

5.2.1.3	Confirmar Cadastro . . . . .	46
5.2.1.4	Finalizar Compra . . . . .	47
5.2.1.5	Listar Cupons(QRCode) . . . . .	47
5.2.1.6	Efetuar Compra Offline . . . . .	47
5.2.1.7	Visualizar Cupons(QRCode) . . . . .	47
5.2.2	Projeto . . . . .	48
5.2.3	Solução Adotada . . . . .	48
5.2.3.1	Arquitetura . . . . .	48
5.2.3.2	Aplicativo . . . . .	48
5.2.4	Casos de Teste . . . . .	48
<b>5.3</b>	<b>Sistema Administrativo</b> . . . . .	<b>48</b>
5.3.1	Aplicativo . . . . .	49
<b>5.4</b>	<b>Sistema de Validação de Compra</b> . . . . .	<b>49</b>
5.4.1	Requisitos do Projeto . . . . .	49
5.4.1.1	Leitura de um cupom . . . . .	49
5.4.1.2	Leitura de um cupom válido . . . . .	50
5.4.1.3	Leitura de um cupom inválido . . . . .	51
5.4.2	Projeto . . . . .	51
5.4.3	Solução Adotada . . . . .	51
5.4.3.1	Arquitetura . . . . .	51
5.4.3.2	Aplicação Final . . . . .	52
5.4.4	Casos de Teste . . . . .	52
5.4.4.1	Leitura do Qrcode . . . . .	52
5.4.4.2	Validação do Qrcode . . . . .	53
5.4.4.3	Integração com o eletrônica . . . . .	53
<b>5.5</b>	<b>Subsistemas integrados</b> . . . . .	<b>53</b>
<b>6</b>	<b>PLANO DE INTEGRAÇÃO</b> . . . . .	<b>55</b>
<b>7</b>	<b>CONSIDERAÇÕES FINAIS</b> . . . . .	<b>57</b>

# 1 Introdução

Neste relatório é apresentado o andamento e o desenvolvimento do projeto Auto-Chopp que ocorreu entre os pontos de controle 1 e 2. O foco será dado em questões mais práticas, levantando os resultados obtidos e o que se espera pro decorrer do projeto.

O objetivo desta fase foi o desenvolvimento dos subsistemas separadamente, levando em consideração a segurança e viabilidade de cada um, bem como a sua qualidade final. Eles foram desenvolvidos em paralelo, porém, com foco na integração de cada um, que será a próxima fase do projeto.

Desta forma, este relatório trata sobre os resultados obtidos ao longo das semanas de implementação do ponto de controle 2.



## 2 Energia

### 2.1 Casos de Teste

#### 2.1.1 NoBreak

##### 2.1.1.1 Retificador

- **Pré Condição:**
  - Tensão 220v Alternada;
  - Frequência 60Hz
- **Saída:** Tensão 12v Contínuo
- **Fluxo de Eventos:** Transformação para 16v alternada; Retificação por diodos; Carga capacitiva para diminuição de Ripple; Diodo Zener para regulação de Tensão.

##### 2.1.1.2 Inversor

- **Pré Condição:** Tensão 12V contínua
- **Saída:**
  - Frequência 60Hz
  - Tensão 220v Alternada
- **Fluxo de Eventos:** Chaveamento de Transistores em ciclos diferentes (60Hz); Transformação de 12+12v para 220v Alternado.

### 2.2 Sistema de Refrigeração

Na parte de planejamento do sistema de refrigeração, foram feitos todos os cálculos referentes ao sistema.

O dimensionamento da potência do compressor, necessário no ciclo de refrigeração foi feito a partir da Primeira Lei da Termodinâmica, onde calculou-se a carga térmica necessária a ser retirada do chopp, considerando que a vazão necessária a ser retirada da chopeira é 45l/h, ou aproximadamente,  $0,00125\text{m}^3/\text{s}$ , a Primeira Lei para o compressor será:

$$Q_{(ponto)} = m_{(ponto)} \times (h_1 - h_2) \quad (2.1)$$

Onde:

- Q = Quantidade de calor
- m = fluxo mássico (vazão multiplicada pela densidade)
- h = entalpia

Manipulando, têm-se:

$$h_1 - h_2 = Cp_{méd} \times (T_1 - T_2) \quad (2.2)$$

Onde:

- $Cp_{mádi}$  = Calor Específico a pressão constante
- T = Temperatura do fluido

então, substituindo na primeira equação teremos:

$$Q_{ponto} = m_{ponto} \times Cp_{méd} \times (T_1 - T_2) \quad (2.3)$$

Consideramos como fluido para os cálculos a água, visto que o chopp tem bastante água na sua composição e que não foram encontrados dados para o calor específico do chopp, a aproximação é aceitável. O valor de calor específico a pressão constante encontrado para a água é de 4,18KJ/Kg. A temperatura considerada na entrada do trocador de calor é a mesma que a temperatura ambiente de aproximadamente 25°C, e a requerida na saída dele para estar de acordo com os requisitos é de aproximadamente 1°C. Dessa forma, o fluxo mássico:

$$m_{ponto} = p \times V \quad (2.4)$$

$$m_{ponto} = 1000 \times 0,00125 \quad (2.5)$$

$$m_{ponto} = 0,0125 \quad (2.6)$$

Logo, a carga térmica a ser retirada do fluido, no trocador de calor, pelo gás que sai do compressor será:

$$Q_{ponto} = 0.0125 \times 4.18 \times (25 - 1(-1)) = 1.35\text{KW} \quad (2.7)$$

Ou então:

$$46.32.48\text{Btu/h} \quad (2.8)$$

Analizando o catálogo (nome), para essa carga térmica seria necessário um compressor de  $\frac{1}{2}$  Hp.

#### 2.2.0.1 Simulação do sistema de Refrigeração

Para a segurança do funcionamento do sistema de refrigeração usou-se o Software CoolPack para a simulação do sistema como um todo. Nesse software dados construtivos físicos dos seguintes elementos foram inseridos: Evaporador, Condensador, moto-compressor e tubulação.

A figura 2 ilustra a simulação realizada no software juntamente com os dados das temperaturas em cada estado. Essas temperaturas serão importantes para o devido entendimento da troca de calor entre evaporador e o líquido externo ao sistema a ser refrigerado.

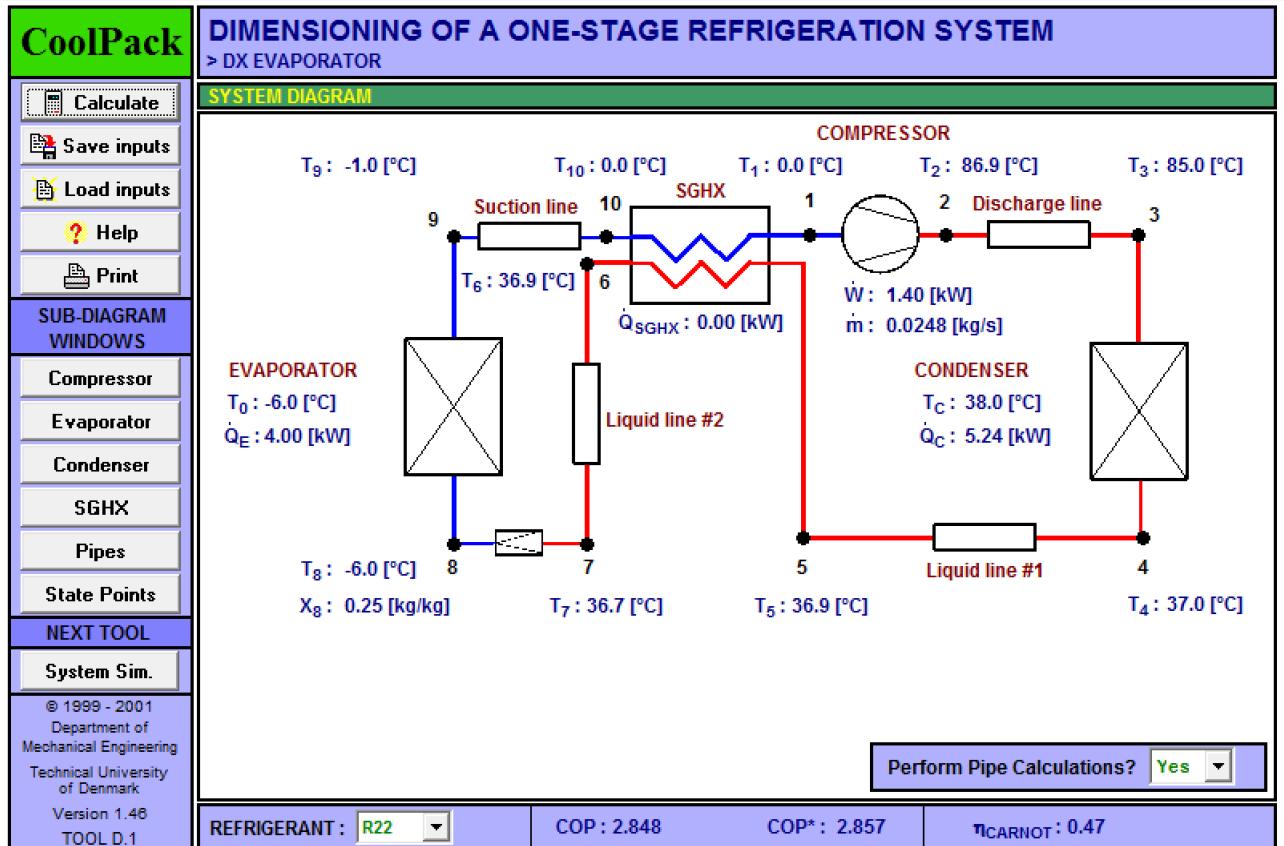


Figura 1 – Simulação do sistema de Refrigeração. Fonte: Própria.

Observa-se na imagem acima que foi inserido o dado correto quanto ao tipo de gás em que o compressor opera. Tem-se como resultado importante a quantidade de energia, ou seja, calor transferida no evaporador que corresponde a 4.00 kW. Esse dado será validado nos testes do sistema já montado.

### 2.2.1 Montagem e Construção do Sistema de Refrigeração

A montagem foi feita com apenas um chiller no protótipo por motivos de custo, sendo assim, a potência do compressor foi reduzida pela metade, ou seja,  $\frac{1}{4}$  Hp. O compressor foi cedido pelo professor Rander, para viabilizar que o protótipo fosse fabricado, as especificações desse compressor são as seguintes:

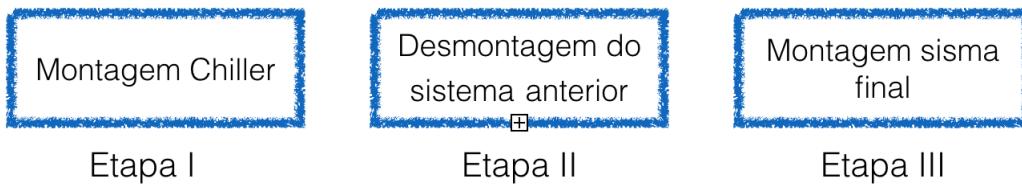


Figura 2 – Processo de Montagem do Sistema de Refrigeração. Fonte: Própria.

#### 2.2.1.1 Descrição das etapas

Para a montagem do sistema de refrigeração teve que seguir as seguintes etapas:

##### 2.2.1.1.1 Manufaturação do Chirller

Neste processo, os materiais utilizados foram: uma panqueca de alumínio  $\frac{3}{8}$  e 15 metros, uma mangueira atóxica trançada  $\frac{3}{4}$ , 2 kits de engate rápido para mangueira, 2 reduções de  $\frac{3}{4}$  para  $\frac{1}{2}$ , 2 tês de PVC com extremidades de  $\frac{3}{4}$  e central de  $\frac{1}{2}$ , 2 espigões de mangueira de  $\frac{3}{4}$  e 4 nipes de união de  $\frac{1}{2}$ , 2 braçadeiras de  $\frac{3}{4}$  e 2 adaptadores de gás de  $\frac{1}{2}$  para  $\frac{3}{8}$ . Primeiro, lavou-se a mangueira com água e sabão, feito isso, desenrolou-se a panqueca em todo o seu comprimento, de forma que ela ficasse bem reta.

Na segunda parte do processo, o tubo de alumínio foi inserido dentro da mangueira, ainda de forma que que a estrutura ficasse reta. Depois, enrolou-se a estrutura no molde, que tem 30cm de diâmetro, prendendo ele com lacres de plástico. A imagem a seguir ilustra esse processo, feito por membros do grupo:

Foi feito um desenho esquemático do Chirller no software CatiaV5 para possibilitar as simulações, ele está ilustrado na figura 4.



Figura 3 – Moldagem do chirller. Fonte: Própria.

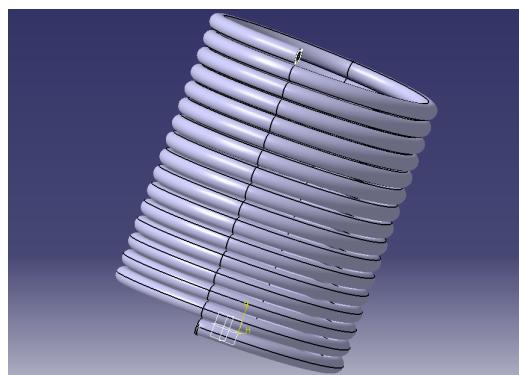


Figura 4 – Desenho esquemático do chirller. Fonte: Própria.

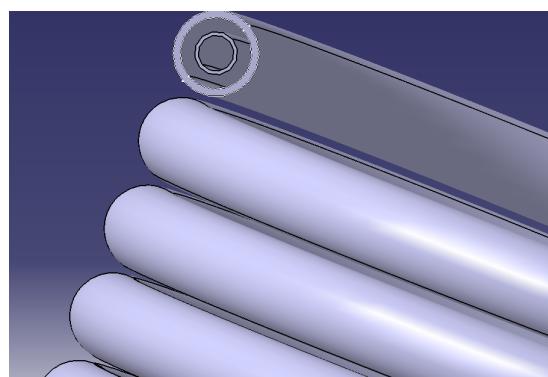


Figura 5 – Vista em corte da mangueira com o tubo de alumínio inserido. Fonte: Própria.

Na terceira parte da passagem, passou-se fita veda-rosca em todos os acessórios de tubulação, para evitar vazamentos, e montou-se o tê adaptado que possibilita a entrada e saída de chopp no Chirller de contra-fluxo. Neste processo, coloca-se o espião em uma

extremidade de  $\frac{3}{4}$  do tê, um nipple de união de  $\frac{1}{2}$  na extremidade de  $\frac{1}{2}$  do tê, um engate rápido no nipple. Na outra extremidade de  $\frac{3}{4}$ , coloca-se uma redução de  $\frac{3}{4}$  pra  $\frac{1}{2}$ , coloca-se um nipple de união e por fim, coloca-se um adaptador de gás no nipple. Depois de montado o tê adaptado ficou da seguinte maneira:



Figura 6 – Tê adaptado para entrada e saída de chopp. Fonte: Própria.

#### 2.2.1.1.2 Desmontagem do Sistema

Nessa etapa teve-se o devido cuidado para a desmontagem do sistema. Retirou-se o gás R22 com o equipamento próprio para tal ação, considerando assim a lei ambiental quanto a emissão de gases poluentes na atmosfera. Dessa forma, na válvula de serviço do compressor a válvula de descarga do fluido refrigerante encaminhando-o para o devido recipiente de armazenamento de gás a vacuo. Desse modo realizou-se o procedimento de recolhimento passivo. Esse procedimento é indicado para quantidades pequenas de gas. Ele pode ser retirado na sua forma gasosa ou líquida. Isso acontece devido a diferença de pressão entre os dois sistemas.

Esse procedimento foi realizado a fim de atender a Resolução CONAMA no 267, de 14 de setembro de 2000. Essa resolução proíbe a utilização e consequentemente emissão de substâncias que destroem a camada de ozônio.

#### 2.2.1.1.3 Montagem do Sistema

Após a devida manufatura do Chiller e a desmontagem do sistema escolheu-se uma base e alocou-se os componentes de acordo com a figura 7



Figura 7 – Sistema aberto e alocação dos dispositivos. Fonte: Própria.

A base para o sistema consiste em uma de material madeira. Esse material foi escolhido devido às suas propriedades térmicas e pela fácil disponibilidade desse.

Realizou-se testes de pressão no motocompressor e constatou-se que esse encontrava-se com baixa capacidade de compressão. Sendo assim realizou-se a limpeza do motor e também a troca de do filtro do sistema para o filtro secador do tipo Darfur. Este filtro impede com que partículas indesejadas passem pela a tubulação e cheguem no motocompressor. Além disso, como o filtro secador foi trocado optou-se, por medidas preventivas, trocar o filtro capilar.

A tubulação do Chiller escolhido consiste de alumínio e as demais tubulações do sistema são de cobre. Assim sendo, tornando inviável a solda desses dispositivos como solução. A partir disso, optou-se pela aquisição da junção de gás (nipel) com a cabeça em forma de cilindro ilustrado na figura 8.

Assim com a peça ilustrada acima realizou-se o processo de flangeamento dos tubos. Esse processo consiste em alargar a espessura do tubo de cobre para que o tubo de alumínio seja inserido e juntado pela peça acima não havendo vazamento de gás. O processo de flangeamento é ilustrado na figura 10.

Para a continuação da montagem do sistema, realizou-se a solda do tubo capilar e filtro secador. Assim, com o sistema todo soldado e montado foi feita a recarga de gás. Antes de abastecer o sistema, ligou-se o motocompressor para a realização de uma câmara de vácuo dentro das tubulações. Depois dessa verificação conectou-se o manifold nos diferentes pontos do sistema, parte de baixa e alta pressão e conectou-se a mangueira de abastecimento permitindo a passagem de fluido refrigerante. A carga foi realizada até que o sistema não suportasse mais gás dentro desse. A figura 11 mostra a recarga do sistema com o fluido refrigerante R22.

Realizou-se os testes e notou-se que o condensador aplicado no sistema estava muito pequeno para o evaporador escolhido. Assim sendo, repetiu-se todos os passos acima quanto a soldagem dos tubos e a recarga de gás. Tendo assim o sistema de refrigeração abaixo.

Com o sistema pronto realizou-se testes e a partir desses montou-se a seguinte tabela para o sistema de refrigeração:

Tabela 1 – Testes Iniciais do Sistema de Refrigeração

Capacidade Efetiva de Chopp	2,5 Litros
Tempo de Aquecimento do Sistema	29 minutos
Tempo de Refrigeração	20 - 37 minutos



Figura 8 – Nipel de gás. Fonte: Própria.



Figura 9 – Tubos Desconectados. Fonte: Própria.



Figura 10 – Processo de Flangeamento. Fonte: Própria.



Figura 11 – Recarga do sistema com gás R22. Fonte: Própria.

## 2.3 Sistema de Proteção



# 3 Eletrônica

A construção de uma máquina de venda autônoma de chopp depende da construção de diversos subsistemas, que correspondem aos requisitos do projeto. Assim, seguem os requisitos do projeto.

- Controle de temperatura
- Abertura do reservatório de copos
- Controle de saída de chopp

Os códigos utilizados neste trabalho encontram-se em controle de versão na ferramenta GitHub e podem ser facilmente acessados em: <[https://github.com/autochopp/embedded\\_electronics](https://github.com/autochopp/embedded_electronics)>.

## 3.1 Requisitos do Projeto

### 1. Controle de temperatua

**Dado** que a leitura da temperatura do sensor de temperatura

**Quando** a temperatura estiver fora da faixa estabelecida

**Então** acionar relé de controle do compressor

### 2. Abertura do reservatório de copos

**Dado** que a leitura da temperatura do sensor seja maior 2°C

**Quando** a temperatura estiver fora da faixa estabelecida

**Então** acionar ou não relé

### 3. Controle da saída de chopp

#### a) Identificar a presença do copo na base

**Dado** que o usuário foi solicitado para colocar o copo na base

**Quando** os contadores do *timeout* se iniciarem

**Então** verificar se o copo se encontra na base

#### b) Controlar a inclinação do copo

**Dado** que a compra foi liberada

**Quando** o copo se encontra na base

**Então** acionar motores que inclinam a base

c) Estimar volume de chopp no copo

**Dado** que o usuário escolheu um volume de copo

**Quando** houver tiragem de chopp

**Então** esperar o tempo necessário para tirar o volume esperado

**Ou** esperar o sinal de parada do sensor de presença

d) Estimar volume restante no barril

**Dado** que se sabe o volume presente no barril em um dado instante

**Quando** houver tiragem de chopp

**Então** subtrair valor da tiragem do valor do volume anterior

e) Garantir o colarinho escolhido

**Dado** que o usuário escolheu um determinado colarinho

**Quando** a compra foi liberada e o chopp foi tirado

**Então** acionar ou não motor que controla o colarinho

## 3.2 Casos de teste

Neste tópico está descrito como foram elaborados os casos de testes para os subsistemas da área eletrônica, tais testes visam buscar defeitos na sequência de funcionamento e na estrutura de componentes.

### 3.2.1 Controle de Temperatura

- Resumo: Este caso descreve como o subsistema de controle de temperatura irá atuar em resposta a estímulos e suas ações.
- Pré-condição: Existir um resfriamento da serpentina, o líquido presente deve estar em contato com o sensor de temperatura.
- Entrada: Temperatura lida através do sensor DS18B20.
- Ação: Acionar e desligar o compressor responsável pelo resfriamento do chopp.
- Resultados Esperados: Entregar o chopp ao sistema de saída de chopp na temperatura correta.

### 3.2.2 Abertura do Reservatório de Copos

- Resumo: Será aqui descrito como e quando o reservatório de copos irá ser acionado.
- Pré-condição: Copos no compartimento e chopp gelado.

- Entrada: Validação de *QR code* autenticada via software.
- Ação: Atuar, através de módulos relés, nas solenoides que atuam no compartimento de copos.
- Resultados: Copo liberado para ser posicionado pelo usuário.

### 3.2.3 Controle de Saída de Chopp

- Resumo: Caso de teste para o controle de saída de chopp idealizando seu funcionamento.
- Pré-condição: Chopp gelado entregue pelo sistema de resfriamento, copo liberado e posicionado.
- Entrada: Posicionamento do copo no suporte.
- Ação: Inclinação do suporte para correta tiragem e leitura da altura de chopp presente no copo.
- Resultados: Chopp com servido com a quantidade requisitada de colarinho e volumes corretos.

## 3.3 Funcionamento

A Figura 12 representa o funcionamento e a interação dos sistemas de eletrônica por meio de uma máquina de estados. Nela, as etapas de liberação do copo, verificação da posição do copo, inclinação do copo, tiragem do chopp, retorno do copo a posição inicial e inserção do colarinho são mostradas.

Inicialmente, a máquina aguarda a solicitação de tiragem de chopp, bem como a confirmação/autenticação de compra. Após ter isso confirmado, os dados referentes ao tipo de tiragem são processados (tamanho do copo, quantidade de colarinho e tipo de chopp) e as configurações para uma tiragem correta são feitas.

Logo após isso, a máquina aciona a solenoide que libera o copo apropriado ao tamanho de chopp selecionado e aguarda que o usuário posicione o copo no suporte da máquina. O sistema foi pensado de modo que o usuário tem um tempo limite para posicionar o copo no suporte, após o qual a tiragem é desconsiderada e a máquina volta ao seu estado inicial de espera.

Caso o copo seja posicionado no suporte, inicia-se o processo de tiragem de chopp. Nesse processo, o suporte do copo é inclinado por meio do acionamento do motor e a válvula que libera o chopp é acionada. Passa-se a próxima etapa no caso de um de dois

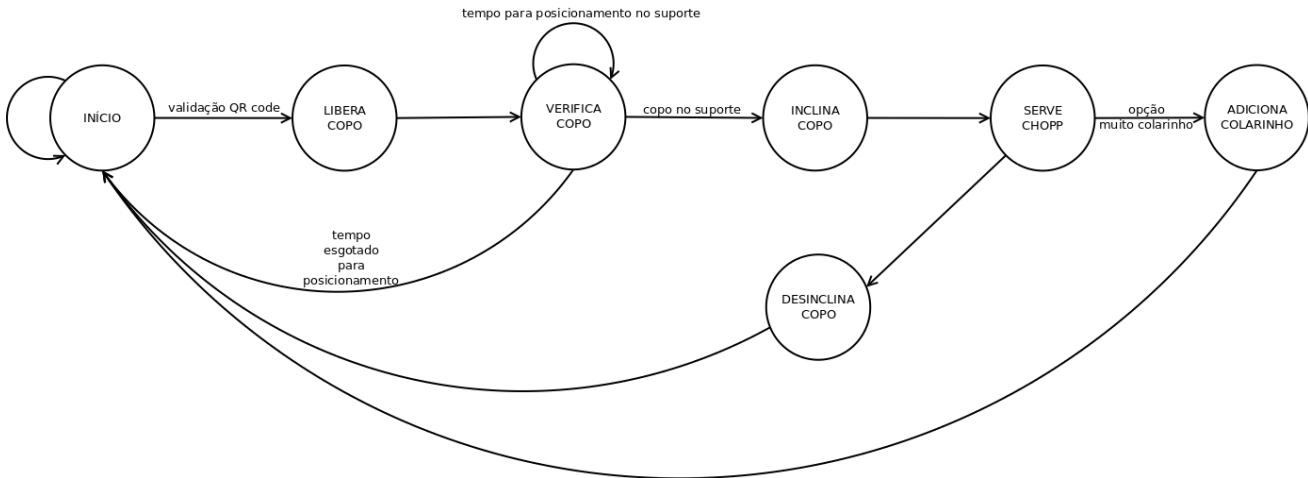


Figura 12 – Máquina de estados que representa o funcionamento do subsistema de tiragem de chopp.

eventos ocorrer: o tempo máximo de tiragem ser excedido ou o volume tirado chegar ao limiar de altura estabelecido no copo.

Se o usuário tiver selecionado colarinho, o motor que libera uma maior entrada de ar no sistema e cria o colarinho é acionado. Após isso, a válvula de chopp é fechada, o copo retorna a sua posição inicial e máquina retorna ao seu estado de espera por uma nova requisição.

O sistema de refrigeração encontra-se funcionando independentemente dos processos de venda/tiragem de chopp e ocorre de forma contínua e ininterrupta.

## 3.4 Solução adotada

### 3.4.1 Controle de temperatura

Para realizar o controle da temperatura, permitindo que o chopp esteja sempre gelado quando da retirada, é utilizado o sensor de temperatura DS18B20 presente na Figura 13. Ele permite a leitura de temperatura mesmo em ambientes úmidos, tal qual esta aplicação. O sensor descrito mede temperaturas entre  $-55^{\circ}\text{C}$  e  $125^{\circ}\text{C}$  com erro de  $\pm 0.5^{\circ}\text{C}$ . O sensor encontra-se posicionado na saída de chopp, que permite uma leitura mais fidedigna da temperatura real do chopp, uma vez que na saída o chopp já circulou por toda a serpentina e está em equilíbrio térmico. O modo como esse sensor é conectado pode ser visto na Figura 14.

A medição da temperatura permite a ativação do compressor mantendo a temperatura na faixa ideal entre  $-1$  e  $1^{\circ}\text{C}$ . Tal ativação será realizada pelo acionamento de um módulo relé ligado ao compressor.

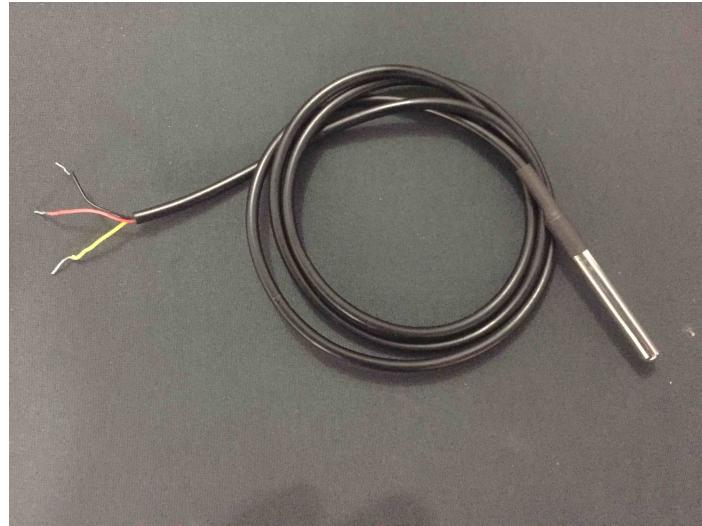


Figura 13 – Sensor de temperatura DS18B20. Fonte: Própria.

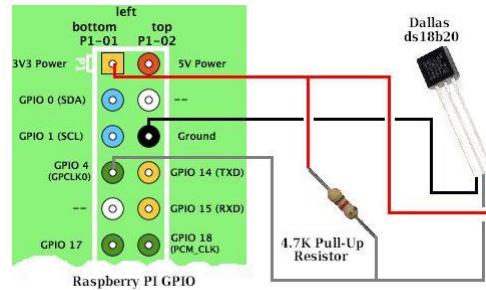


Figura 14 – Esquemático do sensor de temperatura. Imagem da internet.

### 3.4.2 Controle de saída de chopp

#### 3.4.2.1 Presença do copo

Há a necessidade de se verificar a presença do copo no sistema para que o chopp possa ser tirado. Para tanto, inicialmente utilizou-se o sensor ultrassônico HC-SR04, porém, o mesmo inviabilizou seu uso devido a complexidade de sua comunicação com o computador (*Raspberry pi*).

Como solução mais direta e que evitaria tal nível de complexidade, optou-se por utilizar um sensor de trilha, que se comunica de forma direta com o computador e retorna um valor binário para a presença ou não do copo. Essa escolha traz como consequências a necessidade de uma tira branca no copo e a necessidade de uso de um copo transparente, para não interferir com os sensores.

#### 3.4.2.2 Estimar volume no copo

Para que se pudesse ter um controle do volume presente no copo e consequentemente volume restante nos reservatórios, inicialmente pensou-se em utilizar dois sensores

(por redundância), sendo eles o sensor de fluxo YFS201 e uma célula de carga.

Obtiveram-se resultados satisfatórios na montagem do sensor de fluxo, porém verificou-se um erro muito grande em suas medidas, o qual chegava a dez porcento nos testes efetuados. Já quanto ao sensor de carga, não houve sucesso em sua implementação, apresentando resultados quase que aleatórios.

Após discutir o presente problema com professores e orientadores, outra solução foi proposta. A solução proposta se utiliza de diodos emissores de luz e fototransistores posicionados a uma distância conhecida. Sabendo-se a bitola do tubo utilizado e, contando-se o tempo entre os acionamentos, pode-se calcular o fluxo passante, porém também não se obteve sucesso nessa abordagem

A solução final encontrada foi se utilizar de sensores de trilha posicionados ao longo do copo, juntamente com um acionamento por tempo. O acionamento por tempo garante que não se passe do volume total do copo, em caso de falha, e os sensores de trilha garantem o volume solicitado. Isso foi projetado de modo que se garantisse uma maior precisão e eficácia na tiragem do chopp, dadas pelos sistemas redundantes.

#### 3.4.2.3 Garantir o colarinho escolhido

O controle do colarinho é feito através da atuação de motores, estes movimentam a alavanca que controla a saída de chopp e colarinho. Desta forma é necessário realizar medições de tempo para cada tamanho de colarinho selecionável, servindo assim a quantidade. Esse acionamento é feito por tempo e relaciona-se com a entrada de ar no sistema.

### 3.4.3 Atuação dos motores

Dois dos mais importantes subsistemas que contribuem diretamente com a experiência do usuário são a inclinação do copo e a tiragem automática do chopp, para tais tarefas fez-se o uso de motores de passo. Para o controle dos motores usou-se dos pinos da *Raspberry pi*.

Os motores precisam ser alimentados com 12V, para isso usa-se uma fonte de energia externa. Isso é dado pelo consumo de corrente dos motores, que é superior a aquela fornecida pelos pinos da *Raspberry pi*. Assim, fez-se necessário o uso de um *driver* L298N. A Figura 15 mostra o sistema dos motores montado.

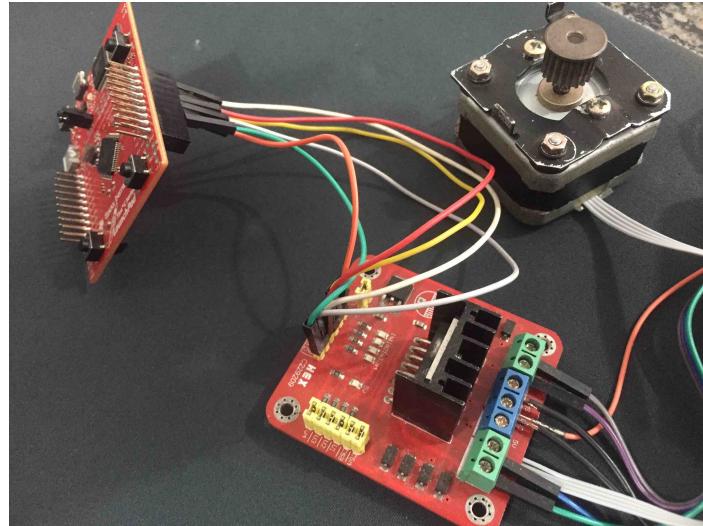


Figura 15 – Sistema de acionamento de motor de passo.

### 3.4.4 Abertura do reservatório de copos

Esse serviço se fez necessário, após a decisão de disponibilizar o copo ao usuário, portanto os copos são armazenados na própria máquina. Para disponibilizar os copos, eles são dispostos de forma que sempre que exista uma requisição de um chopp, um copo caia em um reservatório próprio. Para empurrar os copos são utilizados duas solenoides de modelo TAU-0530, uma delas pode ser vista na Figura 16.



Figura 16 – Solenoide utilizado.

## 3.5 Relatório de teste

### 3.5.1 Relatório Controle de Temperatura

**Título:** Relatório Controle de Temperatura

**Autores:** Gabriel Araújo, Ithallo Guimarães, Lucas Raposo e Oziel da Silva

**Data:** 15/11/2017

#### 3.5.1.1 Introdução

Neste experimento será testado a capacidade e a precisão das medidas realizadas pelo sensor de temperatura DS18B20, utilizado neste trabalho para o acionamento do compressor visando manter a temperatura correta do chopp.

#### 3.5.1.2 Materiais

- Sensor DS18B20
- Resistor de 4.7 KOhms
- Raspberry
- Módulo Relé

#### 3.5.1.3 Procedimento Experimental

Após posicionar o sensor na saída de chopp, ligou-se o sensor a raspberry e utilizou a resistência entre o cado de alimentação VCC e o cabo de dados. Utilizando de código implementado em python foram feitas diversas leituras de temperatura, de posse destes dados a compressor era ativado ou desativado de acordo com as necessidades do projeto. As medições em seguida foram aferidas com um termômetro laser.

#### 3.5.1.4 Resultados e Discussão

Comparando os dados medidos com aqueles do termômetro laser percebeu-se uma pequena diferença, mas nada que ultrapassasse o erro fornecido pelo fabricante, a ativação do compressor através do módulo relé foi realizada dentro da margem de temperatura aplicados.

#### 3.5.1.5 Conclusão

Ao final percebeu-se uma precisão satisfatória do sensor, confirmando assim o seu uso no sistema de controle da temperatura do chopp.

### 3.5.2 Relatório Abertura do Compartimento de Copos

**Título:** Relatório Abertura do Compartimento de Copos

**Autores:** Gabriel Araújo, Ithallo Guimarães, Lucas Raposo e Oziel da Silva

**Data:** 17/11/2017

#### 3.5.2.1 Introdução

Este subsistema é responsável pela liberação do copo ao cliente, o copo será liberado automaticamente cabendo ao cliente apenas posicionar-lo no devido lugar. Coube a eletrônica nesse subsistema a missão de ativar e desativar a solenoide que irá puxar uma porta, desta forma os copos inclinados cairão no compartimento.

#### 3.5.2.2 Materiais

- Solenoide com mola
- Raspberry
- Módulo Relé
- Fonte de alimentação 12V

#### 3.5.2.3 Procedimento Experimental

Para este experimento realizou-se a ligação da solenoide ao módulo relé, o pino responsável pela compartimentação de copos era ativado e desativado como se fossem sucessivas aberturas do compartimento de copo, testando assim além da resposta o comportamento da mola presente na solenoide.

#### 3.5.2.4 Conclusão

A mola apresentou uma boa resposta, mas o experimento apresentaria melhores resultados caso fosse realizado juntamente com o mecanismo de liberação para que o tempo necessário para a passagem de um copo pudesse ser medido com maior fidelidade.

### 3.5.3 Relatório Saída de Chopp

**Título:** Relatório Saída de Chopp

**Autores:** Gabriel Araújo, Ithallo Guimarães, Lucas Raposo e Oziel da Silva

**Data: 17/11/2017**

### 3.5.3.1 Introdução

Este subsistema irá tratar da liberação de chopp, que consiste no processo de verificação do copo, inclinação e ativação da torneira.

### 3.5.3.2 Materiais

- Sensor de trilha
- Raspberry
- Motores de passo
- Fonte de alimentação 12V

### 3.5.3.3 Procedimento Experimental

Para este experimento foram testados dois módulos, o primeiro consiste em medir uma coluna de chopp através do sensor de trilha que detecta a espuma do chopp e a inclinação do copo através dos motores de passo.

Os sensores de trilha foram fixados em um copo e inclinados, foi adicionado ao copo então chopp quando atingida a altura esperada o sensor avisava através de um LED. Os motores foram testados através de giros em diversos ângulos, seu software permitia como entrada um ângulo em graus e então seu eixo executava a rotação solicitada.

### 3.5.3.4 Conclusão

O sensor de trilha conseguiu detectar a borda de espuma, apenas necessita de uma boa calibragem. Os motores também rotacionaram o ângulo desejado faltando apenas definir esse dado juntamente com o pessoal responsável pelo mecanismo.

## 4 Estrutura

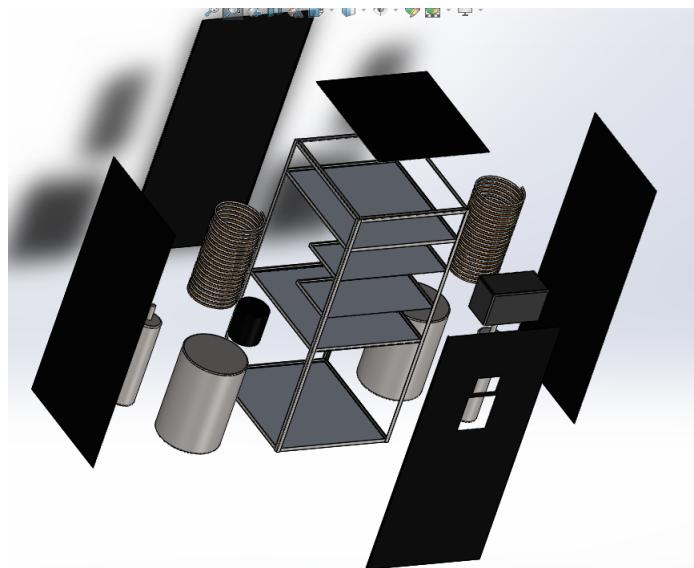


Figura 17 – Vista Explodida. Fonte: Própria.

### 4.1 Levantamento dos esforços a serem levantados

Após a realização do packaging, começaram os trabalhos de integração com os outros subsistemas para levantamento de requisitos. Após algumas reuniões foi verificado que a estrutura deveria ser capaz de suportar o peso de dois barris de chopp de 67,4Kg cada e dois cilindros de CO<sub>2</sub> de 20kg no primeiro nível. O segundo nível suportaria todo o sistema de refrigeração e alimentação do circuito elétrico da máquina que pesaria no total em torno de 50Kg e o terceiro nível seria o reservatório dos copos.

Entre o segundo e terceiro níveis existem dois subníveis que acomodam toda parte eletrônica e de automação da máquina, como haspberry, um motor de passo e algumas placas de circuito.



Figura 18 – Frame Principal. Fonte: Própria.

Segue abaixo algumas informações dos componentes mais pesados e de maior influência nos esforços suportados:

- **Barril:**

Peso: 67.4 *Kg*

Altura: 740 *mm*

Diâmetro: 680 *mm*

- **Sistema de refrigeração e alimentação:**

Peso: 50 *Kg*

Altura máxima: 750 *mm*

Foi verificado que os níveis que suportariam a maior parte dos esforços seriam o primeiro e o segundo. Sendo que o primeiro teria de suportar aproximadamente 1800N (680 de cada barril e 200N de cada cilindro de CO<sub>2</sub>) e o segundo 500N (referente ao peso total de todos os componentes).

## 4.2 Simulação

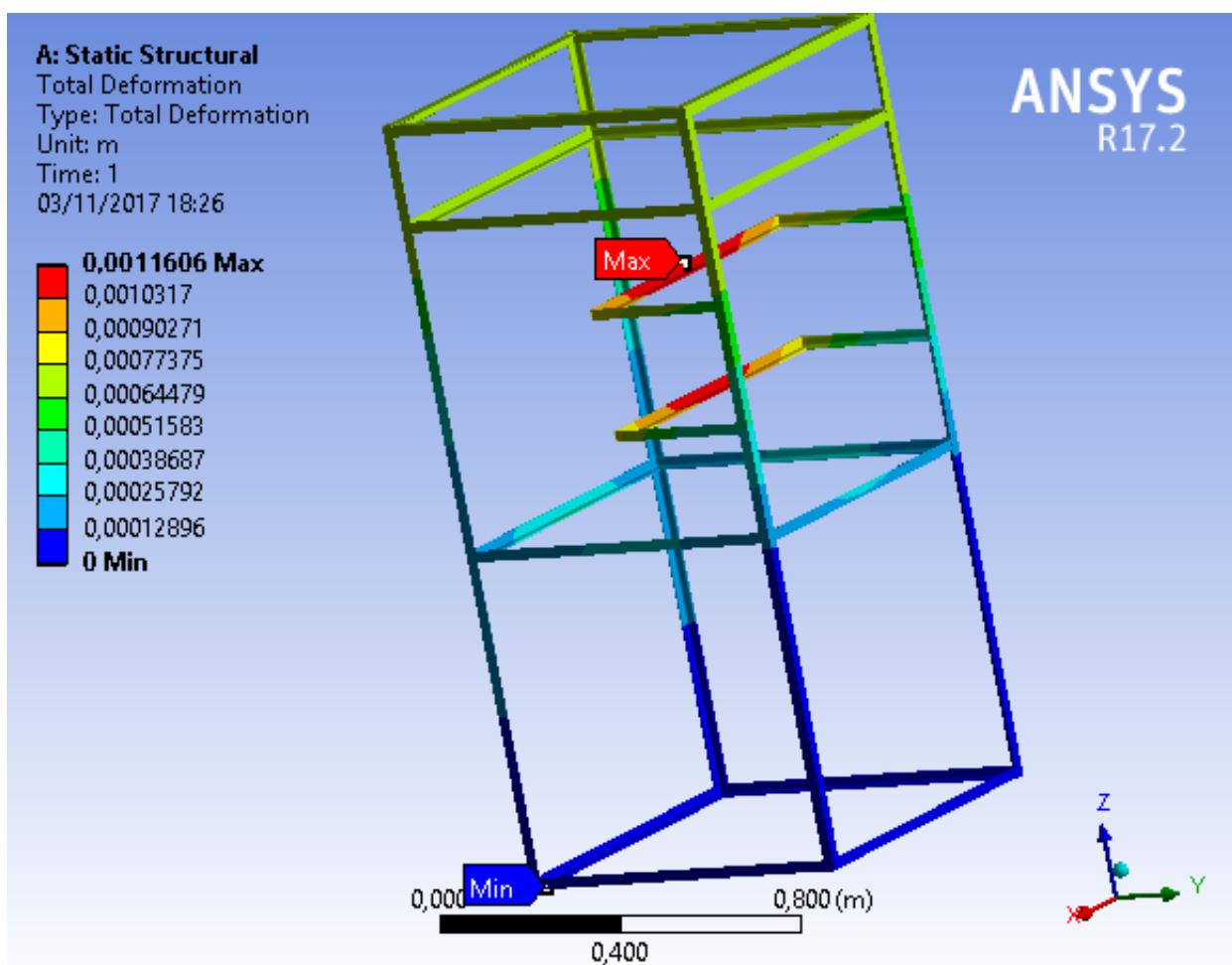


Figura 19 – Deformação Frame. Fonte: Própria.

As simulações do frame e das chapas foram feitas em separado. Para realizar as simulações iniciais foi utilizado o módulo de análise estrutural do Solidworks, pela praticidade e rapidez de implementação de um modelo inicial.

Durante o projeto tudo foi passado para o software Ansys, que gerou resultados mais completos e refinados, mas confirmando a análise prévia simplificada.

Após a simulação do design inicial verificou-se que o frame feito com perfis L de aço 1020 suportaria os esforços sem maiores problemas.

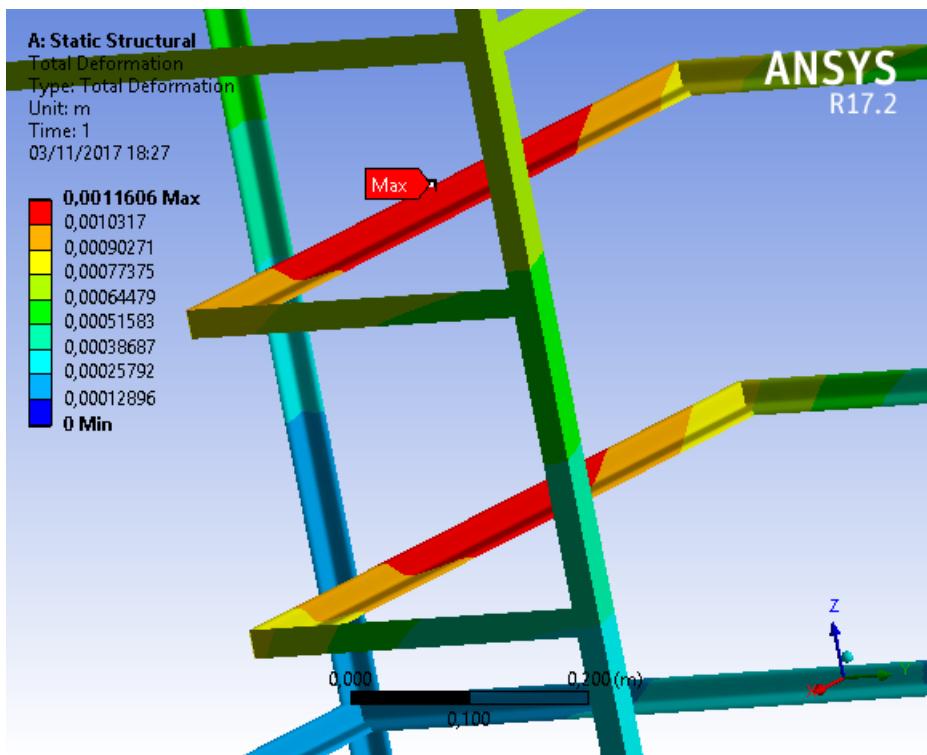


Figura 20 – Deformação Local. Fonte: Própria.

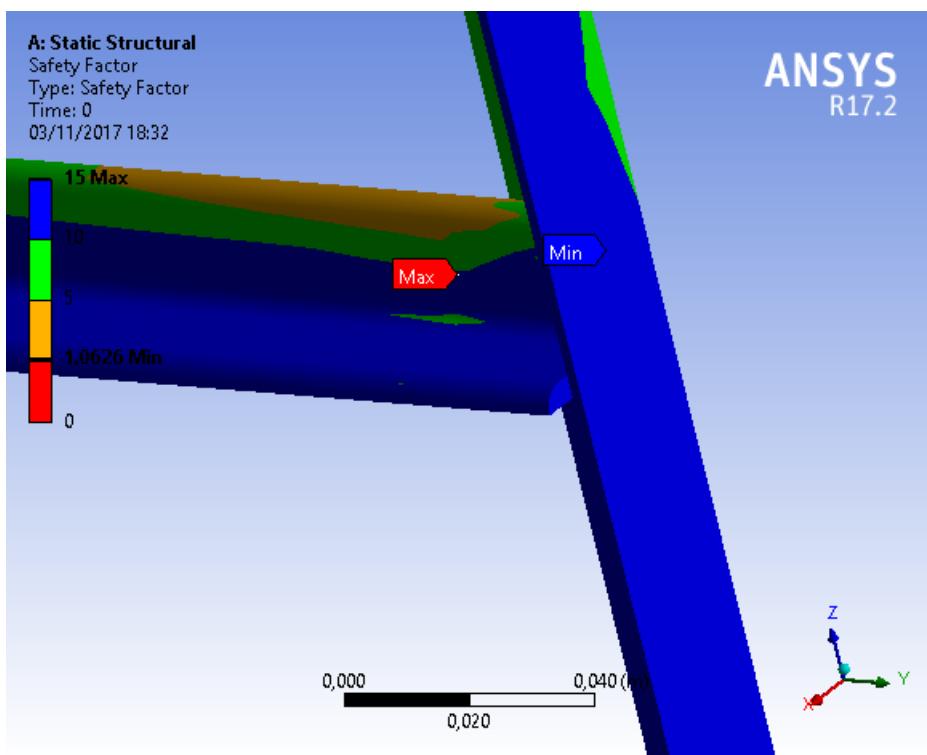


Figura 21 – Fator de Segurança. Fonte: Própria.

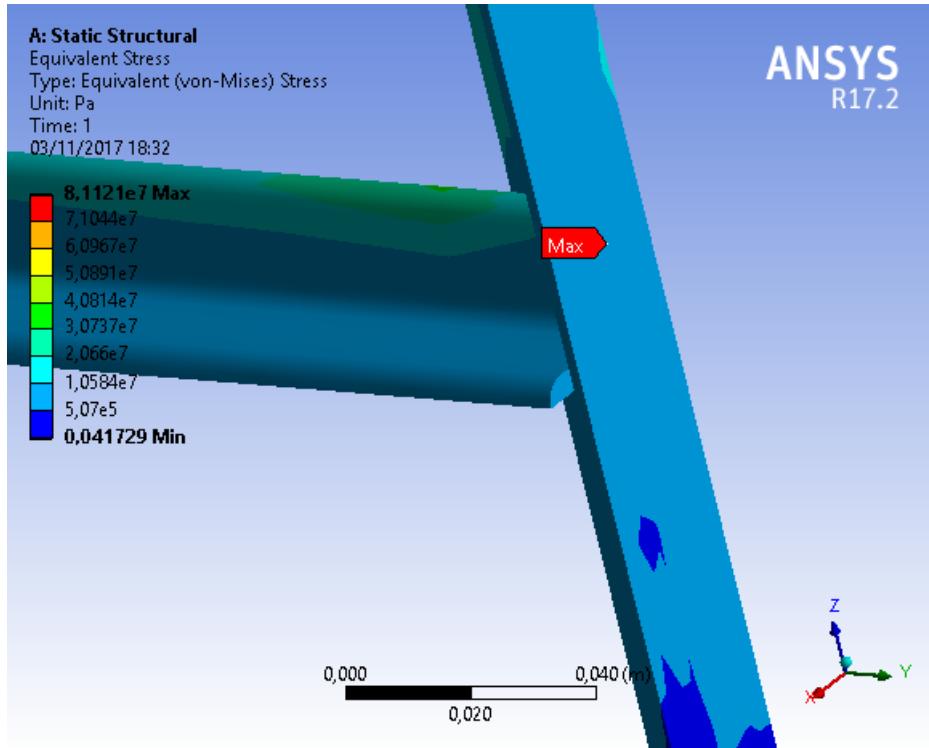


Figura 22 – Tensão Local. Fonte: Própria.

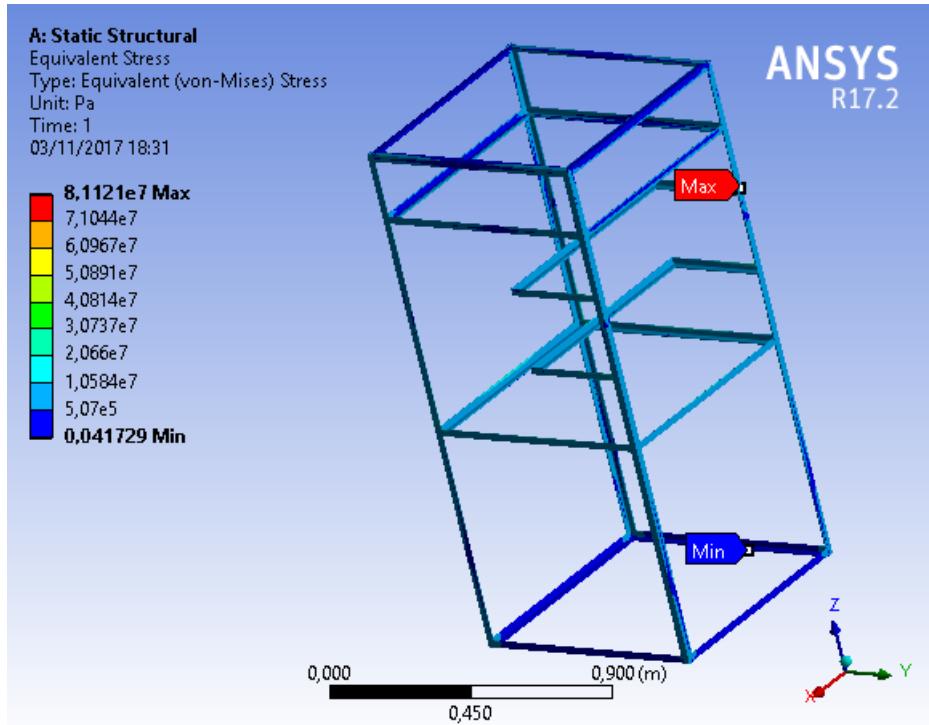


Figura 23 – Tensão. Fonte: Própria.

No que diz respeito às chapas de alumínio que formam as bases dos níveis, percebeu-se que apesar da não ocorrência de escoamento ou ruptura do material, a deflexão da chapa

no primeiro nível estava excessiva (Aproximadamente 8mm de deflexão máxima), de forma que comprometeria a qualidade do produto e de seu funcionamento.

Foi decidido que o primeiro nível não seria feito com chapa de alumínio como decidido anteriormente mas sim, com aço. Houve a opção de apenas aplicar uma treliça a este nível e usar uma chapa de alumínio mais espessa porém o custo ficou muito elevado.

A solução final foi aplicar treliças no primeiro e segundo níveis e usar chapas mais finas (2mm ante 4mm que era usado no projeto inicial). Esta solução foi a que gerou a melhor relação entre peso, custo e resistência mecânica (A deflexão máxima passou a ser de aproximadamente 2mm).

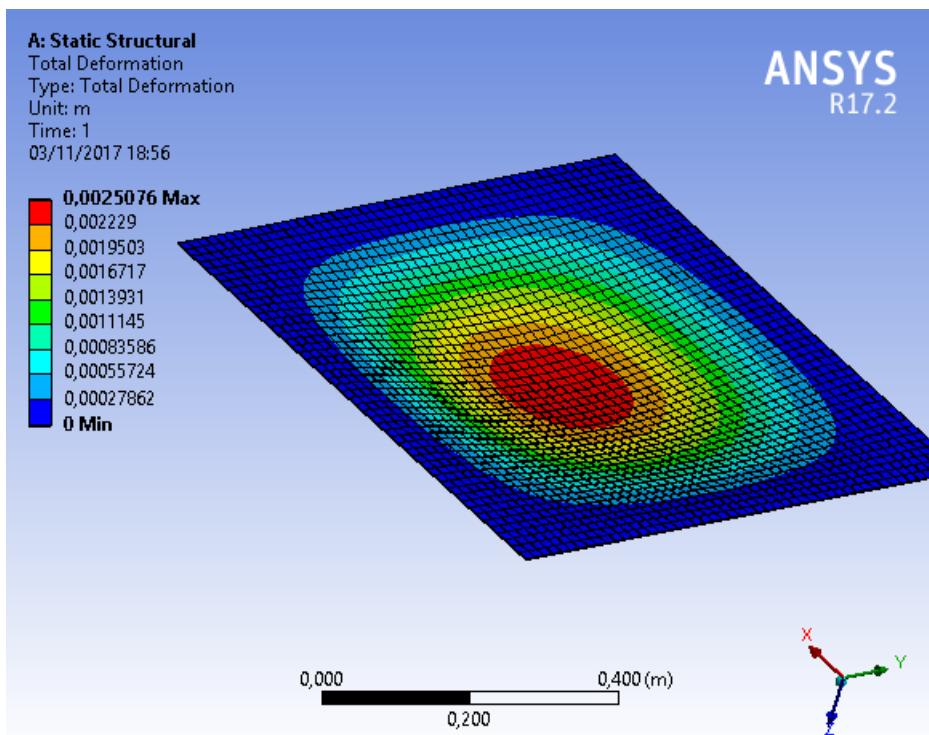


Figura 24 – Deformação da chapa após redimensionamento. Fonte: Própria.

Os demais níveis foram feitos com chapa de alumínio como pensado anteriormente, o que contribuiu para que o peso da estrutura não fosse muito elevado (aproximadamente 70kg).

### 4.3 Fabricação

A fabricação foi pensada desde início, visando sempre simplicidade, para que o projeto seja facilmente executável.

Tudo começou pelo corte dos perfis em L. Na primeira etapa foram feitos os cortes visando apenas acertar os comprimentos necessários para compor cada parte da estrutura.

Na segunda foram feitos cortes em 45 graus para facilitar o encaixe e melhorar o acamento.



Figura 25 – Perfis Cortados. Fonte: Própria.



Figura 26 – Esmerilhadeira usada para acabamento dos cortes. Fonte: Própria.



Figura 27 – Guilhotina usada no corte de chapas. Fonte: Própria.

Após toda a fase de corte foi iniciado o processo de soldagem das bases quadradas do frame. O processo utilizado foi o MIG, devido principalmente à facilidade de acesso ao equipamento e da boa qualidade referente ao resultado final.



Figura 28 – Maquina para soldagem MIG. Fonte: Própria.



Figura 29 – Solda realizada na estrutura. Fonte: Própria.

Após a soldagem de todas as bases, as 4 vigas principais da estrutura foram unidas as bases de forma que fosse possível erguer a estrutura. Foi iniciado, então, o processo de soldagem dos subníveis menores, referentes aos componentes eletrônicos e de automação.



Figura 30 – Frame principal após o término da soldagem. Fonte: Própria.

Assim que terminados os processos de soldagem as placas metálicas foram fixadas

ao frame. A fixação escolhida foi uma fita de alta resistência da 3m. A utilização da fita evitou o exagero no uso de parafusos e soldagem, o que reduziu custo e facilitou qualquer manutenção eventual que tenha de ser realizada na estrutura.

Todo o acabamento final, que é responsável por toda a estética do equipamento, foi feito com chapas de alumínio cortadas e dobradas.

É importante ressaltar que a equipe de estrutura ficou responsável pelo mecanismo de acionamento da base que inclina o copo. Foi projetado um mecanismo simples de biela-mañivela fabricado através de impressão 3d. A manivela eh rotacionada por um motor de passo que empurra a biela contra a base de forma que ela pivota em seu suporte superior.

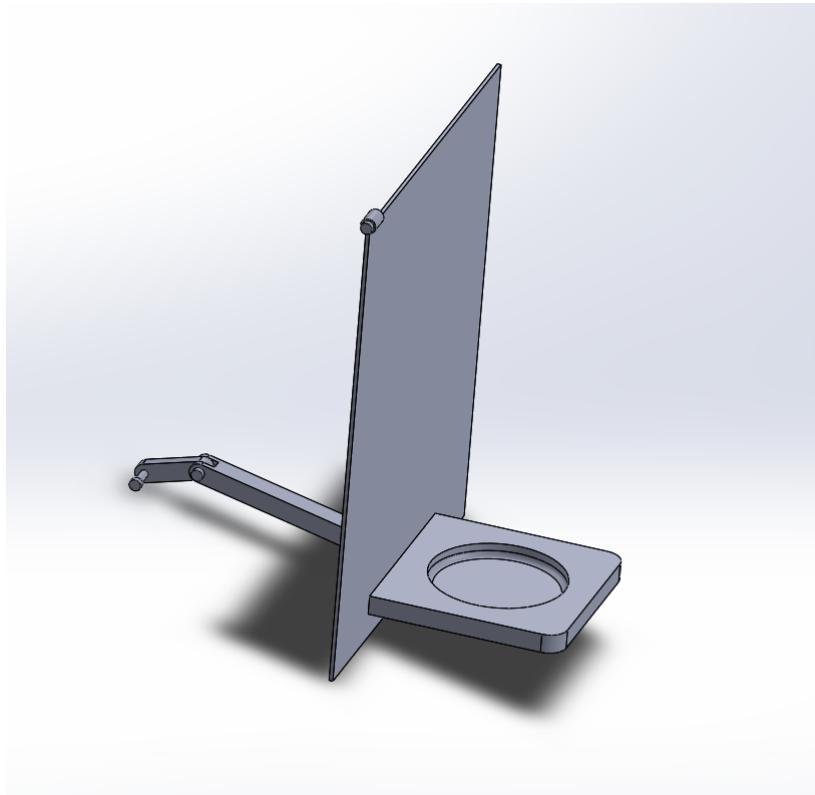


Figura 31 – Mecanismo de inclinação do chopp. Fonte: Própria.

# 5 Software

## 5.1 WebService

### 5.1.1 Requisitos do Projeto

#### 5.1.1.1 Cadastrar usuário

**Dado** que a API receba uma requisição para cadastrar um novo usuário  
**E** todos os campos do usuário sejam válidos  
**Então** o usuário deve ser cadastrado.

#### 5.1.1.2 Autenticar usuário

**Dado** que a API receba uma requisição para autenticar um usuário  
**E** todos os campos do usuário sejam válidos  
**Então** o token correspondente ao usuário deve ser gerado.

#### 5.1.1.3 Realizar Pagamento

**Dado** que a API receba uma requisição para registrar um pagamento  
**E** os dados de pagamento sejam validados junto a API do PagSeguro  
**Então** a transação deve ser efetivada.

#### 5.1.1.4 Gerar Qrcode

**Dado** que a API receba uma notificação da API do PagSeguro  
**E** o status da transação seja alterado para *paga*  
**Então** deve ser gerado um QrCode.

#### 5.1.1.5 Validar qrcode

**Dado** que a API receba uma requisição com um QrCode  
**E** o QrCode seja válido  
**Então** deve ser retornado as informações referente ao QrCode.

### 5.1.1.6 Mostrar chopp

**Dado** que a API receba uma requisição para mostrar os chopp de um usuário

**E** o usuário esteja logado

**Então** deve ser retornado os chopp referentes a esse usuário.

### 5.1.1.7 Salvar dados dos sensores

**Dado** que a API receba uma requisição para salvar os dados dos sensores

**E** os dados sejam válidos

**Então** os dados devem ser salvos.

### 5.1.1.8 Mostrar dados dos sensores

**Dado** que a API receba uma requisição para mostrar os dados dos sensores

**E** o usuário seja um administrador

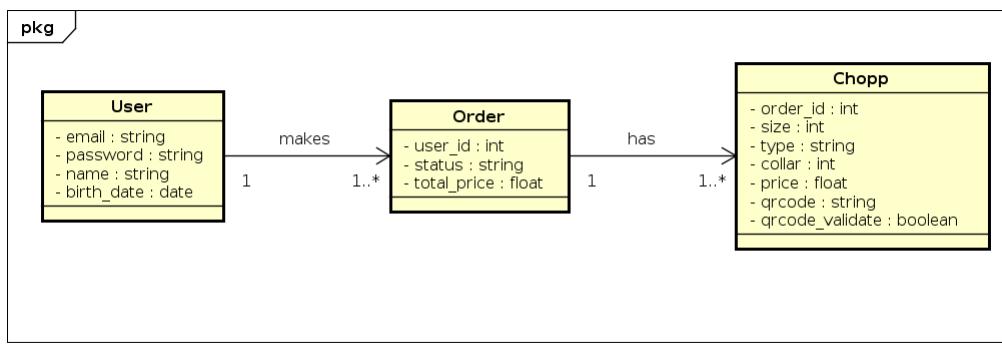
**Então** os dados devem ser mostrados.

## 5.1.2 Projeto

A demanda do problema exige que diferentes hardwares se integrem com uma mesma base de dados. Fundamentalmente, o Smartphone do usuário realiza a compra, e a máquina deve validar o QrCode para que o usuário consuma o chopp.

Para solucionar essa demanda foi proposto uma arquitetura orientada a serviços (SOA), tal que uma webservice REST seja responsável pela persistência e autenticação de todos os dados relevantes do sistema. Sendo assim esse WebService será responsável por se comunicar com todos os outros subsistemas.

O modelo de domínio abaixo traz uma visão inicial das entidades presentes no sistema.



powered by Astah

Figura 32 – Modelo de Domínio. Fonte: Própria.

### 5.1.3 Solução Adotada

Visto que é requisito primordial do sistema o checkout de pagamentos via cartão de crédito, a tecnologia escolhida para implementação desse WebService foi o framework Ruby On Rails<sup>1</sup>, pois a API de pagamentos do PagSeguro fornece suporte a essa tecnologia. A versão utilizada do Ruby foi a versão 2.3.0 e o Rails a versão 5.0.5.

#### 5.1.3.1 Arquitetura

##### 5.1.3.1.1 Models

As classes modelos no Rails representam as entidades e seus relacionamentos, e são responsáveis pelo gerenciamento dos dados no banco. O diagrama de classes abaixo representa as classes modelos do sistema:

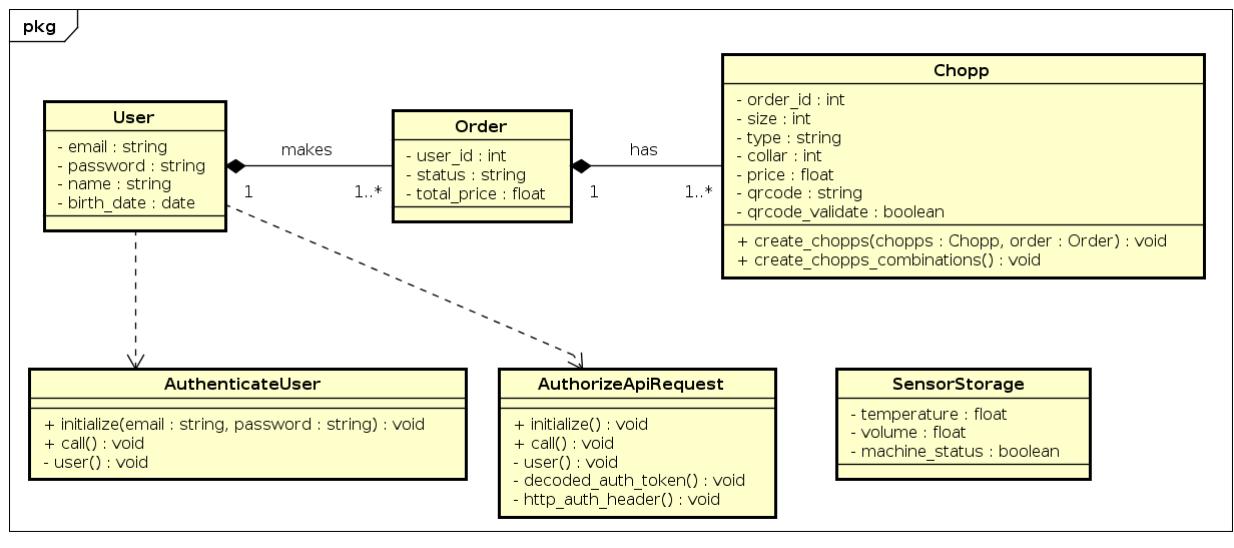


Figura 33 – Diagrama de Classes - Models. Fonte: Própria.

- **User:** A classe *User* guarda as informações referentes a um usuário.
- **Order:** A classe *Order* guarda as informações referentes a um pedido de um usuário específico.
- **Chopp:** A classe *Chopp* guarda as informações referentes a um chopp de um pedido específico.
- **SensorStorage:** A classe *SensorStorage* guarda as informações referentes aos valores lidos dos sensores do módulo embarcado.

<sup>1</sup> <<http://rubyonrails.org/>>

- **AuthenticateUser:** A classe *AuthenticateUser* é responsável por autenticar um usuário na base de dados.
- **AuthorizeApiRequest:** A classe *AuthorizeApiRequest* é responsável por decodificar determinado token de autenticação.

#### 5.1.3.1.2 Controllers

As classes controladoras no Rails são responsáveis por se comunicar com as classes modelos, e são elas que lidam com requisições web do usuário. O diagrama de classes abaixo representa as classes controladoras do sistema:

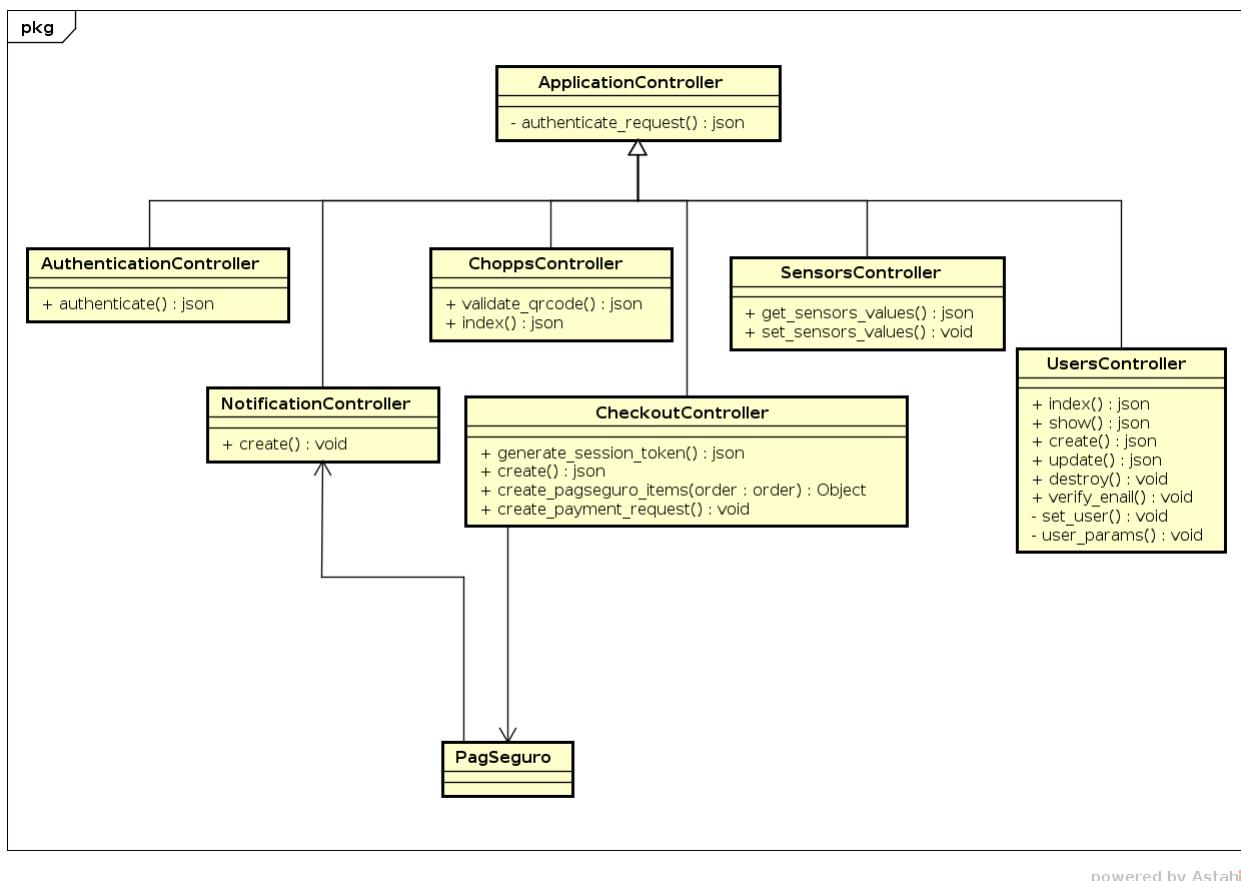


Figura 34 – Diagrama de Classes - Controllers. Fonte: Própria.

- **ApplicationController:** A classe  *ApplicationController* é a controladora padrão do Rails e todas as outras controladoras são herdadas a partir desta.
- **UsersController:** A classe  *UsersController* é responsável por enviar e receber os dados dos usuários.
- **AuthenticationController:** A classe  *AuthenticationController* é responsável por receber os dados do usuário e realizar a autenticação.

- **ChoppsController:** A classe *ChoppsController* é responsável por enviar e receber os dados dos chopps.
- **SensorsController:** A classe *SensorsController* é responsável por enviar e receber os dados dos sensores.
- **CheckoutController:** A classe *CheckoutController* é responsável por receber os dados de pagamento e enviar a API do PagSeguro.
- **NotificationController:** A classe *NotificationController* é responsável por receber notificações da API do PagSeguro quando uma transação muda de status.
- **PagSeguro:** Entidade que representa a API do PagSeguro.

#### 5.1.4 Casos de Teste

Para testar o funcionamento correto da API foi utilizada a ferramenta *Rspec*, que nos permite testar a aplicação unitariamente, através de testes unitários. Com este tipo de teste é possível testar o comportamento de cada método. A imagem abaixo mostra a cobertura de testes:

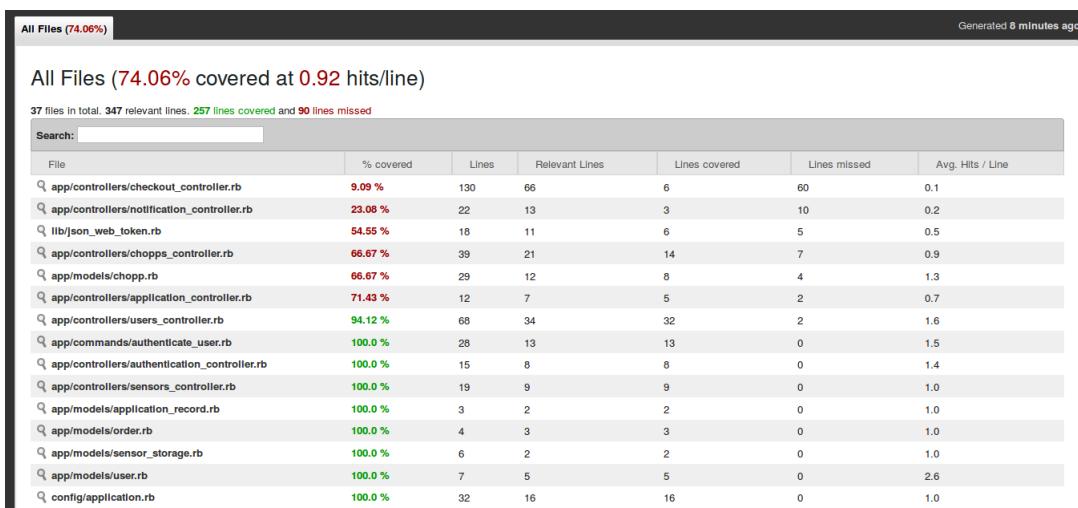


Figura 35 – Cobertura de Testes. Fonte: Própria.

Como a demanda do projeto é muito grande, não foi possível testar todos os métodos unitariamente, logo, também foi utilizada a ferramenta *Postman*. Essa ferramenta nos permite testar se a API está respondendo corretamente as requisições.

### 5.1.5 Gerência e Configuração

- **Deploy:** Para hospedagem do WebService foi usado o *Heroku*<sup>2</sup>. As vantagens em utilizar essa plataforma estão no custo, por ser uma ferramenta grátil e na facilidade em realizar o deploy.
- **Integração Contínua:** Para a integração contínua foi utilizada a ferramenta *TravisCI*. Com esta ferramenta, a cada commit no repositório todos os testes são executados automatizadamente, e assim é possível verificar se aquele commit quebrou a build.

## 5.2 Aplicativo Mobile

Para compra e gerenciamento dos cupons de *chopp*, torna-se necessário um aplicativo *mobile* de forma que seja facilitado a acessibilidade para os usuários.

### 5.2.1 Requisitos do Projeto

#### 5.2.1.1 Comprar Chopp

**Dado** que o usuário esteja logado

**Quando** tenha iniciado a compra de um chopp

**Então** deve ser possível ver as preferências de chope(colarinho, quantidades discretas e pré-definidas)

#### 5.2.1.2 Cadastrar Usuário

**Dado** que o usuário tente entrar no sistema

**E** ainda não tenha se cadastrado

**Quando** tenha iniciado o aplicativo

**Então** terá disponível um formulário com email e senha para cadastro

#### 5.2.1.3 Confirmar Cadastro

**Dada** que o usuário já tenha se cadastrado

**E** recebido um email com link de confirmação

**Quando** clicar no link

**Então** deverá ser levado a uma mensagem de confirmação

---

<sup>2</sup> <<https://www.heroku.com/>>

**E** deve ser possível se autenticar no sistema.

#### 5.2.1.4 Finalizar Compra

**Dado** que o usuário já esteja cadastrado no sistema

**E** tenha clicado para comprar um chopp

**E** tenha acesso a internet

**Quando** finalizado a compra

**Então** deve ser possível inserir os dados do cartão de crédito

**E** ser notificado se a compra foi bem sucedida ou não

**E** em seus tickets devem estar disponíveis para uso na máquina

#### 5.2.1.5 Listar Cupons(QRCode)

**Dado** que o usuário já tenha comprado tickets

**E** esteja autenticado no sistema

**Quando** clicar em uma opção meus tickets

**Então** deve ser possível visualizar o QRCode que representa a unidade de chopp

**E** deverá ter disponível uma opção para usar o ticket

**E** caso já tenha sido usado o ticket, deverá sumir da lista de tickets do usuário

#### 5.2.1.6 Efetuar Compra Offline

**Dada** a compra efetuada de um chopp

**E** um usuário offline

**Quando** o usuário clicar para usar o ticket

**Então** a máquina de chopp deve responder no sistema de interação com mensagem sucesso.

#### 5.2.1.7 Visualizar Cupons(QRCode)

**Dado** que o usuário tenha comprado tickets

**Quando** entrar na tela de visualização de tickets

**Então** deve ser possível ver os tickets comprados segundo as preferências

**E** poder escolher qual queira consumir

### 5.2.2 Projeto

De acordo com os critérios observados do público alvo e do contexto considerado, pensou-se em um aplicativo mobile que se comunica com um Web Service para realizar as operações descritas na especificação acima.

### 5.2.3 Solução Adotada

Para entendimento da solução implementada, será discutido nos tópicos seguintes sobre a arquitetura e os resultados da implementação.

#### 5.2.3.1 Arquitetura

Por fim, decidiu-se desenvolver um aplicativo multiplataforma utilizando o *framework Ionic 3*, baseado em *Angular 4*, uma biblioteca Typescript que facilita a criação de aplicativos *Web* e *Mobile*. Conforme (??), Ionic é um *framework open source* para desenvolvimento de aplicativos *mobile* utilizando tecnologias web(HTML, CSS, JavaScript). O Ionic usa do poder do Angular para trabalhar com componentização, elevando o reaproveitamento de código.

#### 5.2.3.2 Aplicativo

A seguir, de acordo com os requisitos acordados, estão as imagens das principais telas do aplicativo:

### 5.2.4 Casos de Teste

Não foram especificados casos de teste para o aplicativo *mobile*, pois acredita-se que a especificação por exemplo elicitada já serve como caso de teste, uma vez que os cenários não se divergem tanto.

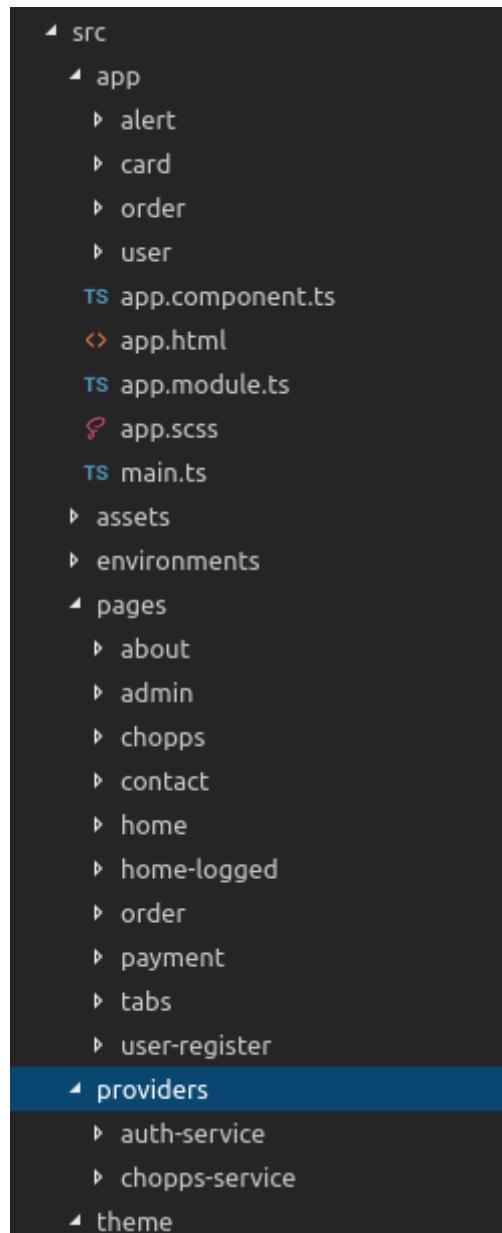


Figura 36 – Estrutura de diretórios do Ionic 3.

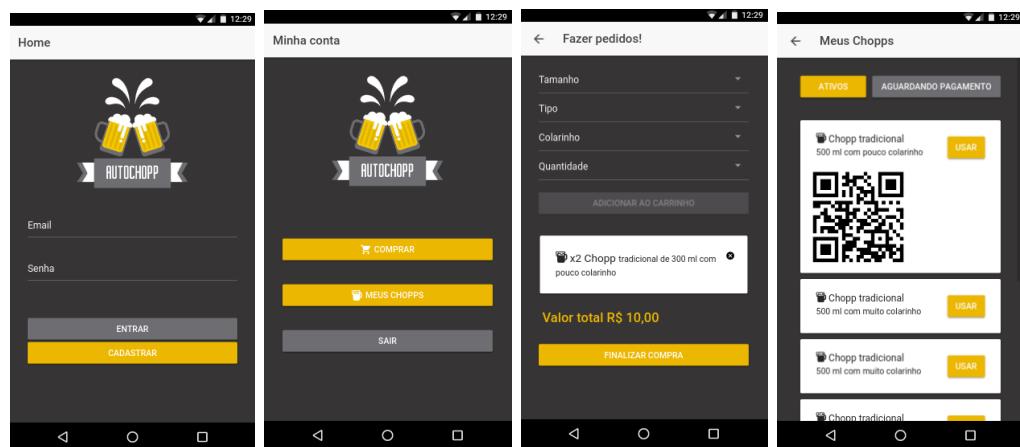


Figura 37 – Telas do Aplicativo. Fonte: Própria.

## 5.3 Sistema Administrativo

O Sistema administrativo está incorporado ao aplicativo do Sistema de Compras, porém apenas usuários administradores possuem acesso. Ao utilizar o aplicativo, o usuário visualiza as seguintes informações referentes ao estado da máquina: a quantidade de chopp que a máquina possui, a temperatura de resfriamento e o status da conexão da máquina com a internet.

Esses valores são recuperados através de requisições GET ao WebService que guarda essas informações recebendo requisições POST a cada 60 seg da aplicação embarcada que lê os dados dos sensores.

### 5.3.1 Aplicativo

A imagem abaixo representa a tela do usuário administrador:



Figura 38 – Tela do Administrador. Fonte: Própria.

## 5.4 Sistema de Validação de Compra

### 5.4.1 Requisitos do Projeto

#### 5.4.1.1 Leitura de um cupom

- **Dado** uma pessoa com cupons comprados
- **Quando** o mesmo selecionar a opção de validar um cupom da tela inicial

- **Então** deve ser instruído a exibir o QRCode na câmera da máquina para efetuar o uso do cupom
- **E** a máquina deve informar se a leitura ocorreu de forma correta ou não.

#### 5.4.1.2 Leitura de um cupom válido

- **Dado** uma pessoa com um cupom válido pelo sistema
- **Quando** o mesmo receber a confirmação de leitura com sucesso do cupom
- **Então** deve ser instruído a pegar o copo na gaveta e posicionar o copo no suporte
- **E** aguardar o processo de serventia do chopp conforme as preferências.

#### 5.4.1.3 Leitura de um cupom inválido

- **Dado** uma pessoa com um cupom invalidado pelo sistema
- **Quando** o mesmo receber a confirmação de leitura sem sucesso do cupom
- **Então** deve ser instruído a tentar novamente a leitura do cupom
- **E** em caso de insucesso aguardar o sistema ser recuperado, ou entrar em contato com o proprietário da máquina.

### 5.4.2 Projeto

O produto exige uma forma de retirada do chopp de forma autônoma, isso implicou no desenvolvimento de uma aplicação gráfica acoplada na máquina que servisse de forma interativa com o usuário fazendo a validação dos cupons e servindo os chopp conforme as preferências.

### 5.4.3 Solução Adotada

Por ser necessária a criação de uma aplicação que não consumisse muitos recursos, optou-se pelo uso do *framework* Kivy juntamente com a linguagem de programação Python, fornecendo um maior suporte ao foco principal da aplicação que é a leitura e reconhecimento de cupons (QrCodes)

A solução foi instalada na própria máquina, sobre o sistema operacional padrão da plataforma Raspberry, o Raspbian, com intuito de evitar incompatibilidade com outros sistemas operacionais e retrabalho.

#### 5.4.3.1 Arquitetura

Por se tratar de uma aplicação com poucas funcionalidades, dados e entidades a serem trabalhadas, não há uma distinção entre models e controllers. O diagrama de classes presente na imagem 39 representa a disposição das classes do sistema.

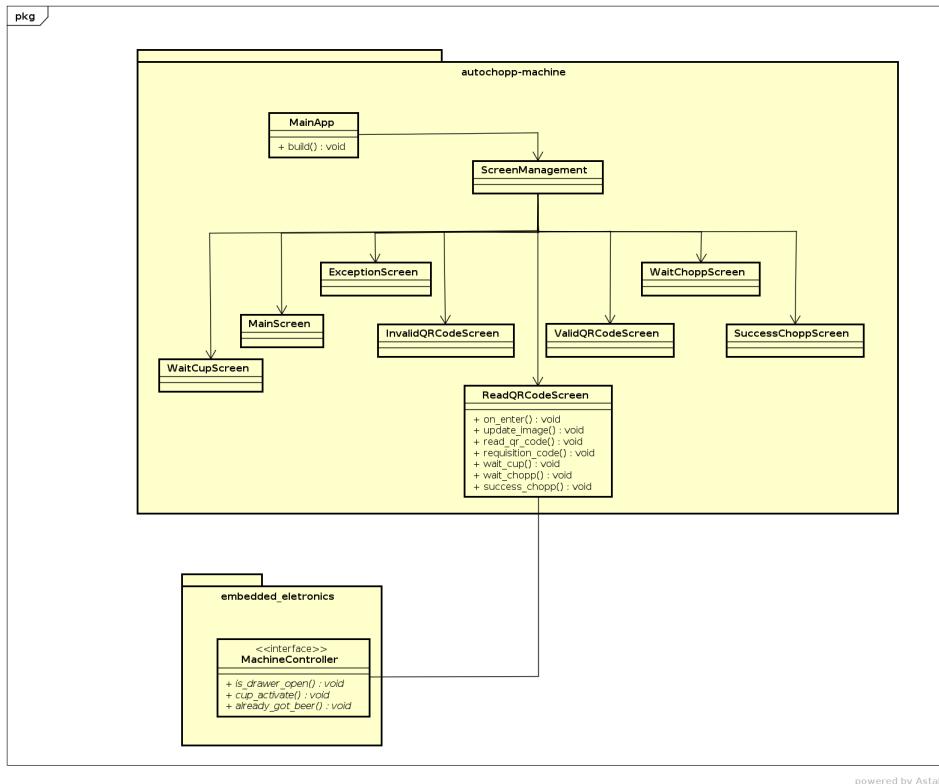


Figura 39 – Diagrama de classes do sistema de validação de compras. Fonte: Própria.

Analisando o diagrama, percebe-se que a maioria das classes estão sem métodos e sem atributos, isso acontece pelo fato de que cada tela a ser renderizada, necessita de uma classe para estar associada. O Kivy faz uma diferenciação entre classes *Screens* e classes *Apps*. A primeira se refere a classes que estão associadas a alguma tela a ser renderizada, enquanto que as classes do segundo tipo são as responsáveis por compilar uma *build* do projeto e executar o mesmo. Nesse caso a classe do tipo *App* é a classe *MainApp*. Outro ponto a ser elucidado é que a transição entre telas é gerenciado pela classe *ScreenManagement* que acopla todas as *Screens* do projeto e as integra permitindo a classe *MainApp* gerar a *build* das mesmas.

Além da parte gráfica e de leitura e validação de cupons, essa aplicação tem como objetivo também a comunicação com a parte de leitura de alguns sensores para a construção do projeto. Com isso, todos os scripts de leituras de sensores foram abrigados no pacote *embedded\_electronics* criando uma interface entre as funções necessárias para o processo de serventia de chopp e a aplicação gráfica. Tais funcionalidades são consumidas na classe responsável pelo gerenciamento da leitura dos cupons junto a câmera.

#### 5.4.3.2 Aplicação Final

COLOCAR PRINTS

### 5.4.4 Casos de Teste

#### 5.4.4.1 Leitura do Qrcode

Para leitura do Qrcode, o usuário deve posicionar um Qrcode em frente a câmera. Para testar esse comportamento, foi verificado se a câmera conseguia identificar a string dentro do Qrcode.

#### 5.4.4.2 Validação do Qrcode

Para validar o Qrcode, a aplicação deve mandar uma requisição *POST* ao Web-Service descrito anteriormente. Para testar este comportamento, foi verificado o retorno da requisição ao WebService.

#### 5.4.4.3 Integração com o eletrônica

Como mostra a figura 39, a classe *MachineController* é a interface que separa a implementação dos componentes embarcados do sistema de validação de compras. Para testar o funcionamento correto do sistema independente do módulo embarcado, foi implementado os métodos da interface com *sleeps*, que simularam o comportamento do módulo embarcado.

## 5.5 Subsistemas integrados

EXPLICAR E COLOCAR DIAGRAMAS DE SEQUÊNCIA



## 6 Plano de Integração

A integração do projeto consiste primeiramente em realizar a junção da estrutura criada com os módulos de maior dimensão, exemplo: compressor, *nobreak* entre outros.

Após essa etapa, o próximo passo é a integração entre as engenharias de *software* e eletrônica. Essa etapa consiste em juntar os sensores, atuadores e *software* construídos ao equipamento desenvolvido pelas engenharias de estrutura e energia. Com todas essas junções realizadas os testes finais serão feitos com a presença e participação de todos os integrantes do projeto.

Com a finalização das etapas anteriores, a frente de trabalho de estrutura tem como objetivo futuro promover o desenvolvimento de componentes físicos que agreguem todos os equipamentos dos demais subsistemas, havendo portanto preocupação com a facilidade de manuseio durante os testes a serem realizados. Portanto o objetivo para o bom andamento do projeto é alocar equipamentos eletrônicos e da Raspberry Pi bem como os equipamentos de refrigeração e segurança conforme suas dimensões e integra-los a estrutura da máquina de tiragem de chopp de maneira ergonômica, se preocupando com a disponibilização dos cabos e sensores, de forma que os mesmos não tenham risco de se danificarem e comprometerem o bom funcionamento do produto.

Iteração 3	1d?	23/08/2017	23/08/2017
▣ Máquina de Chopp Produto	39d?	17/10/2017	08/12/2017
▣ Iniciação	1d?	18/10/2017	18/10/2017
Revalidar projeto de integração	1d	18/10/2017	18/10/2017
Planejar reuniões de integração dos sistemas	1d?	18/10/2017	18/10/2017
▣ Planejamento	4d?	17/10/2017	20/10/2017
Definir casos de teste do sistema integrado	2d?	17/10/2017	18/10/2017
▣ Controle de riscos	3d?	18/10/2017	20/10/2017
Efetuar arrecadação de rendas	1d?	20/10/2017	20/10/2017
Validar prazos	3d?	18/10/2017	20/10/2017
▣ Execução	36d	17/10/2017	05/12/2017
Revisar projeto de integração	7d	17/10/2017	25/10/2017
Integrar projeto	30d	17/10/2017	27/11/2017
Realizar bateria de testes	7d	27/11/2017	05/12/2017
▣ Encerramento	6d	01/12/2017	08/12/2017
Apresentar produto	6d	01/12/2017	08/12/2017

Figura 40 – Cronograma para a fase de integração de subsistemas. Fonte: Própria.



## 7 Considerações Finais

De maneira geral, foi possível observar a profundidade de detalhes e necessidades de implementação de cada subsistema. Em cada uma das frentes de trabalhos separadas por engenharia ou na combinação delas, ficou clara a necessidade de uma visão ampla do projeto, mas focada no trabalho de cada um e no impacto que cada atividade gera no andamento do projeto. Além disso, a realização de reuniões recorrentes, execução de atividades além do horário de aula da disciplina e comunicação constante entre os membros do grupo foram imprescindíveis para execução do projeto até o momento.

Espera-se que na próxima fase do projeto, os subsistemas desenvolvidos até essa etapa, possam ser integrados sem maiores problemas, uma vez que eles foram planejados visando essa integração. E mesmo levando em conta possíveis dificuldades nessa atividade, acreditamos que o projeto possa ser concluído até a data planejada.