

Santa Helena, Paraná

## IV Semana Tecnológica Acadêmica de Ciência da Computação da UTFPR, câmpus Santa Helena

25, 26 e 27 de setembro de 2017.

---

**ANAIIS**

---

<http://sh.utfpr.edu.br/setac/anais2017.pdf>

**Universidade Tecnológica Federal do Paraná, câmpus Santa Helena**

Reitor  
**Luiz Alberto Pilatti**

Vice-reitora  
**Vanessa Ishikawa Rasoto**

Diretor do Câmpus  
**Carlos Alberto Mucelin**

Diretor de Graduação e Educação Profissional  
**Adelmo Lowe Pletsch**

Coordenadora do Curso de Bacharelado em Ciência da Computação  
**Arlete Teresinha Beuren**

Coordenador da IV Semana Tecnológica Acadêmica do Curso de Bacharelado em  
Ciência da Computação  
**Giuvane Conti**

Comissão Científica

**Agnaldo da Costa**  
**Anderson Brilhador**  
**Arlete Teresinha Beuren**  
**Claudio Leones Bazzi**  
**Dênis Lucas Silva**  
**Diego Venâncio Thomaz**  
**Franck Carlos Velez Benito**  
**Giuvane Conti**  
**Leandro Antônio Pasa**  
**Maristela Rosso Walker**  
**Sérgio Antônio Brum Junior**  
**Thiago França Naves**  
**Vítor Alex De Marchi**

Organização e revisão

**Kelyn Schenatto**  
**Vera Vasilévski**

**UTFPR, câmpus Santa Helena**  
Prolongamento da Rua Cerejeira, s.n.  
85892-000, Bairro São Luiz, Santa Helena/PR - Brasil  
Telefone geral +55 (45) 3268-8800  
Coordenação de Ciência da Computação (COCIC)  
[cocic-sh@utfpr.edu.br](mailto:cocic-sh@utfpr.edu.br)

S471s Semana Tecnológica Acadêmica de Ciência da Computação da UTFPR, Câmpus  
Santa Helena (4.: 2017 : Santa Helena, PR)  
Anais ... / Giuvane Conti, Vera Vasilévski e Kelyn Schenatto (orgs.) [recurso  
Eletrônico]. -- Santa Helena: UTFPR, 2017.  
153 p. il. 30 cm

Endereço eletrônico: <http://sh.utfpr.edu.br/setac/anais2017.pdf>

1.Ciência da Computação . 2. Análise numérica . 3. Computação gráfica. 4.  
Redes. 5. Robótica. 6. Inovações Tecnológicas. I. Título. II. Conti, Giuvane, org. III.  
Vasilevski, Vera , org. IV. Schenatto, Kelyn, org. V. Universidade Tecnológica Federal  
do Paraná.

CDD 22.ed. 004.063

Bibliotecária Cleide Bezerra CRB 9<sup>a</sup>/770  
Biblioteca da UTFPR – Câmpus Santa Helena

## Apresentação

Apresentamos os Anais da IV Semana Tecnológica Acadêmica do Curso de Ciência da Computação (SeTAC<sup>2</sup>), que reúnem os trabalhos completos que foram expostos em comunicação oral durante o evento, em outubro de 2017. Neste ano, foram incluídas resenhas aos anais, um pedido de nossa comunidade acadêmica para que, dessa forma, se possibilitasse aos alunos se familiarizar com esse gênero discursivo, bem como incentivá-los a produzi-lo, assim exercitando seu pensamento crítico.

Nesta edição foi realizada, durante a sessão de apresentação oral dos trabalhos, a premiação do concurso para escolha do logotipo oficial da IV SeTAC<sup>2</sup>. O ganhador foi o aluno Jonathan Galdino da Silva, que criou um logotipo impactante, combinando componentes de tecnologia e disseminação de conhecimento. A criação do acadêmico apresenta uma árvore que tem como raiz produtora de conhecimento nossa SeTAC<sup>2</sup> e integra tudo o que é produzido via redes neurais (galhos similares a linhas de circuito impresso) para uma *cloud computing* (folhas). Assim, o desenho que consta na capa destes anais define com precisão o papel da SeTAC<sup>2</sup>: transmitir e compartilhar conhecimento técnico-científico.

Ressaltamos a importância de termos um evento que possibilite a apresentação oral de trabalhos pelos alunos, com posterior publicação dos respectivos artigos e agora também publicação de resenhas. Trata-se de atividades fundamentais para o avanço de pesquisas na área tecnológica e para a construção de conhecimento científico e social para todos profissionais da área de Tecnologia da Informação e áreas relacionadas.

Em todas as etapas do processo, contamos com o indispensável apoio do Centro Acadêmico de Ciência da Computação (CACiC), sempre com disposição, força de vontade e bom humor, que contribuiu para realizarmos um evento descontraído, sem deixar de ser acadêmico, e com participação plena dos alunos do curso. Por tudo isso, agradecemos muito ao CACiC. Agradecemos também aos professores do curso de Bacharelado em Ciência da Computação da UTFPR, câmpus Santa Helena, a participação na SeTAC<sup>2</sup> e o auxílio na organização do evento e na orientação dos alunos. De maneira geral, agradecemos à comunidade santa-helenense, que também nos deu suporte.

Os trabalhos apresentados nestes anais foram produzidos por graduandos e professores do Curso de Bacharelado em Ciência da Computação da UTFPR, câmpus Santa Helena. Os temas abordados encaixam-se nas seguintes áreas: Sistemas de *software*; Controle e Automação, Sistemas Embarcados e Robótica; Visão Computacional, Processamento de Imagens e Reconhecimento de Padrões; e Inteligência e Matemática Computacional, que representam as linhas de pesquisa do curso. Nesta edição, todas essas áreas estão representadas, o que nos orgulha e estimula a fazer um trabalho cada vez melhor.

Sendo assim, a coordenação da IV SeTAC<sup>2</sup> tem a satisfação de partilhar esta segunda edição dos anais com toda a comunidade. Os trabalhos aqui publicados consolidam o crescimento da pesquisa e do pensamento tecnológico dentro de nosso curso, o que é indiscutivelmente relevante para a área de computação e afins.

Prof. Giuvane Conti  
Coordenador da IV SeTAC<sup>2</sup>

## Sumário

### ARTIGOS

Modelo matemático do sinal <i>wi-fi</i> de um <i>access point</i> via método dos mínimos quadrados .....	6
Nathalia Vieira Mota de Oliveira Evandro Alves Nakajima	
Cálculo de área desmatada por meio de métodos numéricos .....	19
Vera Lúcia V. S. Araújo Tatiane Tambarussi	
Otimização de rotas e redução do custo por meio do algoritmo de Dijkstra .....	36
Leonardo Nogueira da Silva Tatiane Tambarussi	
Uma avaliação empírica dos efeitos de ataques de negação de serviço sobre redes definidas por <i>software</i> .....	48
Márcio José Araújo Luca Prediger Euclides Peres Farias Junior Michele Nogueira Lima	
Conceitos de qualidade de <i>software</i> aplicados ao desenvolvimento de um sistema <i>web</i> .....	61
Luca Prediger Márcio José Araújo Suzan Kelly Borges Piovesan	
Construção de um motor gráfico com base em um modelo formal simples.....	72
Rafael Campos Nunes Davi Marcondes Rocha	
Unidade móvel autônoma para aspirador de pó doméstico .....	82
Gabriel Lenin Silva Lima Geovani José da Silva Willian de Oliveira Silva Itamar Iliuk	
Braço robótico didático em ABS via comunicação serial.....	100
Antônio César Guardia Mikael Pereira Messias Nelsi Kelm	
Proposta de automação de um sistema de irrigação para estufa utilizando dispositivos microcontrolados .....	118
Jonathan Galdino da Silva Danilo Luiz Cavazim Neto Itamar Iliuk	

**RESENHAS**

Conhecimento tecnológico: a importância para o desenvolvimento social.....	138
Felipe Carvalho Funck	
Pensamento Computacional no Brasil .....	144
Larissa Gabriela Dias Vidal	
O jogo da imitação: o reconhecimento tardio.....	149
Vera Lúcia V. S. Araújo	

# Modelo matemático do sinal *wi-fi* de um *access point* via método dos mínimos quadrados

## Mathematical model of the wi-fi signal of an access point by least squares method

Nathalia Vieira Mota de Oliveira<sup>1</sup>, Evandro Alves Nakajima<sup>2</sup>

Ciência da Computação, Universidade Tecnológica Federal do Paraná (UTFPR-SH)

<sup>1</sup>nathaliaoliveira@alunos.utfpr.edu.br, <sup>2</sup>enakajima@utfpr.edu.br

### RESUMO

Este trabalho descreve um modelo matemático do sinal de um *access point* utilizando como ferramenta o cálculo numérico, mais especificamente por meio do Método dos Mínimos Quadrados, o que permite que a superfície do comportamento do sinal seja representada conforme o afastamento do *access point*. Para tal, é capturada a força do sinal em diferentes pontos do ambiente onde se encontra o *access point*. Pela aplicação do método, é obtido então um polinômio de segundo grau em duas variáveis. Dele pode-se obter a estimativa dos resultados em pontos não tabelados, representando então a superfície de aproximação da difusão do sinal *wi-fi*. O modelo proposto mostrou-se eficiente no processo de modelagem, produzindo resultados com taxa de erro de aproximadamente 3% em relação à escala utilizada. A partir dele é possível estimar a força do sinal em qualquer ponto de um ambiente, além de conhecer os limites do alcance do sinal.

**Palavras-chave:** Método dos mínimos quadrados; aproximação de superfícies; comportamento de sinal *wi-fi*; matemática aplicada.

### ABSTRACT

This paper describes a mathematical model of the signal of an access point using numerical calculus as tool and, more specifically, the Least Squares Method, which allows the surface of the behavior of the signal to be represented according to the distance of the access point. For this, the signal strength was captured at different points in the environment where the access point was located. By applying the method, a second-degree polynomial was obtained in two variables. From it, the results can be obtained in non-tabulated points, representing the surface of the approximation of the wi-fi signal. The proposed model proved to be efficient in the modeling process, producing results with error rate around 3% according to the used scale. Using it makes it possible to estimate the signal strength on any point of the environment, allowing the boundaries it reaches to be known.

**Keywords:** Least squares method; surface approach; wi-fi signal behavior; applied mathematics.

### Referência:

OLIVEIRA, N. V. M.; NAKAJIMA, E. A. Modelo matemático do sinal *wi-fi* de um *access point* via método dos mínimos quadrados. In: SEMANA TECNOLÓGICA ACADÊMICA DE CIÊNCIA DA COMPUTAÇÃO DA UTFPR-SH, 4., 2017, Santa Helena, PR. Anais..., UTFPR, Santa Helena, p.6-18. Disponível em <<http://sh.utfpr.edu.br/setac/anais2017.pdf>>.

## INTRODUÇÃO

**N**a sociedade moderna, é fácil observar e concluir que muitas pessoas permanecem grande parte de seu tempo conectadas à Internet, mais precisamente às redes *wi-fi*, presentes em residências, instituições e estabelecimentos comerciais. Essas redes surgiram nos anos 90, para

suprir a necessidade de se estar conectado a partir de dispositivos móveis, que se popularizavam na época. Hoje, estão por toda parte.

A partir da observação do fenômeno da popularização das redes *wi-fi*, percebeu-se a oportunidade de conhecer mais sobre o comportamento desse sinal, usando a matemática, que é grande base para áreas e disciplinas tecnológicas, foco da universidade, onde se desenvolve esta pesquisa.

Muitas vezes, no intuito de resolver problemas reais, é viável utilizar ferramentas de matemática que podem ser adaptadas a determinada situação, e para tal é preciso também modelar matematicamente o próprio problema. Nesse sentido, este trabalho explora o cálculo numérico, por tratar-se da subárea da matemática mais utilizada em problemas reais, devido às suas soluções por aproximação. No cálculo numérico, encontram-se tradicionais métodos, como o Método dos Mínimos Quadrados, que diferentemente dos outros métodos de interpolação, tem superfície de aproximação compreendida de maneira mais global e menos pontual, sendo adequado para previsões em pontos desconhecidos.

Assim, o objetivo deste estudo é modelar a superfície de aproximação do comportamento de um sinal *wi-fi*. O processo é dividido em duas fases: a coleta da força do sinal em pontos específicos de um ambiente e, a partir dela, a aplicação do Método dos Mínimos Quadrados, donde se pode conhecer a superfície procurada.

## 1 REFERENCIAL TEÓRICO

É comum que, durante a realização de experimentos científicos, um conjunto de pares de dados  $(x_i, f(x_i))$  – onde  $x_i$  é um dado fixo como, por exemplo, localização, tempo, temperatura, etc. e  $f(x_i)$  é um dado que depende de  $x_i$ , em geral dado por uma função desconhecida – seja obtido como resultado. Uma vez que se deseja trabalhar e manipular esse resultado, pode ser necessário conhecer o valor em um certo ponto, não obtido por meio experimental. Nesses casos, podem ser utilizados recursos de aproximação por meio de funções conhecidas, como polinômios com diferentes graus de aproximação e por meio de diferentes métodos, com o objetivo de obter uma função que passe pelos pontos ou o mais próximo possível deles.

Para aproximação de valores  $(x, y)$  que não constem em uma tabela, alguns métodos tradicionais de interpolação polinomial podem ser utilizados. Porém, quando se deseja extrapolar esses dados, estimando a  $f(x)$  de um ponto fora do intervalo conhecido, a interpolação não é aconselhável. Em contrapartida, o Método dos Mínimos Quadrados trabalha de forma a identificar a curva que mais bem se ajusta aos pontos tabelados, com boa margem de segurança e aproximação da função desconhecida (RUGGIERO; LOPES, 1996).

### 1.1 O MÉTODO DOS MÍNIMOS QUADRADOS PARA UMA VARIÁVEL

Para substituir uma função  $f(x)$ , conhecida por meio de pontos dados, por uma curva  $g(x)$  que melhor se ajuste a esses pontos de tal modo que a distância entre as duas seja a menor possível, é preciso fazer uma escolha apropriada para essa  $g(x)$  (FRANCO, 2010).

O método escolhido consiste, em um primeiro momento, em plotar os pontos  $(x_i, f(x_i))$  em um gráfico cartesiano, a fim de que se possa observar o comportamento da curva. A esse gráfico, dá-se o nome de Diagrama de Dispersão. O método dos Mínimos Quadrados, em sua perspectiva discreta, possibilitará a escolha de uma função  $g(x)$ , por meio de coeficientes  $a_0, a_1, \dots, a_n$ , que serão os parâmetros para essa aproximação.

Para o problema proposto, é utilizada uma aproximação polinomial  $g(x) = P_m(x)$ , na forma:

$$P_m(x) = a_0 + a_1 x + \dots + a_m x^m.$$

Define-se  $Q$  como o quadrado da soma dos desvios entre os valores da função e o polinômio calculado:

$$Q = \|f - P_m\|^2.$$

Sejam  $x_0, x_1, \dots, x_n, n+1$  pontos distintos, deve-se encontrar o polinômio de grau máximo  $m$ , com  $m \leq n$ , tal que  $Q$  é o mínimo possível. Sendo assim, para a determinação dos coeficientes, usa-se o produto escalar:

$$(f, g) = \sum_{k=0}^n f(x_k) \cdot g(x_k),$$

onde, fazendo  $y_i = f(x_i)$ , obtém-se:

$$\begin{aligned} Q &= \|f - P_m\|^2 = (f - P_m, f - P_m) \\ &= \sum_{k=0}^n [f(x_k) - P_m(x_k)]^2 = \sum_{k=0}^n (y_k - P_m(x_k))^2 \\ &= \sum_{k=0}^n (y_k - (a_0 + a_1 x_k + \dots + a_m x_k^m))^2. \end{aligned}$$

Deve-se então encontrar os coeficientes de  $P_m(x)$  de modo a minimizar  $Q$ . Esse processo é descrito detalhadamente para um polinômio de grau dois e de duas variáveis na metodologia deste trabalho.

A menos que seja sugerido o produto escalar a ser utilizado, emprega-se o produto escalar usual do  $\mathbb{R}^n$ , isto é:

$$(x, y) = \sum_{i=1}^n x_i y_i,$$

onde  $x = (x_0, x_1, \dots, x_n)^t$  e  $y = (y_0, y_1, \dots, y_n)^t$ .

Para encontrar os coeficientes de  $P_n(x)$  de modo a minimizar  $Q$ , define-se

$$p = \begin{bmatrix} P_m(x_0) \\ P_m(x_1) \\ \vdots \\ P_m(x_m) \end{bmatrix}, \quad u_0 = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \quad u_i = \begin{bmatrix} x_0^i \\ x_1^i \\ \vdots \\ x_n^i \end{bmatrix}, \quad i = 1, 2, \dots, m.$$

Nota-se que é possível reescrever  $p$  na forma:

$$p = a_0 \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + a_1 \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} + a_2 \begin{bmatrix} x_0^2 \\ x_1^2 \\ \vdots \\ x_n^2 \end{bmatrix} + \dots + a_m \begin{bmatrix} x_0^m \\ x_1^m \\ \vdots \\ x_n^m \end{bmatrix},$$

o que induz à forma matricial:

$$p = \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^m \\ 1 & x_1 & x_1^2 & \cdots & x_1^m \\ \vdots & \cdots & & & \\ 1 & x_m & x_m^2 & \cdots & x_m^m \\ \vdots & \cdots & & & \\ 1 & x_n & x_n^2 & \cdots & x_n^m \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix}.$$

Os coeficientes  $a_0, a_1, \dots, a_m$  do polinômio procurado são então dados pelo seguinte sistema linear:

$$\begin{bmatrix} (u_0, u_0) & (u_1, u_0) & \cdots & (u_m, u_0) \\ (u_0, u_1) & (u_1, u_1) & \cdots & (u_m, u_1) \\ \vdots & \vdots & \ddots & \vdots \\ (u_0, u_m) & (u_1, u_m) & \cdots & (u_m, u_m) \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} (y, u_0) \\ (y, u_1) \\ \vdots \\ (y, u_m) \end{bmatrix}.$$

## 1.2 REDES SEM FIO PADRÃO 802.11

Em meados da década de 90, deu-se o surgimento das redes sem fio padrão IEEE 802.11, popularmente conhecidas como *wi-fi*. Seu surgimento veio a atender o anseio de se ter equipamentos conectados à Internet e entre si, independentes de cabos e fios.

O sinal *wi-fi* é composto por ondas de rádio, que necessitam de um emissor e um receptor para se estabelecer uma comunicação, além de um padrão definido, a fim de ser compatível com ambos. As redes 802.11 operam entre as frequências 2,4-2,5GHz e atendem a clientes diversos, como *notebooks* e telefones móveis, por meio de infraestruturas chamadas de pontos de acesso (*access point*) (TANENBAUM; WETHERALL, 2011). As ondas *wi-fi* são medidas em Decibel miliwatt (dBm), que é utilizado no setor de telecomunicações para unidades de potência absoluta, mediante uma relação logarítmica.

## 2 METODOLOGIA

Para atender à modelagem proposta, faz-se então uma adaptação do método para duas variáveis,  $x$  e  $y$ , que representam a posição da coleta de dados.

A fim de aproximar uma função  $f(x, y)$  por um polinômio  $Z = p(x, y)$  em duas variáveis, deve-se fazer:

$$Z = p(x, y) = A + Bx + Cy + Dx^2 + Exy + Fy^2$$

Por se tratar do caso discreto, tem-se então:

$$Z = p(x_i, y_i) = A + Bx_i + Cy_i + Dx_i^2 + Ex_i y_i + Fy_i^2 \quad e \quad f(x_i y_i) = Z_i.$$

Para que  $p(x, y)$  seja uma boa aproximação, a distância

$$d(p(x_i y_i), f(x_i y_i)) = \| p(x_i, y_i) - f(x_i, y_i) \|$$

deve ser mínima, onde  $\| \cdot \|$  é a norma definida pelo produto interno usual no conjunto das funções contínuas em  $\mathbb{R}$  (BOLDRINI, 1980). Isso é equivalente a dizer que a função distância ao quadrado das funções  $f$  e  $p$

$$Q(A, B, C, D, E, F) = d(p(x_i y_i), f(x_i y_i))^2 = \| p(x_i y_i) - f(x_i y_i) \|^2$$

deve ter o menor valor possível.

Como  $Q(A, B, C, D, E, F) \geq 0$ , e por ser a função distância ao quadrado, tem-se que, se um ponto crítico (STEWART, 2007) é um extremo, então é um ponto de mínimo. Assim, para encontrar esse ponto crítico, deve-se calcular o gradiente de  $Q(A, B, C, D, E, F)$ , considerando que

os pontos  $x_i, y_i$  e  $Z_i$  são fixos e que as variáveis são  $A, B, C, D, E, F$ . Sendo assim, sabendo que

$$Q(A, B, C, D, E, F) = \sum_{i=1}^n (Z - Z_i)^2,$$

tem-se:

$$\begin{aligned} \frac{\partial Q}{\partial A}(A, B, C, D, E, F) &= \frac{\partial}{\partial A} \left( \sum_{i=1}^n (Z - Z_i)^2 \right) = \sum_{i=1}^n \frac{\partial}{\partial A} (Z - Z_i)^2 \\ &= \sum_{i=1}^n 2(Z - Z_i) \frac{\partial Z}{\partial A} \\ &= 2 \sum_{i=1}^n (Z - Z_i). \end{aligned}$$

Procedendo da mesma forma para as demais variáveis, obtém-se:

$$\begin{aligned} \frac{\partial Q}{\partial B}(A, B, C, D, E, F) &= 2 \sum_{i=1}^n (Z - Z_i)x_i, \\ \frac{\partial Q}{\partial C}(A, B, C, D, E, F) &= 2 \sum_{i=1}^n (Z - Z_i)y_i, \\ \frac{\partial Q}{\partial D}(A, B, C, D, E, F) &= 2 \sum_{i=1}^n (Z - Z_i)x_i^2, \\ \frac{\partial Q}{\partial E}(A, B, C, D, E, F) &= 2 \sum_{i=1}^n (Z - Z_i)x_iy_i, \\ \frac{\partial Q}{\partial F}(A, B, C, D, E, F) &= 2 \sum_{i=1}^n (Z - Z_i)y_i^2. \end{aligned}$$

Assim, se  $v = (A, B, C, D, E, F)$ , o gradiente de  $Q$  em  $v$  é

$$\nabla Q(v) = \left( \frac{\partial Q}{\partial A}(v), \frac{\partial Q}{\partial B}(v), \frac{\partial Q}{\partial C}(v), \frac{\partial Q}{\partial D}(v), \frac{\partial Q}{\partial E}(v), \frac{\partial Q}{\partial F}(v) \right).$$

Logo, para encontrar o mínimo de  $Q$ , deve-se obter o ponto crítico de  $Q$ , isto é, os valores de  $A, B, C, D, E, F$  que satisfazem as equações seguintes:

$$\frac{\partial Q}{\partial A}(v) \Rightarrow \sum_{i=1}^n (A + Bx_i + Cy_i + Dx_i^2 + Ex_i, y_i + Fy_i^2 - Z_i) 1 = 0 \quad (1)$$

$$\frac{\partial Q}{\partial B}(v) \Rightarrow \sum_{i=1}^n (A + Bx_i + Cy_i + Dx_i^2 + Ex_i, y_i + Fy_i^2 - Z_i) x_i = 0 \quad (2)$$

$$\frac{\partial Q}{\partial C}(v) \Rightarrow \sum_{i=1}^n (A + Bx_i + Cy_i + Dx_i^2 + Ex_i, y_i + Fy_i^2 - Z_i) y_i = 0 \quad (3)$$

$$\frac{\partial Q}{\partial D}(v) \Rightarrow \sum_{i=1}^n (A + Bx_i + Cy_i + Dx_i^2 + Ex_i, y_i + Fy_i^2 - Z_i) x_i^2 = 0 \quad (4)$$

$$\frac{\partial Q}{\partial E}(v) \Rightarrow \sum_{i=1}^n (A + Bx_i + Cy_i + Dx_i^2 + Ex_i, y_i + Fy_i^2 - Z_i) x_i y_i = 0 \quad (5)$$

$$\frac{\partial Q}{\partial F}(v) \Rightarrow \sum_{i=1}^n (A + Bx_i + Cy_i + Dx_i^2 + Ex_i, y_i + Fy_i^2 - Z_i) y_i^2 = 0 \quad (6)$$

isto é,

$$\begin{aligned}
 (1) &\Rightarrow \sum A + Bx_i + Cy_i + Dx_i^2 + Ex_i y_i + Fy_i^2 &= \sum Z_i \cdot 1 \\
 (2) &\Rightarrow \sum Ax_i + Bx_i^2 + Cx_i y_i + Dx_i^3 + Ex_i^2 y_i + Fx_i y_i^2 &= \sum Z_i \cdot x_i \\
 (3) &\Rightarrow \sum Ay_i + Bx_i y_i + Cy_i^2 + Dx_i^2 y_i + Ex_i y_i^2 + Fy_i^3 &= \sum Z_i \cdot y_i \\
 (4) &\Rightarrow \sum Ax_i^2 + Bx_i^3 + Cx_i^2 y_i + Dx_i^4 + Ex_i^3 y_i + Fx_i^2 y_i^2 &= \sum Z_i \cdot x_i^2 \\
 (5) &\Rightarrow \sum Ax_i y_i + Bx_i^2 y_i + Cx_i y_i^2 + Dx_i^3 y_i + Ex_i^2 y_i^2 + Fx_i y_i^3 &= \sum Z_i \cdot x_i \cdot y_i \\
 (6) &\Rightarrow \sum Ay_i^2 + Bx_i y_i^2 + Cy_i^3 + Dx_i^2 y_i^2 + Ex_i y_i^3 + Fy_i^4 &= \sum Z_i \cdot y_i^2.
 \end{aligned}$$

Assim, fazendo-se

$$X_0 = Y_0 = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \quad X_j = \begin{bmatrix} x_1^j \\ x_2^j \\ \vdots \\ x_n^j \end{bmatrix}, \quad Y_j = \begin{bmatrix} y_1^j \\ y_2^j \\ \vdots \\ y_n^j \end{bmatrix} \quad e \quad \bar{Z} = \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_n \end{bmatrix},$$

obtém-se

$$\begin{aligned}
 (1) &\Rightarrow AX_0Y_0 + BX_1Y_0 + CX_0Y_1 + DX_2Y_0 + EX_1Y_1 + FX_0Y_2 = \bar{Z} X_0Y_0 \\
 (2) &\Rightarrow AX_1Y_0 + BX_2Y_0 + CX_1Y_1 + DX_3Y_0 + EX_2Y_1 + FX_1Y_2 = \bar{Z} X_1Y_0 \\
 (3) &\Rightarrow AX_0Y_1 + BX_1Y_1 + CX_0Y_2 + DX_2Y_1 + EX_1Y_2 + FX_0Y_3 = \bar{Z} X_0Y_1 \\
 (4) &\Rightarrow AX_2Y_0 + BX_3Y_0 + CX_2Y_1 + DX_4Y_0 + EX_3Y_1 + FX_2Y_2 = \bar{Z} X_2Y_0 \\
 (5) &\Rightarrow AX_1Y_1 + BX_2Y_1 + CX_1Y_2 + DX_3Y_1 + EX_2Y_2 + FX_1Y_3 = \bar{Z} X_1Y_1 \\
 (6) &\Rightarrow AX_0Y_2 + BX_1Y_2 + CX_0Y_3 + DX_2Y_2 + EX_1Y_3 + FX_0Y_4 = \bar{Z} X_0Y_2.
 \end{aligned}$$

Isso significa que, para encontrar os coeficientes de  $p(x, y)$ , basta resolver o seguinte sistema linear:

$$\begin{bmatrix} X_0Y_0 & X_1Y_0 & X_0Y_1 & X_2Y_0 & X_1Y_1 & X_0Y_2 \\ X_1Y_0 & X_2Y_0 & X_1Y_1 & X_3Y_0 & X_2Y_1 & X_1Y_2 \\ X_0Y_1 & X_1Y_1 & X_0Y_2 & X_2Y_1 & X_1Y_2 & X_0Y_3 \\ X_2Y_0 & X_3Y_0 & X_2Y_1 & X_4Y_0 & X_3Y_1 & X_2Y_2 \\ X_1Y_1 & X_2Y_1 & X_1Y_2 & X_3Y_1 & X_2Y_2 & X_1Y_3 \\ X_0Y_2 & X_1Y_2 & X_0Y_3 & X_2Y_2 & X_1Y_3 & X_0Y_4 \end{bmatrix} \cdot \begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix} = \begin{bmatrix} \bar{Z} X_0Y_0 \\ \bar{Z} X_1Y_0 \\ \bar{Z} X_0Y_1 \\ \bar{Z} X_2Y_0 \\ \bar{Z} X_1Y_1 \\ \bar{Z} X_0Y_2 \end{bmatrix}.$$

## 2.1 DISTRIBUIÇÃO DAS MEDIÇÕES NO ESPAÇO ESCOLHIDO

Para a aplicação do método desenvolvido neste estudo, foi escolhido o restaurante universitário da UTFPR câmpus Santa Helena, que se caracteriza por ser um ambiente *indoor*, sem a presença de grande número de paredes e aparelhos, o que causa mínima interferência na transmissão

do sinal *wi-fi*, porém o ambiente não é totalmente livre dela, já que o salão conta com alguns pilares de sustentação.

A relação entre as variáveis *x*, *y* e *z* é feita de maneira que *x* e *y* representam a posição na qual a medição é realizada a partir do *access point*. A escala de uma unidade para cada 60cm foi adotada, sendo assim, os eixos *x* e *y* compreendem valores de -9 a 9 (metros), ortogonais entre si e, tendo *z* como a força do sinal naquele ponto, em dBm, a aproximadamente um metro e meio de distância, verticalmente, abaixo do aparelho. A escolha dos pontos foi feita de forma aleatória, conforme a disposição estrutural do restaurante.



Figura 1: Fotos do restaurante universitário da UTFPR-SH, destacando o *access point*.

## 2.2 EMISSÃO DO SINAL WI-FI

Os sinais *wi-fi* medidos foram emitidos por meio de um *access point* da marca Cisco, modelo Cisco Aironet 1140 Series Access Point, com as especificações CISCO AIR-LAP1142N-T-K9. O aparelho segue o padrão IEEE de *wi-fi* (802.11) e fica localizado no alto da região central do ambiente, onde, em seu entorno, foram feitas as medições.



Figura 2: Exemplo do modelo do *access point* utilizado.

## 2.3 CAPTAÇÃO DO SINAL WI-FI

Para a captação do sinal emitido pelo *access point*, utilizou-se um aparelho de celular *smartphone*, da marca Motorola, modelo Moto G3 e sistema operacional Android, juntamente com o aplicativo para *smartphone* *wi-fi Analyzer*, que permite a visualização das redes disponíveis, com uma interface gráfica que representa a potência do sinal, com um intervalo que vai de -40 a -100 dBm.



Figura 3: Interface do aplicativo utilizado.

## 2.4 APLICAÇÃO DO MÉTODO DESENVOLVIDO PROPOSTO E REPRESENTAÇÃO GRÁFICA DOS RESULTADOS

Como ferramenta para a aplicação do Método dos Mínimos Quadrados, optou-se pelo editor de planilhas Excel, já que nele é possível listar os pontos e realizar os cálculos do começo ao fim do processo. Uma vez que a variável  $z$  tem sua unidade com diferente padrão de intervalo das demais, opta-se pela técnica de Mudança de Variáveis, a fim de possibilitar os cálculos. Para melhor visualização, os pontos foram plotados na ferramenta do *site* Geogebra 2017, em gráficos bidimensional e tridimensional, os quais caracterizam o diagrama de dispersão, pois possibilitam que se tenha melhor noção da distribuição dos pontos coletados. Já o gráfico 3D, constituído pelos pontos coletados e da superfície do polinômio de aproximação, foi plotado no *software* Maple 2016.

## 3 RESULTADOS E DISCUSSÃO

### 3.1 VISUALIZAÇÃO DOS PONTOS COLETADOS

Sendo a obtenção dos dados e a visualização de sua disposição o primeiro passo na modelagem do problema, conforme figuras abaixo, é possível observar o comportamento de um paraboloide elíptico, de equação normal

$$z = \pm \frac{x^2}{a^2} \pm \frac{y^2}{b^2}.$$

### 3.2 ORGANIZAÇÃO DOS DADOS NO EXCEL

A fim de utilizar o Excel como ferramenta para efetuar os cálculos necessários, primeiramente é preciso organizar os dados coletados conforme demonstrado na primeira seção deste trabalho, e  $Z'$  é o resultado do processo de mudança linear da variável  $Z'$ , feito para converter o

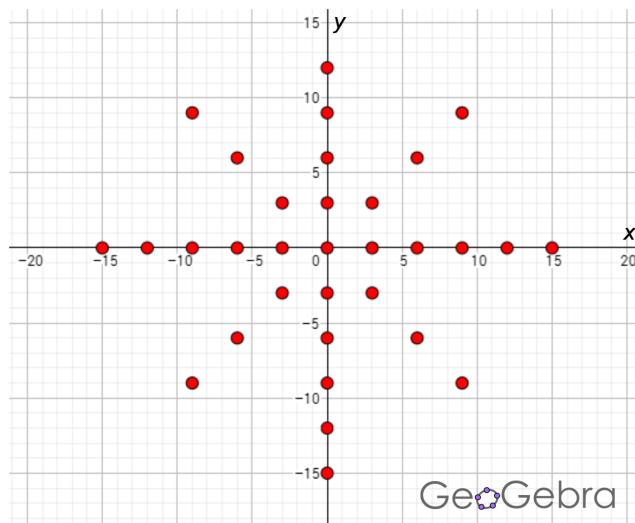


Figura 4: Gráfico 2D dos pontos coletados.

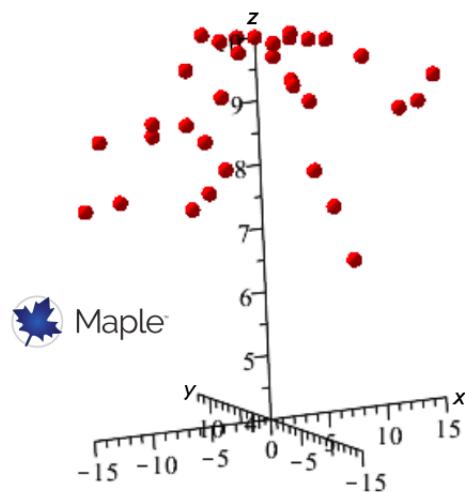


Figura 5: Gráfico 3D dos pontos coletados.

intervalo da força do sinal de  $[-100, -40]$  para  $[0, 10]$ , deixando tais valores mais próximos de X e Y e, assim, facilitando sua visualização.

X	Y	Z	Z'
-15	0	-54,43	7,5950
-12	0	-54	7,6666
-9	0	-47	8,8333
-6	0	-42,33	9,6116
-3	0	-40	10,0000
0	0	-40	10,0000
3	0	-40	10,0000
6	0	-41	9,8333
9	0	-43	9,5000
12	0	-48,25	8,6250
15	0	-45,5	9,0833
0	-3	-40	10,0000
0	-6	-43,4	9,4333
0	-9	-50,8	8,2000
0	-12	-53,6	7,7333
0	-15	-58	7,0000
0	3	-40,6	9,9000
0	6	-46,86	8,8566
0	9	-51,67	8,0550
0	12	-50,67	8,2216
3	3	-42,8	9,5333
6	6	-42	9,6666
9	9	-49	8,5000
3	-3	-40	10,0000
6	-6	-46	9,0000
9	-9	-45,4	9,1000
-3	-3	-40,5	9,9166
-6	-6	-50,5	8,2500
-9	-9	-51,71	8,0483
-3	3	-40	10,0000
-6	6	-49,71	8,3816
-9	9	-50,5	8,2500

Figura 6: Tabela dos valores das variáveis X, Y, Z, Z'.

Com os dados coletados, utiliza-se o Excel para calcular os produtos escalares  $X_i Y_i$ , que compõem a matriz dos coeficientes do sistema linear a ser resolvido, como segue:

X0	Y0	Prod.	X1	Y0	Prod.	X2	Y0	Prod.	X0	Y2	Prod.
1	1	1	-15	1	-15	225	1	225	1	0	0
1	1	1	-12	1	-12	144	1	144	1	0	0
1	1	1	-9	1	-9	81	1	81	1	0	0
1	1	1	-6	1	-6	36	1	36	1	0	0
1	1	1	-3	1	-3	9	1	9	1	0	0
1	1	1	0	1	0	0	1	0	1	0	0
1	1	1	3	1	3	9	1	9	1	0	0
1	1	1	6	1	6	36	1	36	1	0	0
1	1	1	9	1	9	81	1	81	1	0	0
1	1	1	12	1	12	144	1	144	1	0	0
1	1	1	15	1	15	225	1	225	1	0	0
1	1	1	0	1	0	0	1	0	1	9	9
1	1	1	0	1	0	0	1	0	1	36	36
1	1	1	0	1	0	0	1	0	1	81	81
1	1	1	0	1	0	0	1	0	1	144	144
1	1	1	0	1	0	0	1	0	1	225	225
1	1	1	0	1	0	0	1	0	1	9	9
1	1	1	0	1	0	0	1	0	1	36	36
1	1	1	0	1	0	0	1	0	1	81	81
1	1	1	0	1	0	0	1	0	1	144	144
1	1	1	3	1	3	9	1	9	1	9	9
1	1	1	6	1	6	36	1	36	1	36	36
1	1	1	9	1	9	81	1	81	1	81	81
1	1	1	3	1	3	9	1	9	1	9	9
1	1	1	6	1	6	36	1	36	1	36	36
1	1	1	9	1	9	81	1	81	1	81	81
1	1	1	-3	1	-3	9	1	9	1	9	9
1	1	1	-6	1	-6	36	1	36	1	36	36
1	1	1	-9	1	-9	81	1	81	1	81	81
1	1	1	-3	1	-3	9	1	9	1	9	9
1	1	1	-6	1	-6	36	1	36	1	36	36
1	1	1	-9	1	-9	81	1	81	1	81	81
<b>Soma =</b>			<b>32</b>	<b>Soma =</b>			<b>0</b>	<b>Soma =</b>			<b>1494</b>
<b>Soma =</b>				<b>Soma =</b>				<b>Soma =</b>			<b>1269</b>

Figura 7: Tabelas de produtos escalares da matriz dos coeficientes.

Os demais produtos são feitos analogamente, obtendo-se então a matriz A dos coeficientes:

32,0000	0,0000	-15,0000	1494,0000	0,0000	1269,0000
0,0000	1494,0000	0,0000	0,0000	0,0000	0,0000
-15,0000	0,0000	1269,0000	0,0000	0,0000	-3375,0000
1494,0000	0,0000	0,0000	190350,0000	0,0000	31752,0000
0,0000	0,0000	0,0000	0,0000	31752,0000	0,0000
1269,0000	0,0000	-3375,0000	31752,0000	0,0000	139725,0000

Figura 8: Matriz A dos coeficientes.

Além desses valores, é preciso ainda calcular os valores dos produtos no vetor solução, isto é:

X0	Y0	Z'	Prod.	X1	Y0	Z'	Prod.	X0	Y1	Z'	Prod.
1	1	7,5950	7,5950	-15	1	7,5950	-113,9250	1	0	7,5950	0,0000
1	1	7,6666	7,6666	-12	1	7,6666	-91,9992	1	0	7,6666	0,0000
1	1	8,8333	8,8333	-9	1	8,8333	-79,4997	1	0	8,8333	0,0000
1	1	9,6116	9,6116	-6	1	9,6116	-57,6696	1	0	9,6116	0,0000
1	1	10,0000	10,0000	-3	1	10,0000	-30,0000	1	0	10,0000	0,0000
1	1	10,0000	10,0000	0	1	10,0000	0,0000	1	0	10,0000	0,0000
1	1	10,0000	10,0000	3	1	10,0000	30,0000	1	0	10,0000	0,0000
1	1	9,8333	9,8333	6	1	9,8333	58,9998	1	0	9,8333	0,0000
1	1	9,5000	9,5000	9	1	9,5000	85,5000	1	0	9,5000	0,0000
1	1	8,6250	8,6250	12	1	8,6250	103,5000	1	0	8,6250	0,0000
1	1	9,0833	9,0833	15	1	9,0833	136,2495	1	0	9,0833	0,0000
1	1	10,0000	10,0000	0	1	10,0000	0,0000	1	-3	10,0000	-30,0000
1	1	9,4333	9,4333	0	1	9,4333	0,0000	1	-6	9,4333	-56,5998
1	1	8,2000	8,2000	0	1	8,2000	0,0000	1	-9	8,2000	-73,8000
1	1	7,7333	7,7333	0	1	7,7333	0,0000	1	-12	7,7333	-92,7996
1	1	7,0000	7,0000	0	1	7,0000	0,0000	1	-15	7,0000	-105,0000
1	1	9,9000	9,9000	0	1	9,9000	0,0000	1	3	9,9000	29,7000
1	1	8,8566	8,8566	0	1	8,8566	0,0000	1	6	8,8566	53,1396
1	1	8,0550	8,0550	0	1	8,0550	0,0000	1	9	8,0550	72,4950
1	1	8,2216	8,2216	0	1	8,2216	0,0000	1	12	8,2216	98,6592
1	1	9,5333	9,5333	3	1	9,5333	28,5999	1	3	9,5333	28,5999
1	1	9,6666	9,6666	6	1	9,6666	57,9996	1	6	9,6666	57,9996
1	1	8,5000	8,5000	9	1	8,5000	76,5000	1	9	8,5000	76,5000
1	1	10,0000	10,0000	3	1	10,0000	30,0000	1	-3	10,0000	-30,0000
1	1	9,0000	9,0000	6	1	9,0000	54,0000	1	-6	9,0000	-54,0000
1	1	9,1000	9,1000	9	1	9,1000	81,9000	1	-9	9,1000	-81,9000
1	1	9,9166	9,9166	-3	1	9,9166	-29,7498	1	-3	9,9166	-29,7498
1	1	8,2500	8,2500	-6	1	8,2500	-49,5000	1	-6	8,2500	-49,5000
1	1	8,0483	8,0483	-9	1	8,0483	-72,4347	1	-9	8,0483	-72,4347
1	1	10,0000	10,0000	-3	1	10,0000	-30,0000	1	3	10,0000	30,0000
1	1	8,3816	8,3816	-6	1	8,3816	-50,2896	1	6	8,3816	50,2896
1	1	8,2500	8,2500	-9	1	8,2500	-74,2500	1	9	8,2500	74,2500
<b>Soma =</b>				<b>Soma =</b>				<b>Soma =</b>			
<b>286,794</b>				<b>63,9312</b>				<b>-104,151</b>			

Figura 9: Tabelas dos produtos na primeira, segunda e terceira linhas do vetor solução.

As demais linhas são feitas de forma análoga. Por fim, pode-se utilizar o método da matriz inversa por meio do excel (FRANCO, 2010), para resolver o sistema:

$$\begin{array}{|c|c|c|c|c|c|} \hline 32,0000 & 0,0000 & -15,0000 & 1494,0000 & 0,0000 & 1269,0000 \\ \hline 0,0000 & 1494,0000 & 0,0000 & 0,0000 & 0,0000 & 0,0000 \\ \hline -15,0000 & 0,0000 & 1269,0000 & 0,0000 & 0,0000 & -3375,0000 \\ \hline 1494,0000 & 0,0000 & 0,0000 & 190350,0000 & 0,0000 & 31752,0000 \\ \hline 0,0000 & 0,0000 & 0,0000 & 0,0000 & 31752,0000 & 0,0000 \\ \hline 1269,0000 & 0,0000 & -3375,0000 & 31752,0000 & 0,0000 & 139725,0000 \\ \hline \end{array} \cdot \begin{array}{|c|c|c|c|} \hline A & B & C & D \\ \hline E & F & G & H \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline 286,7950 & & & \\ \hline 63,9300 & & & \\ \hline -104,1500 & & & \\ \hline 12835,2000 & & & \\ \hline -50,6250 & & & \\ \hline 10398,2700 & & & \\ \hline \end{array}$$

Figura 10: Sistema linear a ser resolvido.

obtendo-se a matriz inversa

0,0804	0,0000	-0,0007	-0,0005	0,0000	-0,0006
0,0000	0,0007	0,0000	0,0000	0,0000	0,0000
-0,0007	0,0000	0,0009	0,0000	0,0000	0,0000
-0,0005	0,0000	0,0000	0,0000	0,0000	0,0000
0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
-0,0006	0,0000	0,0000	0,0000	0,0000	0,0000

Figura 11: Matriz inversa de A.

e fazendo-se a multiplicação com o vetor solução

0,0804	0,0000	-0,0007	-0,0005	0,0000	-0,0006		286,7950	9,8443
0,0000	0,0007	0,0000	0,0000	0,0000	0,0000		63,9300	0,0428
-0,0007	0,0000	0,0009	0,0000	0,0000	0,0000		-104,1500	-0,0010
-0,0005	0,0000	0,0000	0,0000	0,0000	0,0000		12835,2000	-0,0076
0,0000	0,0000	0,0000	0,0000	0,0000	0,0000		-50,6250	-0,0016
-0,0006	0,0000	0,0000	0,0000	0,0000	0,0000		10398,2700	-0,0133

Figura 12: Produto da matriz inversa pelo vetor solução.

onde o valor dos coeficientes obtido é:

$$A = 9,8443 \quad B = 0,0428 \quad C = -0,0010 \quad D = -0,0076 \quad E = -0,0016 \quad F = -0,0133$$

portanto, o polinômio de segundo grau em duas variáveis que aproxima o problema é

$$P_2(x, y) = 9,8443 + 0,0428x - 0,0010y - 0,0076x^2 - 0,0016xy - 0,0133y^2$$

o qual, aplicado em cada ponto  $(x, y)$  coletado, permite finalmente obter a superfície de aproximação da difusão do sinal.

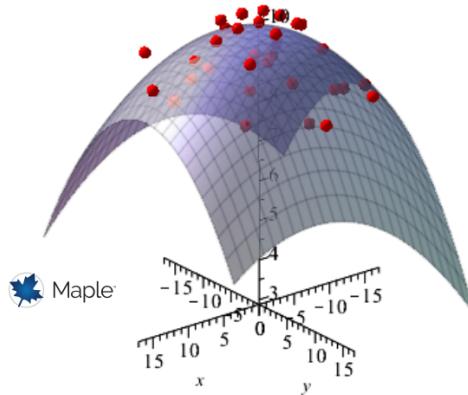


Figura 13: Paraboloide elíptico que representa a difusão do sinal wi-fi.

A superfície encontrada é um paraboloide elíptico e representa de maneira bem definida o sinal wi-fi. Comparando-se os dados obtidos pelo polinômio com os dados da coleta, observa-se erro máximo de 0,7042 e uma média dos erros de 0,3127.

Além disso, encontrando o polinômio que aproxima a curva do sinal wi-fi e igualando-o a 0, obtém-se os limites desse sinal, isto é, o alcance máximo do *access point*.

Z'	7,5950	7,6667	8,8333	9,6117	10,0000	10,0000	10,0000	9,8333	9,5000	8,6250	9,0833	10,0000	9,4333	8,2000	7,7333	7,0000
P	7,4879	8,2335	8,8419	9,3132	9,6473	9,8443	9,9041	9,8267	9,6122	9,2605	8,7716	9,7279	9,3724	8,7778	7,9442	6,8716
E	0,0115	0,3213	0,0001	0,0891	0,1244	0,0242	0,0092	0,0000	0,0126	0,4039	0,0971	0,0741	0,0037	0,3339	0,0445	0,0165
Z'	9,9000	8,8567	8,0550	8,2217	9,5333	9,6667	8,5000	10,0000	9,0000	9,1000	9,9167	8,2500	8,0483	10,0000	8,3817	8,2500
P	9,7217	9,3600	8,7593	7,9195	9,7671	9,2850	8,3980	9,8020	9,4122	8,6749	9,5165	8,7839	7,6463	9,5391	8,8863	7,8861
E	0,0318	0,2533	0,4960	0,0913	0,0547	0,1457	0,0104	0,0392	0,1699	0,1807	0,1601	0,2850	0,1616	0,2125	0,2547	0,1324

Figura 14: Tabela com os valores coletados  $Z'$ , os resultados do polinômio  $P$  e o erro ponto a ponto  $E$ .

A partir da superfície encontrada, pode-se aproximar o raio de alcance ou a força desse sinal em um ponto do intervalo determinado.

Dentro da disponibilidade de aparelhos *access point* da universidade, o restaurante universitário possui o ambiente mais aberto e com menor número de elementos em relação aos outros, porém com algumas colunas que podem ter influenciado a obtenção dos valores.

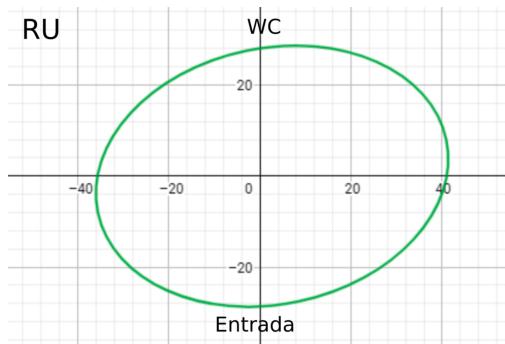


Figura 15: Elipse que ilustra os limites do sinal wi-fi.

Colunas, paredes e divisórias, assim como outros elementos, agem como refletores do sinal wi-fi. Assim, Uma medição entre paredes tende a sofrer discrepância dos valores medidos. Esse caso pode vir a ser tema de outro artigo ou de uma extensão deste.

## CONSIDERAÇÕES FINAIS

Este trabalho apresentou uma aplicação do método dos mínimos quadrados à um problema prático, nesse caso, o modelo do sinal wi-fi de um access point. Outros conceitos matemáticos vistos no decorrer do curso de Ciência da Computação foram apresentados, tais como o gradiente de uma função e o produto escalar. Sabe-se que conceitos matemáticos estão diretamente relacionados ao desenvolvimento de elementos comuns do cotidiano das pessoas, como engenharia, astronomia, softwares e tecnologias em geral. Entende-se que os resultados apresentados possam somar na compreensão da aplicação da matemática em casos mais comuns. A partir deste cenário, outros podem ser explorados, como a análise do sinal em ambientes totalmente abertos ou ambientes com presença de móveis e paredes de diferentes tamanhos e texturas, onde a dispersão ocorre de diferentes maneiras.

## REFERÊNCIAS

- BOLDRINI, J. L.; COSTA, S. I. R.; FIGUEIREDO V. L.; WETZLER H. G. **Álgebra Linear**. 3.ed. São Paulo: Harper Row do Brasil, 1980.
- IEEE, 802.11-2012. **Standard for Information technology**. Telecommunications and information exchange between systems Local and metropolitan area networks-Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Piscataway, 2012. Disponível em: <<http://standards.ieee.org/getieee802/download/802.11-2016.pdf>>. Acesso em: jun. 2017.
- FRANCO, N. B. **Cálculo numérico**. São Paulo: Pearson Prentice Hall, 2010.
- GEOGEBRA. **Graphing calculator**. International GeoGebra Institute 2017. <<https://www.geogebra.org/>>. Acesso em: jun. 2017.
- MAPLE. **Graphing calculator**. Maplesoft, a division of Waterloo Maple Inc. 2017.
- MICROSOFT EXCEL. **Planilha eletrônica**. Microsoft Inc. 2017.
- RUGGIERO, M. A. G.; LOPES, V. L. R. **Cálculo Numérico**: aspectos teóricos e computacionais. 2.ed. São Paulo: Pearson Makron Books, 1996.
- STEWART, J. **Cálculo**. 5.ed. v.2. São Paulo: Thomson Learning, 2007.
- TANENBAUM, A. S.; WETHERALL, D. **Redes de Computadores**. 5.ed. São Paulo: Pearson Prentice Hall, 2011.

# Cálculo de área desmatada por meio de métodos numéricos

## An estimation of a deforested area by numerical methods

Vera Lúcia V. S. Araújo<sup>1</sup>, Tatiane Tambarussi<sup>2</sup>

Ciência da Computação, Universidade Tecnológica Federal do Paraná (UTFPR-SH)

<sup>1</sup>[vera.2015@alunos.utfpr.edu.br](mailto:vera.2015@alunos.utfpr.edu.br), <sup>2</sup>[tambarussi@utfpr.edu.br](mailto:tambarussi@utfpr.edu.br)

### RESUMO

O objetivo deste trabalho é apresentar um cálculo aproximado de uma área de mata preservada de uma região do município de Santa Helena, próxima ao Lago de Itaipu. O cálculo aproximado é feito por métodos numéricos, como interpolação de Newton e integração numérica (3/8 de Simpson), com auxílio de softwares matemáticos e de computação gráfica para tratamento dos dados. A região calculada foi obtida por meio do Google Maps. Segundo os cálculos realizados, a área preservada no local pesquisado corresponde a aproximadamente 503.162,04m<sup>2</sup> e a área desmatada, a 907.630,24m<sup>2</sup>, em uma área total de 1.410.792,29m<sup>2</sup>.

**Palavras-chave:** Métodos numéricos; interpolação; integração numérica; área geográfica.

### ABSTRACT

The purpose of this work is to present an approximation of the calculation of a forest area preserved in a region of the municipality of Santa Helena, near Lake Itaipu. The approximate calculation is made by numerical methods, such as Newton interpolation and numerical integration (3/8 Simpson), with the aid of mathematical software and computer graphics for data processing. The calculated region was obtained through Google Maps. According to the calculations made, the area researched corresponds to approximately 503,162.04m<sup>2</sup> and the deforested area, to 907,630.24m<sup>2</sup>, in a total area of 1,410,792.29m<sup>2</sup>.

**Keywords:** Numerical methods; interpolation; numerical integration; geographical area.

### Referência:

ARAÚJO, V. L. V. S.; TAMBARUSSI, T. Cálculo de área desmatada por meio de métodos numéricos. In: SEMANA TECNOLÓGICA ACADÊMICA DE CIÊNCIA DA COMPUTAÇÃO DA UTFPR-SH, 4., 2017, Santa Helena, PR. Anais..., Santa Helena: UTFPR, 2017. p.19-35. Disponível em: <<http://sh.utfpr.edu.br/setac/Anais2017.pdf>>.

### INTRODUÇÃO

Muitas vezes, alunos de cursos de Ciências Exatas, como engenharias e Ciência da Computação, questionam a real utilidade de se aprender cálculo numérico. A princípio, o que parece não estar clara é a relação dessa teoria matemática com a realidade, o que não é difícil de se obter, pois a sociedade atual é pautada por dificuldades e problemas cada vez maiores e constantes, tendo em vista o grande volume populacional existente e as demandas que isso implica. Assim, a sociedade carece de métodos científicos para conhecer sua realidade, para transformá-la e para desenvolver tecnologia, e isso obrigatoriamente inclui matemática.

É de conhecimento geral que, nas últimas décadas, com o avanço tecnológico e a popularização dos computadores, os métodos numéricos têm sido mais amplamente utilizados em várias áreas do conhecimento, tendo em vista a possibilidade de fazer cálculos complexos rapidamente e com elevada precisão, utilizando-se softwares.

Além de *softwares* matemáticos, diversos outros, como os *softwares* gráficos, podem ser utilizados na resolução de problemas reais por aproximação. Muitos desses recursos são acessíveis a todos os usuários da Internet, que podem baixá-los para seu computador ou utilizá-los *online*.

Levando isso em conta, o objetivo deste estudo é encontrar, por aproximação, a porção de área não desmatada em uma região do município de Santa Helena, Paraná, usando métodos numéricos. A região calculada corresponde à mata ciliar situada entre lavouras e uma porção do Lago de Itaipu. Aplicam-se a interpolação de Newton e 3/8 de Simpson e utilizam-se *softwares* para coleta e edição das imagens utilizadas, bem como para efetuar os cálculos necessários.

## 1 REFERENCIAL TEÓRICO-METODOLÓGICO

### 1.1 MÉTODOS NUMÉRICOS

Sabe-se que o cálculo numérico corresponde a um conjunto de ferramentas ou métodos usados para se obter a solução de problemas matemáticos de forma aproximada. Esses métodos se aplicam principalmente a problemas que não apresentam solução exata, portanto, precisam ser resolvidos numericamente. Segundo Franco (2006), o cálculo numérico pode ser empregado em qualquer área das ciências aplicadas.

Os métodos numéricos procuram desenvolver processos de cálculo (algoritmos), utilizando uma sequência finita de operações aritméticas básicas, de forma que certos problemas matemáticos se tornem exequíveis. Esses algoritmos envolvem, em geral, um grande número de cálculos aritméticos (MONTEIRO, 2012).

De maneira geral, a resolução de um problema de qualquer área do conhecimento científico passa por uma fase de observação e entendimento do fenômeno a ser abordado, na qual se constrói um modelo matemático que represente o problema. Após a modelagem, utiliza-se um método exato, quando possível, ou, então, aproximado. Sabe-se que, nesse processo, há um número grande de operações matemáticas elementares envolvidas, assim, uma vez que elas são processadas por equipamentos com capacidade limitada para armazenar dígitos significativos de valores numéricos, erros podem ocorrer, como arredondamento e truncamento – esse último sendo um erro que ocorre cada vez que se substitui um processo matemático infinito por um processo finito ou discreto. Algoritmos aproximados também podem conduzir a erros (ARENALES e DAREZZO, 2014), no entanto, tais erros são pequenos, pois, apesar de limitada, a capacidade desses algoritmos já é grande e fica cada vez maior, conforme a computação evolui.

Há vários métodos numéricos, de forma que é preciso analisar cada um deles, para se decidir quais são os mais apropriados para resolver o problema em questão. Realizada tal análise, decidiu-se aqui conforme segue.

#### 1.1.1 Interpolação Numérica

Segundo Franco (2006), a aproximação de funções por polinômios é uma das ideias mais antigas da análise numérica, e ainda uma das mais usadas, pois os polinômios são facilmente computáveis, suas derivadas e integrais são novamente polinômios, suas raízes podem ser encontradas com facilidade. Por isso, é vantajoso substituir uma função complicada por um polinômio que a represente. A simplicidade dos polinômios permite que a aproximação polinomial seja obtida de vários modos como: Interpolação e Mínimos Quadrados.

A interpolação polinomial é utilizada principalmente em duas situações: quando a expressão de  $f(x)$  não é conhecida (apenas se conhecem os valores numéricos de  $f(x)$  para um conjunto discreto de pontos), ou quando se conhece a expressão da função  $f(x)$ , mas ela é muito complicada, pelo que é útil aproximá-la por um modelo matemático mais simples do tipo polinomial. O problema geral da interpolação polinomial consiste em, dados  $n +$

1 pontos distintos  $x_0, x_1, \dots, x_n$  e respectivos valores  $y_0, y_1, \dots, y_n$ , em que  $y_i = f(x_i)$  determinar um polinômio  $P_n(x)$  de grau n tal que  $P_n(y_i) =, i = 0, \dots, n$  (MONTEIRO, 2012).

A fórmula geral da interpolação de Newton é a seguinte (Equação 1), que também é denominada de diferenças divididas:

$$P_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots + f[x_1, x_2, \dots, x_n](x - x_0) \dots (x - x_{n-1}).$$

Equação 1

Diferença dividida de  $f(x)$  é uma função simétrica de seus argumentos, ou seja, independe da ordem dos pontos (FRANCO, 2006). Assim, sejam  $x_0, x_1 \dots x_n, n + 1$  pontos distintos no intervalo  $[a,b]$  e sejam  $f_0, f_1 \dots f_n, n + 1$  valores de uma função  $y = f(x)$  sobre  $x = x_k, k = 0, 1, \dots, n$ , define-se:

$$f[x_k] = f(x_k), k = 0, 1 \dots, n;$$

$$f[x_0, x_1, \dots x_n] = \frac{f[x_1, x_2, \dots, x_n] - f[x_0, x_1, \dots x_{n-1}]}{x_n - x_0},$$

Equação 2

onde  $f[x_0, x_1, \dots x_n]$  é a diferença dividida de ordem  $n$  da função  $f(x)$  sobre os pontos  $x_0, x_1, \dots, x_n$ . Assim, tem-se:

$$f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0};$$

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}.$$

Equação 3

### 1.1.2 Integração Numérica

Integrar numericamente uma função  $y = f(x)$  em um dado intervalo consiste em geral em integrar um polinômio  $P_n(x)$  que aproxime  $f(x)$  no dado intervalo, com a vantagem de que, enquanto  $f(x)$  pode ser difícil ou impossível de integrar, um polinômio é sempre de integração direta (FRANCO, 2006). Assim, integração numérica refere-se a um conjunto de métodos que calculam aproximadamente a integral de uma função com uma variável real definida em um intervalo  $[a, b]$  (ARENALES e DAREZZO, 2014), com a e b finitos (Equação 4):

$$I = \int_a^b f(x)dx.$$

Equação 4

Outras vantagens de usar esses métodos são quando  $f(x)$  é conhecida apenas em pontos discretos obtidos por meio de experimentos, e ainda quando a solução analítica da função integranda é conhecida por meio de uma tabela de valores, mas seu cálculo só pode ser obtido aproximadamente (FRANCO, 2006). Assim, tais métodos baseiam-se na aproximação da função

$f(x)$  por outra função, cuja integral é mais fácil de calcular, como por exemplo polinômios interpoladores de  $f(x)$ . São utilizadas as fórmulas simples e compostas do Trapézio, Simpson (um terço e três oitavos) e respectivos erros de truncatura (MONTEIRO, 2012).

A fórmula para 3/8 de Simpson é a seguinte (Equação 5), na qual se tem  $3n + 1$  pontos, para que o número de intervalos seja múltiplo de 3:

$$\int_{x_0}^{x_{3N}} f(x)dx \approx \frac{3}{8}h[f(x_0) + 3(f(x_1) + f(x_2)) + 2f(x_3) + 3(f(x_4) + f(x_5)) + 2f(x_6) + \dots + 2f(x_{3N-3}) + 3(f(x_{3N-2}) + f(x_{3N-1})) + f(x_{3N})].$$

Equação 5

## 1.2 METODOLOGIA

Os *softwares* utilizados para aproximar os cálculos, gerar os gráficos e realizar os cálculos foram a planilha eletrônica Excel e os programas Scilab e GeoGebra. A imagem foi extraída do Google Maps, e tratada (editada) no Gimp e Inkscape, a fim de então se obterem os dados para aplicação dos métodos numéricos. As figuras de 1 a 3, a seguir, indicam o que foi feito com cada programa manipulador de imagem.

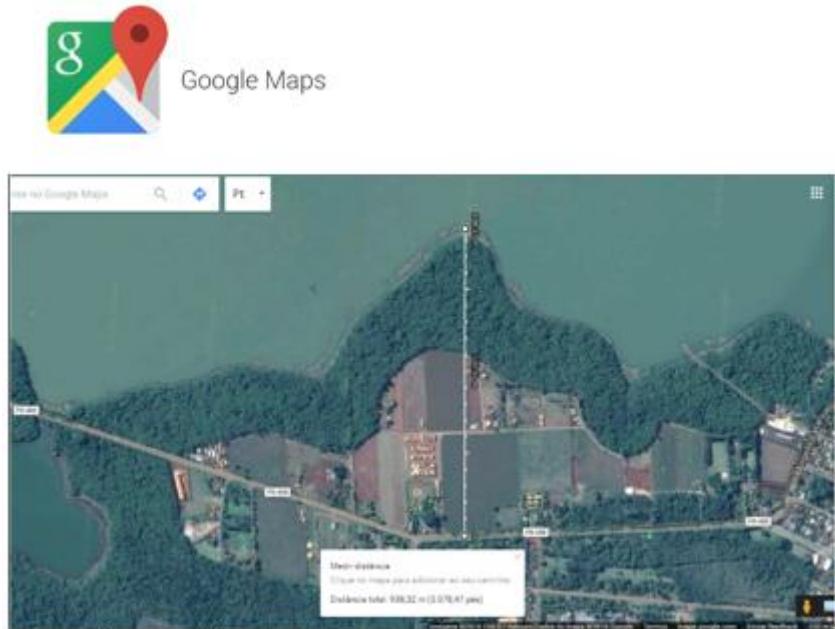


Figura 1: Imagem selecionada para trabalho.



Figura 2: Delimitação da área a ser calculada.

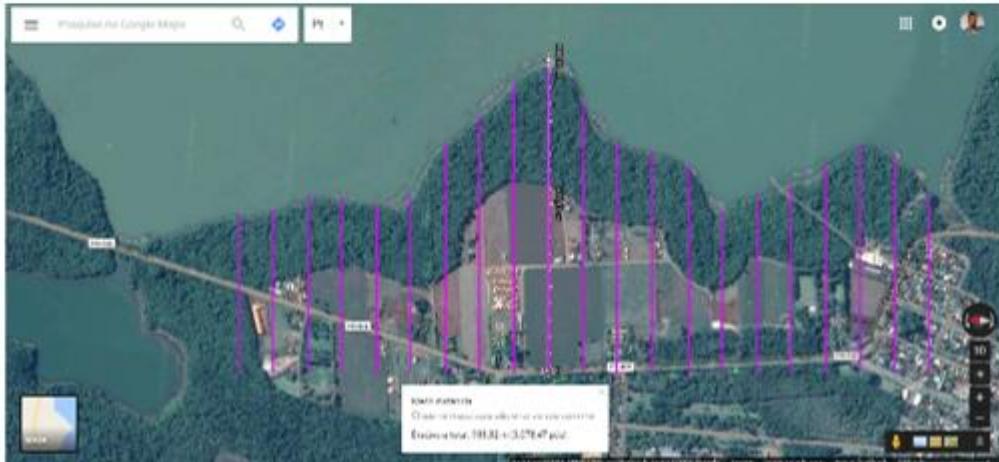


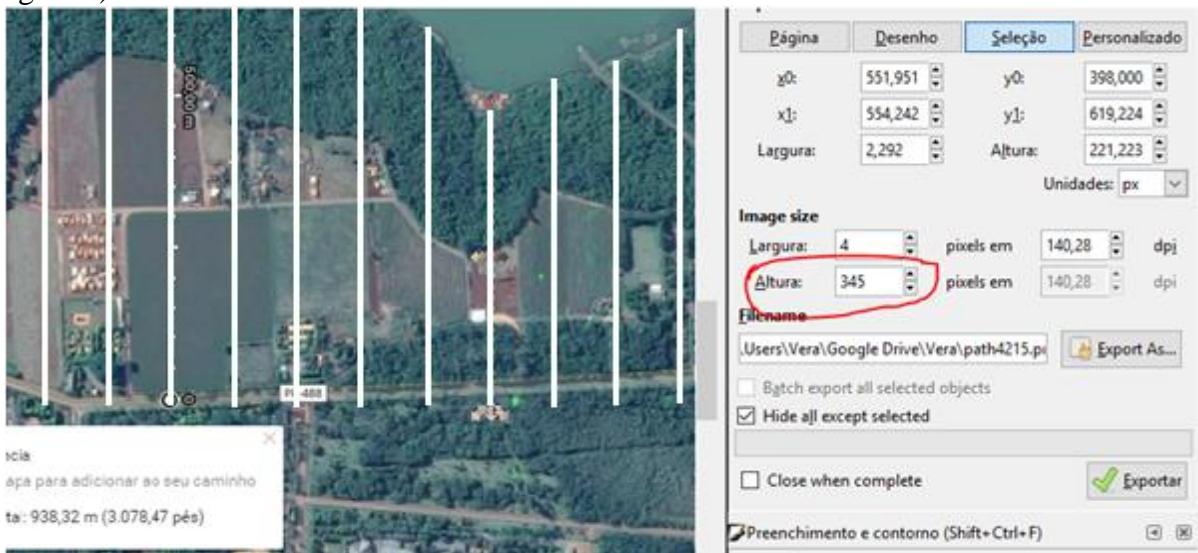
Figura 3: Primeira marcação dos pontos (valores de x com  $h = 100m$ ).

Tomou-se a imagem do Google Maps. Aproveitando a ferramenta de medição do Google, determinou-se a relação entre pixels e metros (Figura 4). Cabe considerar que essa relação depende do *zoom* dado na figura.



Figura 4: Cálculo da relação pixel-metro.

Quando qualquer objeto é selecionado no Inkscape, suas dimensões aparecem em pixels (Figura 5).



$$\frac{345\text{px}}{0,46} = 750\text{m}$$

Figura 5: Dimensões de uma imagem no Inkscape e razão de conversão pixel/m.

A partir disso, marcaram-se as distâncias, inicialmente, de 100 em 100 metros ( $h = 100$ ), contudo, verificou-se que a medida demandava 22 pontos, o que inviabilizava a criação do gráfico a partir das equações geradas pelo software Scilab, utilizando o algoritmo de Newton. Isso ocorria por conta da limitação a 15 casas decimais do software GeoGebra (utilizado para criar as curvas), pois havia coeficientes elevados a -40 nos polinômios interpoladores.

Para contornar o problema, optou-se por aumentar o  $h$  para 200 metros (Figura 6), o que equivale a aproximadamente 92 pixels a partir da escala utilizada. Vale destacar que esse fato pode aumentar o erro no método, pois, à medida que se diminui o espaçamento entre os pontos, o

limitante superior para o erro decresce, indicando valores mais próximos do valor exato da integral (ARENALES e DAREZZO, 2014). No entanto, essa foi a solução possível.

Após esse procedimento, o resultado foi o seguinte, considerando-se os valores (altura, que corresponde ao início – parte inferior – e ao término – parte superior – da área preservada) em metros, da estrada até a margem do lago:

Tabela 1: Definição dos pontos (x,y).

Pontos(x)	Altura(y)
0	456,2783
200	508,9804
400	528,7457
600	535,8196
800	579,7391
1000	779,5739
1200	900,3543
1400	882,7739
1600	573,1522
1800	483,1174
2000	555,5848
2200	237,1652



Figura 6: Fragmento da marcação definitiva dos pontos ( $h = 200\text{m}$ ).

Considerando-se a primeira linha vertical como posicionada em 0 (Figura 6), a partir dela, a cada linha acrescentam-se 200 metros. Então, dividiram-se os valores da Tabela 1 em três conjuntos: o primeiro e o segundo com 5 pontos e o último com 4 pontos. Logo, pelo Scilab foram obtidos três polinômios interpoladores, sendo dois de grau quatro e um de grau três. A Figura 7 mostra um dos polinômios.

The screenshot shows the Scilab 5.5.2 Console window. The command `Newton(X,Y)` was run, resulting in the following output:

```

Y =
306.52174 271.73913 273.91304 313.04348
-->Newton(X,Y)

Tabela de diferenças divididas:

306.521739      -0.172600      0.000457
271.739130      0.010881      0.000455
273.913044      0.193819
313.043478

Polinomio interpolador: ans =

$$306.52174 - 0.2649757x + 0.0004590x^2 - 2.940D-09x^3$$


```

The Variable Browser panel on the right shows variables: ans (1x1 Polyno...), Y ([307, 2...]), and X ([0, 202...]). The Command History panel shows the command and its execution results.

Figura 7: Terceiro polinômio interpolador (grau 3).

Os polinômios gerados para a parte de cima da figura selecionada (área total, delimitada pelo lago e pela estrada) são:

$$f(x) = 456.27826 + 0.3803136x - 0.0007401x^2 + 0.000001x^3 - 0.0000000001351x^4 + 0.000000000001049x^5$$

$$g(x) = 779.57391 + 4.0735856(x - 1027.9729877997465) - 0.0277528(x - 1027.9729877997465)^2 + 0.0000643(x - 1027.9729877997465)^3 - 6.496(10^{-8})(x - 1027.9729877997465)^4 + 0.0000000002413(x - 1027.9729877997465)^5 + 26.32157474310293$$

---


$$h(x) = 483.11739 + 0.05109(x - 1984.89345) + 0.00268(x - 1984.89345)^2 - (0.00001(x - 1984.89345)^3 - 4.64273)$$

Equação 6, 7 e 8

Quanto aos polinômios gerados para a parte de baixo da figura (da estrada até o início da mata preservada), são eles:

$$f(x) = 404.35 + 0.24(x - 6.08) - 0.01(x - 6.08)^2 + 0(x - 6.08)^3 - 3.67(10^{-8})(x - 6.08)^4 + 0(x - 6.08)^5 - 18.25$$

---


$$g(x) = 541.30435 - 0.0986378(x - 1076.539786133119) + 0.0020067(x - 1076.539786133119)^2 - 0.0000095(x - 1076.539786133119)^3 + 0.000000008296(x - 1076.539786133119)^4 + 71.7559925472841$$

---


$$h(x) = 306.52174 - 0.2649757(x - 1808.5040690252827) + 0.000459(x - 1808.5040690252827)^2 - (0.0000000294(x - 1808.5040690252827)^3 - 88.55207189416302)$$

Equação 9, 10 e 11

Esses polinômios foram inseridos no GeoGebra, a fim de obter o gráfico das funções correspondentes. A partir das funções interpoladoras, aplicou-se, com a ajuda do MsExcel, o método de integração numérica de 3/8 de Simpson (Figura 8), que retorna o valor da área abaixo de cada uma das três curvas, limitadas por intervalos. As três áreas foram somadas e obteve-se o valor da área entre a margem do rio e a estrada.

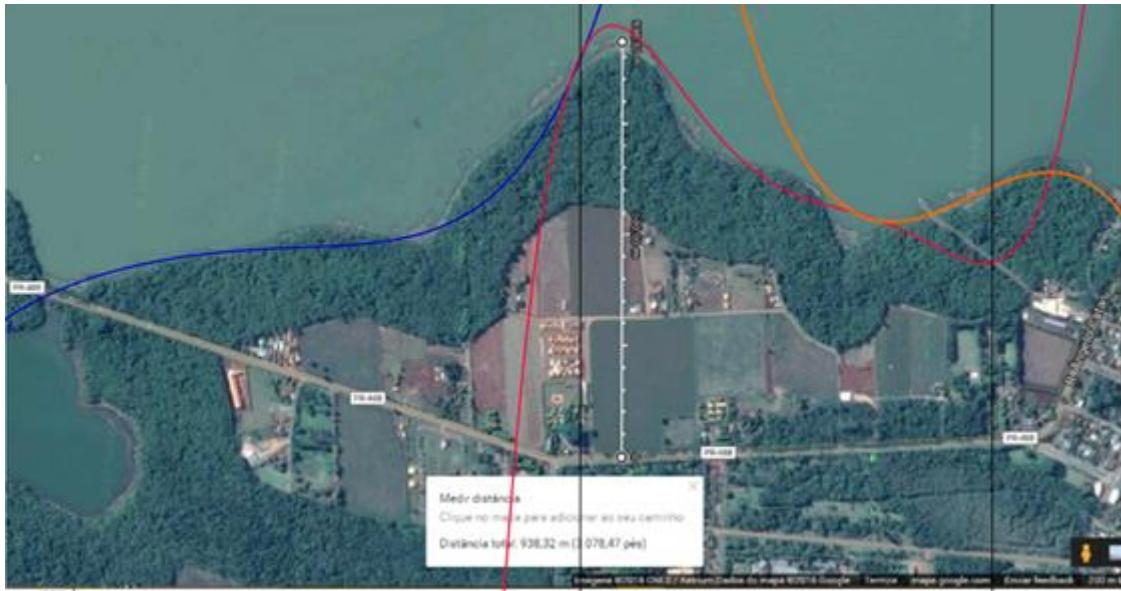


Figura 8: Curvas dos polinômios interpoladores – parte superior.

O mesmo procedimento, com a geração de três curvas pelo GeoGebra, foi repetido para a parte inferior da curva, que vai da estrada até o início da mata (Figura 9). Obteve-se o valor da soma das outras três integrais, correspondente a essa área.

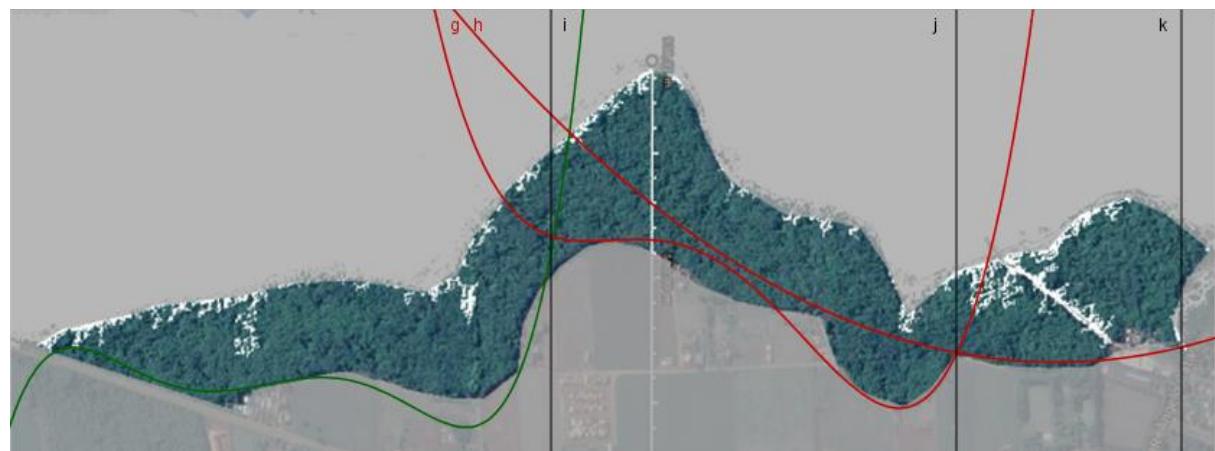


Figura 9: Curvas dos polinômios interpoladores – parte inferior.

A diferença entre o resultado da parte superior e da parte inferior corresponde à área cujo cálculo é o objetivo deste estudo.

Os resultados detalhados da aplicação de 3/8 de Simpson para os três polinômios da parte de cima (total) e os três da parte abaixo da área preservada encontram-se no Anexo 1.

## 2 RESULTADOS E DISCUSSÃO

Os resultados do método de 3/8 de Simpson estão dispostos na Tabela 2:

Tabela 2: Dados de área obtidos.

Áreas	m <sup>2</sup>
Parte de cima (total)	1.410.792,287414
Parte de baixo	907.630,243658
Área preservada acima da rodovia	503.162,04

Assim, foram preservados apenas aproximadamente 503.162m<sup>2</sup> de mata. Nesse caso, em torno de 35% do total. Cabe lembrar que o método revela uma aproximação que pode ser um tanto imprecisa, devido à ferramenta de medição utilizada, ou seja, a régua do Google. Então, calculou-se o erro com base nas medições da própria ferramenta, marcando-se a mesma área:

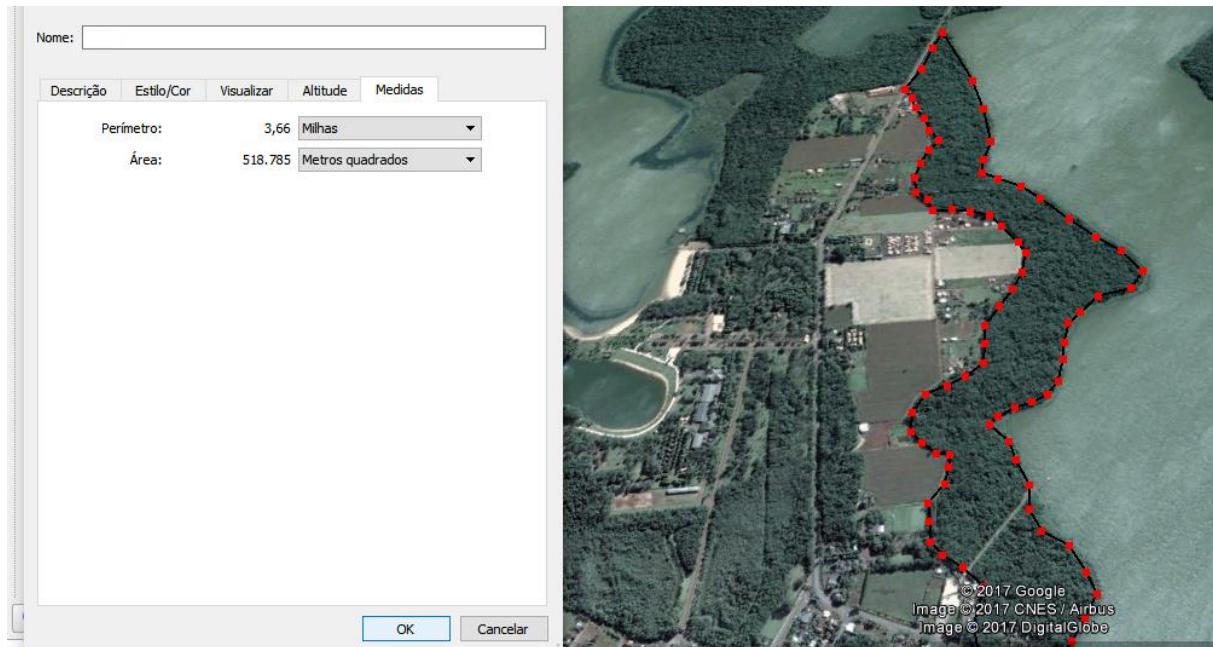


Figura 10: Erro calculado com base na medição de área da ferramenta Google Maps.

A ferramenta indica diferença de 15.123m<sup>2</sup>, o que representa erro de mais ou menos 3,011%. Esse percentual confirma a eficácia do método numérico aplicado, pois, apesar de não se utilizarem ferramentas específicas de geoprocessamento, por se tratar de um exemplo didático, o erro foi bastante baixo, portanto, plenamente aceitável, dadas as condições mencionadas.

## CONSIDERAÇÕES FINAIS

Este trabalho apresentou uma aplicação prática dos métodos numéricos, ou seja, o cálculo aproximado de uma determinada área a partir de uma imagem. Usaram-se *softwares* para tratamento da imagem e para a realização dos cálculos. Os métodos aplicados foram interpolação polinomial (Newton) e integração numérica (3/8 de Simpson), os quais permitiram obter a aproximação desejada, ou seja, uma estimativa de quanto foi preservado em determinada região do município de Santa Helena, Paraná. Os *softwares* utilizados mostraram-se adequados, posto que facilitaram o processo e permitiram chegar ao resultado esperado, por meio de cálculos e manipulação de imagens. Vale ressaltar que o trabalho com imagens torna o processo mais claro, mais visível, então, permite relação direta com a prática.

A partir disso, entende-se que este artigo colabora, por meio de exemplo prático, embora didático, para desmistificar a ideia errônea e ilusória, mas ainda presente nos bancos escolares e mesmo universitários, de que a matemática é uma ciência puramente abstrata e teórica, desconectada da realidade.

## REFERÊNCIAS

- ARENALES, Selma Helena de Vasconcelos; DAREZZO, Artur. **Cálculo numérico:** aprendizagem com apoio de *software*. São Paulo, SP: Thomson Learning, 2008.
- INKSCAPE. 0.92. Gratuito e de código aberto licenciado sob a GPL.<<https://inkscape.org/pt-br/>>. Acesso em: abr. 2017.
- FRANCO, Neide Maria Bertoldi. **Cálculo numérico**. São Paulo, SP: Pearson Prentice Hall, 2006.
- GEOGEBRA. **The graphing calculator for functions, geometry, algebra, calculus, statistics and 3D Math!** Privacy License 2017, International GeoGebra Institute. Disponível em: <<https://www.geogebra.org>>. Acesso em: abr. 2017.
- GNU IMAGE MANIPULATION PROGRAM – GIMP. Creative Commons Attribution-ShareAlike 4.0 International License. Disponível em: <<https://www.gimp.org>>. Acesso em: set. 2017.
- GOOGLE MAPS. Google Brasil. 2017. <<https://www.google.com.br/maps>>. Acesso em: maio. 2017.
- MICROSOFT EXCEL. **Planilha eletrônica**. Microsoft Inc. 2013.
- MONTEIRO, Maria Teresa Torres. **Métodos Numéricos**: exercícios resolvidos aplicados à Engenharia e outras Ciências. Minho: Departamento de Produção e Sistemas, UMinho, 2012. Disponível em: <[http://repository.sdum.uminho.pt/bitstream/1822/14965/6/livro\\_mn.pdf](http://repository.sdum.uminho.pt/bitstream/1822/14965/6/livro_mn.pdf)>. Acesso em: nov. 2016.
- SCILAB. **Software for numerical computation**. 2010-2017. Open source, rollApp, Inc. Disponível em: <<https://www.rollapp.com/app/scilab>>. Acesso em: out. 2016.

**Anexo 1 – Cálculos realizados no MsExcel.**

Polinômios da parte de cima da figura e cálculo de 3/8 de Simpson

0-1065       $f(x) = 456.27826 + 0.3803136x - 0.0007401x^2 + 0.000001x^3 - 0.000000001351x^4 + 0.000000000001049x^5$

	x	f(x)	k	k*f(x)
1	0	456,2783	1	456,2783
2	50,71429	473,784	3	1421,352
3	101,4286	488,1507	3	1464,452
4	152,1429	499,8922	2	999,7844
5	202,8572	509,3921	3	1528,176
6	253,5715	516,9462	3	1550,838
7	304,2857	522,8048	2	1045,61
8	355	527,2149	3	1581,645
9	405,7143	530,4623	3	1591,387
10	456,4286	532,9139	2	1065,828
11	507,1429	535,0602	3	1605,181
12	557,8572	537,5569	3	1612,671
13	608,5715	541,2678	2	1082,536
14	659,2858	547,3066	3	1641,92
15	710,0001	557,0794	3	1671,238
16	760,7144	572,3265	2	1144,653
17	811,4286	595,1652	3	1785,496
18	862,1429	628,1318	3	1884,395
19	912,8572	674,2234	2	1348,447
20	963,5715	736,9409	3	2210,823
21	1014,2857	820,3307	3	2460,992
22	1065	929,027	1	929,027
		Soma:		32082,73
		3/8 de Simpson <sup>1</sup> :		19,01786
		Resultado <sup>2</sup> :		<b>610.144,8</b>

$$^1 \quad (3/8)*((x22-x1)/21) = (3/8)*((1065-0)/21)$$

$$^2 \quad \text{soma} * 3/8 \text{ de Simpson} = 32082,73 * 19,01786$$

1065-1806

$$g(x) = 779.57391 + 4.0735856 (x - 1027.9729877997465) - 0.0277528(x - 1027.9729877997465)^2 + 0.0000643(x - 1027.9729877997465)^3 - 6.496 \cdot 10^{-8} (x - 1027.9729877997465)^4 + 0.0000000002413(x - 1027.9729877997465)^5 + 26.32157474310293$$

	x	f(x)	k	k*f(x)
1	1065	921,8228	1	921,8228143
2	1114,4	988,6628	3	2965,988254
3	1163,8	987,3205	3	2961,961586
4	1213,2	945,6816	2	1891,36313
5	1262,6	884,6889	3	2654,066658
6	1312	819,195	3	2457,584978
7	1361,4	758,8137	2	1517,627467
8	1410,8	708,7722	3	2126,316517
9	1460,2	670,7625	3	2012,28738
10	1509,6	643,7937	2	1287,587406
11	1559	625,0438	3	1875,131503
12	1608,4	610,7115	3	1832,13445
13	1657,8	596,8678	2	1193,735691
14	1707,2	580,3086	3	1740,925656
15	1756,6	559,4055	3	1678,216622
16	1806	534,9589	1	534,9589251
		Soma:		29651,70904
		3/8 de Simpson:		18,525
		Resultado:		<b>549.297,9099</b>

1806-2295 
$$h(x) = 483.11739 + 0.05109(x - 1984.89345) + 0.00268(x - 1984.89345)^2 - (0.00001(x - 1984.89345)^3 - 4.64273)$$

	x	f(x)	k	k*f(x)
1	1806	621,6392	1	621,6392
2	1846,75	558,2092	3	1674,628
3	1887,5	517,4436	3	1552,331
4	1928,25	495,2823	2	990,5647
5	1969	487,6652	3	1462,996
6	2009,75	490,5323	3	1471,597
7	2050,5	499,8234	2	999,6468
8	2091,25	511,4785	3	1534,436
9	2132	521,4375	3	1564,313
10	2172,75	525,6404	2	1051,281
11	2213,5	520,027	3	1560,081
12	2254,25	500,5373	3	1501,612
13	2295	463,1112	1	463,1112
		Soma:		16448,24
		3/8 de Simpson:		15,28125
		Resultado:		<b>251.349,6</b>

## Polinômios da parte de baixo da figura e cálculo de 3/8 de Simpson

0-1032       $f(x) = 404.35 + 0.24(x - 6.08) - 0.01(x - 6.08)^2 + 0(x - 6.08)^3 - 3.67(10^{-8})(x - 6.08)^4 + 0(x - 6.08)^5 - 18.25$

	x	f(x)	k	k*f(x)
1	0	384,4053	1	384,4053
2	49,14286	385,5127	3	1156,538
3	98,28572	367,7153	3	1103,146
4	147,4286	343,304	2	686,608
5	196,5714	320,5365	3	961,6095
6	245,7143	304,2238	3	912,6713
7	294,8572	296,3169	2	592,6337
8	344	296,4937	3	889,481
9	393,1429	302,7457	3	908,2371
10	442,2857	311,9649	2	623,9298
11	491,4286	320,5302	3	961,5906
12	540,5715	324,8946	3	974,6838
13	589,7143	322,1717	2	644,3434
14	638,8572	310,7225	3	932,1674
15	688	290,7422	3	872,2266
16	737,1429	264,847	2	529,6941
17	786,2858	238,6608	3	715,9825
18	835,4286	221,402	3	664,2059
19	884,5715	226,4701	2	452,9401
20	933,7143	272,0327	3	816,0981
21	982,8572	381,6123	3	1144,837
22	1032	584,6726	1	584,6726
		Soma:		17512,7
		3/8 de Simpson:		18,42857
		Resultado:		<b>322.734,1</b>

1032-1888 | 
$$g(x) = 541.30435 - 0.0986378(x - 1076.539786133119) + 0.0020067(x - 1076.539786133119)^2 - 0.0000095(x - 1076.539786133119)^3 + 0.00000008296(x - 1076.539786133119)^4 + 71.75599925472841$$

	x	f(x)	k	k*f(x)
1	1032	622,3066	1	622,3066
2	1089,0667	612,1212	3	1836,363
3	1146,1333	612,9073	3	1838,722
4	1203,2000	615,5913	2	1231,183
5	1260,2667	613,2109	3	1839,633
6	1317,3333	600,9155	3	1802,747
7	1374,4000	575,9662	2	1151,932
8	1431,4667	537,7356	3	1613,207
9	1488,5334	487,7079	3	1463,124
10	1545,6000	429,4788	2	858,9576
11	1602,6667	368,7558	3	1106,267
12	1659,7334	313,3577	3	940,0732
13	1716,8000	273,2153	2	546,4305
14	1773,8667	260,3705	3	781,1116
15	1830,9334	288,9773	3	866,9319
16	1888,0000	375,3009	1	375,3009
		Soma:	18874,29	
		3/8 de Simpson:	21,4	
		Resultado:	403.909,8	

1888-2362     $h(x) = 306.52174 - 0.2649757(x - 1808.5040690252827) + 0.000459(x - 1808.5040690252827)^2 - (0.00000000294(x - 1808.5040690252827)^3 - 88.55207189416302)$

	x	f(x)	k	k*f(x)
1	1888	376,9085427	1	376,9085
2	1940,667	368,0644313	3	1104,193
3	1993,334	361,7602133	3	1085,281
4	2046,001	357,9933119	2	715,9866
5	2098,668	356,7611499	3	1070,283
6	2151,335	358,0611504	3	1074,183
7	2204,002	361,8907364	2	723,7815
8	2256,669	368,2473309	3	1104,742
9	2309,336	377,1283569	3	1131,385
10	2362,003	388,5312374	2	777,0625
		Soma:		9163,807
		3/8 de Simpson:		19,75013
		Resultado:		<b>180.986,3</b>

# Otimização de rotas e redução do custo por meio do algoritmo de Dijkstra

## Route optimization and cost reduction through Dijkstra algorithm

Leonardo Nogueira da Silva<sup>1</sup>, Tatiane Tambarussi<sup>2</sup>

Ciência da Computação Universidade Tecnológica Federal do Paraná (UTFPR-SH)

<sup>1</sup>[leosil.2015@alunos.utfpr.edu.br](mailto:leosil.2015@alunos.utfpr.edu.br), <sup>2</sup>[tambarussi@utfpr.edu.br](mailto:tambarussi@utfpr.edu.br)

### RESUMO

Este artigo apresenta uma solução para uma rota de transporte rodoviário, a fim de otimizá-la e reduzir custos de transporte de uma importadora e distribuidora de farinha de trigo do estado do Paraná. O algoritmo de Dijkstra, que calcula a melhor rota, tendo como parâmetro seus custos, foi implementado. Então, gera-se um grafo que contém em seus vértices o destino e a origem, e em suas arestas cada rota com seus respectivos custos. O algoritmo recebeu como parâmetro uma rota de entrega específica da empresa, com origem em Foz do Iguaçu, Paraná e destinos em Marília, São Paulo e Avaré, São Paulo. Os dados das rotas desse percurso foram coletados por meio do aplicativo Google Maps, e o custo da rota foi calculado a partir do consumo médio do veículo. Custos de pedágios foram coletados a partir do site Mapeia. O algoritmo gerou o grafo final, obtendo a melhor rota.

**Palavras-chave:** Logística; melhor rota; Pesquisa Operacional; Algoritmo de Dijkstra.

### ABSTRACT

*This article presents a solution for a road transport route, for optimizing it and reducing transport costs of a wheat flour importer and distributor, located in the state of Paraná, Brazil. The Dijkstra algorithm was employed, since it calculates the best route, based on its costs. Then, a graph that contains in its vertices the destination and the origin, and in its edges each route with its respective costs was generated. The algorithm received as a parameter a company specific delivery route, from Foz do Iguaçú, Paraná to Marilia, São Paulo, and Avare, São Paulo. Route data for this route was collected by using GoogleMaps app, and its cost were calculated from the vehicle consumption average. Toll costs were collected from the Mapeia website. The algorithm generated the final graph, obtaining the best route.*

**Keywords:** Transport routes; Dijkstra algorithm; best route; Operational Research.

### Referência:

SILVA, L. N.; TAMBARUSSI, T. Otimização de rotas e redução do custo por meio do algoritmo de Dijkstra. In: SEMANA TECNOLÓGICA ACADÊMICA DE CIÊNCIA DA COMPUTAÇÃO DA UTFPR-SH, 4., 2017, Santa Helena, PR. *Anais...*, Santa Helena: UTFPR, 2017. p.36-47. Disponível em: <<http://sh.utfpr.edu.br/setac/Anais2017.pdf>>.

### INTRODUÇÃO

Atualmente no Brasil, o custo de transporte é um problema tanto para indústria quanto para o consumidor. Assim, conduzir seus produtos de forma estrategicamente eficiente, diminuindo o custo a cada entrega, é o que toda empresa busca para conseguir se manter no mercado.

A maior parte do transporte no Brasil dá-se por meio rodoviário, mas existe grande desgaste nas estradas. A malha rodoviária brasileira possui várias rotas alternativas, e todas elas possuem custo operacional (combustível, pedágio, distância). Assim escolher a melhor rota é um desafio para o setor logístico, que é encarregado dessa tarefa.

A partir disso, este trabalho apresenta uma proposta para otimizar os custos e o tempo desse processo de transporte mediante conceitos computacionais, utilizando-se o algoritmo de Dijkstra, para escolher a melhor rota, a fim de otimizar o caminho de entrega de uma importadora de farinha de trigo, e assim minimizar seu custo.

## 1 REFERENCIAL TEÓRICO-METODOLÓGICO

### 1.1 LOGÍSTICA

A logística sempre procura alto desempenho com planejamento e um gerenciamento eficiente. Segundo Carvalho (2002, p.31), logística é:

A parte do gerenciamento da cadeia de abastecimento que planeja, implementa e controla o fluxo e armazenamento eficiente e econômico de matérias-primas, materiais semi-acabados e produtos acabados, bem como as informações a eles relativas, desde o ponto de origem até o ponto de consumo, com o propósito de atender às exigências dos clientes.

Observa que tal conceito diz que o objetivo das organizações é gerar maior satisfação para o cliente, porém, com o modelo logístico mal estruturado que o Brasil possui, onde se tem estradas desgastadas e congestionadas, isso se torna um desafio.

Em 1950, houve a escolha do modelo rodoviário, mas isso se deu sem planejamento, tanto de infraestrutura como de logística de longo prazo, em que a expansão do modelo acompanhou o desenvolvimento urbano, o que trouxe problemas para o setor do transporte. Nesse sentido, Barat (2008, p.64) destaca que:

A deficiência das ferrovias e portos, além da malha rodoviária ainda pouco extensa para as dimensões do país, tornaram o sistema de transportes um dos principais estrangulamentos, tanto às complexas logísticas de abastecimento e escoamento, que surgiram com as novas cadeias produtivas, quanto ao próprio desenvolvimento econômico.

Hoje no Brasil, o transporte tornou-se o maior custo para o setor logístico. Além desse custo de transporte, também há taxas, impostos e as exigências da legislação ambiental, o que faz a redução de custos ainda mais difícil e importante para o setor logístico.

### 1.2 ALGORITMO DE DIJKSTRA

O algoritmo de Dijkstra, criado pelo holandês Edsger Dijkstra<sup>1</sup> em 1956 e publicado em 1959, é um dos algoritmos que calculam o caminho de custo mínimo entre vértices de um grafo. Escolhida a aresta A como raiz da busca (cidade origem), esse algoritmo calcula a distância mínima desse vértice até todos os demais vértices do grafo, ou seja, as demais cidades, no caso. Ele funciona da forma que segue (GOLDBARG e GOLDBARG, 2012).

Seja  $G(S, A)$  um grafo orientado em que  $s$  é um vértice e  $a$  é uma aresta de  $G$ :

1. Atribua valor zero à estimativa do custo mínimo do vértice  $s$  (cidade de origem) e infinito às demais estimativas;
2. Atribua um valor qualquer aos precedentes (o precedente de um vértice  $t$  é o destino no caminho de custo mínimo de  $s$  para  $t$ );

---

<sup>1</sup> Edsger Wybe Dijkstra (1930-2002) foi um cientista da computação holandês conhecido por suas contribuições nas áreas de desenvolvimento de algoritmos e programas, de linguagens de programação (que lhe conferiu o Prêmio Turing em 1972, por suas contribuições fundamentais), sistemas operacionais e processamento distribuído (WIKIPÉDIA, 2017).

3. Enquanto houver vértice aberto:

3.1 Seja **k** um vértice ainda aberto cuja estimativa seja a menor dentre todos os vértices abertos:

- Feche o vértice **k**.

3.2 Para todo vértice **j** ainda aberto que seja sucessor de **k**, faça:

- Some a estimativa do vértice **k** com o custo do arco (custo da rota) que une **k** a **j**;
- Caso essa soma seja melhor que a estimativa anterior para o vértice **j**, substitua-a e anote **k** como precedente de **j**.

### 1.3 AS ROTAS DE TRANSPORTE

A importadora e distribuidora que faz parte deste estudo importa farinha da Argentina e a distribui no Brasil. Para isso possui um caminhão que realiza as entregas de trigo para seus clientes, que se localizam no interior de São Paulo e no sul de Minas Gerais. Toda entrega tem como partida a cidade de Foz do Iguaçu, PR, onde a importadora fica localizada. Seu destino é variado, conforme a venda de farinha. Após realizar as entregas, o caminhão deve seguir com destino a Avaré-SP, onde carrega piso, e volta para Foz do Iguaçu-PR.

A empresa disponibilizou informações do caminhão, contidas no Quadro1:

Quadro 1: Informações do caminhão da empresa.

<b>Marca</b>	<b>Volvo</b>
Modelo	FH 12
Potência	520cv
Ano	2010
Reboque	Carreta simples, carga seca <sup>2</sup>
Número total de eixos	6
Peso bruto	48t
Tara	15t
Lotação	33t

Para aplicação do algoritmo, foi feita a simulação de entrega no percurso e sua origem, até seu destino. Foram localizadas três rotas. A Tabela 1 detalha a Rota 1:

Tabela 1: Informações da Rota 1.

Partida	Foz do Iguaçu
Destino	Marília
Via	
Rota 1	Via BR 369
Quilometragem a ser percorrida	696 km
Custos	
Número total de pedágios	8
Média do custo por eixo	R\$ 12,18
Total de pedágios X custo por eixo	R\$ 97,4
Quantidade de eixos	6

<sup>2</sup> Carga seca refere-se aos produtos não perecíveis, que podem ser transportados independentemente do horário e das condições climáticas, sem que sejam causados danos à carga.

A imagem a seguir (Figura 1) traz o mapeamento da Rota 1, feito no aplicativo Google Maps.

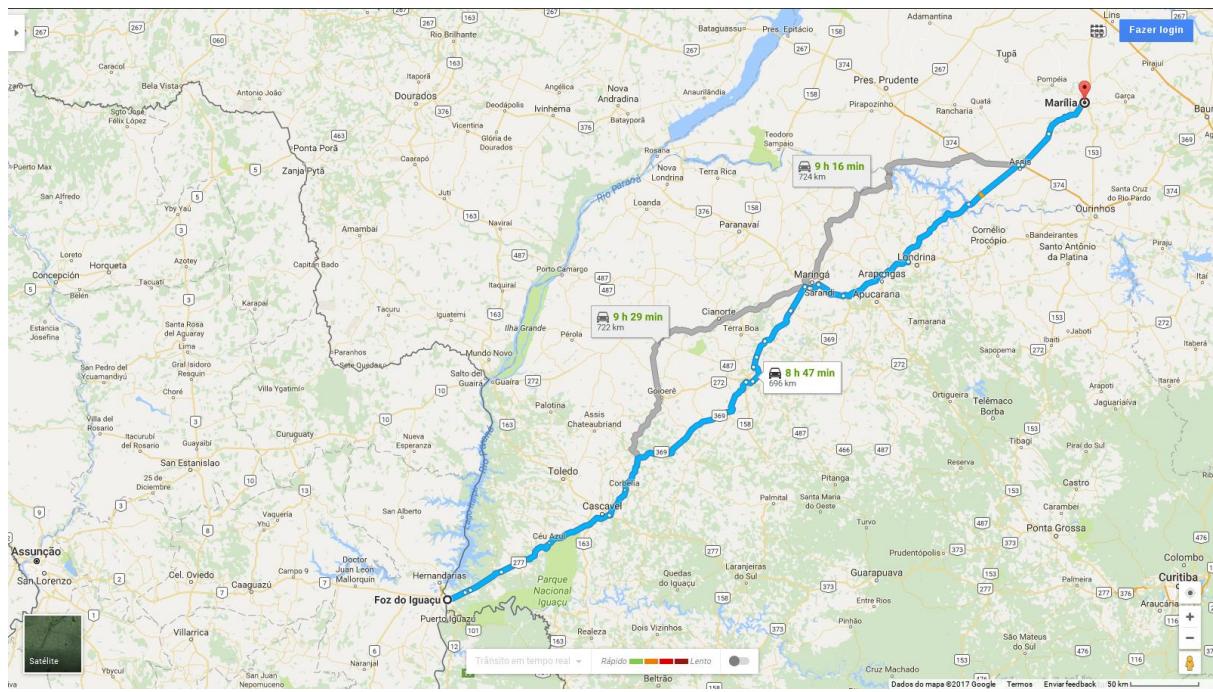


Figura 1: Mapeamento da Rota 1.

**Fonte:** Google Maps (2017).

A Tabela 2 detalha a Rota 2:

Tabela 2: Informações sobre a Rota 2.

Via	
Rota 2	Rod. João Jorge Saad
Quilometragem a ser percorrida	763 km
Custos	
Número total de pedágios	1
Média do custo por eixo	R\$14,80
Total de pedágios X custo por eixo	R\$14,80
Quantidade de eixos	6
Valor médio do dísel nos estados de São Paulo, Paraná e Minas Gerais	R\$2,88

A imagem a seguir (Figura 2) traz o mapeamento da Rota 2, feito no aplicativo Google Maps.

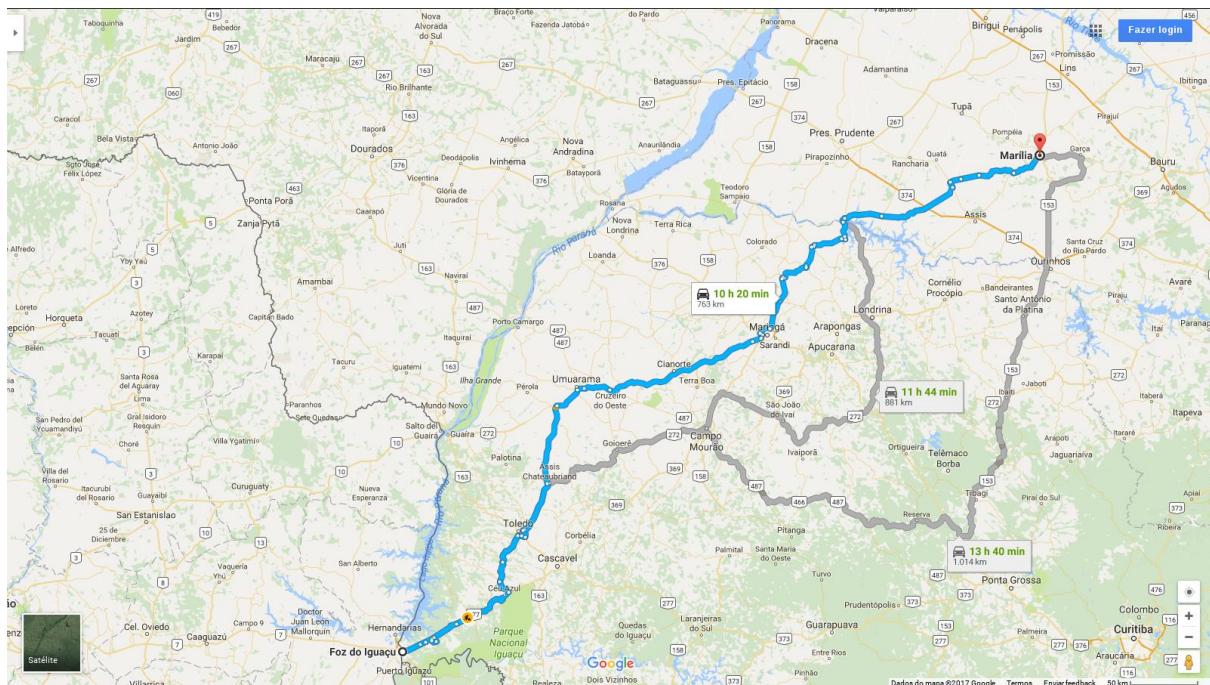


Figura 2: Mapeamento da Rota 2.

**Fonte:** Google Maps (2017).

Por último, eis as informações da Rota 3:

Tabela 3: Informações sobre a Rota 3.

Tabela 3: Informações sobre a Rota 3.	
Via	
Rota 3	BR 277
Quilometragem a ser percorrida	725 km
Custos	
Número total de pedágios	6
Média do custo por eixo	R\$11,44
Total de pedágios X custo por eixo	R\$68,67
Quantidade de eixos	6
Valor médio do dísel nos estados de São Paulo, Paraná e Minas Gerais	R\$2,88

Os valores para o dísel foram obtidos da Agência Nacional do Petróleo (ANP, 2017a) e (ANP, 2017b).

A imagem a seguir traz o mapeamento da Rota 3 feito no aplicativo Google Maps.

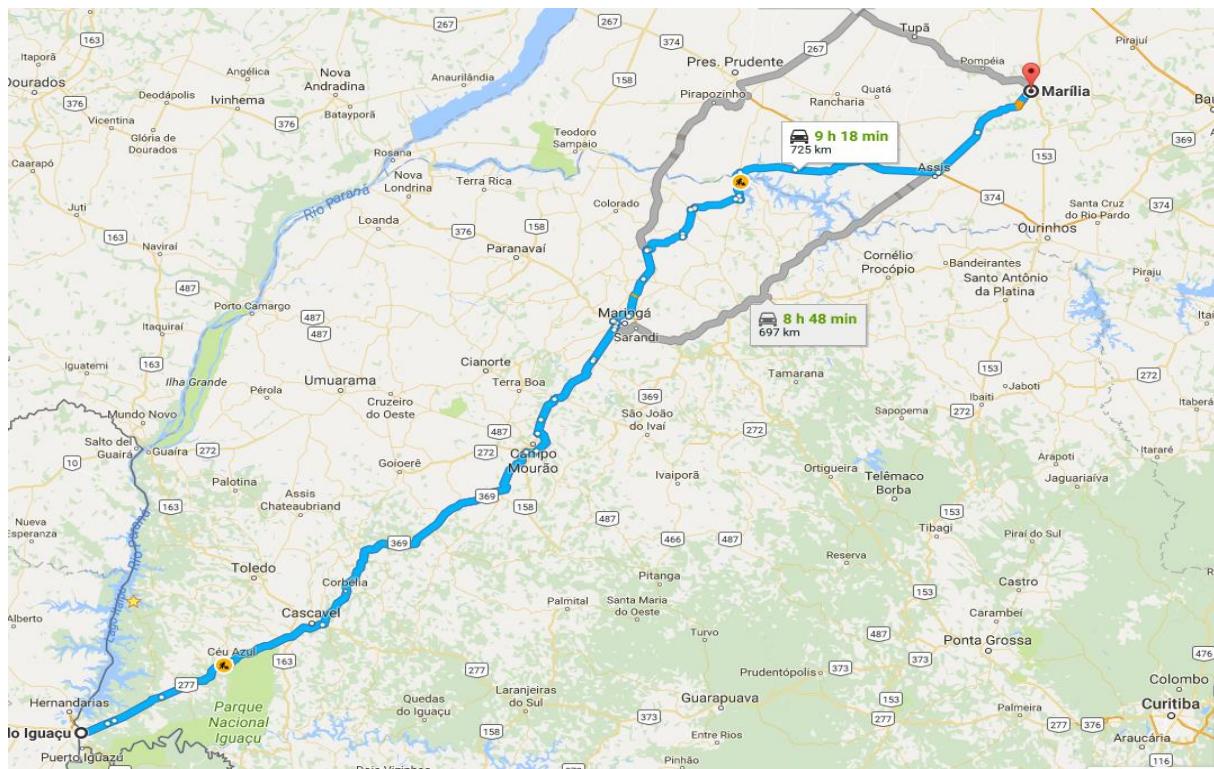


Figura 3: Mapeamento da Rota 3.

Fonte: Google Maps (2017).

Após a entrega no primeiro destino, que é a cidade de Marília no Estado de São Paulo, o caminhão deve partir para Avaré, que também se localiza no estado de São Paulo, para carregar piso para voltar para Foz do Iguaçu, no estado do Paraná, como mencionado. Nesse percurso entre Marília e Avaré, foi localizada somente uma rota, que possui custo total de R\$ 271,67. A Figura 4 mostra o mapeamento da rota.

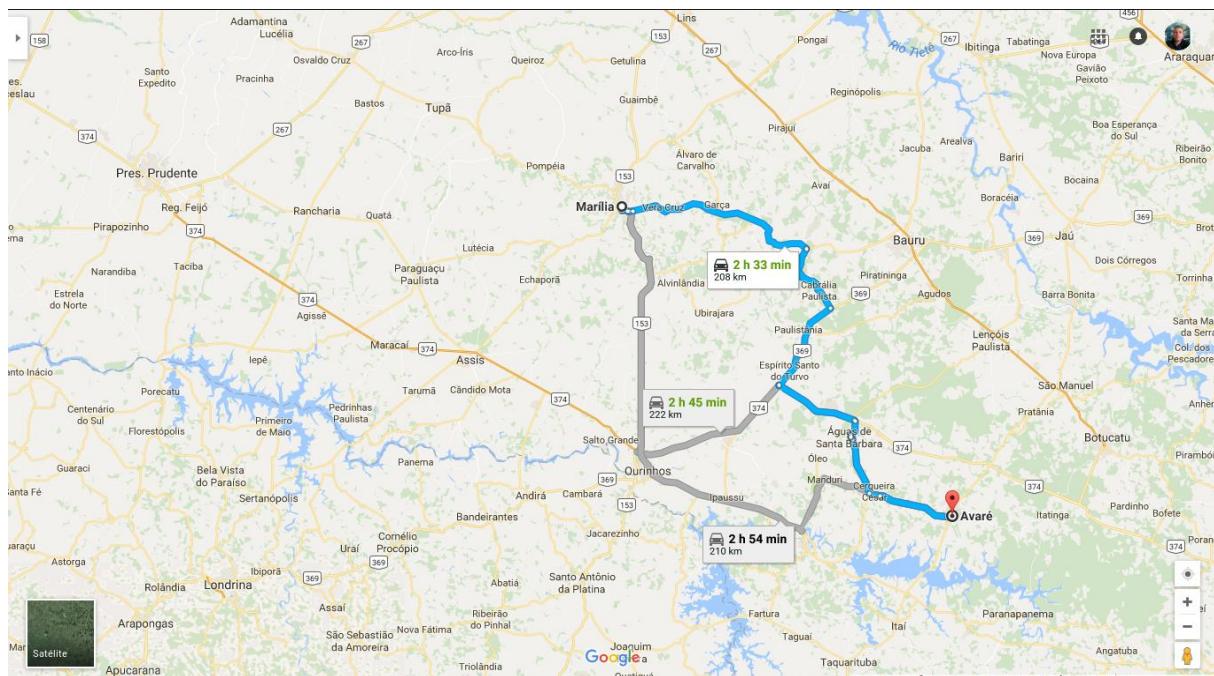


Figura 4: Mapeamento da Rota 4.

Fonte: Google Maps (2017).

Para seu retorno para Foz do Iguaçu, existem duas possíveis rotas, a rota de volta 1 (Tabela 4) e a rota de volta 2 (Tabela 5).

Tabela 4: Informações da Rota 5.

Partida	Avaré
Destino	Foz do Iguaçu
<b>Via</b>	
Rota 1	Via BR 369
Quilometragem a ser percorrida	824 km
<b>Custos</b>	
Número total de pedágios	10
Média do custo por eixo	R\$ 11,60
Total de pedágios X custo por eixo	R\$ 69,60
Quantidade de eixos	6
Valor médio do dísel nos estados de São Paulo, Paraná e Minas Gerais	R\$ 2,88

A imagem a seguir (Figura 5) traz o mapeamento da Rota 5, feito no aplicativo Google Maps.

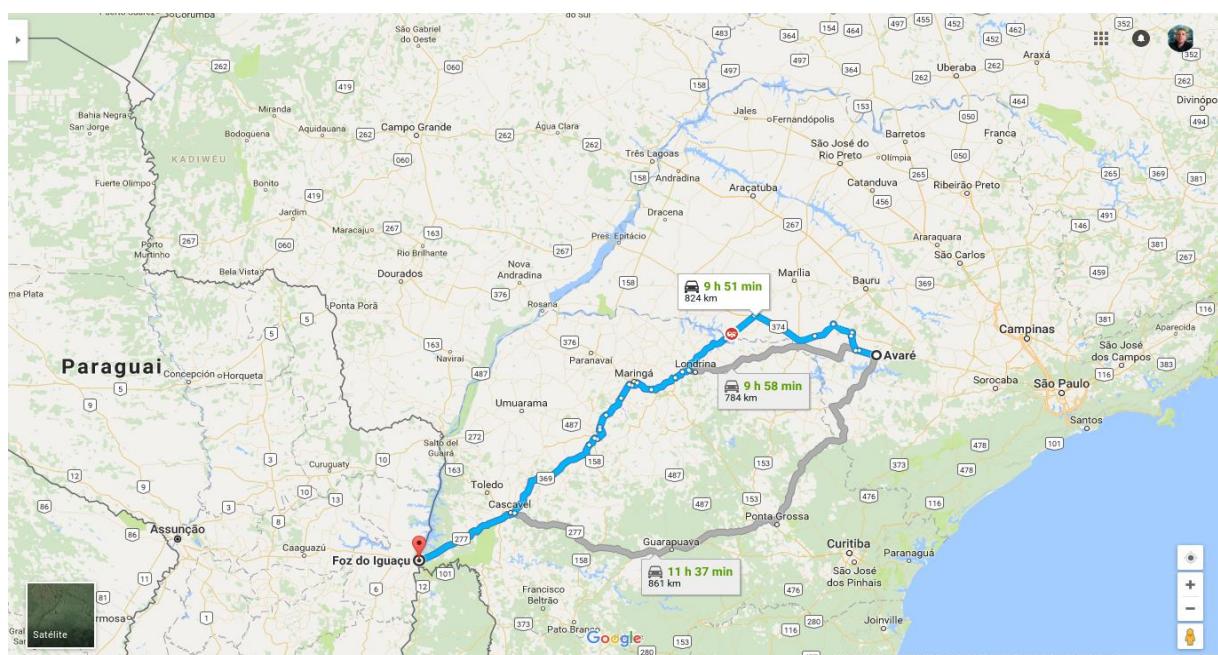


Figura 5: Mapeamento Rota 5.

Fonte: Google Maps (2017).

A Tabela 5, a seguir, apresenta os custos da rota de volta 2.

Tabela 5: Informações da Rota 6.

Via	
Rota 2	Rod. João Jorge Saad
Quilometragem a ser percorrida	903 km
Custos	
Número total de pedágios	3
Média do custo por eixo	R\$ 13,33
Total de pedágios X custo por eixo	R\$ 40,00
Quantidade de eixos	6
Valor médio do dísel nos estados de São Paulo, Paraná e Minas Gerais	R\$ 2,88

A imagem a seguir (Figura 6) traz o mapeamento da Rota 6, feito no aplicativo Google Maps.

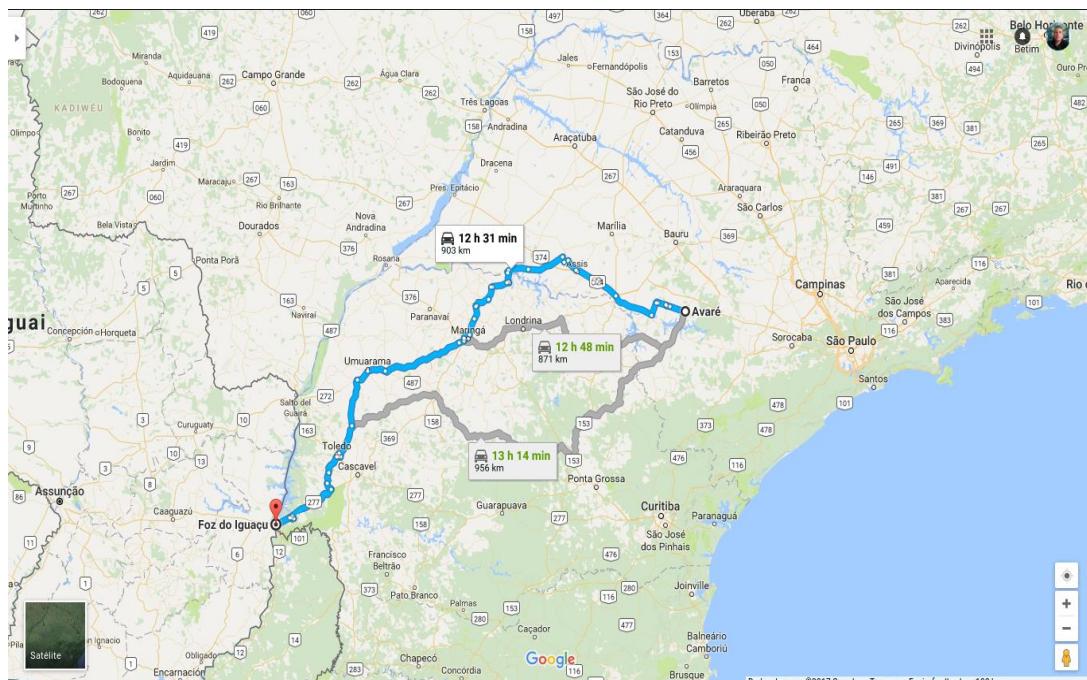


Figura 6: Mapeamento da Rota 6.

Fonte: Google Maps (2017).

#### 1.4 PREPARAÇÃO DO ALGORITMO DE DIJKSTRA

Para implementar o algoritmo, foi construído um grafo. O grafo da Figura 7 possui três vértices. O vértice Foz do Iguaçu, de índice 0, aloca a origem da viagem, nas outras duas, aloca-se o destino: Marília, para o vértice de índice 1, e Avaré para o vértice de índice 2. As rotas são alocadas nas arestas. A Figura 7 contém a representação do grafo.

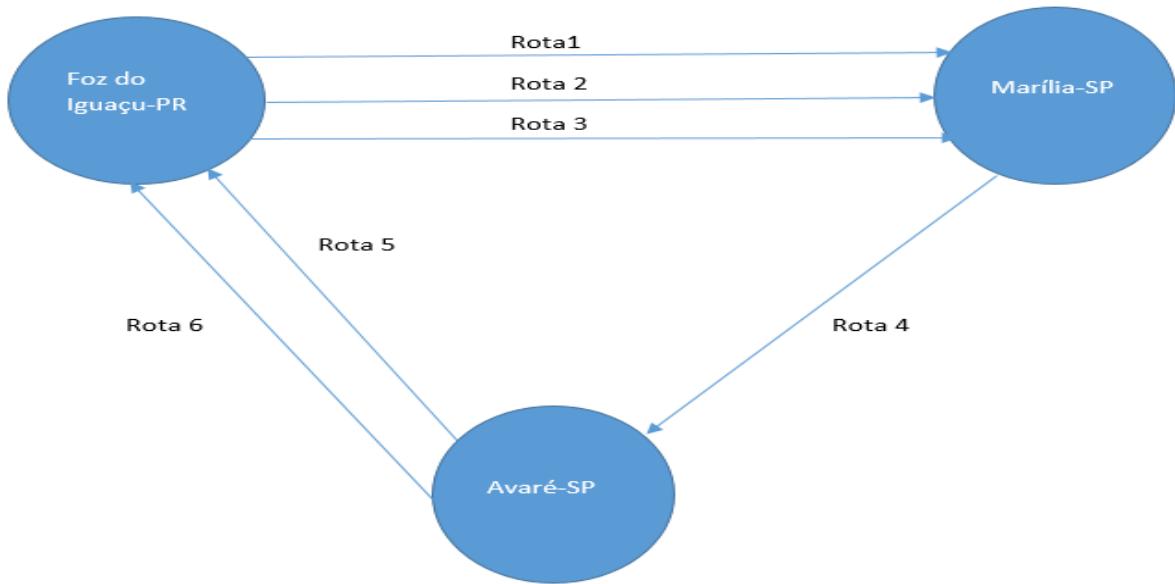


Figura 7: Representação do grafo.

Para realizar a implementação do algoritmo, deve-se calcular o custo final de cada rota. Os custos finais são obtidos com as seguintes fórmulas:

$$\begin{aligned}
 & \textbf{TOTAL DE PEDÁGIOS} \\
 & = \\
 & (\text{NÚMERO TOTAL DE PEDÁGIO} \times \text{CUSTO POR EIXO} \times \text{QTD EIXOS})
 \end{aligned}$$

e

$$\begin{aligned}
 & \textbf{TOTAL COMBUSTÍVEL} \\
 & = \\
 & (\text{KM A SER PERCORRIDO} \div \text{CONSUMO VEÍCULO}) \times (\text{VALOR DÍSEL})
 \end{aligned}$$

Assim,

$$\textbf{CUSTO FINAL} = \textbf{TOTAL DE PEDÁGIOS} + \textbf{TOTAL COMBUSTÍVEL}$$

Aplicando as fórmulas, os seguintes custos foram obtidos:

Tabela 6: Custos das rotas.

Rota	Custo (R\$)
Rota 1	1495,50
Rota 2	1087,63
Rota 3	1359,81
Rota 4	271,67
Rota 5	1496,29
Rota 6	1422,10

Assim fica o grafo após o cálculo das rotas:

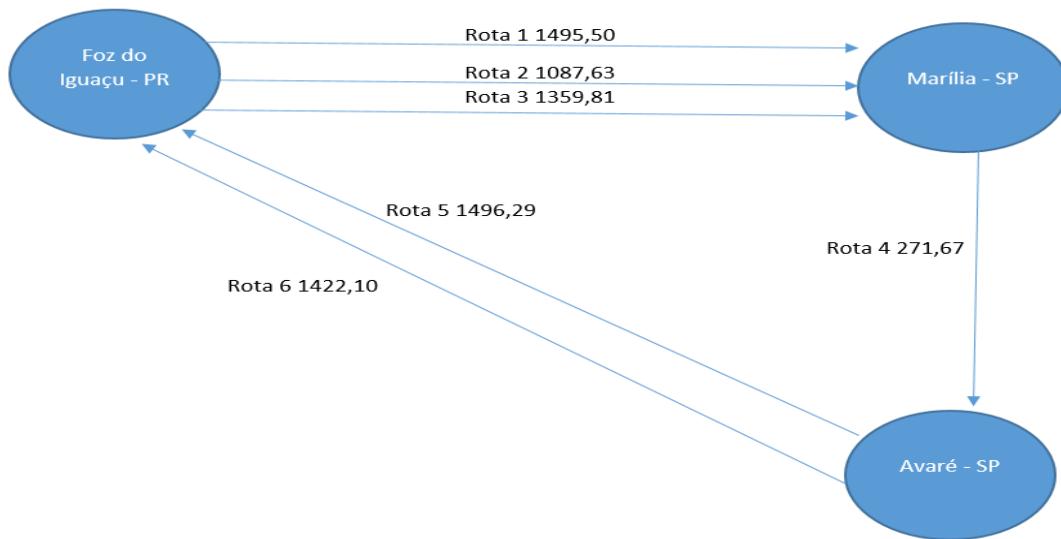


Figura 8: Grafo com as rotas calculadas.

## 2 RESULTADOS E DISCUSSÃO

Primeiramente, deve-se atribuir valor zero à estimativa do custo mínimo para a cidade de Foz do Iguaçu - PR e infinito às demais:

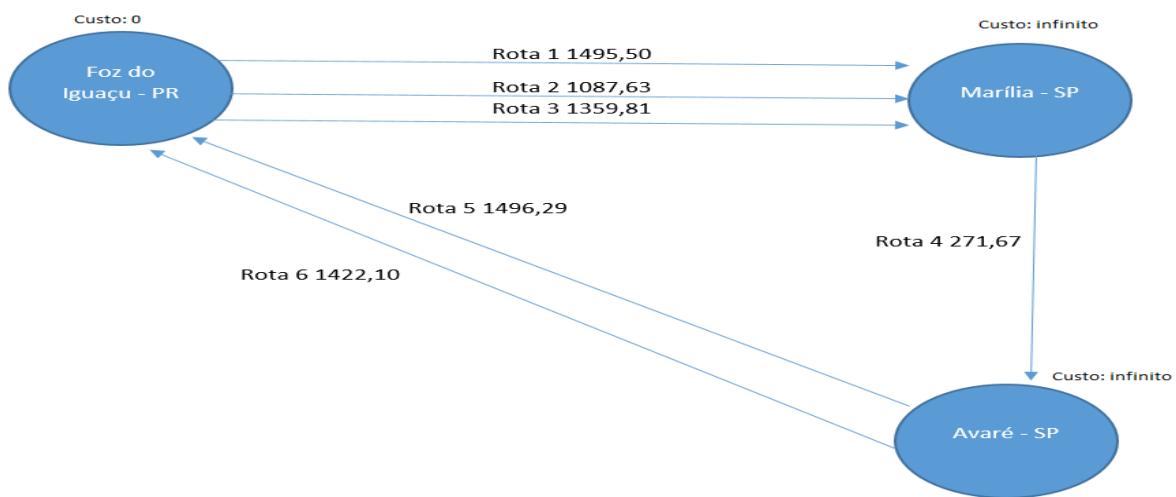


Figura 9: Grafo com custos inicializados.

Em seguida, enquanto houver vértice aberto:

- Seja **k** um vértice ainda aberto, cuja estimativa seja a menor dentre todos os vértices abertos:
  - **k** = custo de Foz do Iguaçu, que é igual a zero;
- Para todo vértice **j** ainda aberto que seja sucessor de **k**, faça:
  - Some a estimativa do vértice **k** com o custo do arco (custo da rota) que une **k** a **j**;
  - Caso essa soma seja melhor que a estimativa anterior para o vértice **j**, substitua-a e anote **k** como precedente de **j**;

A menor soma entre **k** e **j** é a Rota 2, com o valor R\$ 1087,63.

Tabela 7: Soma entre j e k.

J (i)	Vértice	K	Soma (j + k)	Resultado
J (0)	Rota 1 1495,50	0	1495,50 + 0	1495,50
J (1)	Rota 2 1087,63	0	1087,63 + 0	1087,63
J (2)	Rota 3 1359,81	0	1359,81 + 0	1359,81

- **k** = custo de Marília, que é igual à menor soma entre **k** e **j** anteriores, R\$1087,63;

Entre Marília e Avaré, só existe uma rota que deve ser percorrida, então **k** recebe o custo anterior de Marília, somado ao custo da Rota 4, que é R\$ 271,67. Continuando com o algoritmo:

- **k** = 1359,23;
- Para todo vértice **j** ainda aberto, que seja sucessor de **k**, faça:
  - Some a estimativa do vértice **k** com o custo do arco (custo da rota) que une **k** a **j**;
  - Caso essa soma seja melhor do que a estimativa anterior para o vértice **j**, substitua-a e anote **k** como precedente de **j**.

A menor soma entre **k** e **j** é a Rota 6, com o valor R\$2781,33.

Tabela 8: Soma de j e k.

J(i)	Vértice	K	Soma (j + k)	Resultado
J(0)	Rota 5 1496,29	1359,23	1495,50 + 1359,23	2855,52
J(1)	Rota 6 1422,10	1359,23	1087,63 + 1359,23	2781,33

Por meio da aplicação do algoritmo, foi obtido um grafo (Figura 11) com a melhor rota para essa viagem, com os custos reduzidos. No entanto, essa solução considera que as estradas de todas as rotas têm condição igual, ou seja, a melhor rota ainda não leva em conta a condição física de estrada, e sim seus custos base (pedágio, combustível).

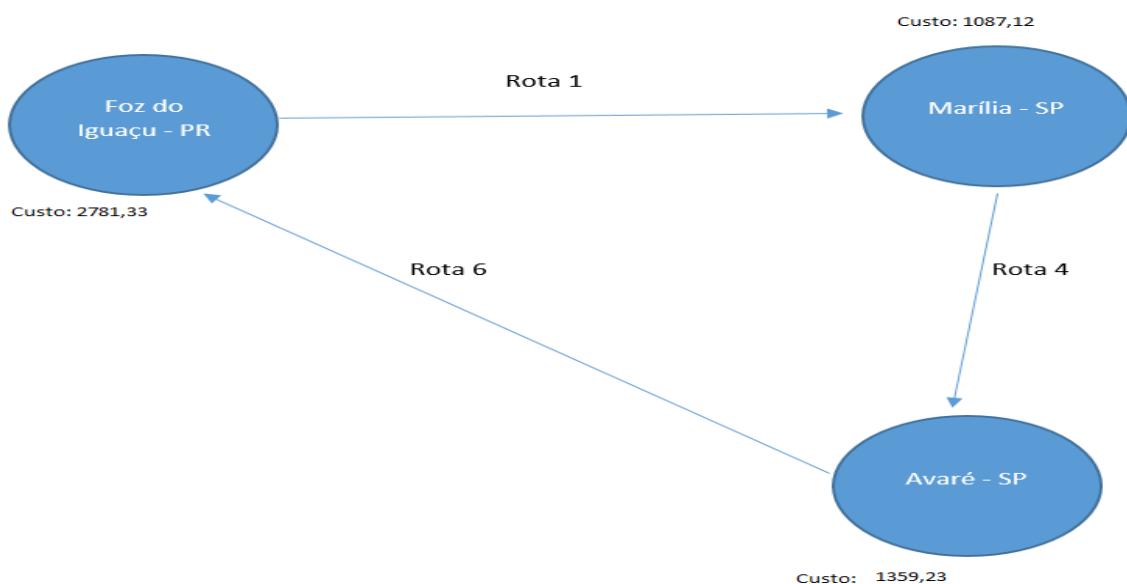


Figura 11: Grafo final.

Percebe-se que o algoritmo de Dijkstra otimizou a rota de transporte e reduziu os custos da viagem da importadora de trigo. A partir do algoritmo, obteve-se o melhor custo dessa viagem, que é de R\$ 2781,33, utilizando a Rota 2, Rota 4 e Rota 6.

## CONSIDERAÇÕES FINAIS

O objetivo do trabalho foi alcançado, pois o algoritmo de Dijkstra forneceu uma rota de transporte otimizada e reduziu os custos da viagem de uma importadora de trigo. Para trabalhos futuros, sugere-se construir um aplicativo móvel que utilize tal algoritmo para realizar o mapeamento e a escolha da melhor rota em função do custo, uma vez que os aplicativos atuais buscam a melhor rota apenas em função da quilometragem a ser percorrida, sem levar em conta o custo da viagem.

## REFERÊNCIAS

- AGÊNCIA NACIONAL DO PETRÓLEO – ANP. Diesel – **Resumo por estado**. 2017a. Disponível em: <[http://www.anp.gov.br/preco/prc/Resumo\\_Por\\_Estado\\_Index.asp](http://www.anp.gov.br/preco/prc/Resumo_Por_Estado_Index.asp)>. Acesso em: 20 jun. 2017.
- AGÊNCIA NACIONAL DO PETRÓLEO – ANP. Diesel – **Resumo por município**. 2017b. Disponível em: <[http://www.anp.gov.br/preco/prc/Resumo\\_Por\\_Municipio\\_Index.asp](http://www.anp.gov.br/preco/prc/Resumo_Por_Municipio_Index.asp)>. Acesso em: 20 jun. 2017.
- BARAT, Josef. **Logística, transporte e desenvolvimento econômico**: A visão Histórica. v.1. São Paulo: CLA Cultural, 2008.
- CARVALHO, José Meixa Crespo. **Logística**. 3.ed. Lisboa: Edições Silabo, 2002.
- GOLDBARG, Marco Cesar; GOLDBARG, Elizabeth Ferreira Gouvêa. **Grafos**: conceitos, algoritmos e aplicações. Rio de Janeiro, RJ: Elsevier, 2012.
- GOOGLE MAPS. **Rotas**. Disponível em: <<http://www.google.com.br/maps>>. Acesso em: 15 jun. 2017.
- MAPEIA. **Custos dos pedágios**. Disponível em <<http://www.mapeia.com.br>>. Acesso em: 15 de jun. 2017.
- WIKIPÉDIA. **Edsger Wybe Dijkstra**. Disponível em: <[https://pt.wikipedia.org/wiki/Edsger\\_Dijkstra](https://pt.wikipedia.org/wiki/Edsger_Dijkstra)>. Acesso em: set. 2017.

# Uma avaliação empírica dos efeitos de ataques de negação de serviço sobre redes definidas por *software*

## An empirical evaluation of the effects of denial of service attacks on software defined networks

Márcio José Araújo<sup>1</sup>, Luca Prediger<sup>2</sup>, Euclides Peres Farias Junior<sup>3</sup>, Michele Nogueira Lima<sup>4</sup>

Ciência da Computação, Universidade Tecnológica Federal do Paraná (UTFPR-SH)

*marcioaraújo@alunos.utfpr.edu.br*<sup>1</sup>, *luca@alunos.utfpr.edu.br*<sup>2</sup>,

*euclidesjunior@utfpr.edu.br*<sup>3</sup>

Universidade Federal do Paraná (UFPR)

*michele.nogueira@ufpr.br*<sup>4</sup>

### RESUMO

O objetivo deste artigo é analisar o comportamento da SDN em ambientes simulados diante de ataques distribuídos de negação de serviço (DDoS), por meio de métodos estatísticos, em especial a análise de sobrevivência. A abordagem principal aplicada à análise dá-se em função do tempo de vida da rede sob ataque, com coleta de características, como limitações e configurações de banda e número de atacantes. As análises aplicadas à rede permitem obter estimativas do tempo de vida tanto do controlador, dos hosts (que são configurados como vítimas), bem como da vazão da rede sobre ataques dessa natureza. Conclui que o ambiente virtualizado, utilizando SDN, mostra-se alternativa efetiva para avaliação de ataques DDoS.

**Palavras-chave:** Redes definidas por *software*; Tunelamento GRE; ataques DDoS; Mininet; análise de sobrevivência.

### ABSTRACT

The purpose of this article is to analyze the behavior of SDN in simulated environments in relation to distributed denial of service (DDoS) attacks, using statistical methods, in particular, the survival analysis. The main approach applied to the analysis is based on the lifetime of the network under attack, collecting characteristics, such as limits and variants of the band, and number of attackers. The analysis applied to the network provides an estimate of controller life, hosts (configured as victims), as well as network throughput on attacks of this nature. It concludes that the virtualized environment, using SDN, is an effective alternative for evaluating DDoS attacks.

**Keywords:** Software defined network; GRE tunelling; DDoS attaks; Mininet; Survival analysis.

#### Referência:

ARAÚJO, M. J.; PREDIGER, L.; FARIAS JR., E. P.; LIMA, M. N. Uma avaliação empírica dos efeitos de ataques de negação de serviço sobre redes definidas por *software*. In: SEMANA TECNOLÓGICA ACADÊMICA DE CIÊNCIA DA COMPUTAÇÃO DA UTFPR-SH, 4., 2017, Santa Helena, PR. *Anais...*, Santa Helena: UTFPR, 2017. p.48-60. Disponível em: <<http://sh.utfpr.edu.br/setac/Anais2017.pdf>>.

## INTRODUÇÃO

Redes de computadores é uma das áreas de pesquisa da Computação com evolução exponencial nos últimos anos, tanto na necessidade de prover conectividade quanto em atender às demandas de diversas áreas, tais como sociais, empresariais, acadêmicas e científicas.

As Redes Definidas por Software (SDNs – Software Defined-Networks) são redes emergentes que surgiram no processo de inovação tecnológica, com a finalidade de desacoplar as tarefas de roteamento dos equipamentos (roteadores) de forma a designar a responsabilidade do

processamento do roteamento para um controlador (KREUTZ, RAMOS e VERRISSIMO, 2013). Como complemento às definições desse tipo de rede, Nunes et al. (2014) afirmam que a SDN foi uma inovação tecnológica das redes, de forma a permitir o desenvolvimento de programação, para controlar os caminhos que os dados devem percorrer na rede. A realocação do *hardware* para o ambiente físico, utilizando um computador, facilita a implantação de novos protocolos e aplicativos, bem como a visualização e o gerenciamento direto das redes e a consolidação das várias caixas intermediárias aplicadas ao controle de *software*. Em vez de aplicar políticas e executar protocolos em uma convolução de dispositivos dispersos, a rede é reduzida a *hardware* de encaminhamento “simples” e ao(s) controlador(es) de rede de decisão (NUNES et al., 2014).

Apesar disso, aspectos de segurança computacional, tais como as vulnerabilidades e riscos devem ser tratados, pois podem comprometer a qualidade e a segurança nas SDNs. No contexto de segurança computacional, diversos estudos e experimentos são desenvolvidos, a fim de mitigar e minimizar ao máximo as vulnerabilidades e riscos nas redes de computadores.

Nesse sentido, o objetivo deste artigo é analisar o comportamento da SDN em ambientes simulados diante de ataques distribuídos de negação de serviço (DDoS), por meio de métodos estatísticos, em especial, a análise de sobrevivência. Desse modo, o paradigma SDN é aplicado sobre duas instâncias de redes, via Mininet, emulador de redes que cria uma rede virtual de *hosts*, *switches*, roteadores e *links*. As duas redes estão conectadas por protocolo de encapsulamento GRE Tunneling (tunelamento), por se tratar de uma configuração que possibilita a integração entre redes (HAMEED e KHAN, 2017).

Aplicações estocásticas são de suma importância em pesquisas de diversas áreas do conhecimento, principalmente em redes, e neste trabalho são fator principal na tomada de decisão, para resolver o problema ocasionado pelo ataque. As ferramentas de ataques DDoS desempenham importante papel neste trabalho, uma vez que, por meio delas, torna-se possível compreender e analisar o comportamento das redes nessas situações.

## 1 REVISÃO DA LITERATURA

### 1.1 ARQUITETURA DA REDE DEFINIDA POR SOFTWARE (SDN)

A SDN possui uma arquitetura em que as decisões de encaminhamentos de pacotes deixam de ser determinadas pelos dispositivos da rede, e assim passam a ser decididas por um dispositivo especial denominado Controlador SDN. Para que esse controlador funcione perfeitamente, devem-se utilizar protocolos de comunicação, e o mais utilizado nessa arquitetura, em conjunto com o controlador, é o protocolo OpenFlow (NUNES et al., 2014).

O protocolo OpenFlow possui interface simples de programação. Ele permite o controle da tabela de encaminhamento, que será utilizada pelos dispositivos da rede. Esse protocolo foi proposto pela Universidade de Stanford, cujo objetivo inicial era atender à demanda de validação de novas propostas de arquitetura e protocolos de rede para atuar em equipamentos comerciais (MCKEOWN et al., 2008). Portanto, o protocolo OpenFlow possibilita a escalabilidade na construção de tecnologia capaz de promover a inovação no núcleo da rede, por meio de execução de redes de testes, em paralelo com as redes de produção. Por isso, ele promove a criação de redes SDN, permitido utilizar elementos como *switches*, roteadores, pontos de acessos, computadores e demais dispositivos conectáveis (MCKEOWN et al., 2008).

Casado et al. (2009) definem a arquitetura de redes SDN em quatro pilares essenciais:

- Os planos de controle e de dados são desacoplados. A funcionalidade de controle é removida dos dispositivos de rede, que se tornarão elementos de encaminhamentos simples (pacotes);
- As decisões de encaminhamentos são baseadas em fluxos em vez de destinos. Dessa forma, um fluxo é amplamente definido por um conjunto de valores de campos de pacotes que atuam como um correspondente (filtro);

- A lógica de controle é movida para uma entidade externa, o chamado Controlador SDN ou NOS. A plataforma NOS é um ambiente de *software* que é executado em tecnologia de servidor de *commodities* e fornece os recursos essenciais e abstrações de forma a facilitar a programação de dispositivos de encaminhamentos com base em uma visão de rede logicamente centralizada e abstrata. O objetivo é atender de forma semelhante a um sistema operacional tradicional;
- A rede é programável por meio de aplicativos de *software* em execução no NOS, que interage com os dispositivos de plano de dados subjacentes. Essa é uma característica fundamental da SDN, considerada sua principal proposta de valor.

Com base na discussão apresentada neste texto, de forma sucinta, apresenta-se então a arquitetura da SDN, ilustrada na

Figura 1. Por meio dela, verifica-se que uma SDN contempla um sistema de controle (*software*), cujo objetivo é controlar o mecanismo de encaminhamento dos elementos de comutação da rede, via uma interface de programação bem definida. Assim, os elementos de comutação são exportados para uma interface de programação, permitindo ao *software* inspecionar, definir e alterar as entradas da tabela de roteamento do comutador, por meio do OpenFlow (GUEDES et al., 2012). Dessa forma, o OpenFlow, além de atuar como interface entre as camadas de controle e infraestrutura, também define as regras de encaminhamento. Isso possibilita a criação de SDN com implementações construídas pelo OpenFlow (KREUTZ, RAMOS e VERRISSIMO, 2013).

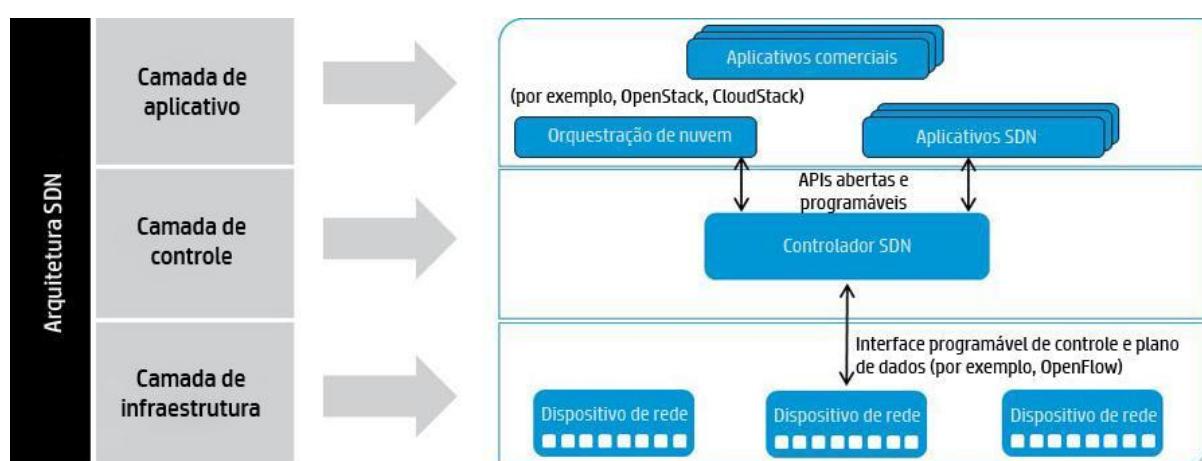


Figura 1: Arquitetura da Rede SDN.  
Fonte: Hewlett-Packard Development Company (2017).

A essência de redes definidas por *software* dá-se por meio da possibilidade de desenvolver programação dos elementos da rede. Entretanto, essa programação restringe-se à manipulação simples de pacotes, baseada no conceito de fluxos, ou seja, sequências de pacotes que compartilham atributos com valores adequadamente predefinidos. Então, a constituição de um fluxo é uma função que comprehende os recursos oferecidos pela interface de programação.

## 1.2 FUNCIONALIDADE DA SDN

Cabe explanar basicamente a funcionalidade da rede SDN, conforme Sezer et al. (2013). A Figura 2 apresenta a arquitetura funcional, a ilustração da infraestrutura de controle e outros elementos de aplicação, que compõem as SDN. Lins (2015) apresenta de forma sucinta a definição dessa infraestrutura, como segue.

O principal componente de uma SDN é o controlador, também chamado de Sistema Operacional da Rede. Esse ambiente define a natureza do paradigma SDN, é responsável por concentrar a comunicação com todos os elementos programáveis da rede e oferece uma visão unificada dela. Existem diversos tipos de controladores aplicáveis às redes SDN.

Na arquitetura SDN, as aplicações passam a ser ativas nas redes, função que a difere das redes convencionais, uma vez que a rede reconhece a aplicação que é executada. Em redes convencionais, as aplicações devem descrever indiretamente seus requisitos para o funcionamento adequado, o que normalmente envolve diversas etapas e processamentos para negociar recursos que estejam disponíveis, bem como definir uma política de controle de forma a atender ao aplicativo em execução. Em SDN, as aplicações podem monitorar o estado da rede de forma a adaptar-se conforme sua estrutura, e o plano de controle é lógico e centralizado, em que o controlador SDN efetua um resumo do estado da rede, de forma a comunicar os aplicativos, bem como faz a tradução dos requisitos das aplicações em execução.

O Limite Norte é uma interface entre as aplicações e os controladores, pois tem a característica de oferecer uma visão abstrata da rede, além de receber as informações do comportamento e requisitos da rede. Já o Limite Sul é a interface definida entre um controlador e a infraestrutura, ou seja, é um dispositivo SDN, pois – por meio de uma linguagem como OpenFlow, em que a API tem o objetivo de fornecer o controle programático de todas as operações, capacidades de notificação – estatísticas e relatórios são disponibilizados e construídos nessa interface. De forma análoga, as interfaces Limite Norte e Limite Sul são implementações construídas de forma aberta, independentes de fornecedores e com interoperabilidade.

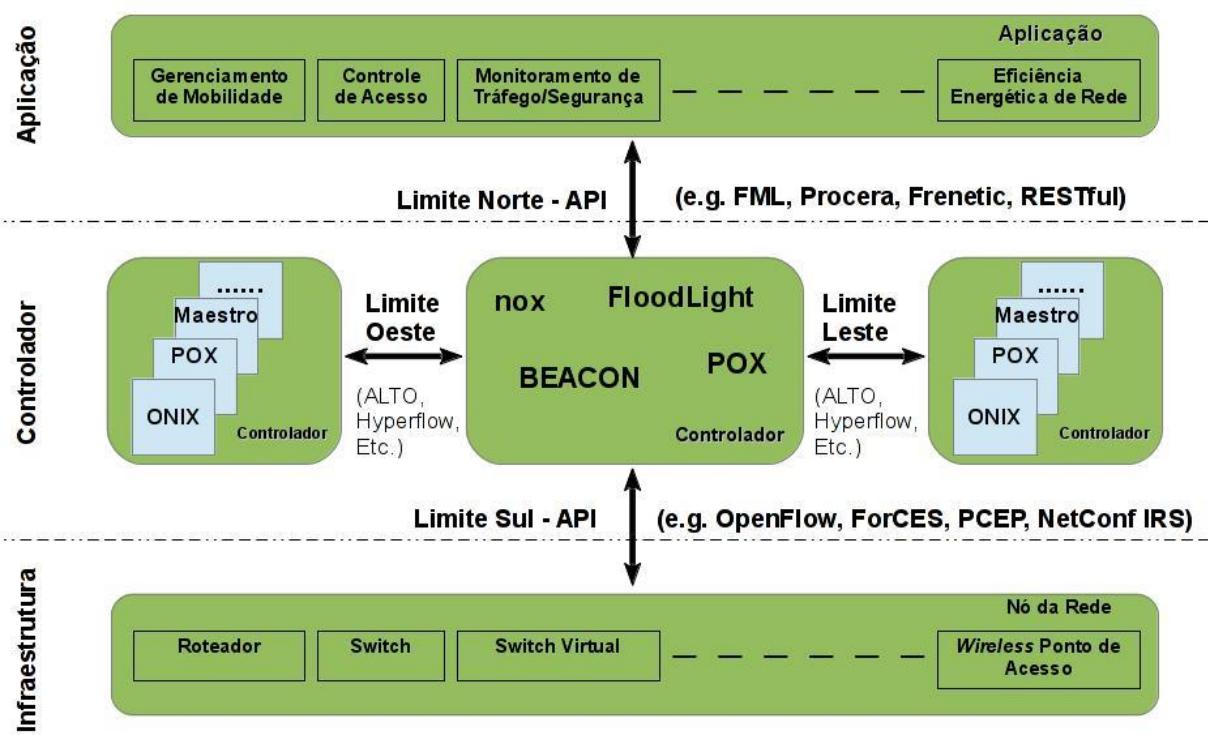


Figura 2: Arquitetura funcional da SDN, ilustrando infraestrutura, controle e outros elementos de aplicação que compõem esse tipo de rede.

Fonte: Adaptado de Sezer et al. (2013).

Dispositivos programáveis, normalmente *switches* e roteadores, recebem a informação do controlador via a interface Limite Sul para montar seu plano de dados. Após o recebimento das informações do controlador, as decisões já podem ser tomadas sobre o que será feito com os pacotes. Além dos dispositivos suportados pelos ambientes livres GPL, existem também muitos

dispositivos comerciais que suportam o padrão OpenFlow, como por exemplo HP, NEC dentre outros. Essa é uma evolução eminente do SDN, pois provocará a tendência de que mais fabricantes adotem esse padrão. As aplicações estatísticas sempre estão presentes no estudos e análises das redes, como é o caso do trabalho desenvolvido por Hermeto (2014), que aplicou a técnica da análise de sobrevivência, com o objetivo de estimar a estrutura probabilística do tempo de vida das Redes de Sensores Sem Fio (RSSF), como forma preventiva de usabilidade da rede. Com essa análise dos tempos de vida de seus nós em simulação, a questão em aberto na literatura que justificou essa abordagem é por quanto tempo a rede permanecerá em funcionamento.

As Redes Definidas por Software são uma realidade e estão sendo utilizadas não somente pela comunidade de *software* livre, mas também pela comunidade acadêmica e por gigantes do mercado, como a HP, CISCO, Juniper, VMWare dentre outros.

### 1.3 ASPECTOS DE SEGURANÇA EM REDES SDN

Como em qualquer tecnologia, a segurança também é elemento essencial na área de redes, principalmente no ambiente SDN. Dessa forma, alguns dos temas pertinentes e importantes para aplicação e estudo, no sentido de prover um mecanismo de segurança em redes, são citados a seguir:

- Ataques de Negação de Serviço ou Denial of Service (DoS);
- Detecção de *malware*;
- Uso de Intrusion Detection Systems (IDS) e Intrusion Prevention System (IPS);
- Análise de vulnerabilidades;
- Segurança na tabela ARP;
- Mitigação de ataques;
- Implementação de *firewall*;
- Autenticação e autorização.

Diversos estudos estão sendo desenvolvidos para atender à segurança em redes de computadores, com técnicas diversas, tanto do ponto de vista de invasão como do ponto de vista de infraestrutura mal construída. Redes SDN, por possuírem gerência por meio de aplicações, que definem a transferência dos pacotes na rede, porém, sem interferir nos atuais protocolos das camadas, como IP, TCP, UDP, ARP e HTTP, requerem especial atenção.

Uma das abordagens necessárias do ponto de vista de segurança em redes SDN é a criação da arquitetura de segurança, cujo objetivo é apoiar a detecção de invasões e ameaças às redes SDN, além de propor o desenvolvimento de recursos para realizar análise de intrusão e análise de ataques, principalmente em ambiente OpenFlow, conforme proposto por Martins e Campos (2016), que apontaram o OpenFlow como uma das vulnerabilidades de desenvolvimento. Todavia, ambientes que necessitam de redundância, alta disponibilidade, escalabilidade e resiliência certamente não podem ficar de fora dos recursos que a segurança computacional em redes SDN deve cobrir. Essa é uma premissa herdada das infraestruturas de redes convencionais, uma vez que todo o ambiente gerencial das redes SDN centra-se no servidor principal ou em um ambiente distribuído de servidores, que administram as redes e seus encaminhamentos, mostrando, assim, em termos de segurança, ser um item que merece atenção.

Como os protocolos são os mesmos em ambas as tecnologias de rede, os problemas já reconhecidos que afetam a infraestrutura das redes convencionais ocorrem em SDN, como é o caso de ataques denominados Man-in-the-Middle (MITM). Esse tipo de ataque é provido com o uso das mensagens ARP fraudulentas, que se posicionam no meio da comunicação e, após a configuração, se estabelece o ataque, fazendo com que o fraudador tenha condições de visualizar todas as mensagens trocadas pelas vítimas (OLIVEIRA, SCHWEITZER e AKIRA, 2016). Isso acontece quando um atacante mal-intencionado modifica o mapeamento de um endereço IP para

seu próprio endereço MAC, na tabela ARP de outros *hosts*. Essa técnica é chamada Fraude de ARP ou ARP Spoofing, pois consiste no envio de respostas ARP, como *broadcast*, em uma rede local e de forma indiscriminada, mesmo que não haja requisições anteriores (OLIVEIRA, SCHWEITZER e AKIRA, 2016).

Arbettu et al. (2016) observam que as estratégias de segurança são fundamentais para a construção de infraestruturas confiáveis e seguras dentro das SDNs, porém, focar no controlador da SDN tem sido prática de diversos pesquisadores que abordam o tema segurança em controladores, como OpenDaylight, ONOS, Rosemary e RYU. A detecção de ataques, como DDoS, usando NOX/OpenFlow, foi proposta por Braga, Mota e Passito (2010) como um dos principais problemas de segurança na Internet na última década. O método proposto pelos autores é baseado em recursos de fluxo de tráfego, no qual a extração das informações é feita com uma sobrecarga muito baixa em comparação com as abordagens tradicionais. Esses recursos são viáveis graças ao ambiente programável NOX como interface de desenvolvimento. Uma vez que toda a infraestrutura está centrada no servidor, Braga, Mota e Passito (2010) utilizaram uma Rede Neural Artificial Não Supervisionada, denominada Self Organizing Maps (SOM), para auxiliar a extração dos resultados dos ataques, de forma inteligente. Eles apresentaram como ferramenta de apoio à tomada de decisão, no contexto da segurança em redes SDN.

Outra abordagem que vem, nos últimos tempos, destacando-se na comunidade científica é a mitigação dos ataques DDoS. Embora o ato de mitigar nem sempre seja solução definitiva para os ataques de DDoS, sua aplicação tem mostrado resultados eficientes. Essa afirmação pode ser observada em Macedo et al. (2016), com uso da computação evolucionária por meio de Algoritmo Genético, como técnica de otimização, para mitigar ataques DDoS em nós clusterizados. Nesse cenário, esses autores utilizaram o esquema denominado Scheme for DDoS Attacks Mitigation with the reOrganization and optimization of the IdM System SAMOS aplicado em IdPs (Provedor de Identidade) e IdM (Gerenciamento de Identidade), que usa algoritmo genético para identificar o número máximo de agrupamentos de IdP, capazes de mitigar a sobrecarga gerada pelos ataques DDoS. Além disso, os autores empregaram técnicas de otimização, a fim de melhorar o uso dos recursos do sistema, minimizando os efeitos de ataques de forma a estender a vida útil do sistema IdM. Já Hameed e Khan (2017) aproveitaram a rede SDN para mitigar de forma colaborativa ataques DDoS. A técnica de mitigação colaborativa depende de múltiplas redes SDN, por isso, o cenário foi construído com o recurso de tunelamento entre redes (GRE Tunneling), que possibilitou o uso do protocolo Controller to Controller (C-to-C). A ideia principal dessa plataforma é criar uma autoridade confiável, que verifique as definições de ataques recebidas através de DDoS, para que os controladores intermediários não passem pela tarefa trabalhosa de verificar e encaminhar individualmente as definições de ataques.

Para a prova de eficiência da técnica estudada neste artigo, foram introduzidas três abordagens diferentes de redes: linear, central e malha. Os resultados da avaliação desses ambientes mostram que o efeito da mitigação é instantaneamente transferido de um destino para outro. Com isso, a mitigação do ataque demora cerca de 2,14 segundos em uma rede linear de oito saltos, além de requerer apenas 290 a 330ms para processar e encaminhar as definições de ataques entre redes adjacentes. Essas técnicas utilizadas são complementares entre si, uma vez que focam mitigação, apesar de os cenários distintos utilizarem o mesmo tipo de rede, mostrando que as redes SDN são dinâmicas e escalares.

As aplicações estatísticas fazem parte das análises desenvolvidas em redes, em especial a abordagem de Análise de Sobrevivência. Nesse sentido, destaca-se o estudo de Hermeto (2014), aplicado para estimar a estrutura probabilística do tempo de vida de uma Rede de Sensores Sem Fio (RSSF), abordando modelos exponenciais, Weibull e log-normal, normalmente utilizados em estudos de análise de sobrevivência, para obter estimativas do tempo de vida de uma rede RSSF real, a partir dos tempos de vida observados de seus nós em simulação. A função de sobrevivência  $S(t)$  ou função de probabilidade fornece a probabilidade de um nó estar vivo após

o instante  $t$ , conforme Cerpa e Estrin (2004). Dessa forma, este trabalho propõe responder a questão em aberto na literatura: por quanto tempo a rede permanecerá em funcionamento?

## 2 METODOLOGIA

A metodologia aplicada neste trabalho dá-se em função de duas abordagens específicas. A primeira é atender ao objetivo de analisar a resiliência de uma rede, e a segunda deve ser analisar a resiliência em uma topologia que tenha no mínimo duas redes conectadas entre si. Dessa forma, optou-se por utilizar GRE Tunneling, cuja funcionalidade atende às necessidades específicas para este trabalho. Outra abordagem que contribui para o desenvolvimento dos experimentos é a aplicação da análise estatística, que deve trazer uma resposta sobre o tempo de vida estimado, dentro de um período, em uma rede sob ataque de DDoS.

O *hardware* utilizado para este trabalho contou com a seguinte configuração:

- Microcomputador HP 16GB RAM, 1TB HD, processador Intel Core i7-4770 - CPU 3.40GHz x 8;
- Sistema Operacional Ubuntu 17.04.
- VirtualBox 5.1.

Os experimentos deste artigo foram realizados em duas redes no ambiente virtual (VirtualBox). Cada rede foi construída em máquinas virtuais (VMs) diferentes, conectadas entre si via GRE Tunneling. Essa configuração foi construída utilizando-se dois controladores e dois nós, com ambos apontando para a mesma porta de comunicação, que nesse caso é a porta 6633, um em cada VM (VM1 = C0 e VM2 = C1), como mostra a **Erro! Fonte de referência não encontrada.**, que apresenta a configuração dos cenários propostos. Pode-se observar dois ambientes, mininet@mininet-vm-1 e mininet@mininet-vm-2.

```

mininet@mininet-vm1: ~
mininet@mininet-vm1: ~ 71x16
from mininet.link import TCLink, Intf
def emptyNet():
    NODE2_IP='192.168.56.16'
    CONTROLLER_IP='192.168.56.15'

    net = Mininet( topo=None,
                   build=False)
    net.addController( 'c0',
                      controller=RemoteController,
                      ip=CONTROLLER_IP,
                      port=6633)

mininet@mininet-vm2: ~ 71x16
from mininet.cli import CLI
from mininet.log import setLogLevel, info
def emptyNet():
    NODE1_IP='192.168.56.15'
    CONTROLLER_IP='192.168.56.16'

    net = Mininet( topo=None,
                   build=False)
    net.addController( 'c1',
                      controller=RemoteController,
                      ip=CONTROLLER_IP,
                      port=6633)

```

Figura 3: Ambiente construído para conexão em tunelamento.

Nesse cenário as configurações contam com:

- VM1 - 42 *hosts*;
- VM1 - 07 *switches*, configurados em *gigalan*;

- VM1 - 01 controlador;
- VM2 - 20 hosts;
- VM2 - 01 switch;
- VM1 - 01 controlador.

O cenário construído pode ser observado no diagrama da **Erro! Fonte de referência não encontrada..**.

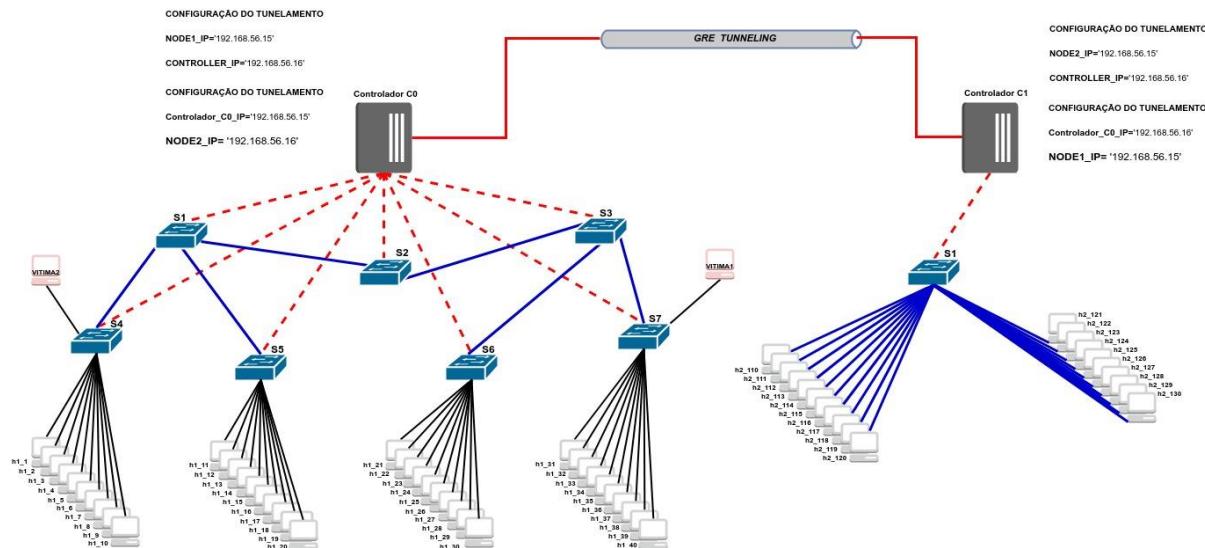


Figura 4: Diagrama das redes construído.

## 2.1 AMBIENTE DE ATAQUES DDoS

Para atender à proposta de ataques DDoS, foram efetuados diversos testes, que abrangem inclusive ataques via Kali Linux, mediante a ferramenta 50 Sukhoi PAK FA Mixed Packet Injector, uma *packet injection* (injeção de pacote) livre, criada pelo brasileiro Nelson Brito, capaz de fazer ataques DoS e DDoS usando o conceito de *stress testing* (MUNDO DOS HACKERS, 2017). O objetivo dessa ferramenta é enviar altíssimo número de requisições de pacotes, de tal forma que o alvo não consiga atender a todas requisições ou atenda-lhes de forma lenta, esperando-se que o canal de comunicação com a máquina alvo caia ou torne-se demasiadamente lento. Essa ferramenta se mostrou eficiente, porém, não foi adequada para o cenário proposto, pois não havia tempo hábil para coleta dos tempos de ataques e de queda do sistema, que ocorreu instantaneamente. Dessa forma, foram testadas outras ferramentas, como o Slowloris, desenvolvida pelo hacker RSnake, que causa DoS por meio de solicitação HTTP muito lenta (SECURITY RADWARE, 2017). Entretanto, essa ferramenta também não permitiu a validação do cenário, pois ambas (T50 e Slowloris) foram muito eficientes no ataque, gerando assim apenas um DoS. A intenção do estudo era realizar um ataque a partir de diversos equipamentos, para a validação do ambiente (HORAS, 2016), portanto, optou-se pela realização de ataques via comando *ping*, alternativa adequada para o cenário e estudo propostos, pois o comando *ping* permite a adição de parâmetros que podem causar lentidão no alvo e necessita de mais de uma máquina atacando, para gerar sucesso na queda do serviço ou servidor.

Os parâmetros utilizados no comando *ping*, para desestabilizar a máquina alvo e coletar os dados, foram:

- -i .0000000001;
- -l 65554
- nome\_do\_host\_tst\_num\_.txt

Assim, o comando “`ping -i 0.00000000001 -l 65554`” foi inserido em um arquivo de *script*, a fim de facilitar sua entrada nas máquinas atacantes e evitar a necessidade de digitá-lo em cada máquina. Dessa forma, para que o cenário fizesse o que se deseja, um *host* denominado VITIMA1 tornou-se servidor HTTP, por meio do comando:

```
mininet $>$ VITIMA1 python -m SimpleHTTPServer 80 &
```

Os ataques DDoS foram alternados, assim em alguns momentos a vítima estava na VM1 e em outros, na VM2. A Figura 5 ilustra a posição da máquina VITIMA1. O ataque foi contínuo, até que ocorresse a parada total do ambiente, após a queda do sistema Mininet. Logo, todo o sistema necessitava ser desconectado por meio do comando “exit”. Em seguida, havia a necessidade de executar o comando “`sudo mn -c`”, para fazer a limpeza dos ambientes criados e evitar vínculo nos testes. Além de desconectados, os ataques foram alternados, ora via VM1, ora via VM2.

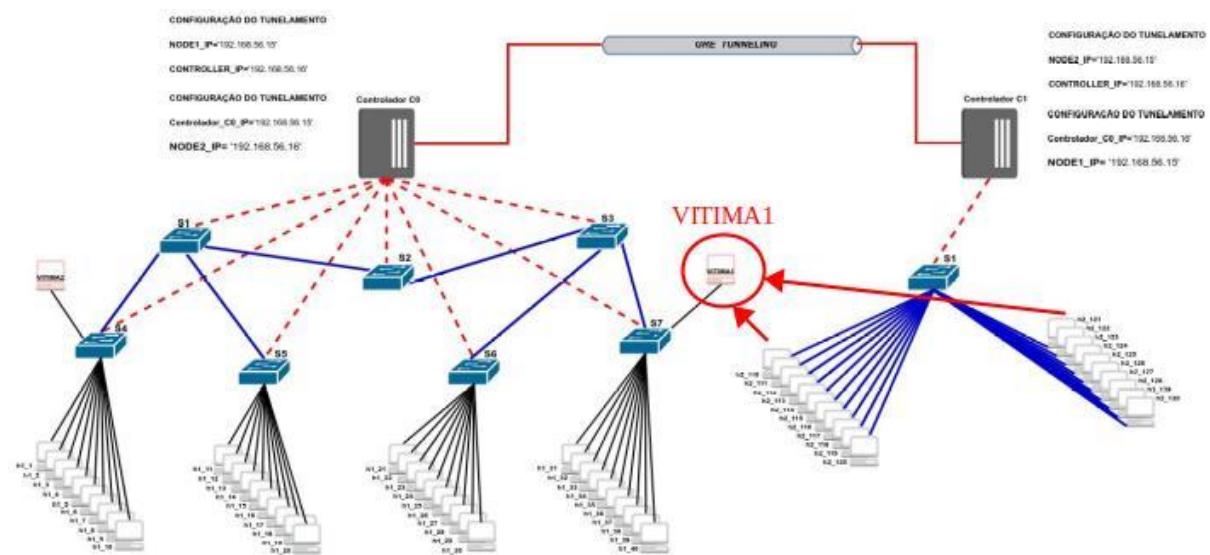


Figura 5: Ilustração dos ataques via VM1 para a VITIMA1.

Após a sequência de ataques, foram executados *scripts* para tratar os dados coletados, por meio do comando SAR, com passagem de parâmetros para coleta dos dados. Dessa forma, geraram-se 10 ciclos de ataques alternados, para se obter a base de análise estatística.

### 3 RESULTADOS

Por meio dos experimentos, constataram-se algumas informações no cenário a partir dos dados coletados. Primeiramente, pode-se observar que a SDN é robusta e adquiriu razoável resistência com a configuração construída, pois o fluxo da vazão de dados entre *hosts* na mesma VM1 e entre *hosts* por meio de tunelamento em VMs diferentes suporta razoável carga de dados, como se pode observar na Tabela 1.

Tabela 1: Dados dos testes.

Teste	Quantia de pacotes	rxpck/s	txpck/s	rxkB/s	txkB/s
1	55	14.630,01	13.553,66	1.400,17	1.294,078
2	208	8.550.183,63	8.380.291,86	818.275,70	802.014,77
3	243	900.562,96	143.846,63	86.180,29	13.751,52
4	56	1.849.618,16	1.766.265,98	177.014,21	169.035,98
5	1246	102.835.871,43	102.633.613,71	9.841.665,48	9.821.588,02
6	375	12.613.324,98	9.000.671,97	1.207.130,27	262.870,36
7	1164	88.074.262,60	85.845.258,76	8.428.926,91	8.215.606,70
8	536	12.766.798,19	10.655.707,07	1.221.719,70	1.019.656,64
9	133	485.964,77	77.377,59	46.502,33	7.361,22
10	974	49.683.113,99	39.767.088,90	4.754.767,55	3.805.283,02

Todavia, mesmo que a máquina vítima resistisse ao ataque, ou seja, permanecesse comunicando-se com os demais *hosts*, houve considerável perda de dados, provocada pelo grande volume simultâneo de requisições a ela direcionadas. A Tabela 2 detalha individualmente os dados de alguns dos *hosts* utilizados em um ataque bem-sucedido e mostra, em sua terceira coluna, a taxa de pacotes perdidos.

Tabela 2: Dados dos pacotes transmitidos via comando *ping*.

Host	Pacotes transmitidos	Pacotes recebidos	Pacotes perdidos
h1_1	524762	145107	72%
h1_10	556058	192961	65%
h1_11	610152	184312	69%
h1_12	293033	133569	54%
h1_13	595985	177531	70%
h1_14	106261	18089	82%
h1_15	2754976	1941045	29%
h1_16	670465	203051	69%
h1_17	291867	64467	77%
h1_18	24867584	18067546	27%
h1_2	22853520	16383295	21%
h1_20	610822	214333	64%

Os dados da Tabela 2 foram extraídos do relatório exibido no momento em que a execução do comando *ping* foi interrompida, após o pressionamento da combinação de teclas *ctrl+c*. Já a Tabela 3 e a Tabela 4 exibem as informações da máquina vítima geradas durante o mesmo ataque, coletadas pelo programa IPtraf-ng, um *sniffer* para monitoramento de rede.

Tabela 3: Transmissão de pacotes na máquina vítima.

<b>Protocolo</b>	<b>Pacotes</b>		
	<b>Enviados</b>	<b>Recebidos</b>	<b>Total</b>
IP	10.537.191	11.879.620	22.416.811
ICMP	10.537.161	11.879.590	22.416.751

Tabela 4: Informações da máquina vítima durante ataque.

<b>Taxa de transferência</b>	<b>Recebidos</b>	<b>Enviados</b>
kbps	5.574,78	4.944,85
pps	8.295	7.358

A Figura 6 registra o momento em que a máquina vítima pára de responder à requisição, solicitando informações HTTP. Na imagem, a requisição foi realizada quando o relógio do sistema marcava 13h00m35s. Nota-se também que a resposta informando a falha surgiu 20 segundos depois, ou seja, já havia lentidão na comunicação.

```

<li><a href="pox/">pox/</a>
</ul>
<hr>
</body>
</html>
100%[=====] 1,006      -.- K/s   in 0s
2017-09-13 13:00:27 (170 MB/s) - written to stdout [1006/1006]

mininet> h1_1 wget -O - VITIMA1
--2017-09-13 13:00:35--  http://10.0.0.100get/
Resolving 10.0.0.100get (10.0.0.100get)... failed: Name or service not known.
wget: unable to resolve host address '10.0.0.100get'
--2017-09-13 13:00:55--  http://10.0.0.100/
Connecting to 10.0.0.100:80... □

```

Figura 6: Falha na solicitação HTTP para a máquina VITIMA1 partindo da máquina h1\_1.

## CONCLUSÕES

O ambiente de simulação Mininet contém recursos capazes de fazer análises de tráfego de rede e permite uma simulação realista de ataques de negação de serviço. As redes utilizadas como objeto de estudo obtiveram bom desempenho, mesmo sobre configuração padrão, sem nenhuma especialização de ambiente, tais como *firewall*, controle de banda, dimensionamento de *hardwares* (caso do *host* denominado VITIMA1).

O cenário permitiu análises do sistema operacional, que apresentou um *throughput* relativamente denso na rede durante os ataques, assim eles não foram suficientes para derrubar a rede, apenas o serviço de Internet por meio do protocolo HTTP.

Apesar disso, pode-se concluir que a utilização do ambiente composto pelo Mininet, acompanhado do tunelamento via GRE, permite a criação de um cenário favorável para o monitoramento de ataques DDoS a SDNs. Tal ambiente favorece a simulação de variados cenários e a extração de resultados, além de contribuir na busca de soluções de prevenção ou contingência para SDNs, sob a iminência de ataques DDoS. Além disso, a criação de ambientes virtuais mostra-se alternativa segura para testes e estimativas de tempo de vida em simulações desses ataques.

## REFERÊNCIAS

- ARBETTU, R. K., KHONDOKER, R., BAYAROU, K., & WEBER, F. Security analysis of OpenDaylight, ONOS, Rosemary and Ryu SDN controllers. **International Telecommunications Network Strategy and Planning Symposium (Networks)**, 17th, 2016. p. 37-44.
- BRAGA, R.; MOTA, E.; PASSITO, A. Lightweight DDoS flooding attack detection using NOX/OpenFlow. **Local Computer Networks (LCN) IEEE Conference**, 35th, 2010. p. 408-415.
- CASADO, M.; FREEDMAN, M. J.; PETTIT, J.; LUO, J.; GUDE, N.; MCKEOWN, N.; SHENKER, S. Rethinking enterprise network control. **IEEE/ACM Transactions on Networking**, 2009. p.1270-1283.
- CERPA, A.; ESTRIN, D. ASCENT: Adaptive self-configuring sensor networks topologies. **IEEE transactions on mobile computing**, v.3, n.3, p.272-285, 2004.
- GUEDES, D.; VIEIRA, L.; VIEIRA, M.; RODRIGUES, H.; NUNES, R. V. Redes Definidas por Software: uma abordagem sistêmica para o desenvolvimento de pesquisas em Redes de Computadores. **SBRC**. Minicurso. 2012. p.160-210.
- HAMEED, S.; KHAN, H. A. **Leveraging SDN for collaborative DDoS mitigation**. Networked Systems (NetSys), 2017 International Conference. IEEE. 2017. p.1-6.
- HEWLETT-PACKARD DEVELOPMENT COMPANY. Arquitetura de Redes Definida por Software. **HP Network**, maio 2017. Disponível em: <<http://h17007.www1.hpe.com/br/pt/solutions/technology/openflow/index.aspx>>. Acesso em: set. 2017.
- HORAS, H. Fazendo um ataque Ddos pelo CMD. **Hacker 24 Horas**, 2016. Disponível em: <<https://hackers24hrs.wordpress.com/2014/07/16/fazendo-um-ataque-ddos-pelo-cmd/>>. Acesso em: 9 jul. 2017.
- KREUTZ, D.; RAMOS, F.; VERÍSSIMO, P. Towards Secure and Dependable Software-defined Networks. In: ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, 2nd. **Proceedings...** Hong Kong: ACM, 2013. p.55-60. Disponível em: <<http://doi.acm.org/10.1145/2491185.2491199>>. Acesso em: set. 2017.
- LINS, T. **Imobilis**. 2015. Disponível em: <<http://www.decom.ufop.br/imobilis/redes-definidas-por-software-software-defined-networks-sdn>>. Acesso em: 13 set. 2017.
- MACEDO, R.; SANTOS, A.; GHAMRI-DOUDANE, Y.; NOGUEIRA, M. A scheme for DDoS attacks mitigation in IdM systems through reorganizations. In: **Network Operations and Management Symposium (NOMS)**. IEEE, 2016. p.298-305.
- MARTINS, J. S.; CAMPOS, M. B. A security architecture proposal for detection and response to threats in SDN networks. **IEEE**, p. 1-4, 2016.
- MCKEOWN, N.; ANDERSON, T.; BALAKRISHNAN, H.; PARULKAR, G.; PETERSON, L.; REXFORD, J. ... TURNER, J. OpenFlow: enabling innovation in campus networks. **ACM SIGCOMM Computer Communication Review**, p.69-74, 2008.
- MUNDO DOS HACKERS. Estudo de caso de Ataques de Negação de Serviço (DDoS). **Mundos hackers**, 2017. Disponível em: <<http://www.mundodoshackers.com.br/ferramenta-para-dosddos-t50>>. Acesso em: 7 jul. 2017.
- NUNES, B. A. A.; MENDONCA, M.; NGUYEN, X. N.; OBRACZKA, K.; TURLETTI, T. A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks. **IEEE Communications Surveys Tutorials**, v.16, n.3, p.1617-1634, 2014.
- OLIVEIRA, R. L. S.; SCHWEITZER, C. M.; AKIRA, A. **Testes de desempenho com o módulo de segurança L3-ARPSec em Redes Definidas por Software**. XI WORKSHOP DE PÓS-GRADUAÇÃO E PESQUISA DO CENTRO PAULA SOUZA. São Paulo, Out. 2016.

SECURITY RADWARE. DDoS Prevention Techniques and DDoS Best Practices. **Security Radware**, 2017. Disponível em: <<https://security.radware.com/ddos-knowledge-center/ddospedia/slowloris/>>. Acesso em: 9 jul. 2017.

SEZER, S.; SCOTT-HAYWARD, S.; CHOUHAN, P. K.; FRASER, B.; LAKE, D.; FINNEGAM, J.; Rao, N. Are we ready for SDN? Implementation challenges for software-defined networks. **IEEE Communications Magazine**, v.51, n.7, p.36-43, 2013.

# Conceitos de qualidade de *software* aplicados ao desenvolvimento de um sistema *web*

## Software quality concepts applied to the development of a web system

Luca Prediger<sup>1</sup>, Márcio José Araújo<sup>2</sup>, Suzan Kelly Borges Piovesan<sup>3</sup>

Ciência da Computação, Universidade Tecnológica Federal do Paraná (UTFPR-SH)

*predigerluca@gmail.com<sup>1</sup>, marcioaraaujo@alunos.utfpr.edu.br<sup>2</sup>, suzankpiovesan@utfpr.edu.br<sup>3</sup>*

### RESUMO

Este trabalho apresenta um planejamento da aplicação de conceitos da qualidade de *software* (QS) na produção de um sistema *web*, utilizando metodologias ágeis, a fim de mostrar como se implementam tais métodos. Prevê a implementação de métodos para controle de tarefas, controle de versionamento e ambiente de programação integrados (IDE) em um ambiente criado para controle de atividades acadêmicas chamado Sweca, o qual ainda está em fase inicial de desenvolvimento, e deverá controlar eventos na UTFPR, câmpus Santa Helena, quando finalizado. O trabalho ressalta a importância da utilização da QS no processo de desenvolvimento de sistemas, para proporcionar maior rapidez e versatilidade na produção de softwares.

**Palavras-chave:** Qualidade de *software*; sistemas *web*; PHP; Scrum.

### ABSTRACT

*This work presents a planning for the application of Quality of Software concepts in the production of a web system, using agile methodologies, in order to show how these methods can be implemented. It predicts the implementation of Task Control, Versioning Control, and Integrated Programming Environment (IDE) methods in an environment created for managing academic activities called Sweca, which is still in the early stages of development, and should manage events at UTFPR, Santa Helena campus, when concluded. The work underscores the importance of using QS in the system development process, to provide greater speed and versatility in software production.*

**Keywords:** Software quality; web system; PHP; Scrum.

### Referência:

PREDIGER, L.; ARAÚJO, M. J.; PIOVESAN, S. K. B. Conceitos de qualidade de *software* aplicados no desenvolvimento de um sistema *web*. In: SEMANA TECNOLÓGICA ACADÊMICA DE CIÊNCIA DA COMPUTAÇÃO DA UTFPR-SH, 4., 2017, Santa Helena, PR. Anais..., Santa Helena: UTFPR, 2017. p.61-71. Disponível em: <<http://sh.utfpr.edu.br/setac/Anais2017.pdf>>.

### INTRODUÇÃO

Qualidade de *software* (QS) – de acordo com a literatura, como Pressman e Maxim (2016), Ramanna, Peters e Ahn (2002), Sanders e Curran (1994) – é uma área do conhecimento da Engenharia de *Software* que se refere às características desejadas de produtos de *software*, à extensão que um produto de *software* em particular possui, ferramentas e técnicas que são usadas para garantir essas características. Também corresponde à capacidade de o produto de *software* satisfazer as necessidades implícitas e explícitas quando usado em condições específicas, pelos usuários.

Em conformidade com a literatura, necessidades explícitas são aquelas definidas no requisito do projeto. Esses requisitos devem definir as condições em que o produto deve ser utilizado e informar seus objetivos, funções e desempenho esperado. Já necessidades implícitas

são aquelas que, embora não expressas no documento de *software*, são necessárias para o usuário. Estão englobados nessa classe os requisitos que não precisam ser declarados por serem óbvios, mas que, pela gravidade das consequências caso sejam ignorados, devem ser levados em conta na produção do *software*.

Atualmente, a qualidade de *software* na produção de *software* não é restrita a grandes corporações. Entende-se que, se alguma empresa, principalmente do ramo computacional, tiver perspectivas de crescimento, deve utilizar tais métodos para que consiga atuar nesse mercado tão competitivo, para obter, além de melhorias no processo de produção, algum diferencial que a faça sobressair perante os concorrentes do mercado. Os métodos ágeis nessa fase, devem ser considerados estratégicos, garantir maior produção e consequentemente maior lucro e satisfação ao cliente, que é o alvo dos investimentos.

A partir disso, o objetivo do presente trabalho é demonstrar um planejamento de como se implantariam os conceitos de QS no desenvolvimento de um sistema de eventos criado pelos autores, cujo nome é Sweca, acrônimo de Sistema Web de Eventos Científicos e Acadêmicos. O produto advindo do referido projeto deve ser colocado em prática para auxiliar eventos acadêmicos que carecem de controle computacional para sua gestão.

Outrossim, considera-se relevante a existência de um documento que norteie novos colaboradores sobre as metodologias e boas práticas, seguindo técnicas de qualidade de *software*. Dessa forma, será possível contribuir para um produto que atenda adequadamente aos propósitos do escopo do projeto.

## 1 REFERENCIAL TEÓRICO-METODOLÓGICO

### 1.1 ASPECTOS TEÓRICOS

#### 1.1.1 Qualidade de *Software*

Segundo Pressman e Maxim (2016), qualidade de *software* é a conformidade a requisitos funcionais e de desempenho, que foram explicitamente declarados, a padrões de desenvolvimento claramente documentados e a características implícitas, que são esperadas de todo *software* desenvolvido por profissionais. Para Ramanna, Peters e Ahn (2002), a qualidade de *software* é avaliada em termos de atributos de alto nível chamados fatores, que são medidos em relação a atributos de baixo nível, chamados de critérios. Já Sanders e Curran (1994) afirmam que um produto de *software* apresenta qualidade, dependendo do grau de satisfação das necessidades dos clientes sob todos os aspectos do produto.

#### 1.1.2 Metodologias Ágeis

Em fevereiro de 2001, entre os dias 11 e 13, no Snowbird Ski Resort nas montanhas de Wasatch, Utah (EUA), ocorreu um encontro entre 17 representantes de diferentes metodologias de desenvolvimento simpatizantes de alternativas ao processo tradicional de desenvolvimento de *software*, que exige grande carga de documentação e planejamento. Foi durante esse encontro que surgiu o Manifesto para o Desenvolvimento Ágil de *Software* (LESLIE, 2015).

Assim, as metodologias ágeis para desenvolvimento de *software*, uma alternativa às chamadas metodologias tradicionais, são compostas por um conjunto de metodologias surgidas a partir de um encontro, que visam a agilizar a produção de *softwares*. Vários são os *frameworks* de processo para nortear o desenvolvimento de *softwares*. Os mais comuns são o Scrum, Crystal, Dynamic Systems Development Method (DSDM), Feature Driven Development (FDD), Extreme Programming (XP) e Kanban (MCLAUGHLIN, 2015). Nesse ponto, vale ressaltar que, apesar de algumas das citadas ferramentas terem sido formalizadas em meados dos anos 90,

como o Scrum e o DSDM, elas ainda não são de conhecimento de muitos gerentes de projeto (LESLIE, 2015).

### 1.1.3 Sweca

O Sistema Web de Eventos Científicos e Acadêmicos (SWECA) é um sistema *web* para controle de eventos. Ele está sendo desenvolvido para suprir a necessidade de controle computacional de eventos da UTFPR, Santa Helena, como a Semana Tecnológica Acadêmica de Ciência da Computação (SeTAC<sup>2</sup>) e a Exposição da Universidade Tecnológica (ExpoUT), ambos anuais. O objetivo de tais eventos é disseminar o conhecimento tecnológico e científico entre profissionais, estudantes, pesquisadores, alunos do curso de Ciência da Computação da UTFPR e de outros cursos da instituição e a comunidade externa, por meio de palestras, minicursos, oficinas, campeonatos e apresentação de trabalhos acadêmicos, nesse último, com posterior publicação dos artigos correspondentes nos anais do evento.

O Sweca é desenvolvido em linguagem PHP, aplicando-se conhecimentos de qualidade de *software* e metodologias ágeis em sua produção. O *software* está sendo criado por meio de módulos que trabalham de forma independente, e, a cada nova implementação, o sistema não apresenta problemas de acoplamento entre os módulos, que executam suas funções em conjunto e harmonicamente. O primeiro evento considerado para teste de utilização do Sweca é a SeTAC<sup>2</sup>. A Figura 1 apresenta a tela inicial da página da SeTAC<sup>2</sup>.



Figura 1: Tela inicial do site da SeTAC<sup>2</sup> (2017).

## 1.2 ASPECTOS METODOLÓGICOS

Mesmo que haja a premissa de que metodologias ágeis devem valorizar mais os indivíduos e interações do que processos e ferramentas, é essencial que processos e ferramentas sejam implantados para garantir o andamento adequado do projeto. Nesse sentido, o Sweca é desenvolvido por uma equipe de colaboradores, composta por 7 membros: 3 desenvolvedores

PHP, 1 *designer web*, 1 gerente de projeto, 1 *product owner* e 1 testador – e com a necessária utilização de técnicas e ambientes colaborativos de desenvolvimento.

### 1.2.1 Controle de Tarefas

Para que haja sincronia entre os itens produzidos por cada membro da equipe, ou seja, para que fique claro para todos qual trabalho cada um está desempenhando, o controle das tarefas do projeto é realizado pelo SeeNowDo (2017). Tal serviço provê às equipes que utilizam metodologias ágeis, como Scrum, colaborar por meio de *taskboards* digitais (são quadros digitais com as tarefas que serão desenvolvidas para a execução do projeto) altamente interativos, conforme mostra a Figura 2.



Figura 2: Controle de tarefas.

Além disso, a utilização dessa ferramenta auxilia na organização das tarefas por parte dos envolvidos no desenvolvimento do projeto. As tarefas são organizadas dentro de um painel de trabalho, bastante intuitivo, usam-se cores e colunas para oferecer todos os dados na mesma tela (POLO, 2010). O sistema SeeNowDo, executado pela *web*, mostra também o tempo que falta para concluir cada uma das atividades cadastradas, como se percebe na Figura 3.

Portanto, o SeeNowDo é um quadro *kanban*<sup>1</sup>, em formato digital, que pode ser acessado pelos membros da equipe de forma remota, tornando a produção e distribuição das atividades mais simples e rápida.

<sup>1</sup> **Kanban** é um termo de origem japonesa e significa literalmente “cartão” ou “sinalização”. Trata-se de um conceito relacionado com a utilização de cartões como *post-it* e outros, para indicar o andamento dos fluxos de produção em empresas de fabricação em série (SIGNIFICADOS, 2017).

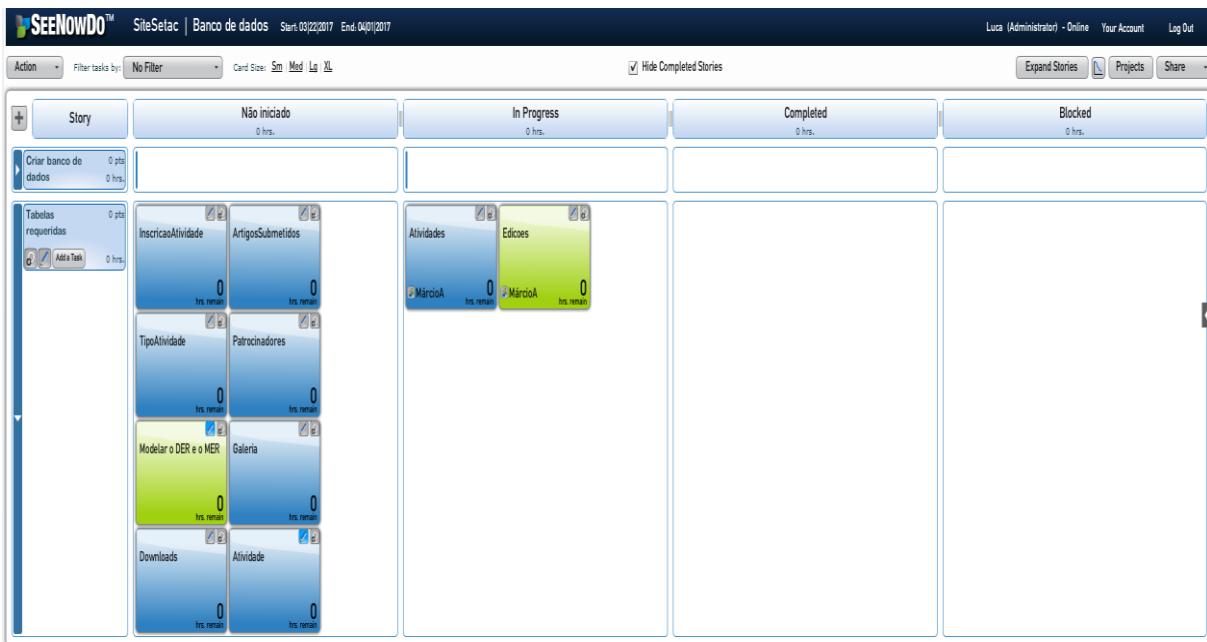


Figura 3: Duração e descrição de atividades.

Ao gerenciamento do desenvolvimento do Sweca foi adotada a metodologia ágil Scrum. Dessa forma, foram definidas as *sprints* que deveriam fazer parte do desenvolvimento inicial do sistema. Essas *sprints* estão descritas na Figura 2. Na sequência, a Figura 3 representa as tarefas de uma *sprint*. Cada tarefa deve ter ao menos um colaborador do projeto como dono temporário da tarefa, geralmente cada coluna representa um tipo de colaborador. O Quadro 1 mostra o tipo de colaborador que geralmente é o responsável (*owner*) pelas tarefas em cada uma das colunas e a respectiva descrição de cada coluna.

Quadro 1: Descrição dos estados das tarefas e papel do respectivo *owner*.

Coluna de estado da tarefa	Tipo de colaborador
<i>Não iniciado</i> : Tarefas que ainda estão aguardando o responsável por seu desenvolvimento.	<i>Gerente de projeto</i> : Responsável por cadastrar a tarefa e seu responsável.
<i>Em progresso</i> : Tarefas já iniciadas pela primeira vez, em correção de erros preliminares detectados pelo testador.	<i>Desenvolvedor</i> : Responsável por desenvolver os códigos necessários à implementação da tarefa.
<i>Em teste</i> : Tarefas com desenvolvimento concluído, prontas para o início da fase de testes.	<i>Testador</i> : Responsável por realizar os testes, a fim de garantir a qualidade do <i>software</i> do ponto de vista do usuário.
<i>Concluído</i> : Tarefas já desenvolvidas e testadas, prontas para ser entregues ao cliente.	<i>Gerente de projetos</i> : Nesse momento, ele apenas aguarda a conclusão das demais tarefas para então finalizar o <i>sprint</i> .
<i>Bloqueado</i> : Tarefas que por algum motivo não podem ser desenvolvidas ou dependem da conclusão de outra tarefa ou recurso, por exemplo.	O <i>gerente de projetos</i> é responsável por resolver as pendências que bloquearam a tarefa.

### 1.2.3 Gestão de Configuração

Ainda no sentido de organizar a produção, a possibilidade de que haja conflitos entre itens desenvolvidos e a necessidade de haver versões do produto com variadas configurações tornam essencial a utilização de uma ferramenta de controle de versão. Nesse sentido, o GitHub (2017), ferramenta escolhida para tal propósito, é uma plataforma de hospedagem, por meio de repositórios de arquivos, para controle de versão e colaboração (Figura 4).

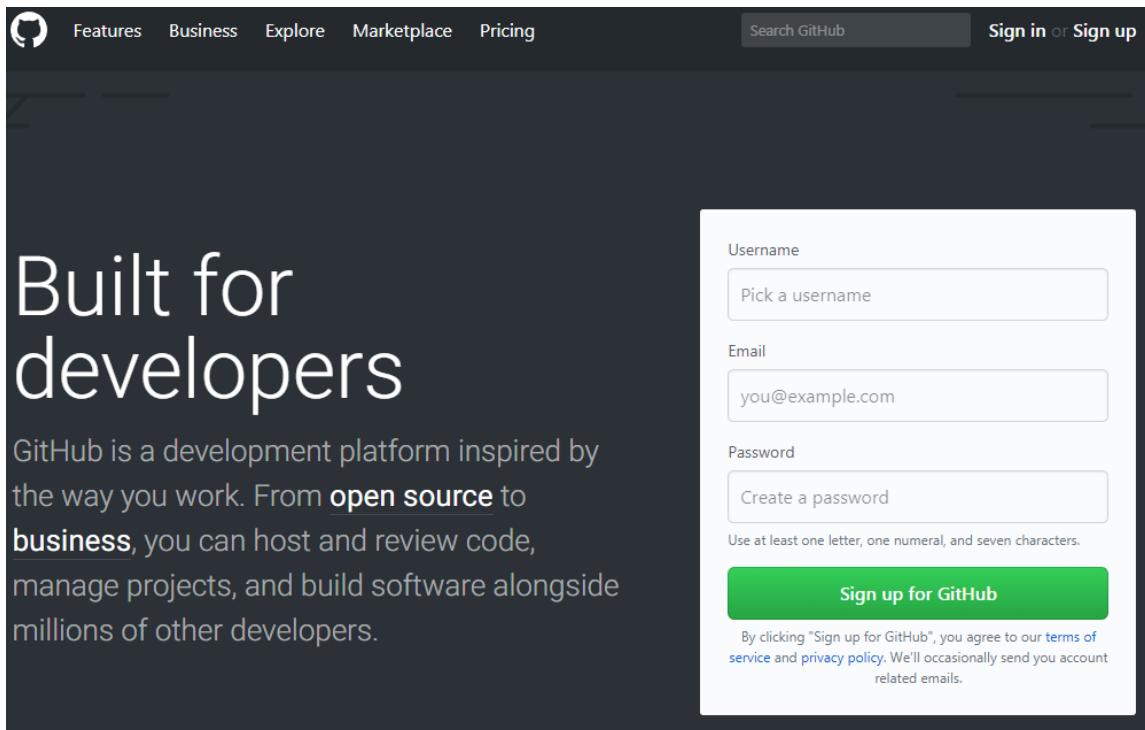


Figura 4: Tela de acesso ao GitHub.

Sua utilização permite que vários colaboradores trabalhem juntos em um projeto a partir de qualquer lugar no mundo, pois é um recurso alocado na *web*. Usando um sistema de versionamento de projeto, o GitHub é na verdade uma rede social ou uma fábrica social de *software*, que desenvolve e promove milhares de códigos fontes pré-armazenados, para as mais diversas finalidades (POZZEBOM, 2015).

Conforme Pozzebom (2015), existem dois tipos de hospedagem de repositórios no GitHub, gratuita e paga. Todos os projetos gratuitos ficam disponíveis publicamente, abertos para qualquer usuário da Internet, ao contrário da versão paga, em que os projetos não ficam disponíveis. Atualmente, utiliza-se a versão gratuita, ou seja, o código-fonte do projeto Sweca está público. Por ser um projeto desenvolvido no âmbito acadêmico, recomenda-se que seja mantida tal característica, a fim de garantir o compartilhamento dos conhecimentos. Entretanto, vê-se como necessária a implantação de uma licença, incluída em todos os arquivos de código, para garantir a autoria e ainda divulgar o nome da instituição no meio computacional.

### 1.2.3 Ambiente de Desenvolvimento Integrado (IDE)

A linguagem de programação para *web* escolhida para desenvolver o Sweca é o PHP, e o GitHub é a ferramenta para controle de versão. Havia a necessidade de definir a IDE que seria utilizada para o desenvolvimento do código. Optou-se pelo NetBeans (2017), que suporta a linguagem PHP e também tem integração nativa ao GitHub, além de ser gratuito e de código aberto para desenvolvedores de *software*. O NetBeans é executado em muitas plataformas, como

Windows, Linux, Solaris e MacOS. Ele oferece aos desenvolvedores as ferramentas necessárias para criar aplicativos profissionais de *desktop*, empresariais, *web* e móveis multiplataforma. Além disso, fornece uma base sólida para a criação de projetos e módulos, possui um grande conjunto de bibliotecas, módulos e Application Program Interface (APIs), ou seja, um conjunto de rotinas, protocolos e ferramentas para a construção de aplicativos de *software*, além de uma documentação vasta. Tais ferramentas auxiliam o desenvolvedor a criar seu *software* de maneira mais rápida (REDAÇÃO OFICINA, 2008). Ainda, como citado, o grande facilitador da ferramenta é a união com o GitHub, que permite editar códigos em diversas máquinas ao mesmo tempo, e a utilização do *plugin* PHP, conforme a Figura 5, que mostra como é feita a conexão com o repositório GitHub.

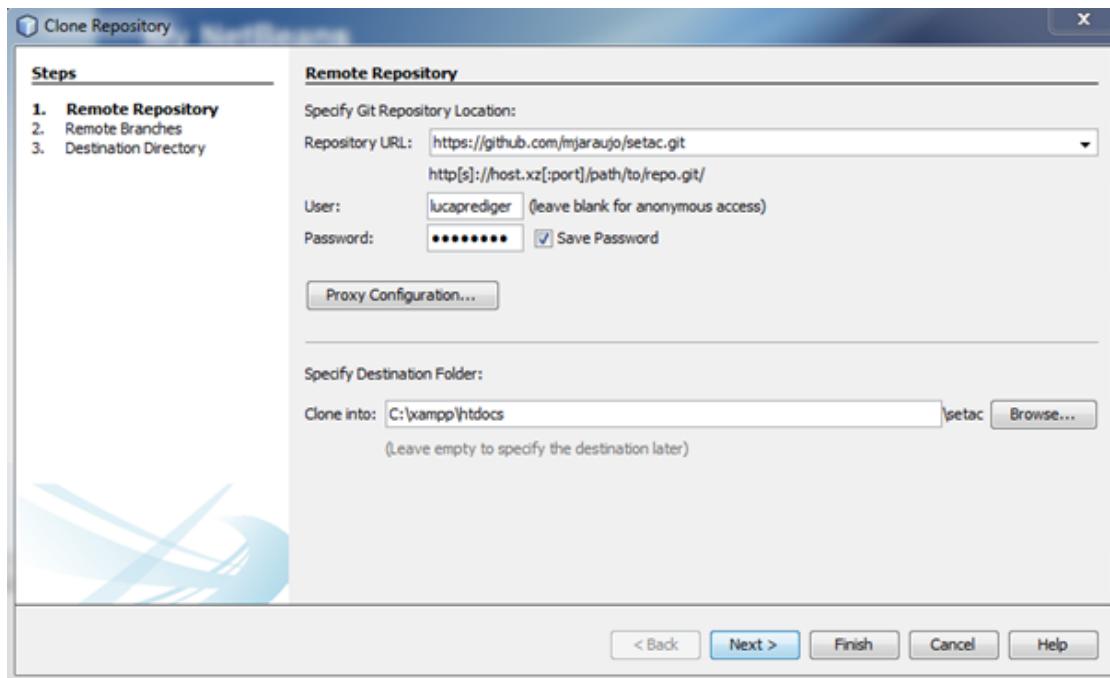


Figura 5: NetBeans e GitHub, conexão.

Após a conexão, a IDE receberá todos os arquivos que estão no GitHub. Desse modo, estará acessível para que se realizem alterações no código, como mostra a Figura 6.

```

<?php
$title = "SeTAC2 - Semana Tecnológica Acadêmica de Ciência da Computação";
require_once('view/include/inc_header.php');
?>


<section class="container" id="conteudo">
    <div class="row">
        <div class="col-md-8">
            <h2 class="topicos">
                SOBRE A SETAC2
            </h2>

            <p class="text-justify">A Semana Tecnológica Acadêmica de Ciência da Computação tem como objetivo disseminar
            <p class="text-justify">O evento busca oportunizar a vinda de palestrantes a fim de expor novas experiências,
        </div>
        <div class="col-md-4">
            <div class="lateral-box">
                <h2 class="titulo-direita">Datas Importantes</h2>
                <ul class="datas-importantes">
                    <li>Junho de 2017 <br> Início do prazo para submissão de trabalhos científicos</li>
                    <li>Setembro de 2017 <br> Fim do prazo para submissão de trabalhos científicos</li>
                </ul>
            </div>
        </div>
    </div>
</section>

```

Figura 6: NetBeans e GitHub, edição.

Desse modo, a integração do GitHub ao projeto Sweca permitiu o compartilhamento dos arquivos de código-fonte, principalmente entre os desenvolvedores, a fim de agilizar as atualizações e modificações dos arquivos. Além disso, a utilização dos recursos *git* disponíveis diretamente no NetBeans facilitou o processo de enviar e solicitar códigos ao repositório *web*. Entretanto, durante as primeiras utilizações desses recursos integrados à IDE houve alguns conflitos e até perda de códigos. Essas falhas ocorreram por falta de experiência dos desenvolvedores com o ambiente do GitHub.

### 1.3 TESTES

Segundo Costa (2013), o teste do *software* é a investigação do *software*, a fim de fornecer informações sobre sua qualidade em relação ao contexto em que ele deve operar. Isso inclui o processo de utilizar o produto para encontrar seus defeitos. Testes são necessários para que falhas sejam detectadas antes que o *software* seja colocado no mercado. Dessa forma, custos com suporte e correção de erros são reduzidos, pois são tratados antes que o produto chegue a seu consumidor final. A partir disso, os testes realizados no Sweca são efetuados pelos desenvolvedores e testadores, para que não ocorra falhas quando o *software* for entregue ao mercado.

Os testes de *software* podem ser divididos em dois grupos, que têm características e focos diferentes: testes de caixa branca e testes de caixa preta. Como o Sweca ainda se encontra em fase de planejamento e iniciação, não há um testador específico, entretanto, conforme o andamento do projeto, haverá a necessidade de que os testes sejam realizados por determinadas pessoas da equipe, que focalizem seu trabalho nos testes, para que eles mostrem resultados satisfatórios, conforme determinam as metodologias do projeto.

#### 1.3.1 Teste de caixa branca

O analista tem acesso ao código-fonte, conhece a estrutura interna do produto analisado. Isso possibilita que sejam escolhidas partes específicas de um componente para ser avaliadas. Essa metodologia de teste é usada com maior frequência durante o desenvolvimento do Sweca.

Os analistas do Sweca têm acesso ao código-fonte e podem assim tratar erros com maior rapidez, pois têm permissão para fazer as alterações necessárias, a fim de corrigir o problema.

Esse tipo de teste, também conhecido como teste estrutural, é projetado em função da estrutura do componente e permite averiguação mais precisa do comportamento dessa estrutura. O acesso ao código facilita o isolamento de uma função ou ação do *software*, o que ajuda na análise comportamental delas (COSTA, 2013).

### 1.3.2 Teste de caixa preta

O analista não tem acesso ao código-fonte e desconhece a estrutura interna do sistema. É também conhecido como teste funcional, pois é baseado nos requisitos funcionais do *software*. Como exemplo, tem-se o teste de uma calculadora, em que o analista só tem acesso à interface gráfica. Desse modo, ele valida os campos (conforme especificações do projeto) e as funções matemáticas e relata ao desenvolvedor algum erro encontrado ou sugestão que tenha.

O foco, nesse caso, é nos requisitos da aplicação, ou seja, nas ações que ela deve desempenhar. Por isso, sem ter acesso ao código-fonte do *software*, somente à interface de saída e seus resultados, o analista valida campos e resultados de funções matemáticas (COSTA, 2013).

### 1.3.3 Testes no Sweca

Somente testes de caixa branca foram sucintamente realizados no Sweca, porque o projeto está em fase inicial de desenvolvimento, e apenas desenvolvedores realizaram testes a fim de verificar a funcionalidade de suas próprias implementações. Todavia, há previsão de gerar um plano de testes para garantir a qualidade do sistema.

## 1.4 QUALIDADE DE SOFTWARE APLICADA AO PROJETO

A Figura 7 apresenta um planejamento das ações de cada colaborador, ou seja, de como cada um utiliza as ferramentas de controle de qualidade de *software*. Trata-se de um fluxograma com a ordem natural das atividades executadas pelos colaboradores do projeto Sweca. Nele, cada ação começa com um número que representa a ordem sequencial crescente de cada atividade desenvolvida pelos 3 tipos de colaborador: gerente de projeto, desenvolvedor e testador.

## 2 RESULTADOS E DISCUSSÃO

Como alguns métodos implantados, como o desenvolvimento ágil, são relativamente novos no mercado, pois surgiram há pouco tempo e estão sendo aplicados às empresas de forma gradativa, alguns colaboradores não os conheciam, e dessa forma tiveram de se adaptar às novas ferramentas. Assim, constatou-se a necessidade de treinamento, o que é comum no meio computacional. O ser humano tende a ter certa insegurança perante as novidades do mercado. Desse modo, houve certo receio de se incorporarem novos métodos ao trabalho, mas, superado esse sentimento, o resultado da implantação dos métodos foi maior rendimento e redução de conflitos entre versões, proporcionados pela utilização de ferramentas como o GitHub e o SeeNowDo. Nas ações seguintes, a equipe teve mais segurança na aplicação dos métodos ágeis.

As tecnologias que foram aplicadas até agora se mostraram satisfatórias. O controle de versionamento GitHub proporcionou à equipe de desenvolvimento o controle das versões do sistema, tratando os conflitos de código, acelerando o andamento do projeto e auxiliando na programação em equipe. Com a integração entre Netbeans e GitHub, houve aumento na produção e na qualidade do *software* produzido, o que proporcionou mais tempo para a equipe tratar e remover possíveis falhas do sistema.

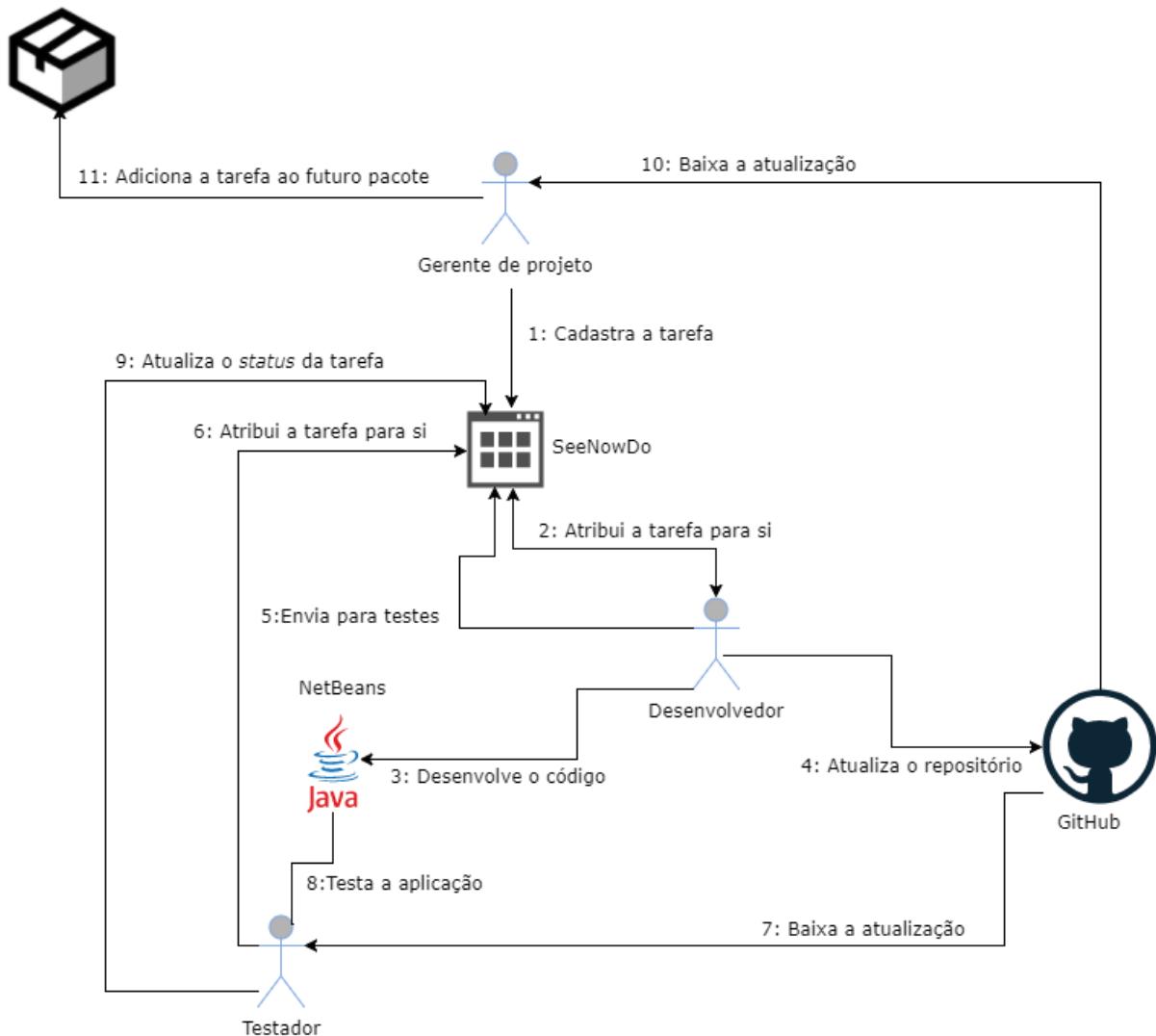


Figura 7: Fluxograma das tarefas executadas por cada tipo de colaborador no Sweca.

A ferramenta SeeNowDo teve papel importante na distribuição das atividades e no tratamento dos conflitos relacionados às atividades desenvolvidas durante o desenvolvimento do *software*. Em resumo, as tecnologias e métodos utilizados até aqui ajudaram na organização do trabalho, seja no controle das tarefas, no acesso ao que estava sendo desenvolvido, no acompanhamento dos prazos – ainda, permitem que novos membros que entram na equipe se atualizem em relação ao trabalho até então realizado, o que agiliza a integração deles ao projeto, facilita a nova fase de trabalho ou a continuidade de um trabalho anterior.

O código-fonte do *software* está disponível em: <<https://github.com/mjaraudo/setac>>, de forma gratuita. Novos membros estão se integrando à equipe e outros métodos e testes devem ser implementados e conduzidos nas próximas versões do sistema.

## CONSIDERAÇÕES FINAIS

Este trabalho mostrou, com resultados satisfatórios, um planejamento e a fase inicial de implantação de tecnologias e metodologias ágeis ao projeto de um sistema para controle de eventos acadêmicos, o Sweca. Por ser um projeto desenvolvido no âmbito acadêmico, recomenda-se que seja mantida a versão pública do Sweca no GitHub, a fim de garantir o compartilhamento dos conhecimentos. Entretanto, vê-se como necessária a implantação de uma licença, incluída em todos os arquivos de código, a fim de garantir a autoria do projeto.

Considera-se relevante a existência de um documento que norteie novos colaboradores sobre as metodologias e boas práticas de desenvolvimento de *software*, seguindo técnicas de qualidade de *software*. Dessa forma, é possível garantir um produto que atenda adequadamente aos propósitos do escopo do projeto.

Para um aprofundamento deste projeto, sugere-se a implementação de ferramentas de testes automatizados como: JMeter para testes de estresse e desempenho e Selenium-IDE para criação de interface gráficas *web* para testes, para que haja testes mais concisos e rápidos.

## REFERÊNCIAS

- COSTA, P. **Conceitos:** testes de caixa branca e caixa preta. Disponível em: <<http://www.base2.com.br/2013/07/24/teste-caixa-branca-caixa-preta/>>. Jul. 2013. Acesso em: jun. 2017.
- GITHUB. **Built for developers.** 2017. Disponível em: <<http://www.github.com>>. Acesso em: set. 2017.
- LESLIE, J. **Agile Project Management Software User Report.** 2015. Disponível em: <<http://www.softwareadvice.com/resources/agile-project-management-user-trends-2015/>>. Acesso em: jun. 2017.
- MCLAUGHLIN, M. **What is Agile Methodology.** Disponível em: <<https://www.versionone.com/agile-101/agile-methodologies/>>. Acesso em: jun. 2017.
- NETBEANS. **Netbeans IDE.** 2017. Disponível em: <<https://netbeans.org/index.html>>. Acesso em: set. 2017.
- POLO, D. J. **Seenowdo – um excelente painel de trabalho para equipes.** 2010. Disponível em: <<https://br.wwwwhatsonline.com/2010/06/seenowdo-um-excelente-painel-de-trabalho-para-equipes/>>. Acesso em: jun. 2017.
- POZZEBOM, R. **O que é GitHub.** 2015. Disponível em: <<https://www.oficinadanet.com.br/post/14791-o-que-github>>. Acesso em: jun. 2017.
- PRESSMAN, R.; MAXIM, B. **Engenharia de Software.** 8.ed. Porto Alegre: McGraw Hill, 2016.
- RAMANNA, S.; PETERS, J. F.; AHN, T. Software quality knowledge Discovery: a rough set approach. In: PROCEEDINGS OF THE 26TH ANUAL INTERNATIONAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE, p.1140-1145, 2002.
- REDAÇÃO OFICINA. **O que é NetBeans.** 2008. Disponível em: <[https://www.oficinadanet.com.br/artigo/1061/o\\_que\\_e\\_o\\_netbeans](https://www.oficinadanet.com.br/artigo/1061/o_que_e_o_netbeans)>. Acesso em: jun. 2017.
- SANDERS, J.; CURRAN, E. **Software quality:** a framework for success in software development and support. Addison Boston, MA: Wesley Longman, 1994.
- SEENOWDO. **When all you need is a shared taskboard.** 2017. Disponível em: <<https://www.seenowdo.com>>. Acesso em: set. 2017.
- SEMANA TECNOLÓGICA ACADÊMICA DE CIÊNCIA DA COMPUTAÇÃO – SeTAC<sup>2</sup>. UTFPR, Campus Santa Helena: Santa Helena, PR: 2017. Disponível em: <<http://sh.utfpr.edu.br/setac>>. Acesso em: set. 2017.
- SIGNIFICADOS. **Kanban.** Disponível em: <<https://www.significados.com.br/kanban/>>. Acesso em: set. 2017.
- SISTEMA WEB PARA EVENTOS ACADÊMICOS – SWECA. 2017. Disponível em: <<https://github.com/mjaraujo/setac>>. Acesso em: set. 2017.

# Construção de um motor gráfico com base em um modelo formal simples

## A game engine based on a simple formal model

Rafael Campos Nunes<sup>1</sup>, Davi Marcondes Rocha<sup>2</sup>

Ciência da Computação, Universidade Tecnológica Federal do Paraná (UTFPR-SH)

<sup>1</sup>[rafaelnunes@alunos.utfpr.edu.br](mailto:rafaelnunes@alunos.utfpr.edu.br), <sup>2</sup>[davirocha@utfpr.edu.br](mailto:davirocha@utfpr.edu.br)

### RESUMO

O objetivo deste trabalho é apresentar um novo modelo conceitual simples de um motor gráfico dentre os presentes na indústria de jogos, como a Unity3D e UnrealEngine. No desenvolvimento do modelo, foram utilizadas bibliotecas para abstrair diferenças entre sistemas operacionais, como a SDL2, e também tecnologias para acesso ao dispositivo de computação gráfica OpenGL. Por fim, demonstra-se que é possível implementar o modelo conceitual de motor gráfico (o renderizador) e também expandi-lo de acordo com a necessidade de projeto.

**Palavras-chave:** Motor gráfico; Renderizador; Modelo de motores; Jogos.

### ABSTRACT

*The goal of this work is to present a simple model to a graphical engine in contrast with others present in the game industry like Unity3D and UnrealEngine. For the development were used libraries to abstract differences amongst operational systems like the SDL2, also technologies to direct access to the graphics processing unit: OpenGL. At last, it was shown that is possible to implement a basic model of a graphical engine (the renderer) and also expand it according to the needs of the project.*

**Keywords:** Game engines; Renderers; Engine models; Games.

### Referência:

NUNES, R. C.; ROCHA, D. M. Construção de um motor gráfico com base em um modelo formal simples. In: SEMANA TECNOLÓGICA ACADÉMICA DE CIÊNCIA DA COMPUTAÇÃO DA UTFPR-SH, 4, 2017, Santa Helena, PR. Anais..., Santa Helena: UTFPR, 2017. p.72-81. Disponível em: <<http://sh.utfpr.edu.br/setac/Anais2017.pdf>>.

### INTRODUÇÃO

Atualmente, emprega-se a expressão “motor de jogo” ou “motor gráfico” para designar uma ferramenta cujo propósito é facilitar o desenvolvimento de jogos ou simulações, por ela oferecer abstração tanto sobre *hardware* como sobre *software*.

A construção de tal ferramenta está inerente ao propósito de o que se deseja obter. Pode-se adotar por exemplo, na construção do motor, um renderizador de projeção isométrica, de perspectiva ou ortogonal. Além disso, outras características podem influenciar o desenvolvimento do motor, como o estilo de arte que se deseja adotar. Para alguns desenvolvedores, deve-se adotar o fotorrealismo, outros adotam uma arte mais cartunista. Outras abstrações da realidade podem ser implementadas, como áudio e sistemas que imitam inteligência.

O presente artigo traz perspectivas de diversos autores sobre a terminologia das palavras motor gráfico. Também apresenta seus componentes básicos, suas respectivas implicações para a construção do modelo conceitual e uma abordagem para o desenvolvimento de um motor gráfico capaz de carregar objetos 3D modelados em *software* de terceiros (Blender), a fim de clarificar características de um motor gráfico.

Tais conceitos e teorias são importantes para se atingir o objetivo deste trabalho, que é demonstrar a construção de um novo modelo conceitual de motor gráfico e sua implementação, o qual utiliza C++ e algumas bibliotecas para abstrair diferenças sobre sistemas operacionais, como a SDL2, e também tecnologias para acesso ao dispositivo de computação gráfica OpenGL.

## 1 REFERENCIAL TEÓRICO-METODOLÓGICO

### 1.1 MOTOR GRÁFICO: ORIGEM E TERMINOLOGIA

Apesar do termo motor gráfico ter surgido na década de 1990, com o motor gráfico da Id Software (GREGORY, 2013), de acordo com Simpson (2002), seu significado e de seus componentes não foram estabelecidos academicamente. A terminologia exata de o que é realmente um motor gráfico é obscura. Há discussão sobre onde se encontra a divisão entre um jogo e um motor gráfico (ANDERSON et al., 2008).

Algumas descrições foram postas, como a de Macedo (2015, p.182), que define um motor gráfico como: “ferramenta cujo propósito é facilitar o desenvolvimento de jogos por oferecer abstração tanto sobre hardware como *software*”. Já Lewis e Jacobson (2002) definem um motor gráfico como uma coleção de módulos cujo conteúdo são códigos de simulação que não interferem diretamente no comportamento de um jogo (lógica do jogo).

Ainda, o termo *game engine*, em português motor gráfico ou motor de jogo, segundo Gregory (2013, p.11): “[...] arose in the mid-1990s in reference to first-person shooter (FPS) games like the insanely popular Doom by id Software”. O jogo Doom, um First Person Shooter (FPS) de ficção científica, foi um dos jogos mais influentes dessa época, porque, segundo Gregory (2013), foi arquitetado com camadas bem definidas entre o cerne do jogo e o próprio jogo. Ele conta com um sistema de renderização tridimensional, o sistema de áudio, recursos gráficos e o sistema de detecção de colisão.

No decorrer das décadas, os motores gráficos aumentaram em complexidade. Kobashikawa (2007) afirma que um motor gráfico, atualmente, é composto por diversas facetas, exigindo trabalho intenso e interação de diversas áreas do conhecimento em seu desenvolvimento. Jason Gregory mostra diversos diagramas que representam as camadas de um motor gráfico em seu livro *Game Engine Architecture* (2013). Pode-se observar, dentre os demais diagramas, o diagrama que representa a arquitetura do motor gráfico Unreal Engine (Figura 1), dividido em blocos, com cada bloco representando um módulo do motor ou partes específicas.

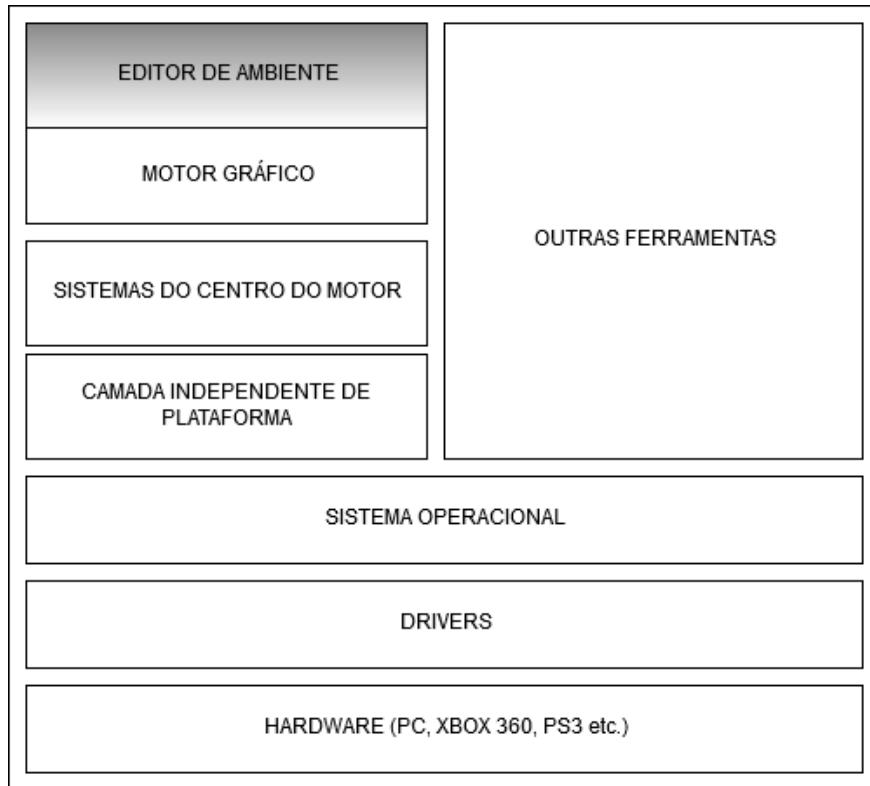


Figura 1: Arquitetura do motor gráfico Unreal.

Fonte: Gregory (2013, p.61).

### 1.1.1 Renderizador

Gregory (2013) explana que a renderização é o processo de geração de imagens a partir da interação entre objetos, luz e um ponto virtual no espaço definido como câmera virtual. De maneira mais simples, renderizar é o ato de compilar e obter o produto final de um processamento digital. Então, é quando toda a sequênciia de imagens montada é condensada em um vídeo (PROFESSOR WEB, 2013).

Kobashikawa (2007) afirma que a tarefa de um renderizador é representar objetos, partículas, sombras dinâmicas, aplicar efeitos de processamento e, além disso, prover abstração sobre como os objetos serão desenhados e quais efeitos estão sendo aplicados a cada desenho. Tudo isso descrito numa *pipeline* para gerar imagens 2D a partir de modelos 3D.

Quase tudo que acontece em uma simulação ou jogo é transmitido ao usuário por meio de imagens, salvo em casos em que a representação acústica é proeminente. Além disso, o renderizador deve ter fidelidade ao que estiver representando, se estiver tratando da realidade.

Renderizar uma cena tridimensional, segundo Gregory (2013), envolve os seguintes passos:

- Uma cena virtual é descrita em termos de superfícies 3D representadas de alguma forma matemática;
- Uma câmera virtual é posicionada e orientada para produzir a visualização desejada;
- Pontos de luz são definidos. Essas fontes proveem todos os raios de luz que irão interagir e refletir nos objetos do ambiente e, eventualmente, refletir na câmera virtual;
- Para cada pixel, o renderizador calcula cor e intensidade dos raios de luz convergindo à câmera virtual. Isso é conhecido como a solução da equação de renderização.

A renderização de primitivas (triângulos) na tela de um computador é feita a partir da comunicação com o dispositivo de vídeo, isto é, a Unidade de Processamento Gráfico (GPU). Pode-se realizar tal feito utilizando-se bibliotecas gráficas, como o DirectX no Windows, Metal no MacOS ou OpenGL, em todas as outras plataformas e também no Windows.

### 1.1.2 Nós Periféricos

Os nós periféricos, conhecidos também como componentes de um motor gráfico (GREGORY, 2013), rodeiam o cerne do motor e podem ser considerados motores, pois proveem interfaces para a realização de diversas funções. Assim como o renderizador, tais nós são complexos e não serão implementados neste trabalho. A título de explicação, um motor gráfico pode ter os seguintes nós periféricos:

- Motor de áudio;
- Motor de física/colisão;
- Sistema de estruturas de dados.

#### 1.1.2.1 Motor de áudio

Com módulo de áudio, é possível interagir com dispositivos e arquivos de áudio de um sistema operacional. A implementação de um motor de áudio implica a utilização de uma interface de programação (API), como OpenAL ou DirectSound. Motores recentes implementam técnicas avançadas de imersão, como o efeito Doppler, e com posicionamento sonoro em 3D. Tais técnicas são implementadas em motores gráficos como Unity3D, em que é possível criar diversas fontes de áudio (UNITY3D, 2017).

#### 1.1.2.2 Motor físico

A maior parte da interação entre objetos de um jogo dá-se pela colisão ou atuação de forças sobre objetos, portanto, justifica-se a necessidade de um motor físico. Ele é responsável por controlar como objetos se comportam no mundo virtual (KOBASHIKAWA, 2007).

Há uma variedade de propriedades da física do mundo real que podem ser implementadas com um motor físico, segundo Gregory (2013):

- Detectar colisões entre objetos dinâmicos e a geométrica da estática do mundo;
- Simular corpos rígidos sob a influência da gravidade e outras forças;
- Simular sistemas de massa mola (Movimento Harmônico Simples);
- Simular estruturas e edificações destrutíveis;
- Simular veículos dirigíveis realistas.

Certamente, há ressalvas quanto ao que implementar. Simulações físicas demandam processamento exaustivo tanto da unidade de processamento central (CPU) como da unidade de processamento gráfica (GPU).

#### 1.1.2.3 Sistema de estrutura de dados

A categorização e organização de objetos do mundo virtual é feita utilizando estruturas de dados. Por exemplo, há o processo de *culling* e *clipping* que é o método de escolher e seccionar objetos que são desenhados na visão da câmera. Jogos como Doom utilizam um algoritmo de particionamento de espaço binário conhecido como Binary Space Partitioning (BSP) para escolher rapidamente o que será renderizado (SIMPSON, 2002).

## 1.2 METODOLOGIA

Para a construção do motor gráfico conceitual representado na Figura 2, o qual foi criado pelos autores e é descrito neste trabalho, utilizou-se como base o modelo Data Driven Architecture (DDA), cuja *arquitetura de software* é moldada a partir dos dados que o motor gráfico controlará. Nesse caso, as informações que se abstraem dos nós do modelo são o renderizador, o interpretador de eventos, a biblioteca gráfica OpenGL (2017) e o carregador de objetos.

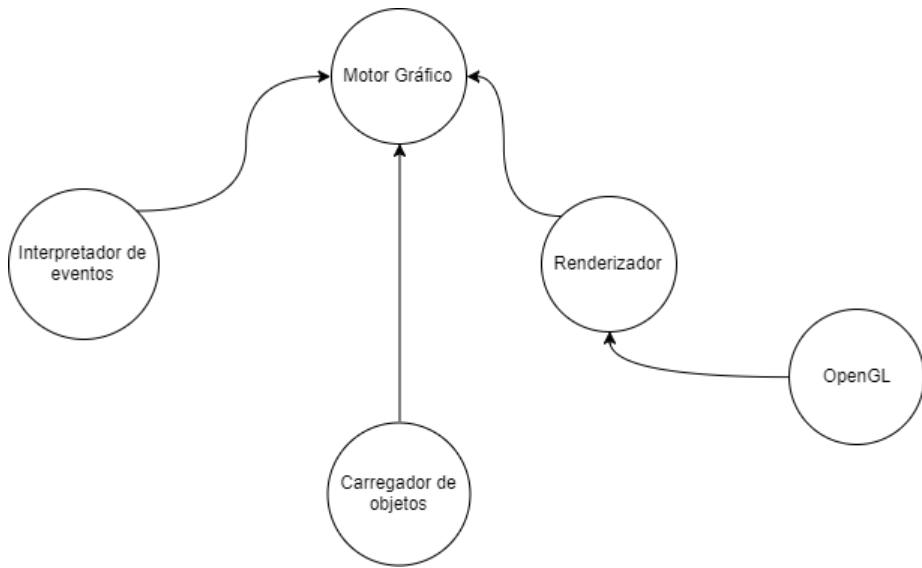


Figura 2: Modelo de motor gráfico conceitual

Os componentes do motor gráfico foram construídos utilizando a linguagem de programação C++. Além disso, são utilizadas bibliotecas para abstrair características do sistema operacional (SDL2) e dar possibilidade de executar o programa em diversas plataformas, como Linux, Windows e MacOS.

### 1.2.1 O Programa Principal

Ao executar o motor gráfico, o programa principal se encarrega de inicializar os sistemas necessários, como o sistema de vídeo sobre o sistema operacional utilizando SDL2 e o contexto OpenGL. Após isso, o objeto é importado e o laço principal entra em execução.

O programa principal é constituído por um laço principal que executa indefinitivamente várias instruções, a menos que o laço seja terminado explicitamente pelo usuário ou sistema operacional. Esse laço será responsável por fazer as chamadas à função de renderização e às funções de interpretação de eventos, como uma tecla pressionada ou o redimensionamento da tela.

O programa criado pode ser descrito pelo seguinte pseudocódigo:

```
1  sdl2.inicializa_sistema_video()
2  sdl2.inicializa_contexto_opengl()
3  objeto = assimp.carrega_objeto()
4  renderizador = sdl2.contexto_opengl()
5  enquanto programa.rodando
6      renderizador.renderiza(objeto)
7      interpreta_eventos()
8  fim enquanto
```

### 1.2.2 O Renderizador

O renderizador é responsável por desenhar objetos repassados a ele via carregador de objetos e posteriormente os enviar aos *buffers* da OpenGL. Além disso, é responsável por abstrair as funções primitivas relacionadas à OpenGL, como a rotação utilizada no programa.

Como citado, um renderizador contém estágios de execução para o desenho de uma imagem. Processos de *culling* e *clipping* não são abordados, já que o objeto está sempre posicionado de forma que esteja contido no *frustrum* da câmera, isto é, na área visível do espaço. A área visível da câmera é definida como a área do trapézio entre os planos descritos na Figura 3. Além disso, é possível observar que os únicos objetos desenhados são o objeto 2 e, parcialmente, o objeto 1.

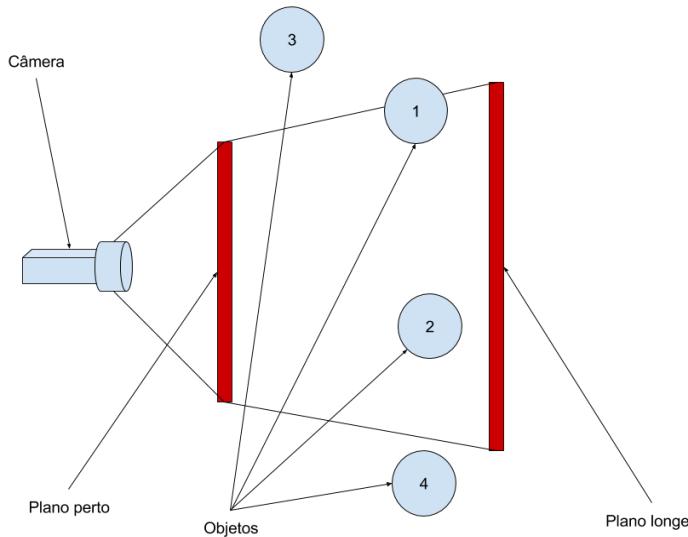


Figura 3: Câmera exibindo seu *frustrum*.

Para a implementação do renderizador, foi criada uma classe. Nela estão contidas funções para renderização de dois objetos geométricos, um triângulo e um cubo, e também para rotação desses objetos.

### 1.2.3 Carregador de Objetos

Objetos modelados em software de terceiros, como o Blender (2017), usado neste trabalho, são representados em arquivos com extensões específicas, como a extensão *.obj*. O importador é o componente do motor gráfico responsável por ler esse arquivo que contém informações sobre a posição de vértices, arestas e faces do objeto e enviar ao motor de renderização.

A importação de objetos é feita interpretando dados do arquivo *.obj*. O arquivo descreve a posição de cada vértice, a componente normal do vértice e as faces do vértice de uma maneira simples como pode ser visto abaixo:

```
1 v 0.0 0.0 0.0
2 v 1.0 0.0 0.0
3 v 0.5 1.0 0.0
4 f 1//2 2//3 3//1
```

Cada *v* do arquivo denota um vértice no espaço com suas respectivas coordenadas, e cada *f* denota uma face, sendo essa denotada pela junção entre três pontos, formando um triângulo.

Carregam-se os vértices do arquivo em um vetor, as faces em outro vetor e envia-se o objeto carregado ao renderizador.

O objeto é enviado ao renderizador utilizando-se a função *render*, que aceita como parâmetro um objeto carregado pelo programa. A função *render* é responsável por enviar à OpenGL os vértices e faces que serão renderizados, utilizando a função:

```
glBufferData(GL_ARRAY_BUFFER, object.vertices.size() * sizeof(glm::vec3),
             &object.vertices[0], GL_STATIC_DRAW);
```

O parâmetro *GL\_ARRAY\_BUFFER* e *GL\_STATIC\_DRAW* são enumeradores da OpenGL, descritos internamente como GLenum, que descrevem o que a renderização deverá fazer e como deverá fazê-lo, respectivamente. O *GL\_ARRAY\_BUFFER* descreve a qual *buffer* os dados deverão ser enviados, nesse caso, um *buffer* de *array*. O *GL\_STATIC\_DRAW* descreve como a OpenGL vai escrever ou acessar os dados enviados a ela. A primeira parte do comando STATIC significa que os dados serão modificados somente uma vez e utilizados várias vezes após isso. A segunda parte, DRAW, indica à OpenGL que os dados armazenados serão utilizados como fonte para a especificação de outros comandos de renderização.

O parâmetro *object* é o objeto carregado na inicialização do programa. O objeto abstrai algumas propriedades, como a quantidade de vértices, faces e vetores normais que um objeto contém.

### 1.3 INTERAÇÃO COM O USUÁRIO

A interação com o usuário ocorre no momento em que ele aperta uma tecla, a qual é interpretada a cada execução do laço principal. Uma estrutura *switch* atua, acionando a instrução dada pelo usuário, isto é, rotaciona o objeto.

Embora o objeto esteja no espaço ( $\mathbb{R}^3$ ), o objeto só pode ser rotacionado em duas direções distintas: a abcissa e a ordenada do plano euclidiano ( $\mathbb{R}^2$ ). A rotação do objeto no eixo x pode ser vista na Figura 4:

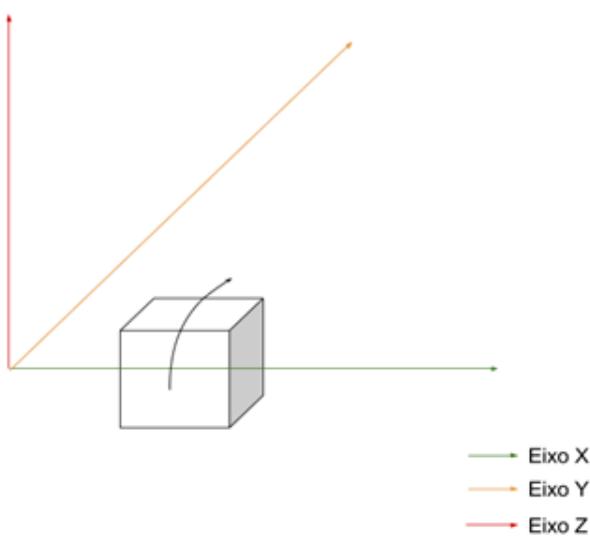


Figura 4: Rotação de um cubo no eixo x.

O algoritmo criado para rotação utiliza abstrações providas no objeto de renderização, a classe *renderer*, e pode ser representado da seguinte maneira:

```
1  tecla_pressionada = sdl.capturar_evento()
2  switch (tecla_pressionada)
3      caso seta_cima:
4          renderizador.rotacione(2, -1, 0, 0)
5      caso seta_baixo:
6          renderizador.rotacione(2, 1, 0, 0)
7      caso seta_esquerda:
8          renderizador.rotacione(2, 0, -1, 0)
9      caso seta_direita:
10         renderizador.rotacione(2, 0, 1, 0)
11     caso esc:
12         finaliza_programa()
13  fim switch
```

É válido ressaltar que a fila de eventos na biblioteca SDL2 é sincronizada. Isso quer dizer que somente uma tecla é interpretada por quadro desenhado. Para contornar esse problema, a título de conhecimento, pode ser utilizada uma fila de eventos não sincronizada, isto é, que é independente da fila de eventos da SDL2.

## 2 RESULTADOS E DISCUSSÃO

Após a descrição e elaboração dos algoritmos citados, utilizando as bibliotecas gráficas SDL2 e OpenGL, pôde-se obter um cubo sendo desenhado pelo renderizador e sendo devidamente rotacionado, à medida que o usuário pressiona teclas direcionais (Figura 5):

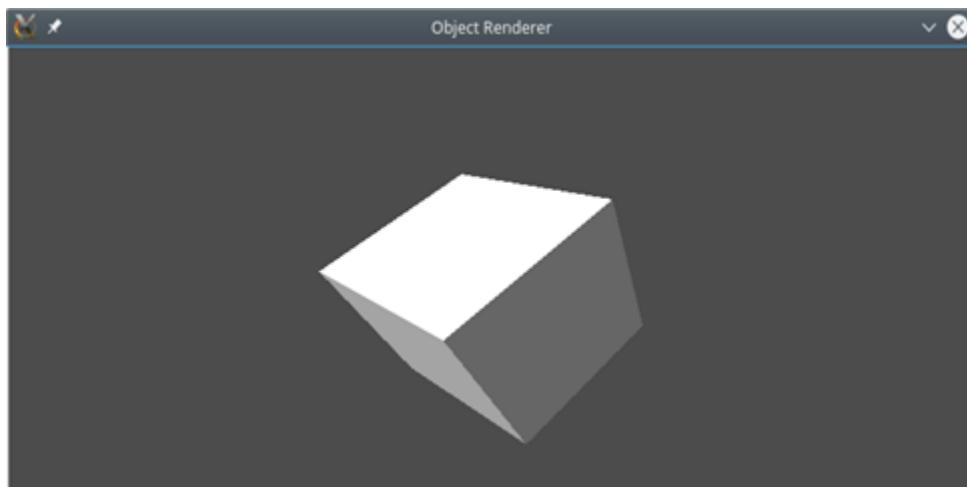


Figura 5: Um cubo sendo renderizado.

O estudo realizado mostrou que, embora o conceito de motor gráfico não esteja bem definido, é possível implementar o modelo de motor gráfico que carrega um objeto e o renderiza ao usuário.

Extensões podem ser adicionadas aos nós da Figura 2, implementando cada uma, cujo resultado seria como mostra a Figura 6. Os nós adicionados são efeitos de pós-processamento, uma implementação DirectX para quando o motor for compilado em Windows, um motor de áudio entre outros nós.

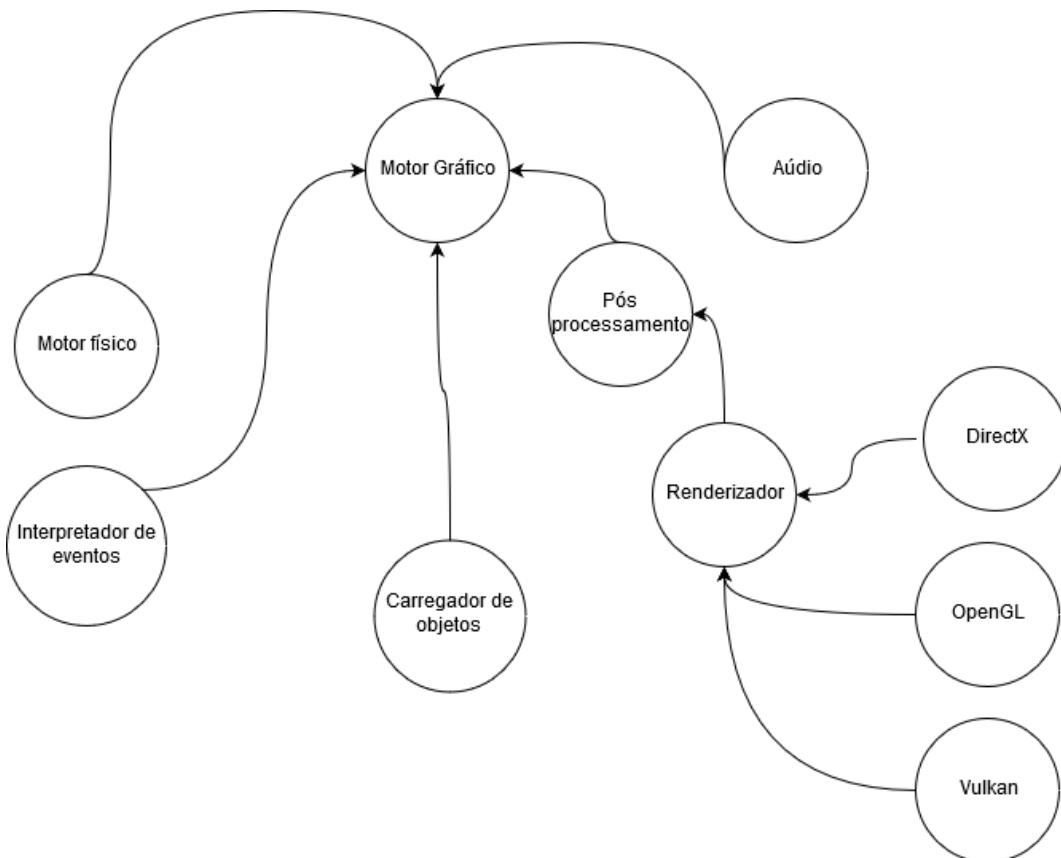


Figura 6: Modelo de motor gráfico com algumas extensões.

Esse exemplo pode ser estendido a motores de física, motores de rede e inteligências artificiais de propósito múltiplo, como classificadores e aprendizado de máquina assistido.

## CONSIDERAÇÕES FINAIS

O código-fonte contendo a implementação do motor está disponível em: <https://github.com/rafaelcn/object-renderer>. Atualmente, o motor pode ser construído, utilizando CMake (2017) como gerador de sistemas de construção. O CMake é responsável por gerar arquivos de construção como o Make (THE IEEE e THE OPEN GROUP, 2016) para compilação em duas plataformas: Linux e Windows.

## REFERÊNCIAS

ANDERSON, Eike; ENGEL, Steffen; COMNINOS, Peter; MCLOUGHLIN, Leigh. **The Case for Research in Game Engine Architecture**. In: PROCEEDINGS OF THE 2008 CONFERENCE ON FUTURE PLAY: RESEARCH, PLAY, SHARE. Toronto e Ontario, Canadá, 2008, p. 228-231.

BLENDER. **Open Source 3D creation**. Disponível em: <<http://www.blender.org/>>. Acesso em: ago. 2017

CMAKE. **Open Source tool to build and test software**. Disponível em: <<https://cmake.org/>>. Acesso em: set 2017.

GREGORY, Jason. **Game Engine Architecture**. 2.ed. New York: CRC Press, 2013.

KOBASHIKAWA, Diana. **Estudo comparativo de ferramentas para motores de jogos RPG**. Disponível em: <[http://revista.uniplac.net/ojs/index.php/tc\\_si/article/download/771/483](http://revista.uniplac.net/ojs/index.php/tc_si/article/download/771/483)>. Acesso em: set. 2017

LEWIS, Michael; JACOBSON, Jeffrey. Game engines in scientific research. **Communications**

**of the ACM**, v.45, n.1, p.27–31, 2002.

MACEDO, Daniel Valente; SERPA, Yvens Rebouças; RODRIGUES, Maria Andréia Formico. Desenvolvimento de Aplicações Gráficas Interativas com a Unreal Engine 4. **Rev. UFRGS**, Rio Grande do Sul, v.22, n.2, p.182-202, 2015. Disponível em: <<http://www.seer.ufrgs.br/rita/article/view/RITA-VOL22-NR2-181>>. Acesso em: set. 2017.

OPENGL. **OpenGL API Documentation**. Disponível em: <<http://docs.gl/>>. Acesso em: set. 2017.

PROFESSOR WEB. **O que é renderizar?** 2013. Disponível em: <<https://oprofessorweb.wordpress.com/2013/12/10/o-que-e-renderizar/>>. Acesso em: set. 2017.

SIMPSON, Jake. **Game engine anatomy 101**. 2002. Disponível em: <<https://www.cs.colorado.edu/~ralex/courses/cs4448/documents/GameEngineAnatomy101.pdf>>. Acesso em: set. 2017.

THE IEEE; THE OPEN GROUP BASE. **Make – Maintain, update, and regenerate groups of programs. The Open Group Base Specifications, Issue 7**, IEEE Std 1003.1-2008, 2016. Disponível em: <<http://pubs.opengroup.org/onlinepubs/9699919799/utilities/make.html>>. Acesso em: set. 2017.

UNITY3D. **Audio.** Unity's Audio features. Disponível em: <<https://docs.unity3d.com/Manual/Audio.html>>. Acesso em: set. 2017.

# Unidade móvel autônoma para aspirador de pó doméstico

## An autonomous mobile unit for a domestic vacuum cleaner

Gabriel Lenin Silva Lima<sup>1</sup>, Geovani José da Silva<sup>2</sup>, Willian de Oliveira Silva<sup>3</sup>, Itamar Iliuk<sup>4</sup>

Ciência da Computação, Universidade Tecnológica Federal do Paraná (UTFPR-SH)  
*gabriellsl2009@hotmail.com<sup>1</sup>, geovani-lol@hotmail.com<sup>2</sup>, will.oliver1994@hotmail.com<sup>3</sup>, itamariliuk@utfpr.edu.br<sup>4</sup>*

### RESUMO

O objetivo deste trabalho é apresentar uma unidade autônoma com aspirador de pó, criado a partir de um circuito com uma placa Arduino Mega 2560, com fins de automatizar o trabalho no ambiente residencial. Utilizam-se elementos como ponte H, placa de ensaio (*protoboard*), bateria, plataforma robótica e outros, para criar um circuito físico integrado, que possibilita a circulação da plataforma móvel. O sistema de sucção é elaborado por meio de *coolers*, um recipiente e um condutor para absorver resíduos. Acopla-se então a plataforma robótica ao sistema de aspiração. São feitas alterações no código, com o *software* de programação do próprio Arduino que, por meio de funções, cria uma sequência de comandos estruturados. Obteve-se a livre locomoção da plataforma sobre uma superfície plana e um processo eficiente do cálculo do servomotor para medir a distância entre a plataforma e um objeto.

**Palavra-chave:** Unidade autônoma; Arduino; aspirador de pó; robôs.

### ABSTRACT

*The aim of this study is expose an autonomous cart with vacuum cleaner, created from a circuit using an Arduino Mega 2050 board, with the purpose of automating the work in residential environment. Using elements how H bridge, protoboard, battery, robotic platform and others, to create an integrated physical circuit, making possible the circulation of the mobile platform. The suction system is elaborate with coolers, a container and a conductor to absorb waste. Merge the robotic platform with the suction system. Are made alterations in the code, using the software of programming of Arduino, that through functions, create a structured commands sequence. The free locomotion of the platform was obtained on a flat surface and an efficient process of calculating the servomotor to measure the distance between the platform and an object.*

**Keywords:** Autonomous Unit; Arduino; vacuum cleaner; robots.

### Referência:

LIMA, G. L. S.; SILVA, G. J.; SILVA, W. O.; ILIUK, I. Unidade móvel autônoma para aspirador de pó doméstico. In: SEMANA TECNOLÓGICA ACADÉMICA DE CIÊNCIA DA COMPUTAÇÃO DA UTFPR-SH, 4., 2017, Santa Helena, PR. *Anais...*, Santa Helena: UTFPR, 2017. p.82-99. Disponível em: <<http://sh.utfpr.edu.br/setac/Anais2017.pdf>>.

### INTRODUÇÃO

O processo de locomoção autônoma agregado à limpeza influencia a rapidez e eficácia da realização do trabalho doméstico. Nessa linha, o avanço tecnológico é um aspecto inerente a um mercado competitivo, pois trabalhos que consistem em repetir mecanicamente a mesma tarefa vêm sendo abolidos e substituídos por outros, relacionados à tecnologia e à Internet.

Robôs são geralmente integrados num sistema que, como um todo, é projetado para executar uma tarefa ou fazer uma operação. Eles são usados em conjunto com outros

dispositivos, periféricos e outras máquinas de fabricação, quando isso não ocorre, têm sua utilidade reduzida (NIKU, 2017).

Considerando-se essa característica do mercado atual de economizar trabalho, percebe-se que ao utilizar dispositivos mecânicos e eletrônicos, é possível construir uma tecnologia projetada para suprir necessidades diárias, como aspirar um local residencial utilizando um robô.

A partir disso, o objetivo deste trabalho é apresentar uma unidade móvel autônoma com aspirador de pó, construída com um circuito que utiliza uma placa Arduino Mega 2560. A função de mantê-la trabalhando na área doméstica por determinado período é feita seguindo-se uma sequência de códigos, de modo que não é necessária a interferência do cliente para definir o caminho a percorrer.

## 1 REFERENCIAL TEÓRICO

Levando-se em consideração os trabalhos desenvolvidos por França (2016) e Storch (2016), foram levantados alguns tópicos para revisão de literatura deste trabalho, como: placa Arduino, controle autônomo, robô, Internet, controle remoto via *bluetooth*, ponte H, sensor de ultrassom, servomotor.

França (2016) e Storch (2016) relatam que é possível, utilizando uma plataforma de esteira ou com eixo e um Arduino Mega 2560, construir uma unidade móvel totalmente controlável por meio do celular, permitindo a interação com o usuário. A partir desse conhecimento, optou-se pela construção de um robô autônomo. Nesse sentido, Niku (2017, p.21) pontua que:

O Roomba™, um robô aspirador de pó, comercialmente disponível há anos, se move de forma autônoma e aleatória sobre a área e aspira a poeira. Ele também encontra o seu próprio local de armazenamento para carregar. Toda sua inteligência é baseada em algumas regras simples: mover-se aleatoriamente, virar à esquerda ou à direita quando atingir um obstáculo, recuar e girar quando estiver em um canto, e encontrar seu local de armazenamento.

Sobre a Internet, Evans (2011) lembra que ela já existe em locais antes inalcançáveis, e com esse alcance a humanidade pode ter um futuro próspero, no qual pessoas se tornem administradoras de seus próprios negócios e recursos, com conhecimento para combinar diversas tarefas, como coletar, analisar, transmitir e distribuir dados em larga escala. O autor destaca a comunicação como essencial para a globalização, e que poderá ser feita entre os objetos mais comuns do cotidiano. Nesse sentido, relata que a Internet das Coisas (Internet of Things – IoT) é a primeira evolução real da Internet, a qual permite demonstrar que a tecnologia afeta diretamente a vida das pessoas, tornando-as mais proativas e menos reativas.

McRoberts (2011) descreve o Arduino como uma ferramenta simples, sem a necessidade de investimento de alto custo, que pode ser usada de distintas maneiras. Serve de entretenimento e base para iniciantes da eletrônica e programação. Os códigos e diagramas compartilhados pela comunidade, que está apta a auxiliar outros desenvolvedores, podem ser modificados ou copiados, o que implica melhor método de aprendizado. Com o Arduino se podem desenvolver objetos interativos independentes, que sejam conectados a um computador ou a uma rede e que recuperem e enviem dados para a Internet.

Segundo Souza (2014) a placa Arduino Mega 2560 é uma placa da plataforma Arduino, que possui recursos para prototipagem e projetos mais elaborados. Ela é baseada no microcontrolador ATmega2560, que possui 54 pinos de entradas e saídas digitais, dentre os quais 15 podem ser utilizados como saída PWM.

PWM, do inglês Pulse Width Modulation, é uma técnica:

[...] utilizada por sistemas digitais para variação do valor médio de uma forma de onda periódica. A técnica consiste em manter a frequência de uma onda quadrada fixa e variar o tempo que o sinal fica em nível lógico alto. [...] exemplos simples do uso do PWM [...] controle de velocidade de motores elétricos, variação de luminosidade de leds, geração de sinais analógicos, geração de sinais de áudio e alguns outros (ARRUDA; PRADO; DELATORE, 2017).

Patsko (2006) relata que a ponte H é um circuito utilizado para controlar um motor DC a partir de sinais gerados por um microcontrolador. Devido à disposição de seus componentes, torna-se extremamente fácil selecionar o sentido da rotação de um motor, apenas invertendo a polaridade sobre seus terminais.

Wendling (2010) retrata que o sensor de ultrassom é um tipo de sensor muito útil na detecção de objetos a uma certa distância, desde que estes não sejam muito pequenos, e capazes de refletir esse tipo de radiação. O princípio de funcionamento desse sensor é o seguinte: um oscilador emite ondas ultrassônicas (em torno de 42kHz), que resultam em um comprimento de onda na ordem de alguns centímetros, o que permite detectar objetos relativamente pequenos.

Ottoboni (2002) descreve o servomotor como uma máquina síncrona composta por uma parte fixa (o estator) e outra móvel (o rotor), com seis polos no estator de alimentação trifásica, com ímãs permanentes dispostos linearmente sobre a face do rotor e um sensor analógico chamado resólver para realimentação de posicionamento e velocidade.

## 2 METODOLOGIA

Para desenvolvimento da unidade autônoma com base no referencial teórico, é preciso uma fonte de energia para alimentar o circuito. Para isso, é utilizada uma bateria 12V, pelo fato de ter maior duração e ser de simples manutenção. Para acoplá-la à base móvel Rover 5, utilizam-se cintas Helleman (Figura 1). A *protoboard* é utilizada para facilitar o processo de entendimento da parte lógica do circuito da unidade móvel. A partir dela, é possível conectar a placa Arduino, utilizada por sua fácil manipulação e maior disposição de saídas, e a ponte H (Figura 2), por meio de fios condutores. Esses componentes, responsáveis por fazer o controle e movimento da plataforma, são fixados por meio de cintas plásticas.



Figura 1: Bateria e base robótica Rover 5 fixados por meio de cintas plásticas.

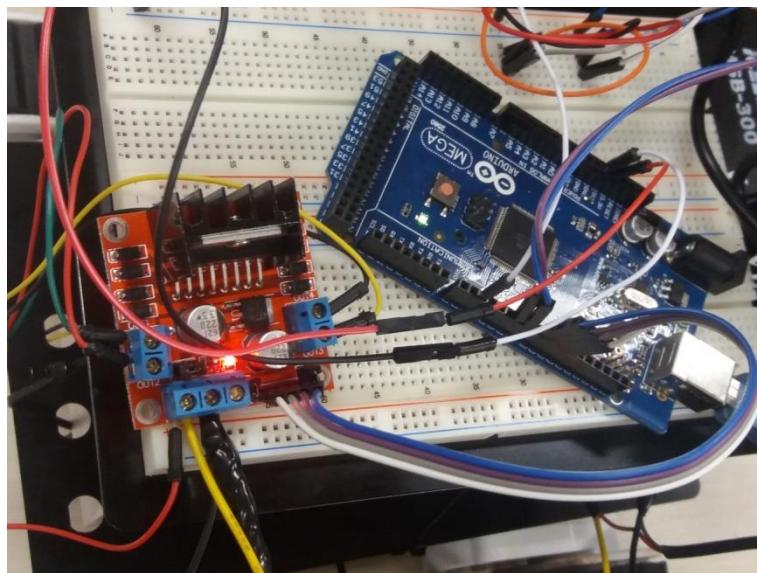


Figura 2: *Protoboard* com Arduino Mega 2560 e ponte H.

Por fim, são implementados o servomotor, utilizado para fornecer maior amplitude do campo visual, e o sensor de ultrassom (Figura 3), por ser o melhor componente disponível para a tarefa de calcular distâncias. Esses componentes são fixados por meio de cintas plásticas e conectados à placa Arduino, para executar o processo de monitoramento do ambiente. Assim é possível chegar à implementação final do circuito e dos componentes à plataforma móvel (Figura 4).

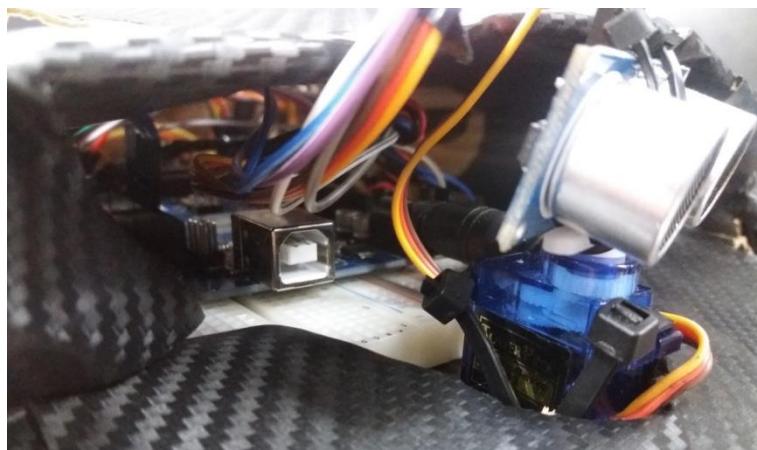


Figura 3: Servomotor e sensor de ultrassom conectados ao Arduino Mega 2560.

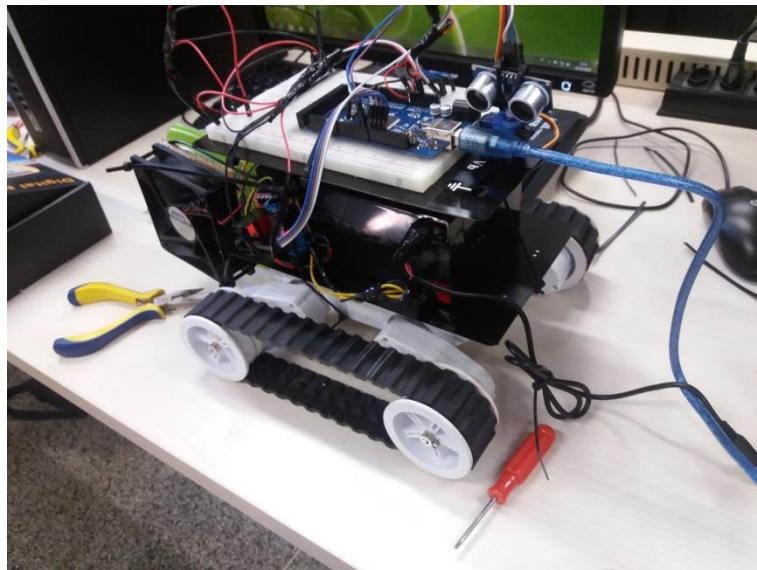


Figura 4: Componentes integrados à unidade móvel.

No desenvolvimento do aspirador, baseado em Fellipe (2014 e 2015), é escolhido um recipiente de médio porte (Figura 5) e recortada sua tampa e parte inferior. No local, é colada uma fina camada de esponja com cola quente, para evitar a saída de detritos sugados pelos *coolers*. É feito um orifício em uma das laterais do recipiente, onde é fixado o canudo ou mangueira com cola quente, para conduzir a sujeira sugada para o recipiente (Figura 5). Nos *coolers* é colada uma borda de papelão (Figura 6), para evitar o atrito com o recipiente, não prejudicar sua rotação e não haver ruído. Os *coolers* são colados ao recipiente (Figura 7) e são alimentados diretamente por uma bateria de 12V.



Figura 5: Recipiente de plástico de médio porte com canudo.

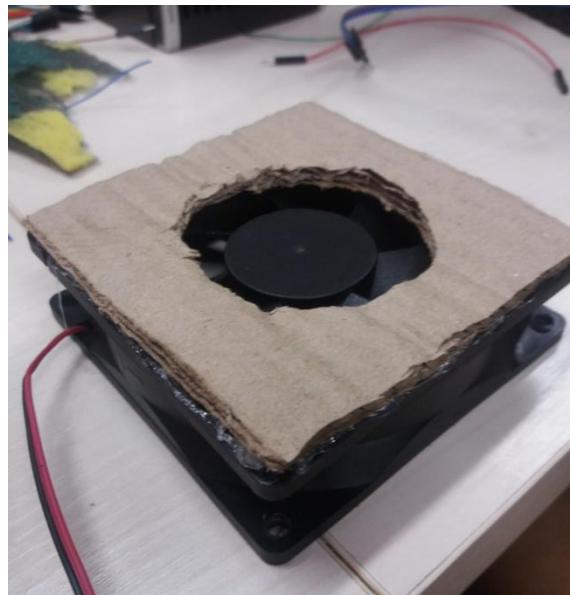


Figura 6: *Cooler* colado com papelão para evitar contato com o recipiente de plástico.



Figura 7: Recipiente colado ao *cooler*.

Com a finalidade de proteger contra impactos e dar um *design* para a unidade móvel, é projetada uma carcaça com papelão recortado e utilizado envelope automotivo de fibra de carbono para revesti-la (Figura 8). Por fim, para unir o aspirador à plataforma móvel, são utilizadas cintas Helleman fixadas na parte lateral da carcaça (Figura 9).

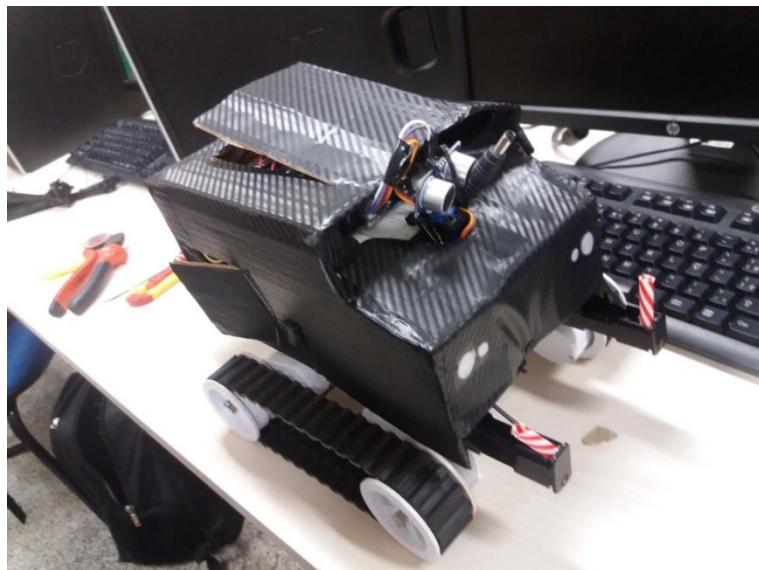


Figura 8: Carcaça de papelão e envelope automotivo de fibra de carbono.

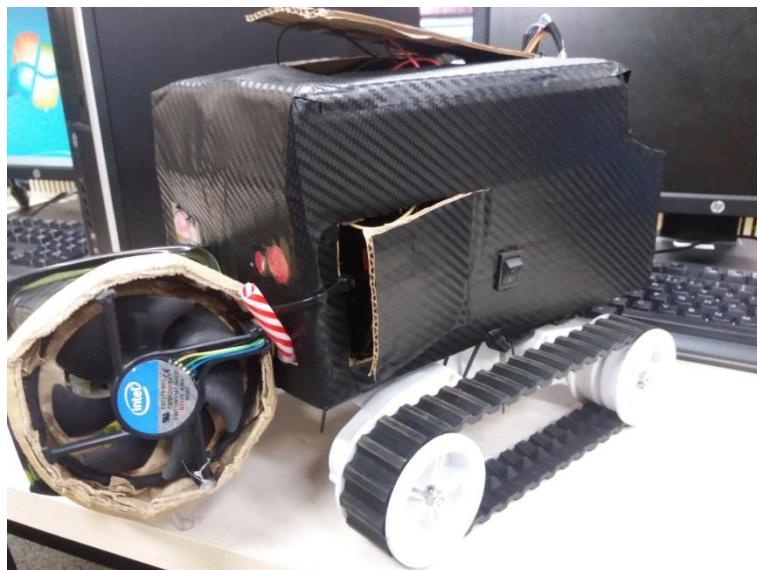


Figura 9: Modelo final da unidade móvel com todos os componentes conectados.

Os materiais utilizados para fazer o circuito estão expostos na Tabela 1.

Tabela 1: Lista de materiais.

Quantidade	Material	Fonte
1	Base robótica Rover 5	Figura 1
1	Bateria 12v	Figura 1
1	Protoboard	Figura 2
1	Ponte H	Figura 2
35	Jumper (cabos)	Figura 2
1	Servomotor	Figura 3
30	Cinta plástica Hellermann	Figura 1
1	Recipiente plástico médio	Figura 5
2	Cooler	Figura 6
1	Mangueira 1,5mm	Figura 9
1	Caixa de papelão	Figura 8

1	Fita isolante 33mm	-
1	Adesivo automotivo de fibra de carbono	Figura 9
1	Arduino Mega 2560	Figura 2
1	Interruptor 2 fases	Figura 9
1	Sensor de ultrassom	Figura 3

A seguir (Figura 10) é apresentada a descrição lógica do circuito, feito em uma ferramenta de modelagem:

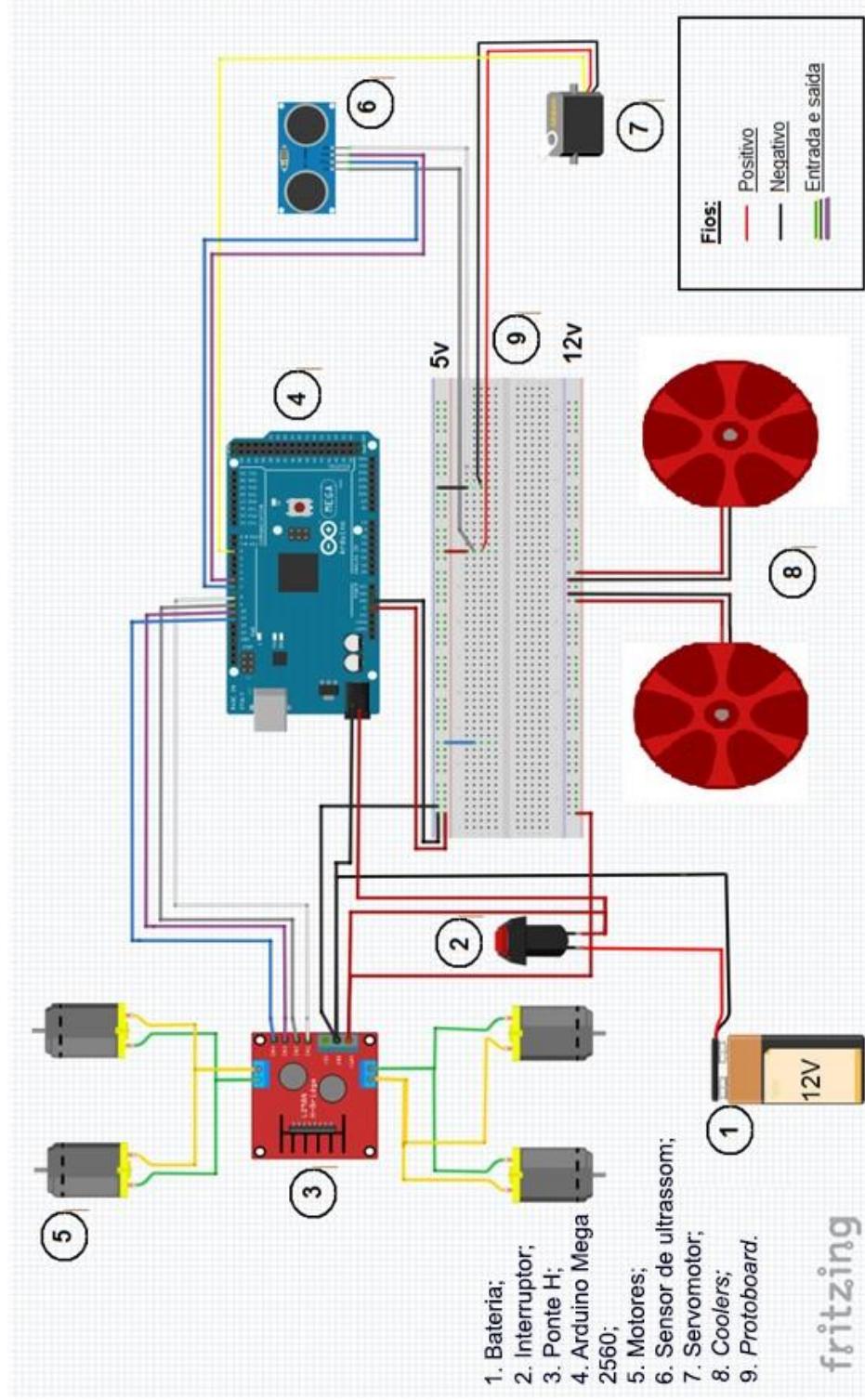


Figura 10: Simulação do circuito feita no Fritzing.

O circuito funciona da seguinte forma. O fio positivo (vermelho) da bateria (1) é ligado no interruptor e o negativo (preto) é ligado diretamente no Arduino e na ponte H.

O interruptor (2) recebe em um de seus polos o fio positivo (vermelho) da bateria e o outro repassa o sinal para o Arduino e para a ponte H, de acordo com as duas fases, uma ligada (a corrente flui normalmente) e outra desligada (a corrente é cortada e o interruptor não emite sinal).

A ponte H (3) recebe os fios negativo (preto) na entrada GND direto da bateria e o positivo (vermelho) na entrada +12V do interruptor (após o chaveamento das fases). As entradas IN1 (branco), IN2 (cinza), IN3 (roxo) e IN4 (azul) recebem os fios de sinal do Arduino para ativação das rodas da plataforma (Figura 10). Existem duas saídas que são ligadas nos motores da plataforma, positivo (verde) e negativo (amarelo), que emitem o sinal para girar seu eixo na direção indicada, ou seja, caso emita polaridade invertida, o motor funcionará de maneira inversa, girando no sentido contrário. A entrada GND tem um fio negativo (preto) para saída, que conecta a ponte H à *protoboard* para igualar as tensões do circuito.

O Arduino Mega 2560 (4) recebe os fios negativos (preto), direto da bateria, e o positivo (vermelho), do interruptor, após o chaveamento das fases, na entrada de energia com tensão de 12V. Na saída de força existem dois fios: negativo (preto) e positivo (vermelho), com tensão de 5V, que são ligados na *protoboard* para alimentar o circuito. Os pinos 2, 6 e 7 são definidos como saída, e emitem sinais ao servomotor (fio amarelo), para determinar a posição, e ao sensor de ultrassom (azul e roxo), para calcular a distância. Também são definidos como saída os pinos 8 (branco), 9 (cinza), 10 (roxo) e 11 (azul) para emitir sinal à ponte H.

Esses motores (5) recebem os fios verde (positivo) e amarelo (negativo) da saída da ponte H, e são ligados em paralelo, ou seja, dos 4 motores, foram ligados dois paralelamente, um de cada lado da plataforma, e funcionam como um motor em conjunto de cada lado.

O sensor de ultrassom (6) recebe dois fios de alimentação, positivo 5V (cinza) na entrada VCC e negativo (branco) na entrada GND da *protoboard*. Também, como entrada, recebe o sinal Echo (roxo) do Arduino para emitir o som, e tem como saída o Trig (azul), que recebe o som emitido e retorna-o ao Arduino para calcular a distância do obstáculo.

O servomotor (7) recebe três fios de entrada: positivo 5V (vermelho), negativo (preto) da *protoboard*, e o fio de controle (amarelo) para determinar os graus em que o servomotor deve girar.

Os coolers (8) recebem dois fios, positivo 12V (vermelho) e negativo (preto), que são ligados em paralelo, ou seja, os dois coolers são ligados com a mesma tensão.

A *protoboard* (9) recebe os fios positivo 5V (vermelho) e negativo (preto) do Arduino para distribuí-los pela parte superior do circuito. Além disso, recebe uma tensão de 12V (vermelho) para alimentar a parte inferior do circuito. A *protoboard* também é responsável por realizar a alimentação do servomotor (vermelho e preto) e do sensor de ultrassom (branco e cinza).

O código responsável pela locomoção da unidade móvel, disponível no Anexo 1, funciona da seguinte forma, foi baseado em Santos (2014) e SeuRobo (2013). Primeiramente, é acionado o servomotor e calculada a distância à frente, à esquerda e à direita. Caso a distância calculada seja maior do que 50 cm, a unidade continua se locomovendo, caso contrário, ela para. Ao parar, calcula-se a distância à frente, à esquerda e à direita, e atribui-se à variável “melhor opção” a direção com maior distância. Em seguida, há outra função (Figura 11), que verifica se o valor armazenado na variável “melhor opção” é maior do que 50 cm. Caso seja maior, a unidade móvel gira para o lado correspondente e locomove-se na nova direção. Caso seja menor, a unidade anda para trás e recalcula a melhor opção.

```
1 // Função para chamar outras funções e definir o que o robô fará
2 void andar(){
3     reposicionaServoSonar();
4     int distancia = lerSonar(); // Ler o sensor de distância
5     Serial.print("distancia: "); // Exibe no serial
6     Serial.println(distancia);
7     if (distancia > 50) {
8         rotacao_Frente(10);
9     }else{
10        rotacao_Parado(10);
11        posicionaCarroMelhorCaminho();
12        andar();
13    }
14 }
```

Figura 11: Código utilizado para movimentar a plataforma móvel autônoma.

O fluxograma (Figura 12) a seguir demonstra a parte lógica utilizada para a movimentação da unidade móvel.

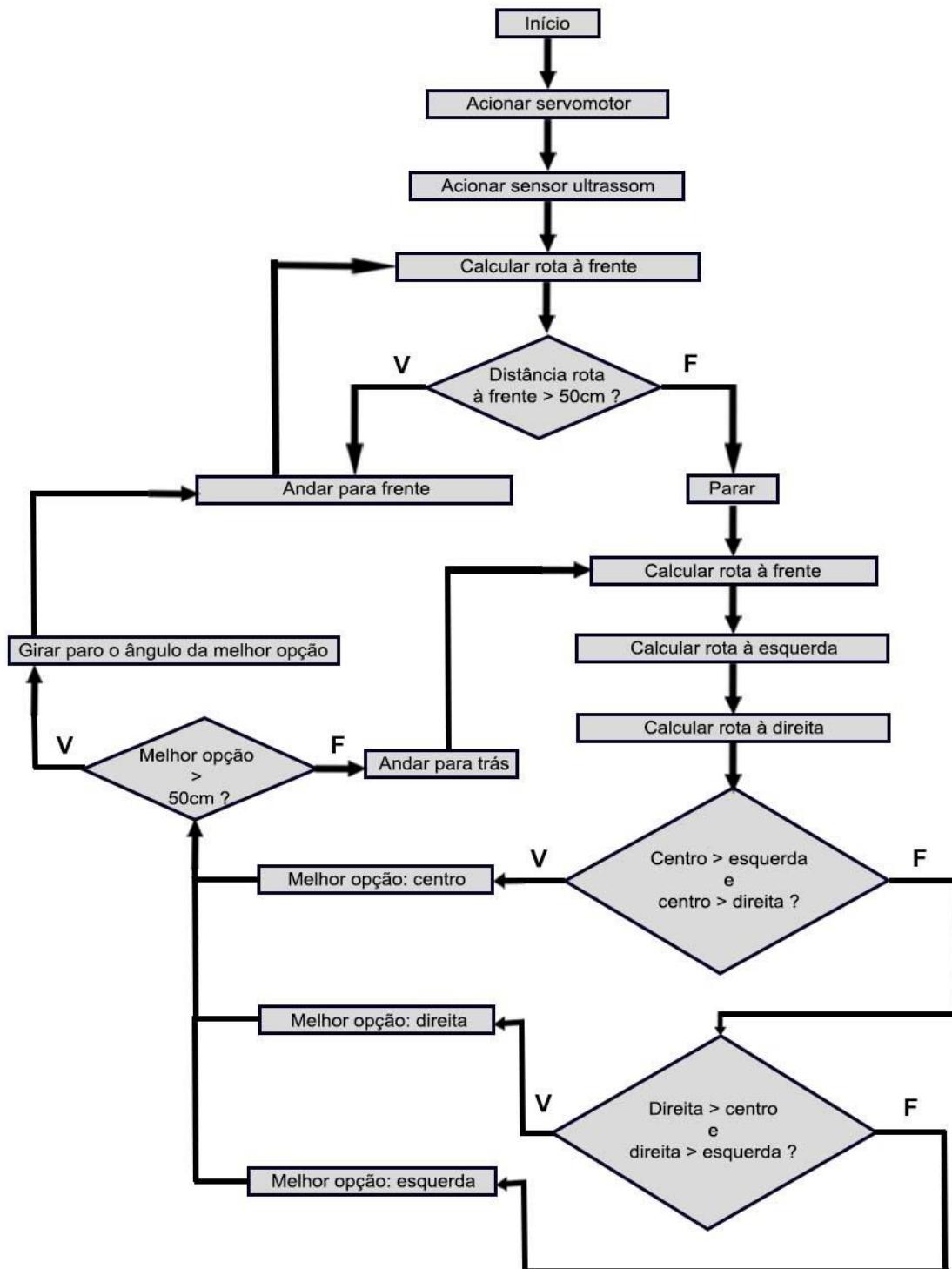


Figura 12: Fluxograma do código utilizado para movimentar a plataforma móvel autônoma.

### 3 RESULTADOS E DISCUSSÃO

Ao final do projeto, constatou-se que a unidade móvel se locomove de forma autônoma. O sistema de sucção do aspirador (Figura 13) funciona, porém não é totalmente efetivo, pois o aspirador suga somente detritos pequenos.

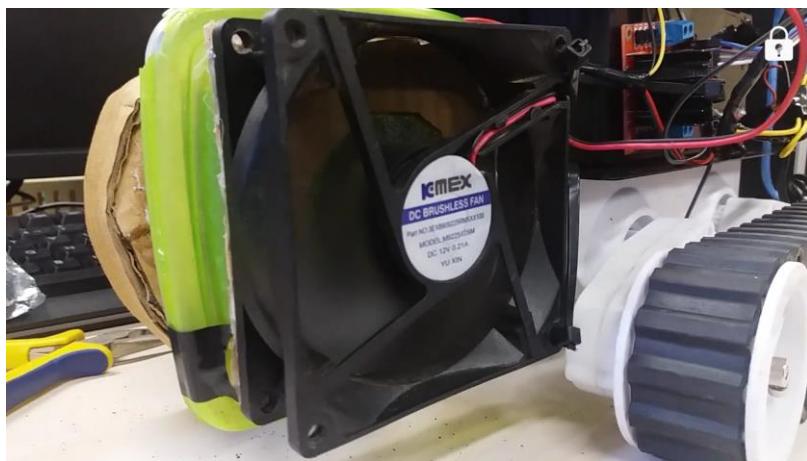


Figura 13: Sistema de sucção do aspirador.

O sensor consegue reconhecer a distância esperada, mas possui dificuldade de identificar itens menores (finos), como pernas de cadeiras e de mesa, pois não consegue obter uma resposta do sinal emitido para calcular a distância.

Pode-se implementar também na unidade móvel o sistema *bluetooth*, para ser guiada por um usuário por meio de um aplicativo *mobile*. Para isso, serão necessários dois resistores de  $1\text{k}\Omega$  e outro de  $2\text{k}\Omega$  e de algum aplicativo como o Bluetooth RC Controller, baixado no celular, para conexão com o carrinho (Figura 14).



Figura 14: Apresentação do sistema de controle via *bluetooth*.

No processo de conduzir a correta passagem da tensão, em princípio, ligou-se a bateria 12V totalmente independente da saída 5V do Arduino (uma para a ponte H e outra para o servomotor e o sensor de ultrassom), o que deixou o circuito aberto, fazendo com que a esteira não ligasse. Para solucionar o problema, foi necessário jampear o fio de energia negativa (terra) das duas tensões.

No desenvolvimento do algoritmo, houve dificuldades para determinar um código de programação para a perfeita locomoção do carrinho, evitando colisões com as paredes e demais objetos. Várias distâncias foram testadas. Por exemplo, com 20 cm de distância, a plataforma colidia com os obstáculos, pois o motor parava tardivamente; com 60 cm de distância, ela parava muito antes dos obstáculos, o sensor percebia obstáculos em todos os lados, e não se

movimentava por um perímetro possível de movimentação. Por fim, chegou-se a um padrão de distância (50 cm, como descrito na metodologia) que harmonizava o espaço de locomoção da unidade.

Construir o aspirador foi o processo mais complicado do projeto, pois o *cooler* não tinha força suficiente para sugar o pó, então foi decidido usar dois *coolers* para solucionar o problema, mas não houve melhora de desempenho, pois a potência adquirida continuava baixa. Foi testado um recipiente plástico de grande porte, o qual não apresentou o resultado esperado, pois o sistema de sucção era insuficiente em relação ao tamanho do recipiente. Então, um recipiente de médio porte foi escolhido para ser agregado ao carrinho. Por entupir, devido ao tamanho da sujeira sugada, o canudo, testado inicialmente com o recipiente de médio porte, foi substituído por uma mangueira de 1,5 mm, para filtrar melhor o ar e evitar entupimento.

## CONSIDERAÇÕES FINAIS

Este trabalho apresentou uma unidade móvel autônoma acoplada a um aspirador de pó, para auxiliar o trabalho doméstico. Obteve-se a livre locomoção da plataforma sobre uma superfície plana e um processo eficiente do cálculo do servomotor para medir a distância entre a plataforma e um objeto. Para melhorar seu desempenho, sugere-se agregar à unidade um aspirador mais potente, para ter a força necessária para sugar por maior período de tempo, até que seu recipiente esteja cheio.

Ainda, pode ser acoplado a ela um sensor infravermelho, com o objetivo de impedir o funcionamento dos *coolers*, quando detectado o acúmulo de sujeira, e um módulo *bluetooth* ou *wifi-shield*, caso se deseje comandar o carrinho e não somente seu funcionamento autônomo. Por fim, um GPS poderia definir o caminho de regresso do carrinho após seu recipiente estar cheio.

## REFERÊNCIAS

- ARRUDA, Paulo César da Costa; PRADO, Álvaro Camargo; DELATORE, Fábio. Estudo do controle de válvulas proporcionais com hardware Arduino para aplicação em veículo rural. In: SICFEI, 2017, Fundação Educacional Inaciana Pe. Sabóia de Medeiros. **Anais...** São Paulo: Centro Universitário da FEI, 2017.
- EVANS, D. **A Internet das Coisas:** como a próxima evolução da Internet está mudando tudo. 2011. Disponível em: <[http://www.cisco.com/c/dam/global/pt\\_br/assets/executives/pdf/internet\\_of\\_things\\_iot\\_ibsg\\_0411final.pdf](http://www.cisco.com/c/dam/global/pt_br/assets/executives/pdf/internet_of_things_iot_ibsg_0411final.pdf)>. Acesso em: 10 set. 2017.
- FELLIPE, L. Canal do YouTube: Leandro Fellipe. **Como fazer um mini aspirador de pó caseiro!** 2014. Disponível em: <<https://www.youtube.com/watch?v=h9X-jmabFE0>>. Acesso em: 10 set. 2017.
- FELLIPE, L. Canal do YouTube: Leandro Fellipe. **Mini aspirador de pó 2.0! Sugestões dos inscritos!** 2015. Disponível em: <<https://www.youtube.com/watch?v=DCxNLPTRFxM>>. Acesso em: 10 set. 2017.
- FRANÇA, M. S. **Plataforma rover 5, controle Bluetooth.** Monografia da disciplina de Circuitos Digitais, 2016. Disponível em: <<http://moodle.utfpr.edu.br/mod/resource/view.php?id=68692>>. Acesso em: 3 out. 2016.
- McROBERTS, M. **Arduino Básico.** São Paulo: Novatec, 2011.
- NIKU, S. B. **Introdução à robótica:** análise, controle, aplicações. 2.ed. Rio de Janeiro: LTC, 2017.
- OTTOBONI, A. Servo-acionamentos. **Mecatrônica Atual**, São Paulo, v.1, n.6, p.7-14, out. 2002. Disponível em: <[http://w3.ufsm.br/fuentes/index\\_arquivos/servo.pdf](http://w3.ufsm.br/fuentes/index_arquivos/servo.pdf)>. Acesso em: 14 set. 2017.

- PATSKO, L. F. **Tutorial montagem da ponte H.** 2006. Disponível em: <[https://www.robocore.net/upload/attachments/ponte\\_h\\_590.pdf](https://www.robocore.net/upload/attachments/ponte_h_590.pdf)>. Acesso em: 14 set. 2017.
- SANTOS, R. **Criando um robô “industrial” com Arduino.** 2014. Disponível em: <<http://www.ronilson.com.br/2014/10/criando-um-robo-com-arduino-ele-e.html>>. Acesso em: 14 set. 2017.
- SEUROBO. **CarAut v1.** Criando um carrinho autônomo (robô) com Arduíno, parte 1/4 (componentes). 2013. Disponível em: <<http://www.riandorobocomarduino.com/2013/10/caraaut-v1-criando-um-carrinho-autonomo.html>>. Acesso em: 10 set. 2017.
- SOUZA, F. **Arduino MEGA 2560.** 2014. Disponível em: <<https://www.embarcados.com.br/arduino-mega-2560/>>. Acesso em: 14 set. 2017.
- STORCH, K. H. **Projeto carrinho Kauê.** Monografia da disciplina de Circuitos Digitais. 2016/1. Disponível em: <<http://moodle.utfpr.edu.br/mod/resource/view.php?id=68692>>. Acesso em: 3 out. 2016.
- WENDLING, M. **Sensores V2.0.** 2010. Disponível em: <<http://www.joinville.udesc.br/portal/professores/saimon/materiais/Sensores.pdf>>. Acesso em: 14 set. 2017.

**Anexo 1 – Código do projeto.**

```
#include <Servo.h>
#include <Ultrasonic.h>
#define echoPin 6 //Pino 13 recebe o pulso do echo
#define trigPin 7 //Pino 12 envia o pulso para gerar o echo
Ultrasonic ultrasonic(7,6);
Servo motor;
int distancia_cm = 0; //variável do valor da distância
//motor A
int IN1 = 8;
int IN2 = 9;
//motor B
int IN3 = 10;
int IN4 = 11;
void setup(){
Serial.begin(9600); // inicializa a comunicação serial para mostrar dados
pinMode(IN1,OUTPUT);
  pinMode(IN2,OUTPUT);
  pinMode(IN3,OUTPUT);
  pinMode(IN4,OUTPUT);
delay(3000);
  pinMode(trigPin, OUTPUT); // define o pino trigger como saída.
  pinMode(echoPin, INPUT); // define o pino echo como entrada.
motor.attach(2);
  motor.write(90);
  rotacao_Parado;
}

// Função principal do Arduino
void loop(){
  andar();
}

// Função para chamar outras funções e definir o que o robô fará
void andar(){
  reposicionaServoSonar();
  int distancia = lerSonar(); // Ler o sensor de distância
  Serial.print("distancia: "); // Exibe no serial
  Serial.println(distancia);
  if (distancia > 50) {
    rotacao_Frente(10);
  }else{
    rotacao_Parado(10);
    posicionaCarroMelhorCaminho();
    andar();
  }
}

// Função para ler e calcular a distância do sensor ultrassônico
int lerSonar(){
  distancia_cm = ultrasonic.Ranging(CM); // Calcula a distância em CM
  delay(300);
  return distancia_cm; // Retorna a distância
}

// Função para calcular a distância do centro
int calcularDistanciaCentro(){
  motor.write(90);
  delay(200);
  int leituraDoSonar = lerSonar(); // Ler sensor de distância
  delay(600);
```

```
leituraDoSonar = lerSonar();
delay(600);
Serial.print("Distancia Centro: "); // Exibe no serial
Serial.println(leituraDoSonar);
return leituraDoSonar;           // Retorna a distância
}

// Função para calcular a distância da direita
int calcularDistanciaDireita(){
motor.write(30);
delay(200);
int leituraDoSonar = lerSonar();
delay(600);
leituraDoSonar = lerSonar();
delay(600);
Serial.print("Distancia Direita: ");
Serial.println(leituraDoSonar);
return leituraDoSonar;
}

// Função para calcular a distância da esquerda
int calcularDistanciaEsquerda(){
motor.write(150);
delay(200);
int leituraDoSonar = lerSonar();
delay(600);
leituraDoSonar = lerSonar();
delay(600);
Serial.print("Distancia Esquerda: ");
Serial.println(leituraDoSonar);
return leituraDoSonar;
}

// Função para pegar as distâncias lidas e calcular qual a melhor distância
char calculaMelhorDistancia(){
int esquerda = calcularDistanciaEsquerda();
int centro = calcularDistanciaCentro();
int direita = calcularDistanciaDireita();
reposicionaServoSonar();
int maiorDistancia = 0;
char melhorDistancia = '0';

if (centro > direita && centro > esquerda) {
    melhorDistancia = 'c';
    maiorDistancia = centro;
} else
if (direita > centro && direita > esquerda) {
    melhorDistancia = 'd';
    maiorDistancia = direita;
} else
if (esquerda > centro && esquerda > direita) {
    melhorDistancia = 'e';
    maiorDistancia = esquerda;
}
if (maiorDistancia <= 50) {
    rotacao_Re(1000);
    reposicionaCarroMelhorCaminho();
}
reposicionaServoSonar();
return melhorDistancia;
}
```

```

// Função para colocar o carrinho na melhor distância, isto é, girá-lo para
a melhor distância
void posicionaCarroMelhorCaminho(){
    char melhorDist = calculaMelhorDistancia();
    Serial.print("melhor Distancia: ");
    Serial.println(melhorDist);
if (melhorDist == 'c'){
    andar();
} else if (melhorDist == 'd'){
rotacao_Direita(10);
} else if (melhorDist == 'e'){
    rotacao_Esquerda(10);
} else{
    rotacao_Re(1000);
}
repositionaServoSonar();
}

// Função para deixar o sensor "olho" do robô no centro
void repositionaServoSonar(){
motor.write(90);
delay(200);
}

// Função para fazer o carro parar
void rotacao_Parado(int tempo)
{
digitalWrite(IN1,HIGH);
digitalWrite(IN2,HIGH);
digitalWrite(IN3,HIGH);
digitalWrite(IN4,HIGH);
delay(tempo);
}

// Função para fazer o robô andar para frente
void rotacao_Frente(int tempo)
{
digitalWrite(IN1,HIGH);
digitalWrite(IN2,LOW);
digitalWrite(IN3,HIGH);
digitalWrite(IN4,LOW);
delay(tempo);
Serial.println("*****");
Serial.println("andarParaFrente");
Serial.println("*****");
}

// Função que faz o robô andar para trás
void rotacao_Re(int tempo)
{
digitalWrite(IN1,LOW);
digitalWrite(IN2,HIGH);
digitalWrite(IN3,LOW);
digitalWrite(IN4,HIGH);
Serial.println("*****");
Serial.println("andarParaTras");
Serial.println("*****");
delay(tempo);

rotacao_Parado(10);
}

// Função que faz o robô virar à direita

```

```
void rotacao_Direita(int tempo)
{
    digitalWrite(IN1,LOW);
    digitalWrite(IN2,HIGH);
    digitalWrite(IN3,HIGH);
    digitalWrite(IN4,LOW);
Serial.println("*****");
Serial.println("girarParaDireita");
Serial.println("*****");
delay(tempo);
}

// Função que faz o robô virar à esquerda
void rotacao_Esquerda(int tempo)
{
    digitalWrite(IN1,HIGH);
    digitalWrite(IN2,LOW);
    digitalWrite(IN3,LOW);
    digitalWrite(IN4,HIGH);
Serial.println("*****");
Serial.println("girarParaEsquerda");
Serial.println("*****");
delay(tempo);
}
```

# Braço robótico didático em ABS via comunicação serial

## Didactic robotic arm in ABS via serial communication

Antônio César Guardia<sup>1</sup>, Mikael Pereira Messias<sup>2</sup>, Nelsi Kelm<sup>3</sup>

Ciência da Computação, Universidade Tecnológica Federal do Paraná (UTFPR-SH)

<sup>1</sup>[guardia@alunos.utfpr.edu.br](mailto:guardia@alunos.utfpr.edu.br), <sup>2</sup>[mikaelmessias@alunos.utfpr.edu.br](mailto:mikaelmessias@alunos.utfpr.edu.br),

<sup>3</sup>[nelsikelm76@gmail.com](mailto:nelsikelm76@gmail.com)

### RESUMO

O objetivo deste trabalho é apresentar um braço robótico didático, criado com base em Arduino e com controle cinemático via comunicação serial. O projeto executado foi proposto por Franciscone (2016). Trata-se de um braço robótico tridimensional, cujas peças são impressas com filamento ABS e movimentadas por quatro servomotores 9g. O controle da movimentação do braço é feito via comunicação serial, que se dá por conexão direta entre o Arduino Mega e periféricos, acionada por meio de dispositivo móvel (sistema operacional Android). O aplicativo de interface entre o dispositivo móvel e os servomotores do braço robótico utilizado é o Arduino Multi Servo Control. Após ajustes, obteve-se êxito no projeto quanto à proposta principal, no entanto, a movimentação do braço pode ser aperfeiçoada, tendo em vista curtas pausas que a tornam um tanto brusca. O braço robótico desenvolvido atinge seu propósito de ser didático e pode ser usado, por exemplo, para introduzir a área da robótica e programação de computadores a crianças.

**Palavras-chave:** Braço robótico; Arduino; impressão 3D; bluetooth.

### ABSTRACT

*The goal of this work is to present a didactic robotic arm, based on Arduino and with kinematic control via serial communication. The project executed was proposed by Franciscone (2016). It is a three-dimensional robotic arm whose parts are printed with ABS filament and moved by 4 9g servomotors. The arm movement control happens via serial communication, by direct connection between the Mega Arduino and peripherals, powered by mobile device (Android operating system). The interface application between the mobile device and the servomotors of the robotic arm is the Arduino Multi Servo Control. After adjustments, the project succeeded regarding the main proposal, however, the arm movements can be improved, considering short breaks that make it somewhat abrupt. The robotic arm developed achieves its purpose of being didactic, and can be used, for example, to introduce Robotics and Computer Programming to children.*

**Keywords:** Robotic arm; Arduino; 3D printing; bluetooth.

### Referência:

GUARDIA, A. C.; MESSIAS, M. P. ; KELM, N. Braço robótico didático em ABS via comunicação serial. In: SEMANA TECNOLÓGICA ACADÊMICA DE CIÊNCIA DA COMPUTAÇÃO DA UTFPR-SH, 4., 2017, Santa Helena, PR. Anais..., Santa Helena: UTFPR, 2017. p.100-117. Disponível em: <<http://sh.utfpr.edu.br/setac/Anais2017.pdf>>.

### INTRODUÇÃO

**N**a sociedade atual, é crescente a necessidade de se realizar tarefas com eficiência e precisão. Há também tarefas a ser realizadas em lugares perigosos ou de difícil acesso, como o fundo do mar, o espaço sideral, desarmar bombas. Para realizá-las, torna-se necessária a presença robôs, que são dispositivos eletromecânicos ou biomecânicos capazes de realizar trabalhos de maneira autônoma ou pré-programada (DIN-UEM, 2017).

A robótica é a área que trata do desenvolvimento dos robôs. Desse modo, estuda o desenvolvimento e a integração de técnicas e algoritmos para a criação de mecanismos automáticos, envolve as engenharias – Mecânica, Elétrica, de Controle e Automação – e várias áreas da computação, como inteligência artificial, visão computacional, processamento de imagens e dados, computação gráfica. Isso faz da robótica uma área interdisciplinar, em que a integração tecnológica é essencial.

A partir disso, este trabalho apresenta a construção de um braço robótico didático, criado com base em Arduino e com controle cinemático via comunicação serial. Um braço robótico didático como esse pode ser usado para introduzir a robótica para crianças, bem como outras tecnologias envolvidas no projeto – como controle por comunicação serial, desenho e impressão das peças em ABS, programação de computadores –, de maneira simples e interessante.

## 1 REFERENCIAL TEÓRICO-METODOLÓGICO

### 1.1 REFERENCIAL TEÓRICO

#### 1.1.1 Controle cinemático

Pode-se entender, a partir de Craig (2005), que a cinemática trata do estudo dos movimentos dos robôs sem considerar as causas que lhes dão origem. Já a dinâmica é o estudo do movimento, tendo em conta as forças e torques que os causam (TENENBAUM, 2006). Para tratar dos movimentos dos manipuladores, é necessário desenvolver técnicas para representar a posição de determinado ponto do braço no tempo. Essa representação depende da posição das juntas e dos elos, sendo necessário ter a base do robô como ponto de referência (SANTOS; WATANABE; CARRARA, 2006).

No caso de braços articulados (ou robôs cartesianos), o grau de liberdade do mecanismo define sua aplicação, sistemas controladores e o sistema mecânico de movimentação, além do controle eletrônico e de *software* (KUMAR, 2016). Vários autores, como Carrara (s.d.), ressaltam que braços articulados são amplamente utilizados na indústria, devido a sua versatilidade em substituir o trabalhador humano e também por serem altamente compactos.

#### 1.1.2 Controle por via de comunicação serial

Em telecomunicação e Ciência da Computação, comunicação serial é a transferência de dados *bit a bit* (um de cada vez), sequencialmente, em um canal de comunicação ou barramento (RUSCHEL, 1996). É usada para a comunicação entre o computador e vários dispositivos: teclado, *mouse*, *modem*, terminais e vários equipamentos de laboratório. É diferente da comunicação paralela, na qual os dados são enviados simultaneamente, podendo ser usada na comunicação com impressoras, escâneres, unidade de disco removível. Cada *bit* representa uma condição de voltagem (ligado – *on* ou desligado – *off*) em um dos pinos da porta serial (DB9 ou DB25), para envio ou recebimento de dados, representando 0 (desligado) ou 1 (ligado) (AL2A, 2016). Para curtas distâncias, barramentos seriais estão se tornando mais comuns, devido ao ponto em que as desvantagens dos barramentos paralelos, como a densidade de interconexão, superam suas vantagens de simplicidade (SANTOS JR.; OLIVEIRA, 2012).

Na comunicação serial, as mensagens digitais são mais longas que alguns poucos *bits*, e por não ser prático tampouco econômico transferir todos os *bits* de uma mensagem simultaneamente, essa mensagem é quebrada em partes menores e transmitida sequencialmente (DENVER, 1995). A transmissão *bit-serial* transforma a mensagem em um *bit* por vez através de um canal. Cada *bit* representa uma parte da mensagem. Os *bits* individuais são então rearranjados no destino para compor a mensagem original. A transmissão *bit-serial* é normalmente chamada de transmissão serial, e é o método de comunicação escolhido por

diversos periféricos de computadores. A transmissão *byte-22* serial converte 8 *bits* por vez através de 8 canais paralelos. Embora a taxa de transferência seja 8 vezes mais rápida do que na transmissão *bit-serial*, são necessários 8 canais, e o custo poderá ser maior do que 8 vezes para transmitir a mensagem (SANTOS JR.; OLIVEIRA, 2012). No braço robótico construído, a comunicação dá-se por conexão direta entre o dispositivo (Arduino Mega) e periféricos (servomotores e módulo *bluetooth*).

### 1.1.2.1 Módulo *bluetooth*

De acordo com ITEAD (2015), o HC-05 é um módulo de comunicação serial *bluetooth* projetado para conexão sem fio. Pode ser utilizado tanto em modo escravo, quando só recebe informações vindas de uma única fonte, quanto em modo mestre, quando recebe e transmite informações. Sua configuração se dá por comandos AT enviados pelo *serial monitor* do Arduino. Os principais comandos utilizados estão no Quadro 1.

Quadro 1: Principais comandos AT utilizados para configuração do módulo *bluetooth*.

Comando	Resposta	Descrição
AT	Ok	Comando de teste para verificar o funcionamento do módulo.
AT+INIT	1. Ok 2. FAIL	Inicializa o perfil SPP para transmissão/recepção de sinal.
AT+RESET	Ok	Reinicia o módulo. Usado após definir configurações como nome, senha, modo de operação etc.
AT+ORGL	Ok	Restaura as configurações de fábrica do módulo.
AT+NAME=PARÂMETRO	Ok	Define um novo nome para o módulo. Nome padrão: HC-05.
AT+PSWD=PARÂMETRO	Ok	Define uma senha para o módulo. Senha padrão: 1234.
AT+ROLE=PARÂMETRO	Ok	Define o modo de operação do módulo: 0. Master; 1. Slave; 2. Slave-Loop.
AT+INQM=PARAM1, PARAM2, PARAM3	1. Ok 2. FAIL	Configura o modo de varredura do módulo. Possui três parâmetros: PARAM1: Modo de consulta; PARAM2: Número máximo de dispositivos a serem encontrados; PARAM3: Tempo máximo para encerrar a varredura.
AT+CMODE=PARÂMETRO	Ok	Define o modo de conexão do módulo: 1. Conexão por endereço fixo; 2. Conexão a qualquer endereço; 3. Slave-loop.

### 1.1.3 Arduino Mega

Conforme Arduino Portugal (2016), Arduino é uma ferramenta para criar computadores que podem sentir e controlar mais o mundo que o computador pessoal. Ele é uma plataforma física de computação de código aberto, baseado numa simples placa microcontroladora, e um ambiente de desenvolvimento para escrever o código para a placa.

Ele permite a automação de projetos eletrônicos e robóticos por profissionais e amadores. Suas unidades são constituídas por controladora Atmel AVR de 8 *bits*, pinos digitais e

analógicos de entrada e saída, entrada USB – o que permite conexão com computadores – ou serial e possui código aberto, que quando modificado, dá origem a outros derivados “ino”, que, por questões comerciais, levam nomes como Netduino, Produino e Garagino. A placa Arduino não possui recursos de rede, mas pode ser combinada com outros Arduinos para criar extensões chamadas de Shields (ARDUINO MEGA 2560, 2015).

O Arduino pode ser usado para desenvolver objetos interativos, admitindo entradas de uma série de sensores ou chaves, e controlando uma variedade de luzes, motores ou outras saídas físicas. Projetos do Arduino podem ser independentes ou podem se comunicar com *software* rodando em computador (como Flash, Processing, MaxMSP). Os circuitos podem ser montados à mão ou comprados pré-montados, e o *software* de programação de código-livre pode ser baixado gratuitamente (ARDUINO PORTUGAL, 2016).

A placa Arduino Mega 2560 possui características adequadas para projetos mais elaborados. Baseada no microcontrolador ATmega 2560, possui 54 pinos de entradas e saídas digitais, e 15 deles podem ser utilizados como saídas PWM. Possui 16 entradas analógicas, 4 portas de comunicação serial (ARDUINO MEGA 2560, 2015).

O Arduino Mega é uma opção compatível com projetos que necessitam de muitos pinos de entrada e saída, além de memória de programa com maior capacidade. A alimentação da placa Arduino Mega pode ser feita tanto por USB como por uma alimentação externa, que é feita por meio do conector *jack* com positivo no centro, em que o valor de tensão da fonte externa deve estar entre os limites 6V a 20V. São recomendados para tensões de fonte externa valores de 7V a 12V (ARDUINO PORTUGAL, 2016).

Quando o cabo USB é ligado a um PC, a tensão não precisa ser estabilizada pelo regulador de tensão. Dessa forma, a placa é alimentada diretamente pela entrada USB. O circuito da USB apresenta alguns componentes que protegem a porta USB do computador, em caso de haver alguma anormalidade (ARDUINO PORTUGAL, 2016).

#### 1.1.4 Servomotores

O servomotor é uma máquina eletromecânica, que apresenta movimento proporcional a um comando, como dispositivos de malha fechada, ou seja, recebem um sinal de controle, que verifica a posição atual para controlar seu movimento, indo para a posição desejada com velocidade monitorada externamente sob *feedback* de um dispositivo denominado taco, ou sensor de efeito Hall, ou *encoder*, ou resólver, ou *tachsin*, dependendo do tipo de servomotor e aplicação. Em contraste com os motores contínuos que giram indefinidamente, o eixo dos servomotores possui a liberdade de apenas cerca de 180° graus (360° em alguns modelos), mas são precisos quanto à sua posição. De um servomotor, são exigidos dinâmica, controle de rotação, torque constante e precisão de posição-posicionamento. As características mais desejadas nos servomotores são o torque constante em larga faixa de rotação (até 4500 rpm), uma larga faixa de controle da rotação e variação (até 1:3000) e alta capacidade de sobrecarga (3 x Mo) (WIKIPEDIA, 2016).

Os servomotores possuem grande aplicabilidade e funcionalidade, que se estendem desde o setor da robótica de pequeno porte até as indústrias e seus dispositivos automáticos. Suas aplicações e particularidades devem ser bem analisadas para uma aplicação funcional (MENDONÇA, 2012).

Esse dispositivo eletromecânico possui uma parte fixa (estator) e outra móvel (rotor), como muitas outras máquinas síncronas. O estator tem bastante semelhança com o de uma máquina elétrica convencional, porém com restrições quanto à alimentação. O rotor é composto por ímãs permanentes, os quais são posicionados alinhadamente sobre o rotor e com o controlador, ou gerador de sinais, chamado de resólver (OTTOBONI, 2002). Ainda segundo Ottoboni (2002), utilizando a tecnologia de ímãs permanentes, os servomotores podem

proporcionar precisão e controle de velocidade e posição, além da grande vantagem de ser possível controlar o torque no eixo, de forma constante e em larga faixa de rotação.

O servomotor 9g (Figura 1), reconhecidamente, adapta-se a projetos com Arduino.

Ilustração	Especificações	
	Modulação	Analógico
	Torque	4,8V: 1,8kg/cm
	Velocidade	0,12 seg/60°
	Peso	9g
	Dimensões (mm)	23x12,2x29
	Tipo engrenagem	Plástico
	Tipo motor	3 polos
	Largura do pulso	500-2400µs
	Tipo de conector	JR

Figura 1: Dados do servomotor.

Fonte: SERVODATABASE.COM, 2016.

### 1.1.5 Impressão em 3D em ABS

As peças do braço robótico aqui descrito foram impressas em impressora 3D, com filamento Acrylonitrile Butadiene Styrene (ABS). A tecnologia de impressão 3D foi criada para materializar projetos que, antes dela, o máximo que se obtinha de interação era simulação por *softwares* de modelagem 3D. Atualmente, o processo mais comum de impressão 3D (também chamada de fabricação aditiva) é o Fused Deposition Modeling (FDM) ou Fuse Filament Fabrication (FFF), que consiste em criar no mundo físico objetos sólidos, desenhados em três dimensões, por depósito de filamento derretido, camada a camada, mediante o fatiamento do objeto virtual via *softwares* dedicados chamados de *slicers* – fatiadores em português (BUMGARNER, 2013).

Com o surgimento da manufatura ou fabricação aditiva, processos de fabricação foram repensados e a materialização de projetos, criação de protótipos, customização de peças e realização de ideias foram levados a um novo patamar. Antes dessa tecnologia, era evidente a dependência de usuários e consumidores dos fabricantes de produtos finais, mas atualmente quem possui uma impressora 3D teoricamente não tem limites de criação, e pode fazer as peças de que precisa (HAUSMAN; HORNE, 2014).

As impressoras 3D utilizam material plástico para a fabricação de peças. Esses plásticos podem ser de diversos materiais, como ABS e Polylactic Acid (PLA), que são os mais utilizados. As peças do braço robótico foram criadas em ABS dadas as características mecânicas do material, como resistência ao impacto, rigidez, estabilidade dimensional e custo reduzido (ADIPLAST, 2016).

O material utilizado no projeto (ABS) foi patrocinado pela empresa F3DB, que gentilmente enviou filamentos com diâmetro de 1,75mm ( $\pm 0,05$  de tolerância), com temperatura de fusão em torno de 230°C e base de pressão aquecida (mesa) a 105°C. Para a modelagem das peças, foi utilizado o *software* gratuito (versão básica) de modelagem 3D Sketchup e, para o fatiamento, o *software* OpenSource Slic3r. As camadas de impressão medem 0,24mm de altura e as peças foram impressas com três perímetros externos, três camadas de base e de topo, além de preenchimento (*infill*) de 25%. A impressora utilizada é de modelo Graber i3, modificada por um dos autores do artigo (acondicionada em uma caixa fechada, para manutenção da temperatura ambiente interna), com extrusora com redução 2:1, *hotend* (componente para derreter o filamento) modelo E3D V.5, com bico de 0,4mm de diâmetro.

## 1.3 REFERENCIAL METODOLÓGICO

### 1.2.1 Descrição do Projeto

O projeto executado foi proposto por Franciscone (2016). Trata-se de um braço robótico tridimensional, cujas peças são impressas em ABS, movimentadas por servomotores, desenhado com o objetivo ser de execução fácil, viável economicamente e de impacto visual. Dentre as possibilidades de controle do braço robótico, optou-se pela comunicação serial *bluetooth*, acionada por meio de dispositivo móvel (sistema operacional Android). O aplicativo de interface entre o dispositivo móvel e os servomotores do braço robótico utilizado é o Arduino Multi Servo Control (2016).

Foram utilizados os seguintes materiais na execução do projeto (Tabela 1).

Tabela 1: Materiais utilizados na execução do braço robótico.

<b>Item</b>	<b>Marca</b>	<b>Modelo</b>	<b>Quant.</b>	<b>Observação</b>
Servomotor	Turnigy	9G	4 un.	
Arduino	Arduino	Mega 2560	1 un.	
Módulo <i>bluetooth</i>		HC-05	1 un.	
Fios conectores		AWG-24		
Parafuso fenda cabeça chata		M3	17 un.	01 un, 20mm 01 un, 24mm 02 un. 30mm 04 un. 15mm 09 un. 10mm
Arruelas		M3	14 un.	
Porca sextavada		M3	17 un.	
Fonte de alimentação		5v/5A	1 un.	ATX

As peças em ABS utilizadas (Tabela 2) foram reproduzidas em conformidade com o projeto original. Ilustrações das peças impressas podem ser verificadas no Anexo 1.

Tabela 2: Peças impressas em ABS utilizadas na execução do braço robótico.

<b>Itens impressos em ABS</b>	<b>Referência</b>	<b>Quant.</b>
Base do motor panorâmico	EBA_01.00.010	1
Base rotativa	EBA_01.00.011	1
Base do motor de elevação e do motor de prolongamento	EBA_01.00.001	1
Haste auxiliar	EBA_01.00.003	3
Braço horizontal	EBA_01.00.005	1
Braço vertical maior	EBA_01.00.004	1
Braço vertical menor	EBA_01.00.002	1
Conexão triangular	EBA_01.00.006	1
Punho	EBA_01.00.012	1
Garra esquerda	EBA_01.00.014	1
Garra direita	EBA_01.00.013	1
Engrenagem de acionamento	EBA_01.00.015	1

<u>Engrenagem auxiliar</u>	EBA_01.00.016	1
<u>Fixador para servo</u>	EBA_01.00.009	1
<b>Total:</b>	<b>16</b>	

O circuito do braço robótico foi executado conforme a Figura 2, a seguir.

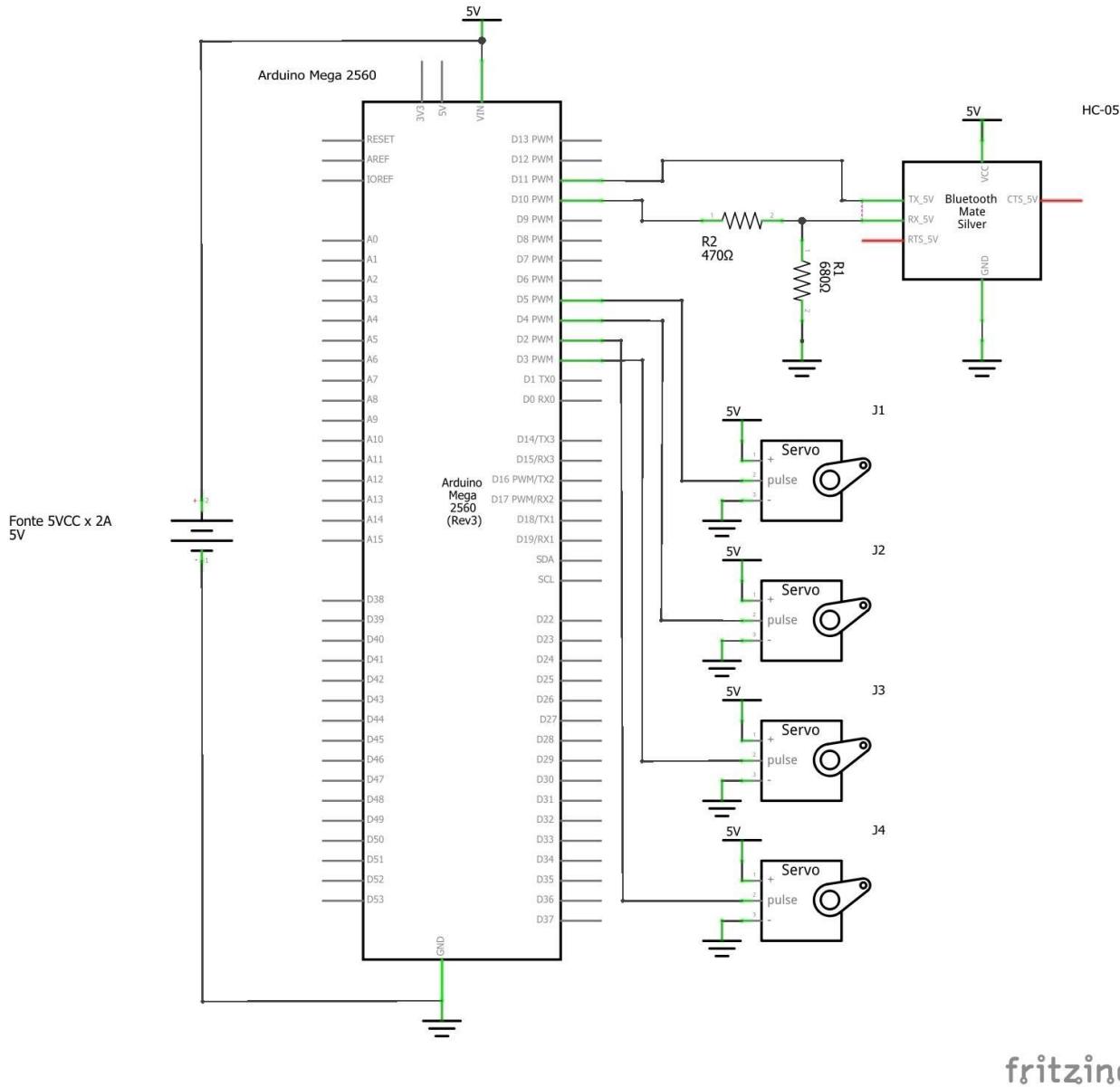


Figura 2: Diagrama do circuito do braço robótico.

fritzing

## 1.2.2 Código

Utilizou-se o código recomendado por Franciscone (INSTRUCTABLES, 2016). O código se inicia com a inclusão das bibliotecas SoftwareSerial.h e Servo.h. A primeira dá suporte à comunicação serial entre dispositivos periféricos e o Arduino, possibilitando a ampliação de aplicações com a plataforma. Originalmente, esse canal de comunicação dá-se pelas portas 0 e 1 do Arduino, mas é facilmente redirecionável no próprio corpo do código, com ressalva de que os pinos aptos a receber parâmetros de comunicação de dados são: 10, 11, 12, 13, 14, 15, 50, 51, 52, 53, A8 (62), A9 (63), A10 (64), A11 (65), A12 (66), A13 (67), A14 (68), A15 (69) do Arduino Mega 2560. Já a segunda biblioteca fornece diretrizes de controle dos servomotores utilizados no projeto. Tal controle estende-se ao posicionamento do eixo do servomotor, que pode variar de 0°

a 180°. O controle da variação da posição é feito por incremento e decremento de ângulos, sempre em forma de dados do tipo inteiro. Em seguida, são atribuídos valores à variável “Servo” como sendo os quatro servos utilizados no projeto (Figura 3).

```

1 //-
2 #include <SoftwareSerial.h>
3 #include <Servo.h>
4 Servo myservo1, myservo2, myservo3, myservo4;
5 //-

```

Figura 3: Fragmento de código 1.

O próximo passo é atribuir aos pinos 10 e 11 do Arduino Mega atributos de recepção e transmissão de dados por protocolo de comunicação serial. Esse canal é nomeado de Bluetooth. Dessa forma, o pino 10 recebe atributo de ser transmissor (TX) e o pino 11, o de receptor (RX) (Figura 4).

```

1 //-
2 SoftwareSerial bluetooth(10, 11); //TX | RX
3 //-

```

Figura 4: Fragmento de código 2.

O próximo bloco de comandos do programa configura os pinos de controle dos servomotores. Pode-se observar a atribuição de um servomotor a cada um dos pinos de 2 a 5 (Figura 5).

```

//-
1 void setup()
2 {
3   myservo1.attach(2);
4   myservo2.attach(3);
5   myservo3.attach(4);
6   myservo4.attach(5);
//-

```

Figura 5: Fragmento de código 3.

No bloco seguinte, são ajustadas as velocidades de comunicação entre o *firmware* e o monitor serial e entre o módulo *bluetooth* e o Arduino. Note-se que são duas taxas de transmissão distintas. Para tanto, utilizou-se a velocidade padrão, que vem nos módulos “de fábrica” (Figura 6).

```

//-
1 //Setup usb serial connection to computer
2 Serial.begin(38400);
3 //Setup Bluetooth serial connection to android
4 bluetooth.begin(38400);
//-

```

Figura 6: Fragmento de código 4.

No próximo bloco de código, vêm os parâmetros das variáveis dos servomotores, bem como os ajustes para a comunicação serial entre o módulo *bluetooth* e o Arduino. Na linha 5, o código aguarda resposta se o módulo *bluetooth* está ou não conectado a um dispositivo de controle do braço robótico, caso não esteja, não há interrupção no *loop* de monitoramento desse estado. Nas linhas seguintes, caso o *loop* seja interrompido pela conexão entre o módulo e o dispositivo de controle, são então requisitadas ao módulo informações sobre o possível acionamento dos motores, que é mostrado na tela do “monitor serial” da IDE do Arduino, por meio da linha “Serial.println(realservo)” (Figura 7).

```

1 //-----
2 void loop()
3 {
4     //Read from bluetooth and write to usb serial
5     if(bluetooth.available()>= 2 )
6     {
7         unsigned int servopos = bluetooth.read();
8         unsigned int servopos1 = bluetooth.read();
9         unsigned int realservo = (servopos1 *256) + servopos;
10        Serial.println(realservo);
11    //-----

```

Figura 7: Fragmento de código 5.

Daí em diante, a cada acionamento dos servomotores, são mostradas na tela do monitor serial as informações de qual servomotor está sendo acionado e em qual posição, em graus, o referido servomotor se encontra naquele momento. Dessa forma, quando o primeiro servomotor for acionado à movimentação, será exibido na tela, como pode ser visto no código abaixo (Figura 8), o seguinte: “1. servo 1 ON”. Quando houver movimentação do servomotor 1, por exemplo, de 15°, serão mostrados na tela do monitor os seguintes caracteres: “2. 1015”. Isso significa que o servomotor 1 partiu da posição 0° e virou 15° a partir de sua origem. A descrição desse código impresso na tela do monitor será, para o servomotor 1: O parâmetro “map” da biblioteca servo.h mapeia o posicionamento do servo no espaço em um perímetro que compreende de 0° a 180°. No código, atribuiu-se ao servomotor 1 um “nome”, que vai de 1000 a 1180, no qual cada unidade entre esses valores corresponde a 1° (Figura 8).

```

1 //-----
2 if(realservo >= 1000 && realservo <1180){
3     int servo1 = realservo;
4     servo1 = map(servo1, 1000,1180,0,180);
5     myservo1.write(servo1);
6     Serial.println("servo 1 ON");
7     delay(10);
8
9 }
10 //-----

```

Figura 8: Fragmento de código 6.

Esse código se repete mais três vezes, ou seja, uma para cada servo.

### 1.2.3 Android Multi Servo Control

O aplicativo Android Multi Servo Control é utilizado para envio dos comandos que acionam os servomotores do braço robótico (Figura 9).



Figura 9: Tela inicial do aplicativo Android Multi Servo Control.

Ao iniciar o aplicativo, é preciso primeiramente ativar o *bluetooth* no aparelho e realizar pareamento com o módulo do Arduino. Em seguida, seleciona-se “Select Device”, para que o aplicativo possa fazer a conexão com o Arduino e enviar os comandos a ser lidos pelos servomotores. Caso a conexão seja bem-sucedida, o rótulo abaixo do ícone muda para “Connected”. Para movimentar o braço, basta deslizar as quatro primeiras barras da esquerda para a direita, em que 0 indica a posição inicial e 180, a posição máxima em que cada servo pode se movimentar. Cada barra indica um dos servos a ser acionado (Figura 10).



Figura 10: Android Multi Servo Control conectado ao Arduino.

À medida que deslizam as barras, o monitor serial do Arduino mostra quais servos foram acionados e quantos graus dentro do perímetro o braço deve se deslocar (Figura 11).

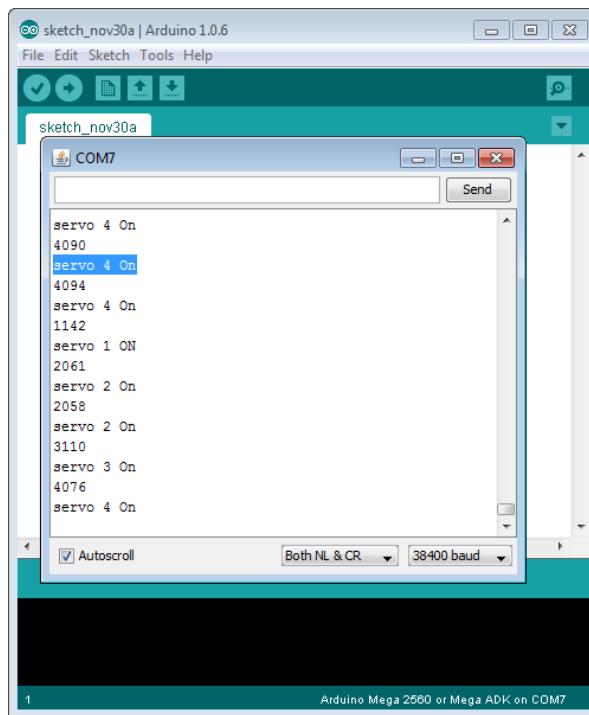


Figura 11: Monitor serial mostrando informações referentes ao servo e sua movimentação.

## 2 RESULTADOS E DISCUSSÃO

Obteve-se êxito no projeto quanto à proposta principal, ou seja, montagem e funcionamento do braço robótico didático (Figura 12 e 13), contudo, alguns ajustes foram necessários para aperfeiçoar os movimentos envolvidos no processo relacionado à carga: segurar, levantar, deslocar e soltar.

As peças impressas, ao ser montadas, mostraram-se harmônicas, mas houve problemas iniciais com as articulações. O modo de fixação das articulações era parafusos M4, e causava enrijecimento das articulações, que se tocavam, fazendo ruídos e ocasionando descontinuidade no movimento, dado o tamanho inadequado do parafuso, que se mostrou grande para ser usado no braço robótico. A solução foi a reimpressão das peças agora adaptadas para parafusos M3, que deixou as articulações livres para movimentar-se. Pode-se argumentar que o projeto que se tentou reproduzir tinha essa falha.

A movimentação do braço precisa ser aperfeiçoada, pois há interrupções que a tornam não-linear, não-contínua, por conta de pequenas pausas durante a movimentação, com retomada brusca do movimento em seguida. Mecanicamente, tem-se que o centro de gravidade do braço robótico desloca-se rapidamente, arrastando o peso do braço para frente. O resultado disso é certa instabilidade no transporte da carga, que pode ocasionar sua queda.

O braço apresenta alguns movimentos involuntários sem interação do usuário. Possível causa pode ser interferência (ruído elétrico) nas linhas de alimentação e de dados. Esses ruídos podem ser devido à baixa blindagem contra ruídos dos fios de conexão dos componentes. Outra causa pode ser interferência de outros dados na transmissão serial entre o aplicativo do celular e o módulo receptor-transmissor *bluetooth*.

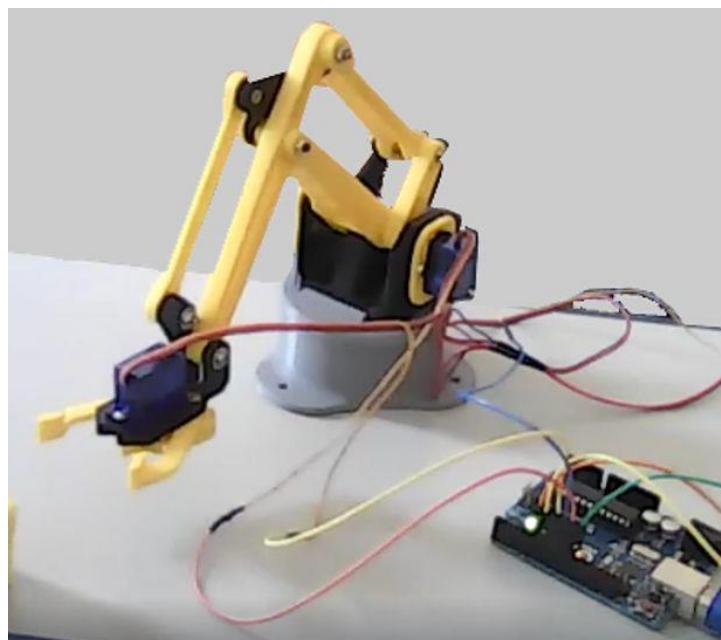


Figura 12: Braço robótico finalizado.



Figura 13: Acionamento da garra do braço robótico com controle via *bluetooth*.

## CONSIDERAÇÕES FINAIS

Este trabalho apresentou e discutiu a execução de um projeto de braço robótico, cujas peças foram impressas em ABS pelo processo de manufatura aditiva. Obteve-se êxito no trabalho, pois, embora tenha havido problemas quanto à movimentação do protótipo, o braço robótico funciona como planejado. Além disso, podem-se melhorar movimentos involuntários. Também, ressalta-se o problema com a comunicação *bluetooth*, que se mostrou instável e passível de interferências, que se pronunciaram na forma de “ruídos”, fazendo com que o braço executasse movimentos sem que houvesse envio de algum comando via *software* de controle. Assim, sugere-se como trabalho futuro eliminar esses movimentos e ruídos, bem como a continuação ou nova execução do projeto com outros tipos de controle, como potenciômetro, computador (*desktop*) ou *joystick*, por exemplo, para que a interferência seja reduzida e abram-se novas possibilidades de aplicação do braço robótico, por meio de outras formas de controle.

Por fim, considera-se que o braço robótico didático pode ser utilizado para incentivar crianças a aprender robótica e programação de maneira prática, divertida e interativa.

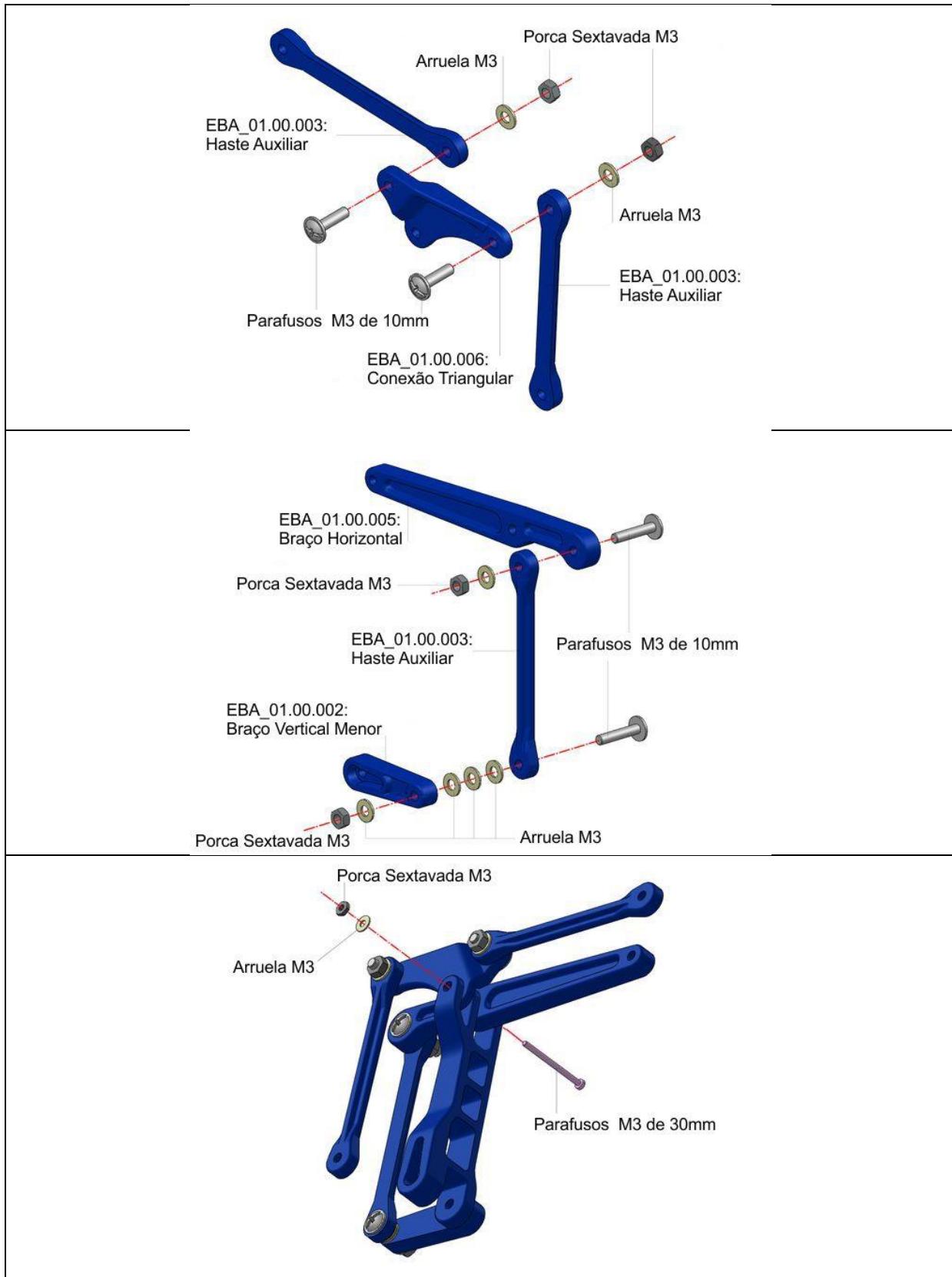
## AGRADECIMENTOS

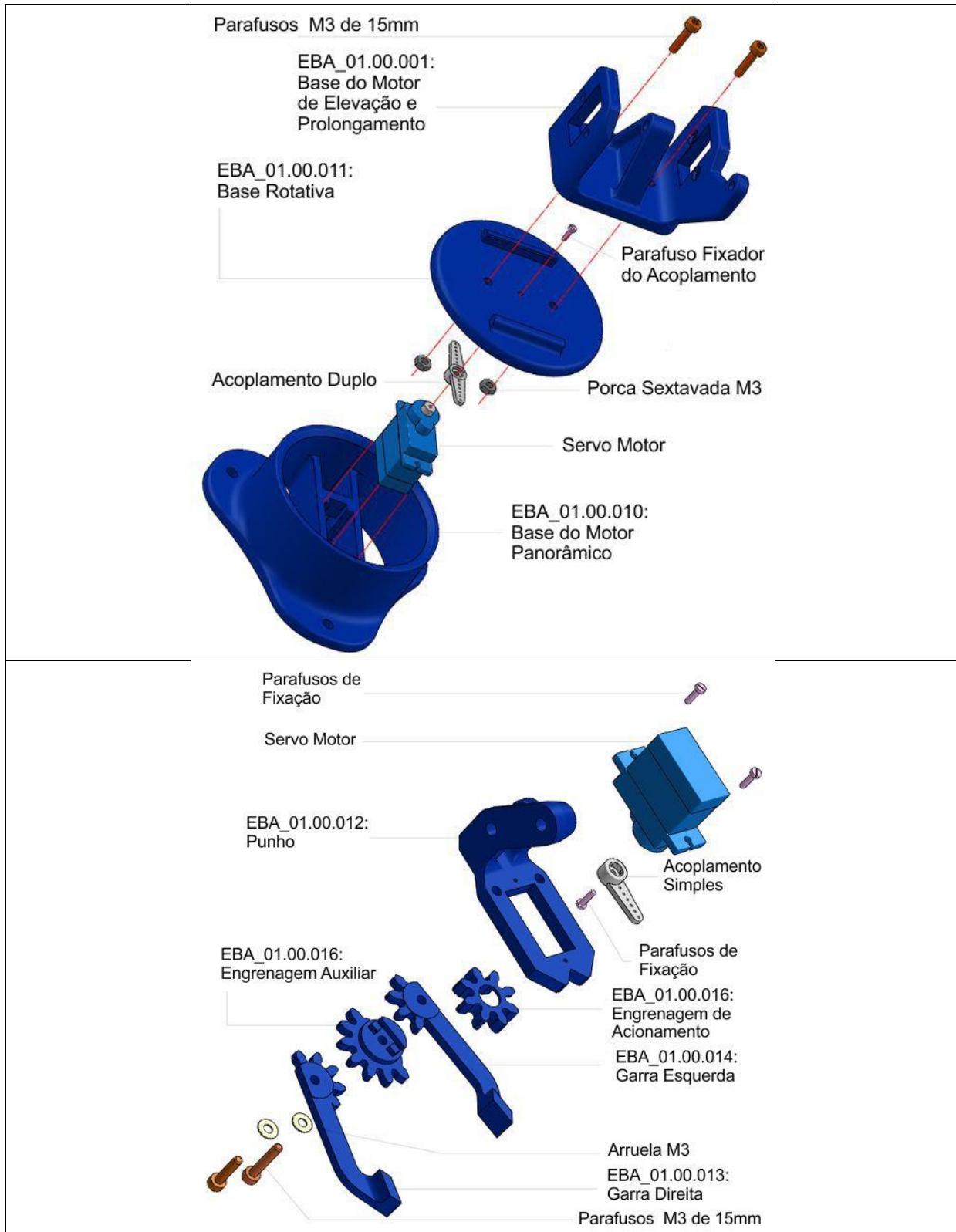
Os autores agradecem ao professor Itamar Iliuk e à acadêmica Vera Lúcia V. S. Araújo as valiosas contribuições técnicas e metodológicas prestadas na execução do projeto.

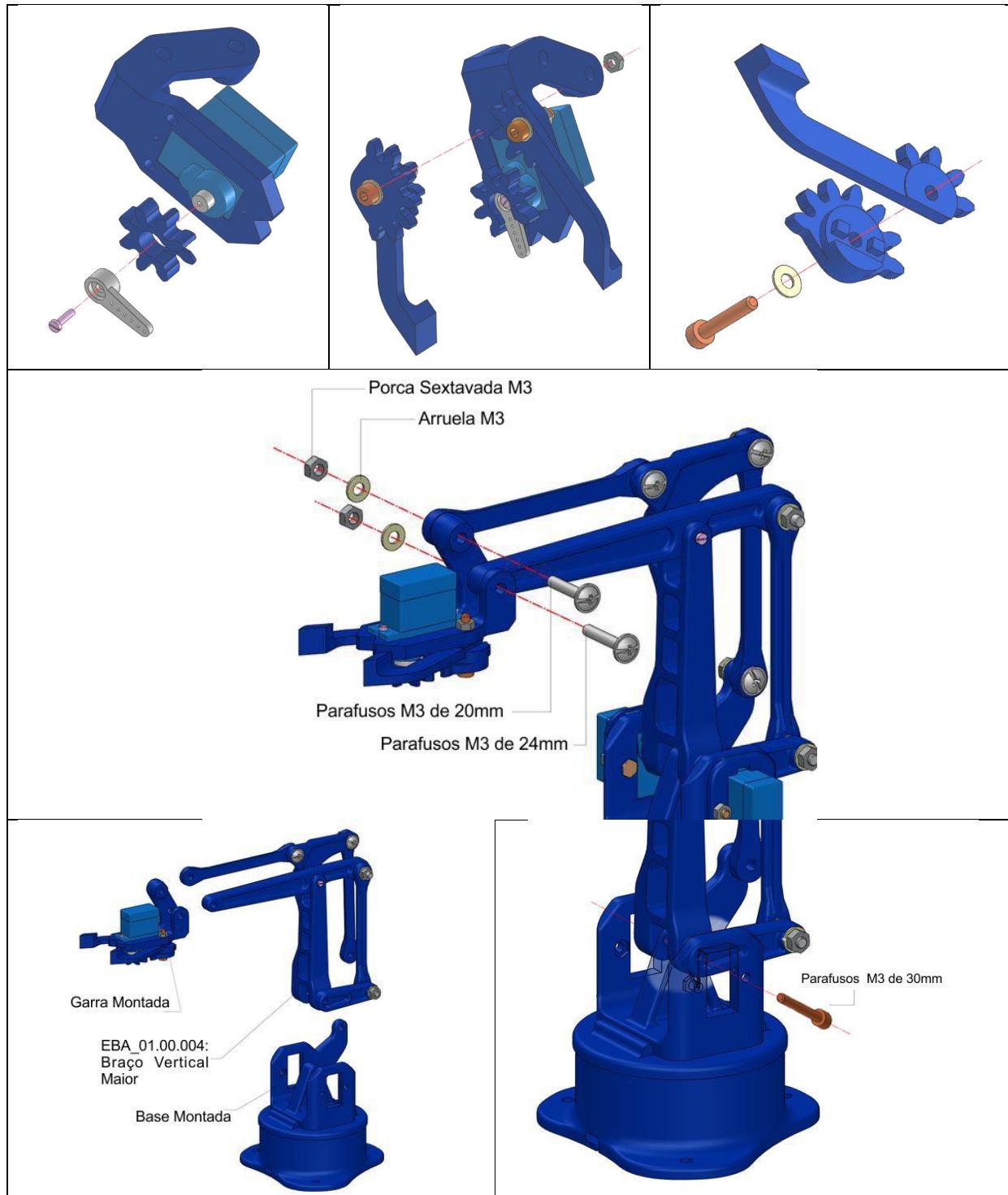
## REFERÊNCIAS

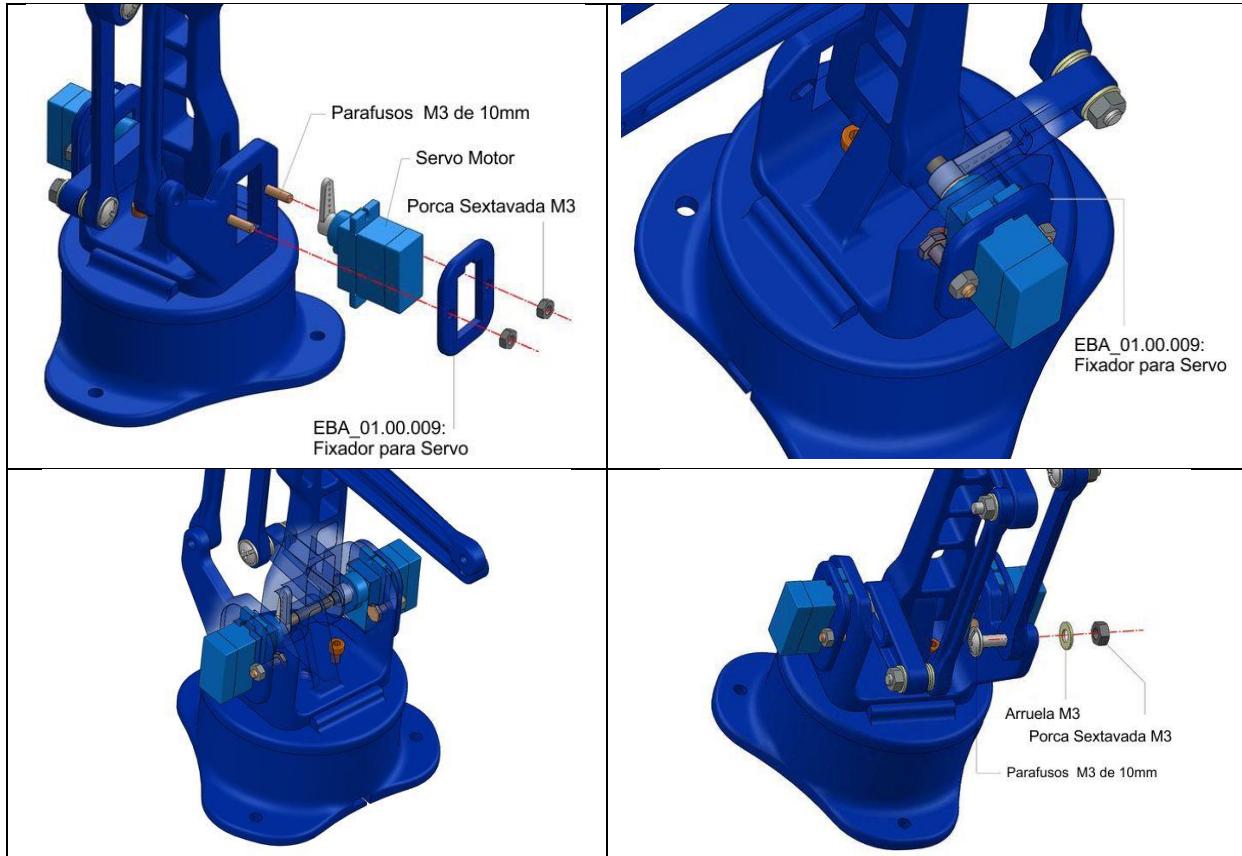
ADIPLAST. Disponível em: <[http://www.adiplast.ind.br/news\\_abs.php](http://www.adiplast.ind.br/news_abs.php)>. Acesso em: nov. 2016.

- ARDUINO MEGA 2560. Disponível em:<[http://arduino.cc/en/Main/ArduinoBoardMega\\_2560](http://arduino.cc/en/Main/ArduinoBoardMega_2560)>. Acesso em: nov. 2016.
- ARDUINO MULTI SERVO CONTROL. Disponível em:<[https://play.google.com/store/apps/details?id=appinventor.ai\\_addmefaster1.BlueCore\\_Tech\\_Multi\\_Servo\\_Control](https://play.google.com/store/apps/details?id=appinventor.ai_addmefaster1.BlueCore_Tech_Multi_Servo_Control)>. Acesso em: nov. 2016.
- ARDUINO PORTUGAL. Disponível em: <<http://www.arduinoportugal.pt/arduino-mega/>>. Acesso em: nov. 2016.
- AUTOMAÇÃO LIVRE PARA LABORATÓRIOS DE ÁGUAS – AL2A. Portas Seriais e Comunicação Serial. Disponível em: <<http://www.c2o.pro.br/automacao/x834.html>>. Acesso em: nov. 2016.
- CARRARA, V. **Apostila de Robótica**. Universidade Braz Cubas. Engenharia Mecânica, Engenharia de Controle e Automação. sem data. Disponível em:<[http://www.joinville.udesc.br/portal/professores/silas/materiais/Apostila\\_de\\_Robotica.pdf](http://www.joinville.udesc.br/portal/professores/silas/materiais/Apostila_de_Robotica.pdf)>. Acesso em: set. 2017.
- CRAIG, J. J. **Introduction to Robotics**: mechanics and control. 3.ed. Pearson Education, Inc., Pearson Prentice Hall: Upper Saddle River, 2005. Disponível em:<[http://www.mech.sharif.ir/c/document\\_library/get\\_file?uuid=5a4bb247-1430-4e46-942c-d692dead831f&groupId=14040](http://www.mech.sharif.ir/c/document_library/get_file?uuid=5a4bb247-1430-4e46-942c-d692dead831f&groupId=14040)>. Acesso em: set. 2017.
- DIN-UEM. **O que é robótica?** Disponível em:<<http://www.din.uem.br/ia/robotica/oqerobo.htm>>. Acesso em: set. 2017.
- BUMGARNER, B. Getting started with a 3D. **Make**, Elsevier Science B.V., Amsterdam, fev. 2003.
- FRANCISCOME, C. **EEZYbotARM**. theGHIZmo in robots. Disponível em:<<http://www.instructables.com/id/EEZYbotARM/>>. Acesso em: out. 2016.
- HAUSMAN, K. K.; HORNE, R. **3D Pinting for Dummies**: A Wiley Brand. Hoboken, NJ: John Wiley & Sons, 2014.
- INSTRUCTABLES. **Android APP to control a 3DPrinted Robot**. Disponível em:<<http://www.instructables.com/id/Android-APP-to-Control-a-3DPrinted-Robot/>>. Acesso em: out. 2016.
- ITEAD Wiki. **Serial Port Bluetooth Module (Master/Slave)**: HC-05. Disponível em:<[https://www.itead.cc/wiki/Serial\\_Port\\_Bluetooth\\_Module\\_%28Master/Slave%29\\_-\\_HC-05](https://www.itead.cc/wiki/Serial_Port_Bluetooth_Module_%28Master/Slave%29_-_HC-05)>. Acesso em: dez. 2016.
- KUMAR, V. **Introduction to Robot Geometry and Kinematics**. 2016. Disponível em:<<https://ucn.instructure.com/courses/7306/modules/items/41796>>. Acesso em: nov. 2016.
- MENDONÇA, V. **Servomotor**: Princípios básicos de funcionamento. Disponível em:<<http://www.ebah.com.br/content/ABAAAAAaLkAB/servomotor-principios-basicos-funcionamento>>. Acesso em: out. 2016.
- OTTOBONI, A. Servo-acionamentos. **Mecatrônica Atual**, São Paulo, v.1, n.6, p.7-14, out.2002. Disponível em:<[http://w3.ufsm.br/fuentes/index\\_arquivos/servo.pdf](http://w3.ufsm.br/fuentes/index_arquivos/servo.pdf)>. Acesso em: maio 2017.
- SANTOS, Flávia Moreira; WATANABE, Emerson Teruhiko; CARRARA, Valdemir. Estudo da cinemática inversa aplicada num braço robótico. **Anais do XII ENCITA 2006**, ITA, p.16-19, outubro, 2006. Disponível em:<<http://www.bibl.ita.br/xxiencita/Comp-10.pdf>>. Acesso em: maio 2017.
- SANTOS JR., E. F.; OLIVEIRA, P. E. **Comunicação ponto a ponto via tecnologia GSM**. 2012. Trabalho de Conclusão de Curso (Graduação em Sistemas para Internet) – Centro Universitário Adventista de São Paulo, Engenheiro Coelho. São Paulo. Disponível em:<<http://unasp-ec.com/sistemas/admin/upload/1411505346TCC.pdf>>. Acesso em: nov. 2012.
- SERVODATABASE.COM. **Servo Specifications and Reviews**. Disponível em:<<http://www.servodatabase.com/servo/towerpro/sg90>>. Acesso em: nov. 2016.
- TENENBAUM, R. A. **Dinâmica aplicada**. 3.ed. Barueri, SP: Manole, 2006
- WIKIPEDIA. **Servomotor**. <<https://pt.wikipedia.org/wiki/Servomotor>>. Acesso em: jun. 2017.

**Anexo 1 – Peças impressas em ABS e montagem.**







# Proposta de automação de um sistema de irrigação para estufa utilizando dispositivos microcontrolados

## An automated irrigation system for a greenhouse using microcontrollers

Jonathan Galdino da Silva<sup>1</sup>, Danilo Luiz Cavazim Neto<sup>2</sup>, Itamar Iliuk<sup>3</sup>

Ciência da Computação, Universidade Tecnológica Federal do Paraná (UTFPR-SH)

<sup>1</sup>[jonathansilva.2016@alunos.utfpr.edu.br](mailto:jonathansilva.2016@alunos.utfpr.edu.br), <sup>2</sup>[daniloneto@alunos.utfpr.edu.br](mailto:daniloneto@alunos.utfpr.edu.br),

<sup>3</sup>[itamariliuk@utfpr.edu.br](mailto:itamariliuk@utfpr.edu.br)

### RESUMO

O trabalho apresenta uma proposta de automação de um sistema de irrigação para estufas utilizando dispositivos microcontrolados. Para a realização deste sistema, foram utilizados um módulo Ethernet, um módulo relé e o Arduino Mega 2560. O sistema é capaz de controlar a estufa, fazendo a leitura de dados de temperatura, umidade relativa do ar e umidade do solo por meio do uso de sensores. Utilizando válvulas solenoides, o sistema é capaz de realizar a irrigação adequada do solo, mantendo-a em 80%. Os dados são enviados para uma plataforma de Internet das Coisas, sendo possível visualizá-los através do *smartphone*.

**Palavras-chave:** Arduino; automação; umidade do solo; estufa.

### ABSTRACT

*This paper presents a proposal for the automation of an irrigation system for greenhouses using micro-controlled devices. To create this system, Ethernet module, a Relay module, and the Arduino Mega 2560 were used. The system is able to control the greenhouse by reading temperature data, relative humidity, and soil moisture by sensors. Using solenoid valves, the system is capable of performing adequate irrigation of the soil, maintaining it at 80%. Data is sent to an Internet of Things platform, being possible to visualize them through the smartphone.*

**Keywords:** Arduino; automation; soil moisture; greenhouses.

### Referência:

SILVA, J. G.; CAVAZIM NETO, D. L.; ILIUK, I. Proposta de automação de um sistema de irrigação para estufa utilizando dispositivos microcontrolados. In: SEMANA TECNOLÓGICA ACADÊMICA DE CIÊNCIA DA COMPUTAÇÃO DA UTFPR-SH, 4., 2017, Santa Helena, PR. *Anais...*, Santa Helena: UTFPR, 2017. p.118-137. Disponível em: <<http://sh.utfpr.edu.br/setac/Anais2017.pdf>>.

## INTRODUÇÃO

Este trabalho tem seu foco em descrever uma proposta de automação de um sistema de irrigação para estufas, o qual utiliza dispositivos microcontrolados. É realizado com Arduino, módulo Ethernet, módulo relé, válvulas solenoide e *protoboards*.

Estufas são locais com o propósito de acumular e conter o calor em seu interior. O sistema proposto, elaborado com uso de programação – linguagem C/C++ – para determinar seu comportamento, recolhe dados via sensores e controla uma estufa, assim, irriga o solo por microaspersores. Os dados são armazenados num banco de dados, mas é possível visualizá-los pela Internet, o que elimina a necessidade de locomoção até o sistema.

## 1 REFERENCIAL TEÓRICO-METODOLÓGICO

### 1.1 REFERENCIAL TEÓRICO

#### 1.1.1 Arduino

Arduino é uma plataforma eletrônica de código aberto, baseada em *hardware* e *software*, fácil de usar. As placas Arduino são capazes de ler entradas, luz em um sensor, um dedo em um botão ou uma mensagem no Twitter, e transformá-las em uma saída, ativar um motor, ligar um diodo emissor de luz (LED – *light emitting diode*), publicar algo *online*. O Arduino nasceu no Ivrea Interaction Design Institute, como uma ferramenta fácil para prototipagem rápida, voltada para estudantes sem formação em eletrônica e programação (ARDUINO, 2017a). Ele dispõe de um espaço de desenvolvimento, o *sketch*, para escrever o código e introduzi-lo no microcontrolador através da comunicação serial USB.

No sistema apresentado aqui, foi utilizada a placa de prototipagem Arduino Mega 2560 (Figura 1). Arduino Mega é desenhado para projetos que exigem mais linhas *input/output*, mais memória de esboço e mais memória Memória de Acesso Randômico (RAM – *random access memory*). Com 54 pinos digitais de entrada e saída, 16 entradas analógicas e um espaço maior para esboço, de acordo com o fabricante, é a placa recomendada para impressoras 3D e projetos de robótica (ARDUINO, 2017b).

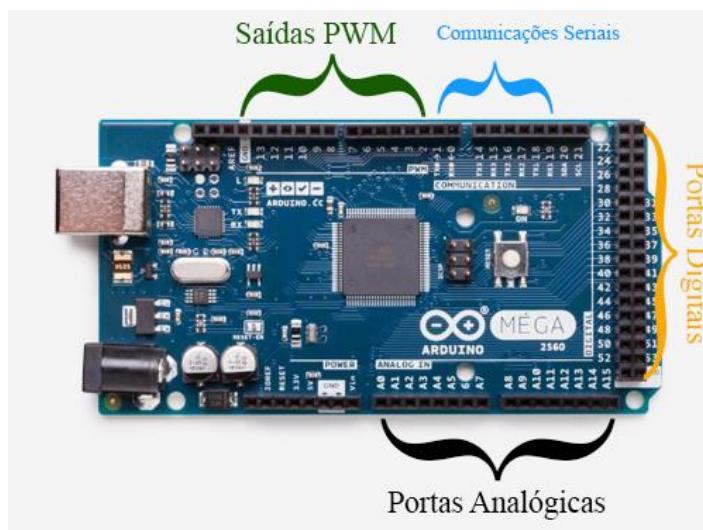


Figura 1: Arduino Mega 2560 e suas pinagens.

Fonte: Arduino (2017b). Adaptado.

#### 1.1.2 Módulo Relé

No sistema descrito aqui, foi necessário utilizar um módulo relé de 8 canais, modelo SRD-05VDC-SL-C. O módulo relé é uma interface com canais, que permite controlar cargas até 220V AC, com tempo de chaveamento de 5ms a 10ms. Tem limite de corrente de 10A por canal e sua tensão de operação é de 5V DC (VCC e GND) (SONGLE RELAY, 2017). A Figura 2 demonstra o uso do módulo relé.

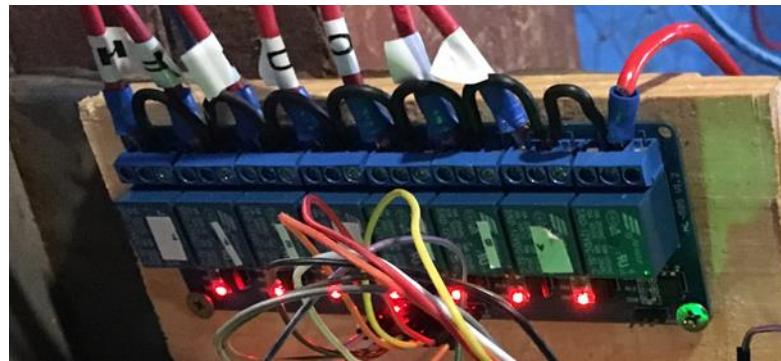


Figura 2: Módulo relé conectado.

### 1.1.3 Módulo de Conexão de Rede

O módulo Ethernet permite que a placa Arduino se conecte à Internet. Esse módulo é constituído pelo chip Ethernet Wiznet W5100. O Wiznet W5100 fornece acesso à rede (IP) nos protocolos TCP (Transmission Control Protocol) e UDP (User Datagram Protocol). O TCP é um protocolo de transporte de fluxo confiável. Nesse contexto, o termo fluxo significa orientado a conexões: deve ser estabelecida uma conexão entre duas extremidades de uma transmissão, antes que uma das duas possa transmitir dados. O UDP é mais simples do que os protocolos de transporte do TCP/IP. Trata-se de um protocolo de processo a processo, que adiciona apenas endereços de porta, controle de erros de soma de verificação e informações de comprimento nos dados da camada superior (FOROUZAN; FEGAN, 2009). Esse módulo suporta até quatro conexões de soquete simultâneas. O chip elabora uma comunicação com a biblioteca Ethernet (ARDUINO, 2017c). Com o propósito de conectar o sistema com a Internet, foi utilizado o módulo Ethernet, demonstrado na Figura 3, devido à necessidade de haver conexão estável para fazer comunicação com a interface *web*. É possível utilizar o módulo *wi-fi* (GOUVEIA; ILIUK, 2016) para realizar conexões sem fio entre o sistema e quaisquer dispositivos que tenham acesso à rede *wi-fi* e que sejam compatíveis com os protocolos de rede 802.11b/g/n. O protocolo 802.11b/g/n possibilita estabelecer conexões nas seguintes velocidades de transmissão: 1 Mb/s, 2 Mb/s, 5,5 Mb/s e 11 Mb/s (INFOWESTER, 2017).



Figura 3: Módulo Ethernet utilizado no sistema.

**Fonte:** FilipeFlop (2017).

### 1.1.4 Interface Web

A interface *web* do sistema de monitoramento é feita pelo aplicativo *online* Dweet, uma plataforma de inscrição e publicação simples para máquinas, sensores, aparelhos e robôs. (DWEET, 2017a). A plataforma Dweet é demonstrada na Figura 4, no *smartphone*.

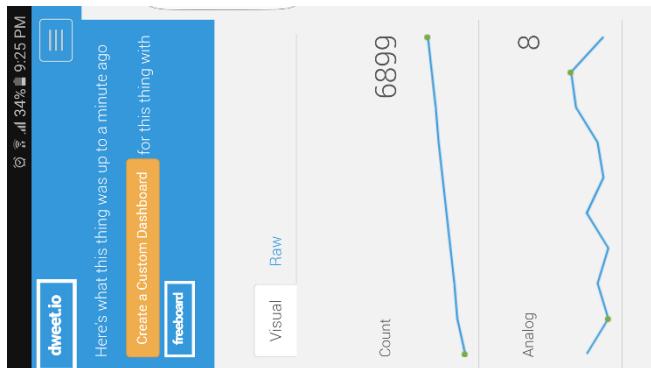


Figura 4: Plataforma do Dweet, que pode ser visualizada pelo celular.

Fonte: Dweet (2017a).

Os dados de umidade, temperatura e umidade do solo são obtidos pelo Freeboard, uma plataforma de código aberto e de assinatura opcional, fácil de integrar com diversas fontes de dados, e que tem produção pronta e *design* elegante (NETWORKWORLD, 2017). Nessa plataforma, é possível criar diversos tipos de gráficos diferentes (Figura 5). É necessário utilizar o método POST, que faz parte do Hyper-Text Transfer Protocol (HTTP) – ou Protocolo de Transferência de Hipertexto –, para transmitir dados a um endereço.

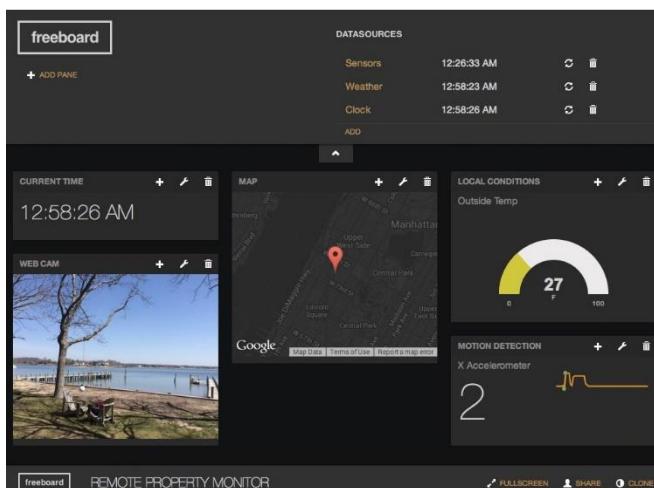


Figura 5: Plataforma Freeboard.

Fonte: Freeboard (2017).

### 1.1.5 Sensor de Umidade e Temperatura do Ar

Para mensurar dados de umidade e temperatura do ar, foi utilizado no sistema proposto, o sensor de temperatura e umidade relativa do ar DHT22, modelo AM2302, com faixa de medição entre 0% e 100% para umidade relativa, e de -40°C a 125°C para temperatura. Esse sensor possui uma tensão de operação de 3V a 5V. A tensão de operação recomendada do DHT22 é de 5V. O sensor é composto por 4 pinos; o VDD, que consiste em alimentar o sensor; o DAS, que consiste em ler e escrever os dados, e o GND, que é o terra. O pino NC desse sensor não é utilizado. A Figura 6 demonstra as pinagens do módulo DHT22.

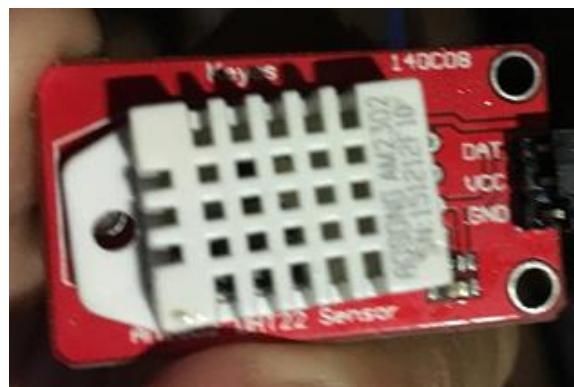


Figura 6: DHT22 e suas respectivas pinagens.

O DHT22 trabalha com tensão de alimentação de 3,3V e o comprimento do cabo não deve ser superior a 100 cm. Caso contrário, haverá queda de tensão da linha, resultando em erro de medição (AOSONG, 2017). A porcentagem de erro varia de acordo com a umidade relativa do ar e da temperatura, podendo ser descrita como na Figura 7. A margem de erro é definida pela umidade e temperatura obtida pelo sensor.

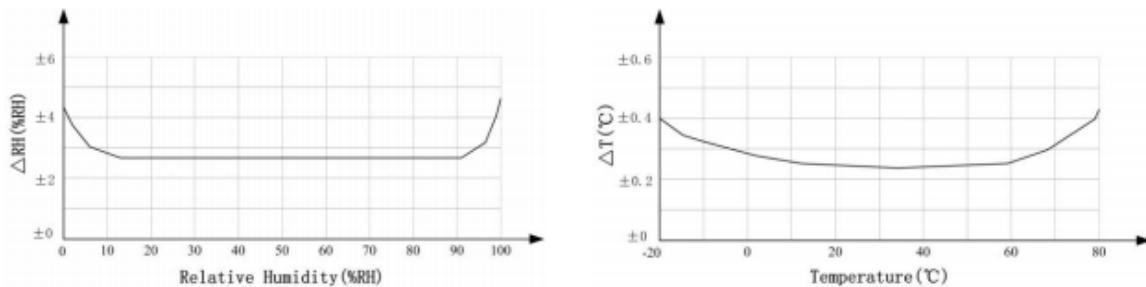


Figura 7: Gráfico da margem de erro.

Fonte: AOSONG (2017).

O gráfico demonstra o erro relativo nas leituras de umidade do ar e de temperatura do ar. A porcentagem de erro na leitura de umidade do ar tende a ser maior se a umidade relativa for maior do que 90% ou menor do que 20%. Nesse intervalo, é de  $\pm 3\%$ . Para a leitura de temperatura de ar, a margem de erro é de  $\pm 0,3^{\circ}\text{C}$  se a temperatura do ar estiver maior que  $15^{\circ}\text{C}$  e menor que  $60^{\circ}\text{C}$ . Fora desse intervalo, a margem de erro é capaz de atingir  $\pm 0,4^{\circ}\text{C}$ .

#### 1.1.6 Módulo Sensor de Umidade do Solo

Para obter a umidade do solo no sistema, foi utilizado o módulo higrômetro YL-69. A resposta desse módulo é uma variação de tensão de saída analógica ou digital. A variação de tensão muda de acordo com a configuração do potenciômetro. O YL-69 possui tensão de operação de 3,3V até 5V e utiliza o comparador LM393 de fácil instalação. Além de saída digital, também possui saída analógica. A Figura 8 mostra o comparador, o qual pode ser conectado a um conversor AD, para obter resultados analógicos mais precisos. A tensão de saída do YL-69 é de no máximo 4,2V (QQTRADING, 2017).

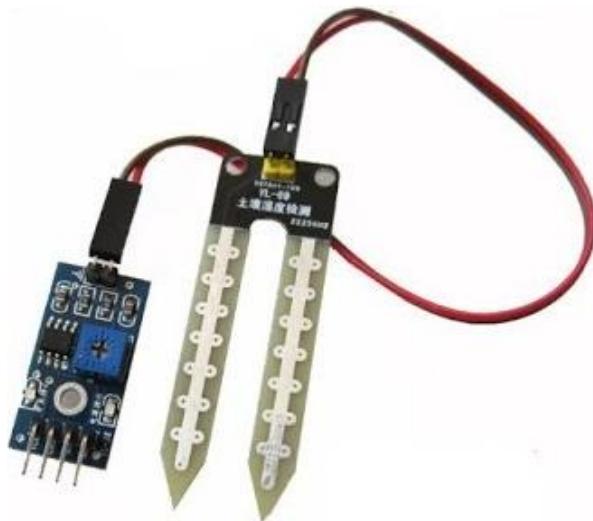


Figura 8: Comparador LM393 (lado esquerdo) e sensor YL-69 (lado direito).  
Fonte: Ponto da Eletrônica (2017).

### 1.1.7 Válvula Solenoide

Para irrigar a estufa do sistema, foram utilizadas sete válvulas solenoides de 220V. A solenoide é constituída por uma bobina de fio no formato de espiral em volta de um pistão. Como acontece com todos os eletroímãs, o campo magnético é criado quando a corrente elétrica passa pelo fio, e a solenoide move o êmbolo, de forma a criar uma pequena abertura. Assim, a pressão que passa por ela opera a vedação da válvula (NGELETRICA, 2017). A Figura 9 demonstra o uso da solenoide no sistema.



Figura 9: Válvula solenoide conectada ao cano de alimentação de água.

Para alimentar o sistema com água, foram utilizadas conexões e peças de material PVC conectadas a cada solenoide. Para realizar a irrigação adequada, foi aplicado o método de irrigação por aspersão. Esse método é realizado por meio de um bico microaspersor, conectado na ponta de cada solenoide, como mostra a Figura 9.

### 1.1.8 Alimentação de Energia

Para alimentar os relés, foi utilizada uma fonte K-Mex ATX 12V, modelo PX-400 modificada (Figura 10). Essa fonte disponibiliza uma tensão de 12V, corrente de 20A e um canal secundário de 5V e 22A. O Arduino, por meio do cabo USB, mantém uma corrente de 5V,

entretanto, para obter estabilidade na tensão, experimentos realizados indicaram que é aconselhável utilizar uma fonte para alimentar o circuito, visto que alimentar o Arduino pela porta USB é instável, ou seja, não mantém a tensão em exatos 5V e sim  $\approx 4,7V$ .

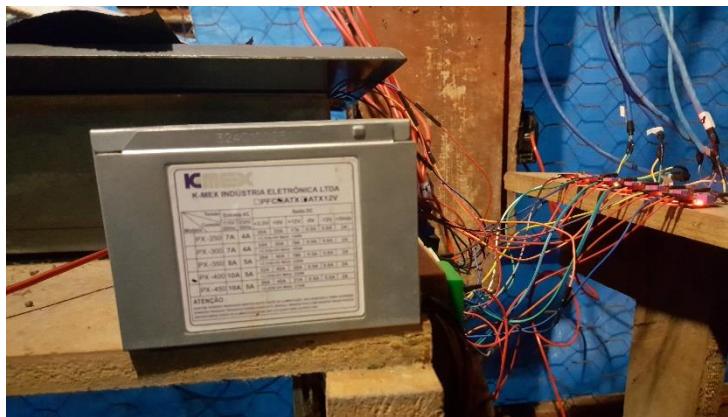


Figura 10: Fonte sendo utilizada para ligar o relé.

## 1.2 METODOLOGIA

### 1.2.1 Materiais Utilizados no Sistema

Além da placa Arduino Mega, do módulo relé e do módulo Ethernet, foram utilizados outros componentes para o desenvolvimento do sistema de irrigação automatizado, que são citados a seguir, na Tabela 1. Todo o protótipo do circuito foi realizado em *protoboards*.

Tabela 1: Materiais utilizados.

Item	Quantidade
Arduino Mega 2560	1
Módulo Ethernet W5100	1
Placa <i>protoboard</i>	3
Módulo relé de 8 canais	1
Válvulas solenoides	7
Sensor de temperatura DHT22	1
Higrômetros YL-69	6
Cabos ( <i>jumpers</i> )	86
Fonte 12V	1

### 1.2.2 Estrutura do Sistema de Irrigação

Essa proposta de automação de um sistema de irrigação foi desenvolvida, usando-se uma placa de prototipagem Arduino para a manipulação dos dados dos sensores. Foram utilizados sensores de umidade e temperatura, umidade do solo, válvulas solenoides e diversos encanamentos de material PVC, conforme a Tabela 1 e a Figura 11, para irrigação da estufa, juntamente de conexões de corrente 220V ou 110V para a alimentação de equipamentos.

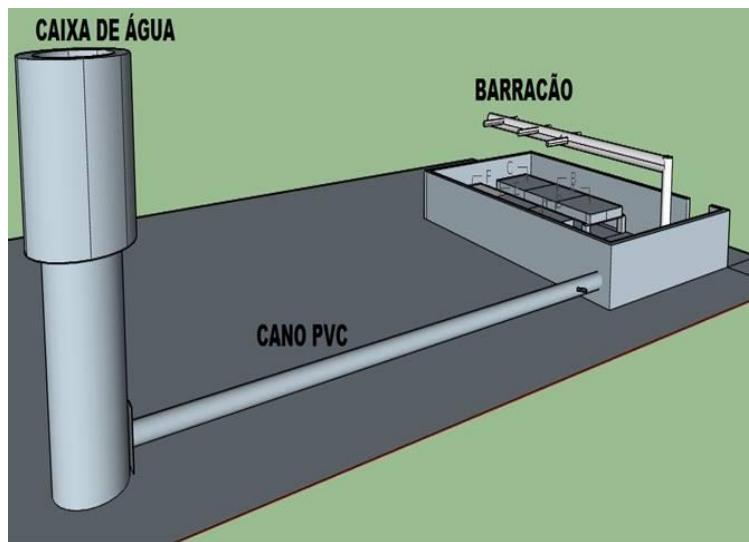


Figura 11: Cano PVC que faz a alimentação de água para a estufa.

### 1.2.3 Software em Java

Foi elaborado um *software*, para realizar a leitura dos dados enviados pelo Arduino, utilizando a linguagem de programação Java, no ambiente de desenvolvimento integrado (IDE) NetBeans. O NetBeans é uma ferramenta de desenvolvimento modular para uma gama de tecnologias de desenvolvimento de aplicativos (NETBEANS, 2017). A interface gráfica é demonstrada na Figura 12.

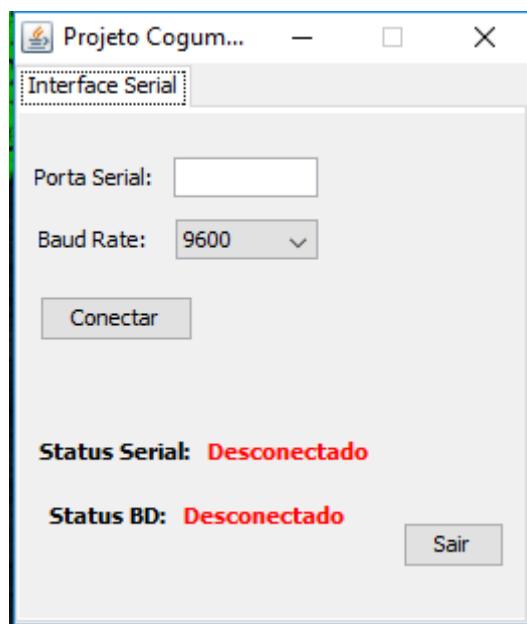


Figura 12: Interface gráfica do *software* programado em Java.

### 1.2.4 Armazenamento dos Dados

O armazenamento de dados foi feito por dois modos. De forma *online*, via Google Spreadsheet. De forma *offline*, por meio da comunicação serial disponível no Arduino, utilizando-se um cabo USB.

#### 1.2.4.1 Modo *online*

Com o propósito de realizar cálculos numéricos com as leituras de dados efetuados pelo sistema, foi utilizado o Google Spreadsheet. Para realizar a inserção desses dados no Google Spreadsheet foi utilizado o PushingBox, o qual é um serviço *cloud* que pode enviar notificações com base em chamadas API. A partir de um pedido, é possível enviar várias notificações, como um *push* (inserir um conteúdo), um *tweet* (forma de publicação de mensagens da rede social Twitter), um *e-mail* em tempo real (PUSHINGBOX, 2017). Foi utilizado o método GET para fazer a transmissão de dados. O método GET é utilizado para passar pequenas informações por meio da barra de endereços (URL). A Figura 13 demonstra o painel do serviço.

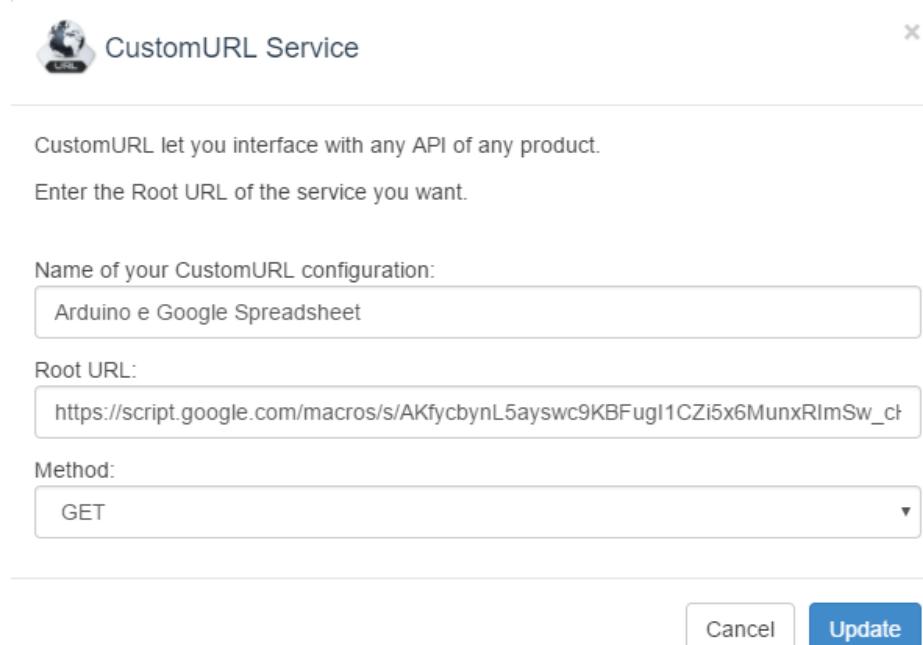


Figura 13: Serviço selecionado na API do PushingBox.

Fonte: PushingBox (2017).

Para estabelecer conexão com o PushingBox, foi utilizado o Google Script, via método de publicação, como um aplicativo *web*. O método se encontra nas ferramentas de *script* do Google Apps. O Google Script é uma linguagem de *script* em nuvem baseada em Javascript, que facilita a automação de tarefas em produtos da Google (2017). *Scripts* são roteiros seguidos por sistemas computacionais que trazem informações quando ações são efetuadas pelo programa. O *script* foi utilizado para obter o exato tempo de comunicação entre o Arduino e o PushingBox e também os valores lidos de cada sensor na aplicação, e inserir os valores na determinada planilha. É necessário informar a identificação da planilha na programação do *script* para que o aplicativo estabeleça comunicação correta com a planilha, conforme Figura 14.

```

1  if (e.parameter == undefined) {
2      result = 'No Parameters';
3  }
4  else {
5      var id = '12gz0lRnVjgrwS4G7mC0N57HOELJw5dMkTkX-sF25';
6      var sheet = SpreadsheetApp.openById(id).getActiveSheet();
7      var newRow = sheet.getLastRow() + 1;
8      var rowData = []; rowData[0] = new Date();
9      for (var param in e.parameter) {
10          Logger.log('In for loop, param=' + param);
11          var value = stripQuotes(e.parameter[param]);
12          switch (param) {
13              case 'temp':
14                  rowData[1] = value; //Value in column B
15                  break;
16              case 'umid':
17                  rowData[2] = value;
18                  break;
19              case 'umidA':
20                  rowData[3] = value;
21                  break;
22              case 'umidB': ...
23              case 'umidC': ...
24              case 'umidD': ...
25              case 'umidE': ...
26              case 'umidF': ...
27              default:
28                  result = "unsupported parameter";
29          }
30      }
31      Logger.log(JSON.stringify(rowData));
32      var newRange = sheet.getRange(newRow, 1, 1, rowData.length); newRange.setValues([rowData]);
33  }
34 }
35 
```

Figura 14: Trecho do código criado.

Fonte: Stackoverflow (2017).

O código consiste numa estrutura de tratamento de requisição (linha 1). Se o Arduino realizar uma requisição, pelo método GET, com os parâmetros indefinidos, o *script* não efetuará a escrita de dados na planilha.

Caso a requisição contenha dados definidos, o *script* estabelece a conexão com a planilha informada (linhas 5 e 6). A última linha preenchida na planilha é obtida pelo *script* (linha 7). A primeira coluna de dados é preenchida com a data e hora do momento da requisição (linha 8). A estrutura de repetição *for* é utilizada para ler os parâmetros enviados pela requisição (linha 9), isto é, enquanto houver parâmetros na requisição, a estrutura lerá cada parâmetro (linha 11) e o armazenará na variável *value*. Caso esse parâmetro seja igual ao informado na estrutura de tomada de decisão *switch* (linha 12), o *script* armazenará o conteúdo do parâmetro (variável *value*) na variável *rowData*.

Se a estrutura da requisição GET for definida, mas não contiver os parâmetros informados no *script*, a requisição será ignorada e o retorno da requisição será efetuado como *unsupported parameter* (linha 38). Essa requisição não efetua alterações na planilha.

Ao final do *script*, na linha 42, ocorre a inserção de uma nova linha na planilha, utilizando a função *setValues*, que contém os dados obtidos. O conteúdo inserido na planilha está armazenado na variável *rowData*.

#### 1.2.4.2 Modo off-line

A comunicação serial é feita mediante a transmissão de informação *bit a bit* sequencialmente, normalmente caracteres ASCII, por uma única via, a grande maioria dos

padrões existem vias de transmissão (TX) e recepção (RX) (MECAWEB, 2017). No Arduino Mega, é possível estabelecer comunicação serial com mais de uma porta, isto é, a placa Mega possui mais de um canal de conexão serial. Após receber os dados de cada sensor, é feita a concatenação em *string* (linhas 4 a 11), que são cadeias de caracteres que se comportam como um vetor, isto é, sua última posição não preenchida por um caractere é identificada como \0, entretanto, ao enviar os dados, foi necessário executar uma estrutura de repetição que fizesse o envio de cada caractere, dado que a comunicação serial é feita *bit a bit*, conforme a Figura 15, linhas 20 a 24, demonstra.

```

1  String data;
2  int tamanhoData;
3
4  data += "INSERT INTO tb_sensores (SENS_TEMP,
5      SENS_UMID,
6      SENS_CA, SENS_CB, SENS_CC, SENS_CD, SENS_CE, SENS_CF) ";
7
8  data += "values ("; data += t;
9  data += ","; data += u; data += ","; data += uA; data += ",";
10 data += uB; data += ","; data += uC; data += ","; data += uD;
11 data += ","; data += uE; data += ","; data += uF; data += "); ";
12
13
14 tamanhoData = data.length();
15 char query[tamanhoData];
16 data.toCharArray(query, tamanhoData);
17
18 int x=0;
19
20 do{
21     Serial.write(query[x]);
22     x++;
23     delay(1);
24 } while (query[x] != '\0');

```

Figura 15: Trecho do código que realiza a concatenação dos dados.

### 1.2.5 Biblioteca de Comunicação Serial do Java

Foi utilizada a biblioteca RXTX para Java. RXTX é uma biblioteca nativa Java que fornece comunicação serial e paralela para o Java Development Toolkit (JDK). O RXTX está sob a licença GNU LGPL, bem como suas distribuições binárias (FIZZED, 2017). Para estabelecer comunicação entre o *software* e a porta serial, foi utilizado o método de comunicação bidirecional com a porta serial. Esse método tem como objetivo estabelecer conexão com o dispositivo serial e depois interagir com ele, recebendo e enviando dados (RXTX, 2017). Os dados obtidos do sistema são lidos pelo *software* via comunicação serial, o qual realiza uma determina consulta de inserção, inserindo os dados na base de dados.

### 1.2.6 Banco de Dados

O banco de dados utilizado nesse sistema, foi o banco de dados MySQL. Apesar de o MySQL ser um *software* pago, a versão de comunidade do MySQL é uma versão gratuita do banco de dados de código aberto mais popular do mundo. Ele está disponível sob a licença GPL, e é apoiado por uma comunidade ativa de desenvolvedores de código aberto (MYSQL, 2017). A conexão com o MySQL foi realizada via MySQL Conector, presente na versão 5.1.23. Quando

conectado, é possível trabalhar com o MySQL no NetBeans, explorando e criando bases de dados, tabelas e executar consultas SQL (NETBEANS, 2017).

### 1.2.7 Sensores de Umidade do Solo

Utilizando-se higrômetro YL-69, a mensuração dos dados é feita de forma sequencial em cada canteiro. Conforme o valor obtido pelo pino analógico, o qual é determinado pelo limite de 0 a 1023, aciona-se a solenoide do respectivo canteiro. A mesma mensuração de dados pode ser realizada pelo pino digital dos sensores, sendo assim, a decisão que leva ao acionamento das solenoides é dada de acordo com a leitura de tensão que o Arduino obtém, podendo ser LOW ou HIGH. Foi utilizado o pino analógico, para obter uma leitura de dados mais precisa, entretanto, os dados lidos são invertidos. Assim, ao retirar o sensor de um meio não umidificado, o valor retornado é 1023 ou próximo a 1023, ao contrário, o meio está umidificado e os valores serão próximos de zero.

Foi utilizado um método de restringir os valores máximos obtidos pelo sensor. Esse método consiste em mergulhar o sensor num copo de água e anotar o valor retornado. Ao retirar o sensor do copo de água, impõe-se um limite no retorno dos dados utilizando a função Map, presente na programação do Arduino. A função Map remapeia um número de um intervalo para outro (ARDUINO, 2017d). Na versão 1.8.1 do Arduino, essa função encontra-se descrita como Map (*value, fromLow, fromHigh, toLow, toHigh*). Com a função Map, é possível realizar a conversão de valores para porcentagem. Sendo assim, em vez de ler valores dentro do limite de 0 a 1023, os valores são interpretados em porcentagem.

### 1.2.8 Sistema *Online*

A visualização dos dados pode ser efetuada pela plataforma Dweet (Figura 16), e sucessivamente pela plataforma Freeboard.

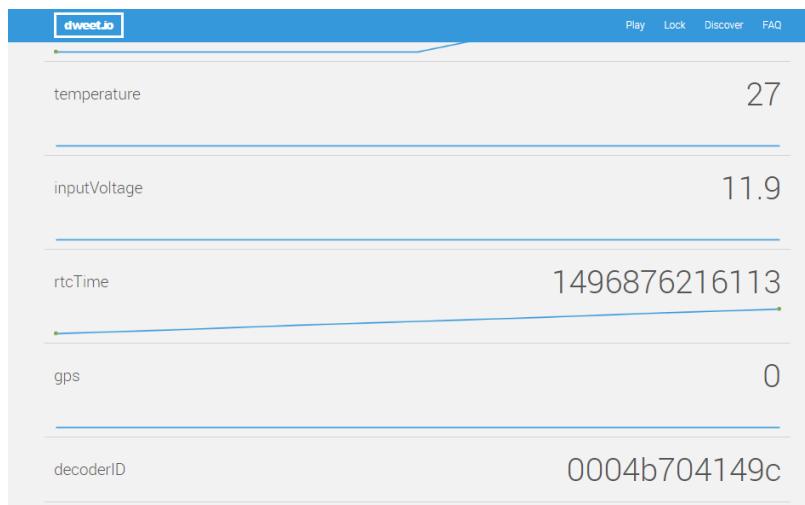


Figura 16: Dados recebidos pela *Dweet API*.

Fonte: Dweet (2017b).

A disponibilização dos dados em ambas as plataformas é feita pelo envio dos dados do Arduino, usando o módulo Ethernet, para o Dweet API. Entretanto, é necessário selecionar o Dweet como fonte de dados no Freeboard, isto é, o Freeboard disponibiliza um painel de controle para estabelecer a conexão com o Dweet. Com essa ferramenta, é possível criar gráficos estatísticos, imagens, importar mapas do Google e indicadores. Isso permite a exibição dos dados de forma efetiva. Também é possível selecionar outros tipos de *widgets*.

Os *widgets* utilizado no sistema foram Gauge e Sparkline, para dados de temperatura, umidade do ar e umidade do solo de cada canteiro.

### 1.2.9 Circuito

O circuito do sistema foi elaborado no *software* Fritzing, que é uma ferramenta *open source* para criar protótipos de projetos eletrônicos. Com essa ferramenta, é possível criar esquemas e projetar placas de circuito impresso (PCB) (SPARKFUN, 2017). A Figura 17 ilustra o esquema.

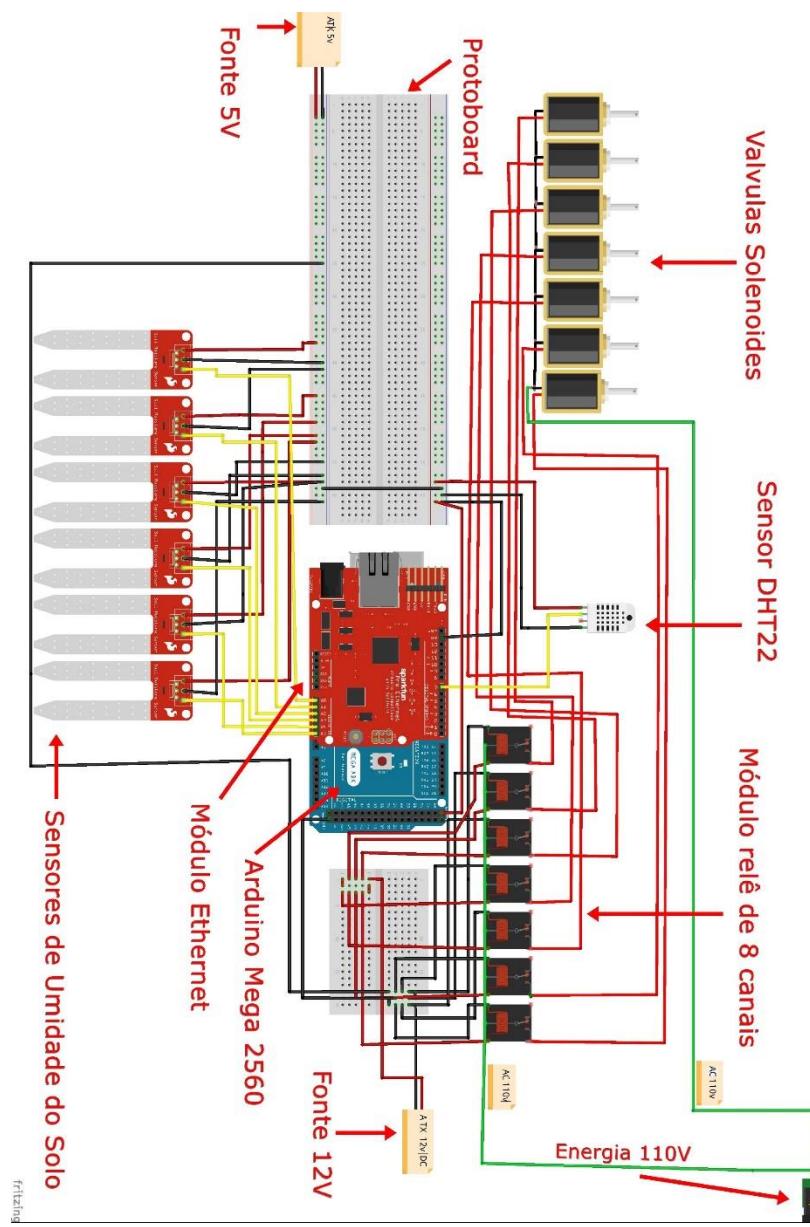


Figura 17: Circuito elaborado no *software* Fritzing.

### 1.3 TESTES

#### 1.3.1 Local

Os testes do sistema foram aplicados numa estufa que está localizada no distrito de São Clemente, na cidade de Santa Helena, Paraná. A estufa possui seis canteiros, caracterizados como (A), (B), (C), (D), (E) e (F), lado a lado, e cada canteiro possui 1m<sup>2</sup> de área. A Figura 18 demonstra o interior da estufa. Foram utilizadas lonas leves de cor azul, com perfurações em todo o material, nos lados direito e esquerdo, com a intenção de manter a umidade no interior da estufa.



Figura 18: Estufa utilizada para testes no sistema.

#### 1.3.2 Dados

O sistema foi ligado no dia 19 de março de 2017, permanecendo ligado até o dia 21 de março de 2017. Durante esse tempo, o sistema realizou a coleta de dados no intervalo de uma hora. Os dados foram caracterizados como temperatura do ar (Temp. Ar), umidade do ar (Umid. Ar), umidade do solo (A), umidade do solo (B), umidade do solo (C), umidade do solo (D), umidade do solo (E) e umidade do solo (F). Todos os dados foram representados em porcentagem. Entretanto, entre os 150 dados coletados nesse intervalo apenas 102 são legíveis devido à queda de energia que ocorreu onde a estufa estava localizada. A Tabela 2 demonstra esses dados.

Tabela 2: Dados coletados pelo sistema no intervalo de 3 dias.

N. <sup>º</sup>	Temp. Ar	Umid. Ar	Umid. Solo (A)	Umid. Solo (B)	Umid. Solo (C)	Umid. Solo (D)	Umid. Solo (E)	Umid. Solo (F)
1	21	88	89	77	81	77	71	73
2	21	88	89	78	81	77	72	73
3	20	86	89	78	81	77	72	74
4	20	89	89	78	81	77	72	74
5	20	89	89	77	81	77	73	73
6	19	89	89	77	81	78	73	73
7	20	91	89	77	81	77	73	74
8	17	94	88	77	79	77	72	73
9	17	94	88	77	79	77	71	72
10	17	93	88	76	79	77	71	72
11	17	92	87	76	79	77	71	72
12	17	92	88	76	79	77	71	72
13	18	92	88	77	79	77	71	72
14	19	91	88	77	79	77	71	71
15	19	86	88	77	79	77	71	72
16	20	83	88	77	79	77	69	71
17	21	76	88	77	79	78	71	72
18	21	71	87	77	79	77	71	72
19	21	67	90	77	79	78	71	72
20	22	65	83	74	78	77	71	73
21	22	63	80	75	79	77	72	73
22	23	61	78	76	79	77	73	73
23	23	61	78	74	78	77	73	74
24	24	57	75	76	79	77	73	73
25	24	57	74	74	79	77	73	74
26	25	55	75	75	79	77	73	74
27	25	52	74	73	79	77	73	74
28	25	51	75	73	79	77	72	73
29	25	49	75	74	79	77	73	73
30	25	48	75	74	79	78	73	74
31	26	48	77	75	79	78	73	74
32	27	46	76	74	79	77	72	74
33	26	46	77	74	80	78	72	74
34	27	45	78	74	80	78	71	73
35	27	46	77	74	80	77	70	74
36	27	46	79	74	80	78	71	74
37	26	45	80	74	80	78	72	74
38	27	44	80	74	80	78	71	73
39	28	45	80	75	81	78	71	73
40	28	44	81	75	81	78	72	74
41	27	43	81	74	80	78	72	75

42	29	42	81	75	80	78	72	74
43	29	42	77	74	80	78	72	74
44	27	57	75	79	80	85	81	80
45	27	55	75	79	80	85	81	80
46	26	50	75	79	80	85	81	-7
47	26	52	75	79	80	85	81	79
48	27	55	76	80	80	85	81	80
49	26	58	77	80	80	86	81	80
50	26	59	77	80	80	86	81	80
51	26	60	77	80	80	86	81	81
52	26	60	77	80	80	86	81	81
53	26	59	77	80	80	86	82	81
54	25	58	61	80	80	86	82	81
55	25	58	80	81	80	84	80	81
56	25	59	80	81	82	85	80	82
57	25	59	80	81	82	85	80	82
58	25	60	81	81	82	85	81	82
59	25	60	81	81	82	85	80	82
60	24	61	81	81	82	85	80	82
61	24	61	81	82	82	86	81	83
62	25	61	82	82	82	86	81	83
63	25	61	82	82	82	86	81	83
64	25	61	82	82	82	86	81	83
65	24	62	82	82	82	86	81	83
66	24	62	82	82	82	86	81	83
67	24	62	82	82	82	86	81	83
68	24	62	82	82	82	86	81	83
69	24	61	83	82	82	86	81	83
70	24	61	83	82	83	87	82	84
71	23	64	83	82	82	86	82	84
72	23	67	84	82	83	87	83	84
73	23	67	84	82	83	87	82	84
74	23	70	84	82	83	87	83	84
75	22	80	84	82	83	87	83	84
76	22	85	84	82	84	87	83	84
77	22	85	84	82	84	86	83	84
78	22	84	84	83	84	86	83	84
79	22	84	84	82	84	86	83	84
80	21	88	84	83	84	86	83	84
81	21	90	84	83	84	86	83	84
82	21	90	84	83	84	86	83	84
83	21	90	84	83	84	86	83	84
84	21	90	84	83	84	86	83	84

85	21	88	84	83	84	86	83	84
86	20	91	84	83	84	86	83	84
87	20	89	84	83	85	86	83	84
88	20	83	84	83	84	86	83	84
89	20	83	84	83	84	86	83	84
90	20	86	84	83	84	86	83	84
91	24	50	85	83	85	85	84	84
92	23	63	85	83	85	84	84	84
93	21	70	85	83	85	85	85	84
94	21	73	85	83	85	85	85	84
95	21	73	85	83	85	85	85	84
96	21	73	85	83	85	85	85	84
97	21	73	85	83	85	85	85	84
98	21	73	85	83	85	85	85	84
99	21	73	85	83	86	85	85	84
100	21	73	85	83	85	85	85	84
101	21	73	85	83	85	85	85	84
102	21	73	85	83	85	85	85	84

Com os dados obtidos nessa coleta, foi possível montar o gráfico da Figura 19, que demonstra a média de cada tipo de dado.

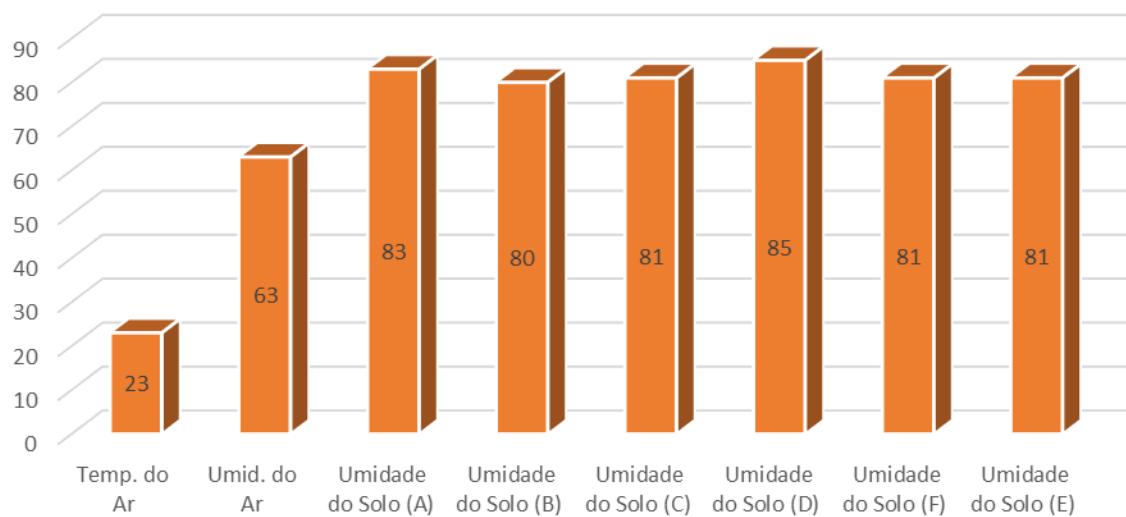


Figura 19: Gráfico das médias de temperatura do ar, umidade do ar e umidade do solo.

Durante o mês de março, no outono, as condições do tempo eram de, no mínimo, 19°C e, no máximo, 31°C (SIMEPAR, 2017) (Figura 20).

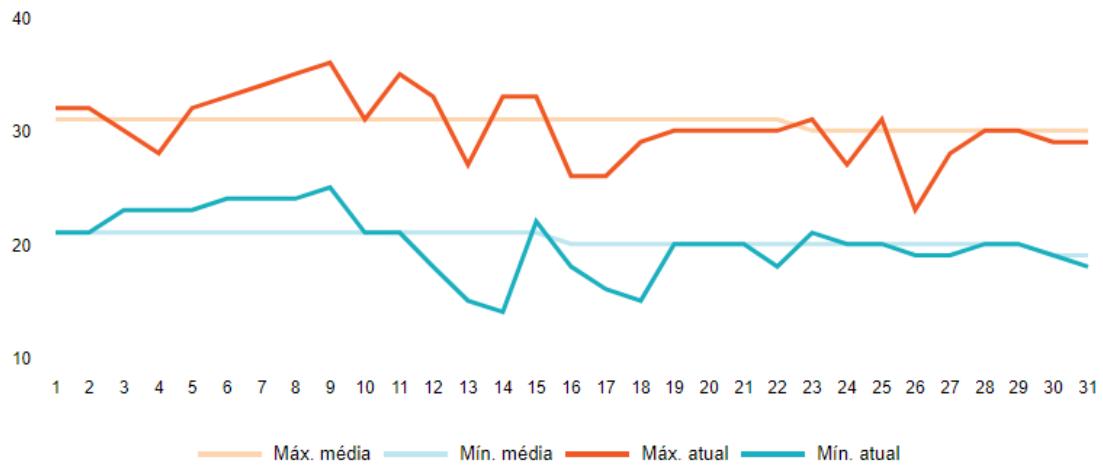


Figura 20: Gráfico da temperatura do ar durante o mês de março.

Fonte: ACCUWEATHER (2017).

Essa temperatura impactou a quantidade de vezes que o solo foi irrigado, isto é, com a temperatura elevada, o solo tende a secar numa velocidade maior. Apesar de o sistema não medir a quantidade de vezes que cada canteiro foi irrigado, o método de irrigação utilizado é suficiente para irrigar o canteiro, evitando desperdício de água.

## 2 RESULTADOS E DISCUSSÃO

Após testes realizados, foi descoberto que o sensor YL-69 não foi desenvolvido para permanecer por muito tempo em locais de alta umidade, isto é, a alta umidade causa dano ao circuito do sensor, conforme mostra a Figura 21. Consequentemente, ocorre perda de dados ou uma leitura de baixa precisão. Apesar de a umidade causar danos ao sensor, ele ainda continuou funcionando, entretanto, com precisão baixa e mensuração de dados não confiável.

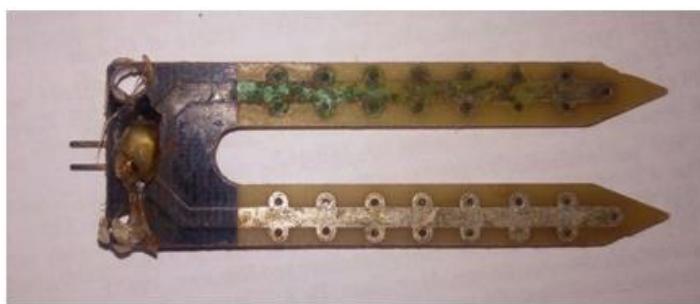


Figura 21: Sensor YL-69 corroído, após 3 dias em alta umidade.

Como o circuito do projeto é grande, foi observado que o Arduino não sustentava o circuito estável por meio de sua tensão de alimentação de 5V. A solução efetiva foi adicionar uma fonte extra, mantendo o Arduino e o módulo Ethernet alimentados. Assim, foi garantida a estabilidade de energia no sistema.

Utilizar o módulo Ethernet como *web server* causaria problemas de conexão e até mesmo perda de dados. Sabendo disso, foi utilizado o *web client* do Ethernet, assim, o chip Ethernet não processa tanta requisição, como o *web server* processaria.

Devido à baixa proteção no material do circuito do sensor de umidade do solo, observou-se que é necessário utilizar um isolante para proteger a área de contato.

Quando o sistema foi testado, pequenos detalhes tiveram de ser corrigidos, ao utilizar o relé. O Arduino deve controlar o relé, isto é, caso falte energia no sistema, o relé tem de

permanecer desligado, esperando o contato do Arduino para acionar a irrigação. Para corrigir esse problema, foi necessário estipular uma dependência entre as válvulas solenoides e o relé, com a utilização do canal normalmente aberto (NA), presente no relé de 8 canais. O canal normalmente aberto permite criar esta dependência pois espera um contato para acionar a válvula solenoide, assim evita o desperdício de água, caso o Arduino desligue.

## CONSIDERAÇÕES FINAIS

Esse trabalho apresentou uma proposta de automação de um sistema de irrigação para estufa utilizando dispositivos microcontrolados, desenvolvido em uma placa de prototipagem Arduino, utilizando sensores de temperatura, umidade e umidade do solo, válvulas solenoides, módulo relé e módulo de *Ethernet*. Após testes realizados em estufa, o sistema mostrou-se capaz de controlar as válvulas solenoides e os sensores com eficácia, entretanto, o sistema depende inteiramente da precisão do sensor de umidade de solo.

Sugere-se, para trabalhos futuros, elaborar um sensor ou utilizar outro modelo, de baixo custo, que faça a leitura da umidade do solo com mais precisão e resistência a variações climáticas, assim possibilitando ao sistema trabalhar em culturas de produção mais específicas.

## AGRADECIMENTOS

Os autores agradecem a Gustavo Girardi ter cedido a estufa para realizar testes no sistema.

## REFERÊNCIAS

- ACCUWEATHER. **Gráfico de temperaturas de março de 2017.** Disponível em: <<https://www.accuweather.com/pt/br/santa-helena/40017/march-weather/40017?monyr=3/1/2017&view=table>>. Acesso em: 21 set. 2017.
- AOSONG. **Módulo de Temperatura e Umidade.** Disponível em: <[http://img.filipeflop.com/files/download/Datasheet\\_DHT22\\_AM2302.pdf](http://img.filipeflop.com/files/download/Datasheet_DHT22_AM2302.pdf)>. Acesso em: 5 jun. 2017.
- ARDUINO. **O que é Arduino?** 2017a. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Acesso em: 5 jun. 2017.
- ARDUINO. **Arduino Mega 2560 e Genuino Mega 2560.** 2017b. Disponível em <<https://www.arduino.cc/en/Main/ArduinoBoardMega2560>>. Acesso em: 5 jun. 2017.
- ARDUINO. **Shield Ethernet para Arduino.** 2017c. Disponível em: <<https://www.arduino.cc/en/Main/ArduinoEthernetShieldV1>>. Acesso em: 28 mar. 2017.
- ARDUINO. **Map Reference.** 2017d. Disponível em: <<https://www.arduino.cc/en/reference/map>>. Acesso em: 6 jun. 2017.
- FOROUZAN, B. A.; FEGAN, S. C. **Protocolo TCP/IP.** 3.ed. Porto Alegre: AMGH, 2009.
- DWEET. **What is Dweetio?** 2017a. Disponível em: <<https://dweet.io/faq>>. Acesso em: 5 jun. 2017.
- DWEET. **What is Dweeting now?** 2017b. Disponível em: <<https://dweet.io/see>>. Acesso em: 5 jun. 2017.
- FILIPEFLOP. **Ethernet Shield W5100 para Arduino.** Disponível em: <<http://www.filipeflop.com/pd-6b62c-ethernet-shield-w5100-para-arduino.html>>. Acesso em: 25 jun. 2017.
- FREEBOARD. Disponível em: <<http://freeboard.io/>>. Acesso em: 25 jun. 2017.
- GOOGLE. **Build web apps and automate tasks with Google Apps Script.** Disponível em: <<https://www.google.com/script/start>>. Acesso em: 18 set. 2017.

GOUVEIA, A. R.; ILIUK, I. **Monitoramento de temperatura e umidade no interior de uma estufa.** In: SEMANA TECNOLÓGICA ACADÊMICA DE CIÊNCIA DA COMPUTAÇÃO DA UTFPR-SH, 3., 2016, Santa Helena, PR. *Anais...* Santa Helena: UTFPR, 2016. p.14-20. Disponível em: <<http://sh.utfpr.edu.br/setac/Anais2016.pdf>>. Acesso em: 20 set. 2017.

INFOWESTER. Disponível em: <<https://www.infowester.com/wifi.php#80211n>>. Acesso em: 25 jun. 2017.

MECAWEB. **Porta Serial.** Disponível em: <[http://www.mecaweb.com.br/elettronica/content/e\\_serial](http://www.mecaweb.com.br/elettronica/content/e_serial)>. Acesso em: 5 jun. 2017.

NETBEANS. **Conectando a um banco de dados MySQL.** Disponível em: <<https://netbeans.org/kb/docs/ide/mysql.html>>. Acesso em: 5 jun. 2017.

NETWORKWOLD. **Painéis da web facilitados.** Disponível em: <<http://www.networkworld.com/article/3131491/internet-of-things/freeboard-web-dashboards-made-easy.html>>. Acesso em: 19 mai. 2017.

NGELETRICA. **Saiba como funciona uma solenoide.** Disponível em: <<http://www.ngeletrica.com.br/destaque-1/saiba-funciona-solenoide/>>. Acesso em: 28 mar. 2017.

PUSHINGBOX. **Uma API para suas notificações.** Disponível em: <<https://www.pushingbox.com/>>. Acesso em: 19 maio 2017.

RXTX. **Comunicação bidirecional com a porta serial.** Disponível em: <[http://rxtx.qbang.org/wiki/index.php/Two\\_way\\_communication\\_with\\_the\\_serial\\_port](http://rxtx.qbang.org/wiki/index.php/Two_way_communication_with_the_serial_port)>. Acesso em: 5 jun. 2017.

QQTRADING. **Sensor de Umidade do Solo YL-69 com Higrômetro HC-38 e placa de módulo LM393.** Disponível em: <<http://qqtrading.com.my/soil-moisture-sensor-yl-69-hygrometer-w-hc-38-module-board-lm393>>. Acesso em: 5 jun. 2017.

SIMEPAR. **Estações do ano.** Disponível em: <[http://www.simepar.br/site/internas/conteudo/meteorologia/estacoes\\_ano.shtml](http://www.simepar.br/site/internas/conteudo/meteorologia/estacoes_ano.shtml)>. Acesso em: 21 set. 2017.

SONGLE RELAY. **RELAY ISO9002.** Disponível em: <[http://img.filipeflop.com/files/download/Datasheet\\_Rele\\_5V.pdf](http://img.filipeflop.com/files/download/Datasheet_Rele_5V.pdf)>. Acesso em: 5 jun. 2017.

SPARKFUN. **What is Fritzing?** Disponível em: <<https://learn.sparkfun.com/tutorials/make-your-own-fritzing-parts/what-is-fritzing>>. Acesso em: 3 out. 2017.

## O conhecimento tecnológico: a importância para o desenvolvimento

### Technological knowledge and its importance for a nation's development

LEITÃO, Dorodame Moura. O conhecimento tecnológico e sua importância. Possibilidades de sua transferência internacional. Brasília, Ci. Inf., n.10, v.2, p. 33-44, 1981. Disponível em: <<http://revista.ibict.br/ciinf/article/viewFile/149/149>>. Acesso em: maio 2017.

Dorodame Moura Leitão é um engenheiro civil, que ingressou na Petrobrás em 1959. Ele foi responsável pela implantação do curso de refino em Mataripe em 1963, atuando como professor e coordenador, e foi um dos responsáveis pela implantação do laboratório de Engenharia Química da Coppe, UFRJ. O Instituto Alberto Luiz Coimbra de Pós-graduação e Pesquisa de Engenharia (COPPE) é o maior centro de ensino e pesquisa em engenharia da América Latina (COPPE, 2017). Em 1971, Dorodame integrou o recém-formado Centro de Pesquisas Leopoldo Américo Miguez de Mello (CENPES), unidade da Petrobrás responsável pelas atividades de pesquisa e desenvolvimento (P&D) e engenharia básica da empresa (WIKIPÉDIA, 2017). Exerceu diversas atividades no Cenpes, dentre as quais, a chefia da Divisão de Tecnologia de Refinação, em 1972. Participou do primeiro planejamento estratégico desenvolvido no Cenpes (MUSEU DA PESSOA, 2017). Dorodame é autor de vários livros relevantes, como *Recordações das lutas pela tecnologia na Petrobrás* (LEITÃO, 2004), e do artigo aqui resenhado: *O conhecimento tecnológico e sua importância*.

O artigo em foco é um texto escrito em 1981, da área de conhecimento e desenvolvimento tecnológico. Tem como objetivo divulgar e debater a importância do conhecimento tecnológico no desenvolvimento de uma sociedade, tanto economicamente como política e socialmente – com destaque para a situação do Brasil, a partir da realidade vivida pelo autor: a da Petrobrás. Vários temas relacionados ao conhecimento tecnológico são tratados no texto, dentre eles: a dependência tecnológica dos países subdesenvolvidos em relação aos países mais desenvolvidos e como o conhecimento atribui poder a quem o detém.

No texto, o autor consegue transmitir para o leitor a importância de se desenvolver tecnologia para o autodesenvolvimento. De acordo com ele, desde os primórdios da humanidade – até os dias atuais – aqueles que detinham o conhecimento exerciam o poder. Era assim com os líderes místicos, que ouviam a voz de Deus, e com os escribas, que redigiam os livros sagrados. Quando o autor correlaciona conhecimento e poder, um marco histórico é citado para comprovar que o conhecimento se correlaciona com desenvolvimento e poder: a Revolução Industrial. A partir dela, o conhecimento teve importância ainda maior, bem como houve aumento na imposição de poder das nações ricas sobre as nações pobres.

Quanto aos avanços tecnológicos na Revolução Industrial, cabe destacar, com base em Iglesias (1981), que o século XVIII foi marcado pelo grande salto tecnológico nos transportes e máquinas. As máquinas a vapor, sobretudo os gigantes teares, revolucionaram o modo de produzir. Por um lado, a máquina substituiu o homem, gerando milhares de desempregados, por outro, baixou o preço de mercadorias e acelerou o ritmo de produção. Na área de transportes, houve a

invenção das locomotivas a vapor (as marias-fumaça) e dos trens a vapor. Foi possível transportar mais mercadorias e pessoas, em tempo mais curto e com custo mais baixo. As principais invenções técnicas da Revolução Industrial foram a lançadeira volante (John Kay), o tear mecânico (Cartwright), máquina a vapor (James Watt) e a locomotiva de Stephenson.

Voltando a Leitão (1981), ele sustenta que, antes da Revolução Industrial, a maioria das tecnologias surgia de maneira empírica, ou seja, de maneira natural, de acordo com a necessidade do ser humano. Com a alta participação da tecnologia nas indústrias, o conhecimento científico, que até então era desenvolvido pelas universidades e a elas ficava restrito, ou seja, não era aplicado em atividades industriais ou com fins lucrativos, passou a ter grande influência na produção de tecnologias, pois começou a ser aplicado na indústria. Aqui cabe comentar que, nesse sentido, a nação que despontou como mais poderosa da época e exerceu maior domínio tecnológico sobre as demais foi a Inglaterra, que possuía grandes reservas de carvão mineral, principal fonte de energia para movimentar as máquinas e as locomotivas a vapor, minério de ferro, principal matéria-prima utilizada no período, mão-de-obra em abundância e capital suficiente para financiar as fábricas e mercado consumidor (IGLESIAS, 1981). Além disso, antes da Revolução Industrial, na Inglaterra havia apenas duas universidades: Oxford e Cambridge, mas a revolução provocou carência de engenheiros, empregados de escritórios e profissionais. Por consequência, a educação tornou-se vital, então, bibliotecas, escolas e universidades foram fundadas por pessoas ou grupos particulares (ENCICLOVERB, 2006), e o conhecimento científico associou-se definitivamente à produção de tecnologia industrial, que a Inglaterra exportava para países que não a possuíam.

Na comparação entre conhecimento e desenvolvimento, o autor cita como os países subdesenvolvidos levam desvantagens com relação aos países desenvolvidos na questão tecnológica. Por ter de importar tecnologia dos países mais industrializados, os países com menos tecnologia sofrem com a falta de conhecimento para dominar completamente as tecnologias importadas. Apesar da importação de tecnologia permitir crescimento econômico maior para esses países, tal ascensão ainda é pouca, quando comparada à de países que produzem tecnologia. Essa dependência tecnológica acontece, inclusive em grande escala, no Brasil.

Um dos grandes motivos de isso acontecer, de acordo com o autor, portanto, levando-se em conta a década de 1970, é que as universidades brasileiras passaram muitos anos quase totalmente afastadas das pesquisas científicas. Na época em que o texto foi escrito, havia poucos programas de pós-graduação no Brasil, e os que havia se dedicavam mais a pesquisas fundamentais, desligadas da realidade nacional. Uma pesquisa mais atual, escrita por Cirani, Campanário e Silva (2012), demonstra grande aumento de cursos de pós-graduação nas universidades brasileiras nas últimas décadas, cerca de 4660, sendo a maioria (3866) de iniciativa pública. Segundo a análise realizada por Soares (2007) sobre o mesmo tema, apesar do aumento existente no investimento em pesquisas acadêmicas, isso não se refletiu no desenvolvimento tecnológico do país, por boa parte em função do investimento ocorrer de forma inconsistente e em quantidade insuficiente, tanto da iniciativa pública quanto privada, que investiu timidamente no desenvolvimento tecnológico. O investimento realizado no país é ainda mais baixo quando comparado com o da Coreia do Sul, onde, com alto investimento na formação de recursos humanos (graduação) pela indústria, o país conseguiu obter o retorno esperado, e o setor produtivo assumiu o papel de financiador das pesquisas, e financia cerca de 75% das pesquisas no país (SOARES, 2007).

Atualmente, apesar de estar longe de ser um dos países referência na produção de tecnologia, no Brasil existem empresas que são referência em desenvolvimento de tecnologia. Um exemplo disso é a própria Petrobrás, que já foi a empresa líder em tecnologia de exploração de petróleo em alto mar. No início de suas atividades de exploração de petróleo, na década de 1960, a empresa importava tecnologia de outros países para a extração de petróleo. As tecnologias importadas eram adaptadas às condições locais de produção. As empresas que forneciam tecnologia para a Petrobras exploravam petróleo em uma profundidade máxima de 400 metros, mas como no Brasil as áreas com poços de petróleo tinham profundidade média de 1000 metros, a empresa teve como única opção desenvolver suas próprias tecnologias (ORTIZ e COSTA, 2007).

A primeira experiência em águas profundas ocorreu com a Petrobras em 1984 [...]. O maquinário empregado nesse segmento foi se desenvolvendo e com o passar do tempo recordes de profundidade foram sendo batidos. Em 2005 já era possível extrair petróleo aproximadamente 6 mil metros de profundidade (EXPLORAÇÃO DE PETRÓLEO EM ÁGUAS PROFUNDAS, 2011).

Esse resultado parece ser consequência da organização, do planejamento e do comprometimento da equipe de engenheiros da Petrobrás, tal como Leitão descreve, 30 anos antes de essa produção tecnológica brasileira se concretizar. Sabe-se que após isso, nos últimos anos, a empresa entrou em crise, provocada por má-administração governamental, sobretudo corrupção, e somente agora começa a recuperar-se (PETROBRÁS, 2017).

De acordo com Dorodame, um dos principais problemas da dependência tecnológica dos países subdesenvolvidos seria que boa parte das dispendiosas tecnologias importadas por esses países não são completamente utilizadas. Uma das formas de resolver esse problema seria pelo processo conhecido como transferência tecnológica, que consiste em utilizar as tecnologias provenientes de outros países, mas adaptadas para as condições locais, para que possam contribuir para o desenvolvimento efetivo desses países. Existem dois tipos de transferência de tecnologia. O mais simples, embora englobe algumas dificuldades, é a transferência horizontal, que seria a transferência interna de tecnologia a uma companhia ou a um país. Esse tipo de transferência, apesar de ser simples, pode gerar alguns problemas, e o mais comum deles é a falta de preparação do usuário que vai receber a tecnologia. O outro tipo é a transferência vertical, que consiste em um processo de transportar atividades de pesquisa e desenvolvimento para um ambiente comercial. A transferência vertical de tecnologia é dividida em cinco níveis, que são: operação industrial (contratação *turn key*, não se sabe ao certo nem como funciona a tecnologia); a construção das indústrias (companhias nacionais vão se capacitando a construir as fábricas); nacionalização da indústria (construção dos equipamentos usados na indústria, com mão-de-obra e matéria-prima nacionais, mas ainda com tecnologia importada); a aquisição dos conhecimentos necessários para a montagem dos equipamentos (conhecimentos necessários ao projeto de detalhamento mecânico, necessário à montagem dos equipamentos e construção de indústria); e então os níveis mais sofisticados e difíceis de conseguir, o da engenharia básica ou comercial (metodologias, programações de computador e conhecimentos empíricos resultantes de anos de experiência); e o dos conhecimentos fundamentais ou da pesquisa tecnológica (interesse em que sejam transmitidas informações sobre os processos, métodos e equipamentos usados na obtenção desses conhecimentos básicos). A Petrobrás passou por todos esses níveis, até se tornar detentora de suas tecnologias.

Por muitos anos, nos países em desenvolvimento, acreditava-se que, tendo um bom nível de pesquisa científica, haveria um desenvolvimento tecnológico natural de acordo com a necessidade do país. Apesar do grande crescimento científico no Brasil, se comparado com a época em que a obra foi escrita, até os dias atuais, o aumento das pesquisas científicas não se refletiu no desenvolvimento tecnológico do país. De acordo com Soares (2007), o Brasil teve crescimento científico alto nos últimos 30 anos, e atualmente responde por 1,3% dos artigos científicos publicados em revistas indexadas internacionais, porém isso não corresponde ao crescimento tecnológico, já que o número de patentes depositadas por universidades e institutos de pesquisas brasileiros em 2007 era de aproximadamente 100 patentes por ano, um número reduzido, quando comparado ao da Coreia do Sul, que chega a 2 mil patentes por ano. Segundo Mazzola e Armellini (2011), aproximadamente 430 pedidos de patentes são realizados no Brasil anualmente, e a maioria é realizada por pessoas físicas ou empresas privadas. O baixo número de pedidos por patentes feito pelos institutos científicos no Brasil demonstra que as pesquisas realizadas estão distantes da realidade do país e das indústrias brasileiras.

O autor conclui que nenhum país em desenvolvimento conseguirá atingir independência econômica, cultural e política sem que se seja alcançada a autonomia tecnológica. Para isso, não seria necessário o bloqueio de importação de tecnologias de outros países, mas sim a capacidade de o país regular o fluxo de tecnologia que circula em seu interior. Para que se possa alcançar autonomia tecnológica, os países em desenvolvimento precisam, de alguma forma, assimilar a tecnologia já existente e, juntamente com a criação de tecnologia própria, resolver problemas internos de maneira mais rápida. Para que essa absorção de conhecimento externo seja realizada com maior sucesso, um dos caminhos encontrados pelo autor é que haja equipes capacitadas para receber as tecnologias importadas, para que as pessoas possam aproveitar ao máximo esses conhecimentos e agrupá-los, ordenadamente, de modo a ser úteis na produção de bens e serviços para a comunidade.

No início do texto, Dorodame menciona que, apesar de seus graves problemas sociais, o Brasil é um dos países menos desenvolvidos que têm melhores condições para chegar ao próximo século em uma situação de desenvolvimento político, econômico e social semelhante à das mais avançadas nações, e que essa expectativa é compartilhada por muitos estudiosos de política mundial. Ele cita um estudo americano, no qual o Brasil, tendo em vista seus recursos naturais, territórios e população, foi classificado em terceiro lugar no mundo em termos de poder potencial, apenas precedido por União Soviética e Estados Unidos. Dorodame acreditava que o desenvolvimento tecnológico seria o fator que pesaria mais decisivamente na transformação desse potencial em realidade – e o grande avanço tecnológico mundial das últimas décadas, sobretudo na área da computação, confirma que ele estava certo. No entanto, passados quase 40 anos da publicação do texto, o Brasil ainda é um (eterno) país do futuro, pois, embora tenha atingido melhores níveis de educação e desenvolvimento social desde então, ainda não pode ser considerado um produtor e exportador de tecnologia, nem uma nação totalmente autônoma nesse sentido. Assim, não explora suficientemente ainda os recursos que o colocaram como um dos países mais promissores no estudo americano, pois o novo século chegou e nossa posição não é semelhante à de nações mais avançadas – está longe disso, especialmente em educação, saúde e segurança. Inclusive, não dispomos de muitas tecnologias (ou não sabemos usá-las), algumas aliás antigas, que nos auxiliariam nessas áreas sociais.

Para que o cenário de dependência tecnológica brasileira mude, algumas medidas importantes podem ser tomadas, como, por exemplo, maior aproximação do

governo, para que as pesquisas se direcionem à solução de problemas voltados para a realidade nacional. Algo que também poderia auxiliar o desenvolvimento tecnológico do país seria a maior aproximação entre empresas e pesquisas – as universidades –, para que os projetos desenvolvidos pudessem ser aplicados na prática e na criação de produtos úteis, pois, como visto, a iniciativa privada brasileira ainda investe timidamente no país. Assim, ainda precisamos todos – governo, empresa, universidade – nos educar para aproveitar nosso potencial e ocupar a posição de país desenvolvido, em que Leitão tanto acredita.

Felipe Carvalho Funck

Graduando em Ciência da Computação pela Universidade Tecnológica Federal do Paraná (UTFPR), câmpus Santa Helena  
*funck@alunos.utfpr.edu.br*

## REFERÊNCIAS

- CIRANI, C. B. S.; CAMPANARIO, M. A.; SILVA, H. H. M. **A evolução do ensino da pós-graduação senso estrito no Brasil:** Análise exploratória e proposições para pesquisa. Campinas, Sorocaba, SP, v. 20, p.163-187, 2015.
- COPPE – Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia da Universidade Federal do Rio de Janeiro. **Apresentação.** Disponível em: <<http://www.coppe.ufrj.br/pt-br/a-coppe/apresentacao>>. Acesso em: set. 2017.
- ENCICLOVERB. **Revolução Industrial.** 2006. <<http://www.klickeducacao.com.br/2006/enciclo/encicloverb/0,5977,IGP-415,00.html>>. Acesso em: set. 2017.
- EXPLORAÇÃO DE PETRÓLEO EM ÁGUAS PROFUNDAS. UnB, Introdução a Engenharia Mecânica. 2011. Disponível em: <<http://unbiem.blogspot.com.br/2011/12/exploracao-de-petroleo-em-aguas.html>>. Acesso em: set. 2017.
- IGLESIAS, Francisco. **Revolução Industrial.** São Paulo: Brasiliense, 1981.
- LEITÃO, D. M. Recordações das Lutas pela tecnologia na Petrobrás. Rio de Janeiro: 2004. Disponível em: <<http://www.aepet.org.br/noticias/pagina/11072/Recordaes-das-lutas-pela-tecnologia-na-Petrobrs>>. Acesso em: set. 2017.
- MAZZOLA, C. R. R. M.; ARMELLINI, F. Desenvolvimento tecnológico do Brasil no cenário mundial: Um estudo exploratório por meio de indicadores de patentes. **Abepro**, Belo Horizonte, MG, out. 2011.
- ORTIZ, J. B. N.; COSTA, A. J. D. A Petrobrás e a exploração de petróleo offshore no Brasil: um approach evolucionário. **Revista Brasileira de Economia**, Rio de Janeiro, v.61, n.1, jan./mar, 2007.
- PETROBRÁS. Apresentamos lucro líquido de R\$ 5 bilhões nos primeiros nove meses de 2017. **Fatos e Dados**, nov. 2017. Disponível em: <<http://www.petrobras.com.br/fatos-e-dados/apresentamos-lucro-liquido-de-r-5-bilhoes-nos-primeiros-nove-meses-de-2017.htm>>. Acesso em: nov. 2017.
- SOARES, M. S. F. Desenvolvimento tecnológico e inovação no Brasil: Desafios na área de biotecnologia. **Novos Estudos – CEBRAP**, São Paulo, jun. 2007.
- WIKIPÉDIA. **Centro de Pesquisas Leopoldo Américo Miguez de Mello.** Disponível em: <[https://pt.wikipedia.org/wiki/Centro\\_de\\_Pesquisas\\_Leopoldo\\_Americo\\_Miguez\\_de\\_Mello](https://pt.wikipedia.org/wiki/Centro_de_Pesquisas_Leopoldo_Americo_Miguez_de_Mello)>. Acesso em: set. 2017.

**Referência:**

LEITÃO, D. M. O conhecimento tecnológico e sua importância. Possibilidades de sua transferência internacional. Brasília, Ci. Inf., n.10, v.2, p. 33-44, 1981. Resenha de: FUNCK, F. C. Conhecimento tecnológico: a importância para o desenvolvimento. In: SEMANA TECNOLÓGICA ACADÊMICA DE CIÊNCIA DA COMPUTAÇÃO DA UTFPR-SH, 4., 2017, Santa Helena, PR. **Anais...**, Santa Helena: UTFPR, 2017. p.138-143. Disponível em: <<http://sh.utfpr.edu.br/setac/Anais2017.pdf>>.

## Pensamento Computacional no Brasil: desafio para a escola

### Computational Thinking in Brazil: a challenge for Education

BORDINI, A.; AVILA, C.; CUNHA, M. M.; Cavalheiro, S. A. C.; FOSS, L. Desdobramentos do Pensamento Computacional no Brasil. In: V CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE), 2016, Uberlândia. *Anais...*, 2016. p.200-209.

O texto “Desdobramentos do Pensamento Computacional no Brasil” foi escrito por Adriana Bordini (doutoranda em Computação), Christiano Avila (docente), Monica Cunha (mestrando em Computação), Simone Cavalheiro (docente de graduação e pós-graduação em Computação) e Luciana Foss (docente de graduação em Ciência da Computação) do Centro de Desenvolvimento Tecnológico da Universidade Federal de Pelotas (UFPel), no ano de 2016.

O texto tem como objetivo apresentar um levantamento de artigos, dos últimos seis anos no Brasil, que objetivam disseminar o pensamento computacional no ensino básico do País. Os dados coletados são embasados em 45 artigos publicados entre 2010 e 2015, em periódicos e anais de congressos brasileiros, com temáticas relacionadas ao Pensamento Computacional. É subdividido em quatro seções: Introdução, Metodologia de Busca, Pensamento Computacional no Brasil e Considerações Finais.

O pensamento computacional é uma expressão primeiramente utilizada pelo matemático americano Seymour Papert, em 1980. Papert é o teórico mais conhecido sobre o uso de computadores na educação, um dos pioneiros da inteligência artificial e criador da linguagem de programação LOGO, em 1967, inicialmente para crianças, em uma época quando os computadores eram limitados e não existia interface gráfica nem Internet. Ele foi influenciado pelas teorias do construcionismo, que significa a construção do conhecimento, a partir da realização de uma ação concreta que resulta em um produto palpável, desenvolvido com o concurso do computador, que seja de interesse de quem o produz (WIKIPÉDIA, 2017).

A introdução traz a definição do pensamento computacional (PC), com base em Wing (2006), o qual é um método para solução de problemas, projetos de sistemas e a compreensão do comportamento humano baseado nos fundamentos e técnicas da Ciência da Computação. Ainda com base em Wing (2006), argumenta o texto que o pensamento computacional é uma habilidade básica a ser desenvolvida em todas as crianças em idade escolar, assim como ler, escrever e fazer as operações aritméticas básicas, e que, dessa forma, o estudo do PC é importante para toda a sociedade, pois são conhecimentos adquiridos para a vida toda, tanto pessoal quanto profissional, que auxiliam na busca de soluções de problemas, em qualquer área.

Voltando ao texto, a introdução apresenta também questões analisadas a partir dos artigos pesquisados, relacionados à área do PC no Brasil. As questões são: 1) Qual tem sido o objetivo de estudo dos trabalhos na área?; 2) Qual é o público-alvo e o número de participantes da pesquisa?; e 3) Quais ambientes ou ferramentas têm sido adotados?; 4) Quais são os resultados reportados?

A Metodologia de Busca inicia com a forma como foram pesquisados os 45 trabalhos analisados no estudo. Para realizar a busca desses artigos, consideraram-se os principais veículos da área de Informática na Educação no Brasil: Simpósio Brasileiro de Informática na Educação (SBIE), Workshop de Informática na Escola

(WIE), Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação (WAlgProg), Revista Brasileira de Informática na Educação (RBIE), Workshop sobre Educação em Computação (WEI), Workshop-Escola de Informática Teórica (WEIT), Revista Novas Tecnologias na Educação (RENOTE) e Workshop de Desafios da Computação Aplicada à Educação (DesafIE).

Dentre os trabalhos pesquisados, identificaram-se sete diferentes objetivos: introdução de conceitos do PC (denotado por INT na Tabela); proposta ou avaliação de metodologias (MET); investigação, apresentação ou avaliação de ferramentas para o ensino de computação no ensino básico (FER); atração de jovens para a área de computação (ATR); desafios de implantação e levantamentos sobre PC (LEV); formação de professores (FOR); e correlação entre as competências de diretrizes nacionais e internacionais com o PC (COR). Foram identificados, em relação ao público-alvo, diferentes focos de atuação: Ensino Fundamental (EF); Ensino Médio (EM); Educação Básica (EB), que envolve alunos do EF e EM; Ensino Técnico (ET); Ensino Superior (ES); Professores (PROFE); e Profissionais (PROFI). Alguns trabalhos não envolvem público-alvo e são denotados por NA na tabela, indicando que o item não se aplica.

A terceira seção – Pensamento Computacional no Brasil – apresenta um resumo geral das respostas para as questões consideradas para a pesquisa:

Qual tem sido o objetivo de estudo dos trabalhos na área?

Resposta: Dos 45 artigos analisados, 20 relatam experiências de projetos que visam a introduzir conceitos fundamentais do PC e que são efetivados, em geral, por meio de cursos, minicursos ou oficinas de linguagens ou ambientes de programação e/ou robótica. Os trabalhos classificados nessa categoria buscam desenvolver nos aprendizes as habilidades do pensamento computacional, como o raciocínio lógico, a capacidade de formulação e resolução de problemas, a promoção da autonomia e da criatividade.

Qual é o público-alvo e o número de participantes da pesquisa?

Resposta: Das experiências de ensino realizadas, a média de participação foi de 30 alunos do EF e 29 alunos no EM – com desvio padrão de 13,7 e 25,5 respectivamente. Nas demais categorias, houve apenas um relato de experiência, incluindo 24 alunos da EB, 6 alunos do ET, 10 alunos do ES e 103 professores.

Quais ambientes ou ferramentas têm sido adotados?

Resposta: A ferramenta mais utilizada em projetos de PC é o ambiente Scratch, que aparece em 12 trabalhos, seguido da Computação Desplugada, com 11 ocorrências. Os trabalhos que adotam ferramentas buscam, além de apresentar os conceitos da computação, o desenvolvimento do raciocínio lógico, da capacidade de formulação e resolução de problemas e a promoção da autonomia e da criatividade.

Quais são os resultados reportados?

Resposta: Os resultados apresentados, em sua maioria, são positivos em relação à efetividade das ações na área do PC, indicando que os alunos tiveram desempenho satisfatório nas atividades e avaliações realizadas, que os jogos utilizados ou desenvolvidos facilitaram a compreensão dos conceitos da computação e que as atividades trabalhadas aumentaram o interesse pela área, entre outros. Nessa seção, o texto traz os softwares e trabalhos, cuja aplicação se mostrou mais efetiva na escola. Ainda, relacionando o desempenho dos estudantes no Enem, relata o texto (2016, p.207) que:

[...] existe uma correlação moderada entre o desempenho dos estudantes no ENEM e a experiência destes em programação. Na análise do teste de PISA com relação aos conceitos do PC, descreve-se que os conceitos de abstração e análise de dados foram cobertos em 100% e 98,76% das questões, respectivamente. Decomposição de problemas foi coberto em aproximadamente 39%, representação de dados em 6%, coleta de dados em quase 4% e algoritmos e procedimentos em aproximadamente 2%. Automação, paralelismo e simulação não foram abordados.

Por fim, o texto conclui que há aumento do interesse pela área da computação, por todos os públicos trabalhados, “inclusive as meninas que foram foco de alguns estudos”. Sabe-se que ainda há carência de público feminino nessa área. Também relata que há maior esclarecimento ou conhecimento sobre o tema PC, sobre as habilidades comuns entre a Matemática e o PC, sobre propostas de metodologias e formas de avaliar o PC. Sobre essa conclusão, percebe-se que não há números para precisá-la, no entanto, são resultados positivos, embora tímidos. Então, o país deve fazer esforços para ampliá-los, e assim alinhar-se ao que outros países já colocaram em prática.

Fora do Brasil, no que se refere à disseminação do pensamento computacional, cabe destacar que países da Europa, como a Estônia e a Grécia, vêm introduzindo a programação ou a Ciência da Computação desde o ensino infantil até o nono ano como parte das disciplinas obrigatórias. Segundo a UK Department for Education (2013), a Inglaterra alterou a disciplina obrigatória de Informática, denominada ICT, que explorava as ferramentas de escritório, substituindo-a por Computing, estruturada no tripé: Ciência da Computação, Tecnologia da Informação e Letramento Digital.

Na maior parte das propostas implantadas ou dos estudos realizados, a ideia é reavivar a programação por meio de atividades como *coding computer science* ou *computer programming*, objetivando a criação de condições para o desenvolvimento do pensamento computacional. Outros países têm preocupação mais ampla do que simplesmente aprender a programar, e estão buscando novas maneiras de explorar os conceitos computacionais no sentido de criar condições para o desenvolvimento do pensamento computacional (VALENTE, 2016).

A Base Nacional Comum Curricular (BNCC), um documento que define o que é ensinado nas escolas do Brasil inteiro, prevê que os estudantes devem, ao longo da formação básica, desenvolver competências cognitivas e socioemocionais para sua formação. Esse documento visa a orientar a elaboração do currículo específico de cada escola, sem desconsiderar as particularidades metodológicas, sociais e regionais de cada uma (BRASIL, 2017). São 10 competências gerais, e uma delas é: 5) utilizar tecnologias digitais de comunicação e informação de forma crítica, significativa, reflexiva e ética nas diversas práticas do cotidiano, incluindo as escolares, ao se comunicar, acessar e disseminar informações, produzir conhecimentos e resolver problemas (PANTELIADES, 2016). Apesar disso, não é o que se constata na realidade da escola brasileira, em que, quando há salas de computadores, os alunos os utilizam para jogar ou assistir filmes, sobretudo nas séries iniciais.

Um estudo realizado por Geraldes et al. (2017) constatou que 1) o computador é muito utilizado para automação das tarefas, já que esta é uma importante habilidade associada ao PC, porém, 2) ao avaliar as ferramentas ou *softwares* mais utilizados pelos professores da educação profissional e tecnológica, observou-se a preferência por aplicativos de processamento de texto, planilhas de cálculo e editores de imagens e multimídia. Ao cruzar esses dois fatores, os autores

inferiram que as tarefas automatizadas são básicas, como digitar textos, fazer cálculos e editar imagens. Assim, eles concluem que, apesar de se fazer uso constante do computador como ferramenta de trabalho, ainda não há indícios de uso do pensamento crítico associado aos fundamentos da computação na busca de soluções para problemas mais complexos na prática pedagógica dos professores da educação profissional e tecnológica.

Outro exemplo refere-se à lógica de programação, que permite realizar um encadeamento coerente de instruções em uma sequência lógica para a resolução de problemas, constituindo um dos saberes elementares para o desenvolvimento do pensamento computacional. Todavia, frequentemente esse conteúdo é ensinado de forma isolada, sem que haja correlação com conceitos de outras áreas do conhecimento ou mesmo sem levar em conta uma contextualização para o aprendiz. Dessa forma, uma vez que o educando não comprehende a aplicabilidade dos conteúdos que lhe foram ensinados, o processo de aprendizagem é prejudicado e esta não se torna significativa (GOMES e MELO, 2013).

O Brasil ainda precisa colocar em prática o pensamento computacional no ensino básico, uma vez que o estudante se desenvolverá melhor a partir dos princípios do PC. Hoje é difícil imaginar uma sociedade sem computadores, pois a computação está presente em tudo. Dessa forma, entende-se que quem estiver alheio a ela estará alheio ao mundo. Nesse sentido, o texto resenhado revela, por meio de pesquisa, a importância da introdução do PC no ensino básico. Recomenda-se sua leitura por acadêmicos e docentes das áreas da computação, por pedagogos e pessoas que estão diretamente ligadas à educação, os quais podem alterar sua metodologia de ensino de modo a contribuir mais para melhorar a efetividade da escola e ampliar as relações entre os conteúdos e a vida em sociedade, já a partir do ensino básico, como deve ser.

Larissa Gabriela Dias Vidal

Graduanda em Ciência da Computação pela Universidade Tecnológica Federal do Paraná (UTFPR), câmpus Santa Helena  
*larissa\_dias\_vidal@hotmail.com*

## REFERÊNCIAS

- BRASIL. Ministério da Educação. **A base**. Guia de leitura. Disponível em: <<http://basenacionalcomum.mec.gov.br/a-base>>. Acesso em: set. 2017.
- GERALDES, W. B.; FERNEDA, E.; MARIZ, R. S.; ALONSO, L. B. N. O Pensamento Computacional no Ensino Profissional e Tecnológico. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 6, 2017. **Anais dos workshops do...**, 2017. Disponível em: <<http://www.brie.org/pub/index.php/wcbie/article/view/7476/5272>>. Acesso em: set. 2017.
- GOMES, T. C. S.; MELO, J. C. B. O Pensamento Computacional no Ensino Médio: Uma Abordagem Blended Learning. In: XXI WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, 2013, Maceió, Alagoas. **Anais...**, 2013.
- PANTELIADES, Daniela. BNCC: Tudo que você precisa saber sobre a Base Nacional Comum Curricular. 2016. Disponível em: <<http://appprova.com.br/bncc-base-nacional-comum-curricular/>> Acesso em: 09 set. 2017.
- VALENTE, J. A. Integração do pensamento computacional no currículo da educação básica: diferentes estratégias usadas e questões de formação de professores e avaliação do aluno. **Revista e-Curriculum (PUCSP)**, v. 14, p. 864-897, 2016.

WIKIPÉDIA. Seymour Papert. 2017. Disponível em:  
<[https://pt.wikipedia.org/w/index.php?title=Seymour\\_Papert&oldid=49093840](https://pt.wikipedia.org/w/index.php?title=Seymour_Papert&oldid=49093840)>. Acesso em: 09 set. 2017.

---

**Referência:**

BORDINI, A.; AVILA, C.; CUNHA, M. M.; Cavalheiro, S. A. C.; FOSS, L. Desdobramentos do Pensamento Computacional no Brasil. In: V CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIÉ), 2016, Uberlândia, MG. *Anais do...*, 2016. p.200-209. Resenha de: VIDAL, L. G. D. Pensamento Computacional no Brasil: desafio para a escola. In: SEMANA TECNOLÓGICA ACADÊMICA DE CIÊNCIA DA COMPUTAÇÃO DA UTFPR-SH, 4., 2017, Santa Helena, PR. *Anais...*, Santa Helena: UTFPR, 2017. p.144-148. Disponível em: <<http://sh.utfpr.edu.br/setac/Anais2017.pdf>>.

---

## O jogo da imitação: o reconhecimento tardio

### The imitation game: a late recognition

O JOGO da imitação. Direção: Morten Tyldum, Produção: Teddy Schwarzman e Grahan Moore. Estados Unidos e Reino Unido: Warner Bros., Black Bear Pictures, Bristol Automotive, 2014.

**O**filme *O jogo da imitação* foi dirigido pelo norueguês Morten Tyldum, com roteiro de Graham Moore. É uma produção da Warner Bros., Black Bear Pictures e Bristol Automotive. Lançado em 2014, tem duração de 1h54min e é uma produção inglesa em associação com os Estados Unidos. Baseia-se no livro biográfico *Alan Turing: The enigma*, de Andrew Hodges (1983). Turing foi interpretado pelo ator britânico Benedict Cumberbatch. *O Jogo da imitação* recebeu indicações para muitos prêmios, e venceu o Oscar de Melhor Roteiro Adaptado. O filme teve muito boa aceitação de público, faturou US\$227 milhões, e é considerado uma produção de alto nível. Recebeu boas críticas, mas também lhe foi cobrado, por conhecedores da vida de Turing, um pouco mais de fidelidade aos fatos.

O filme narra parte da história da vida de Alan Mathison Turing, matemático britânico que nasceu em 1912, em Londres. O filme inicia com a polícia investigando uma invasão à casa de Turing, em 1951, na qual nada foi roubado. A polícia acaba suspeitando da suposta invasão e decide investigar Turing, imaginando que ele seja um espião, uma vez que não há registros do governo sobre ele. Na verdade, quem invadira a casa fora um garoto de programa, o qual ele contratava às vezes. Chamado para interrogatório, Turing revela que participou de um projeto secreto da inteligência britânica em Bletchley Park, durante a II Grande Guerra e conta essa história.

O filme divide-se em três épocas da vida de Turing: sua pré-adolescência, em 1927, a época da II Guerra, 1941, e a atualidade, então, 1951, com a investigação da polícia. As três histórias se intercalam e se relacionam no filme. Em sua pré-adolescência, o garoto Turing se apaixonou por um colega do colégio interno, Christopher Marcom – seu único amigo na escola. No filme, os meninos são afastados um do outro por causa da morte prematura de Marcom, nas férias escolares. Durante a II Guerra Mundial, Turing trabalhou juntamente com um grupo de criptógrafos britânicos para decodificar a máquina Enigma, usada pelos nazistas para criptografar as mensagens de modo a torná-las ininteligíveis para o inimigo. O segredo da Enigma era trocado todos os dias. Líder do grupo, durante dois anos, Turing construiu uma máquina eletromecânica que decifrou os códigos de criptografia da Enigma, e finalmente a guerra encaminhou-se para o fim, com a derrota dos alemães. O nome dado a essa máquina era Christopher. Com ela, reduziu-se a guerra em dois anos e, estima-se, salvaram-se 14 milhões de pessoas.

Ao ser interrogado pela polícia, que faz uma investigação e descobre que os arquivos de Turing foram apagados pelo governo, ele explica o que é o jogo da imitação, que foi teorizado por Turing para testar uma inteligência artificial. No jogo, há interação por meio de perguntas e respostas escritas via uma máquina, um computador, e no final um juiz diz se quem respondeu às perguntas é uma pessoa ou uma máquina. Depois de responder às perguntas da polícia e contar a história de Bletchley Park, Turing diz ao policial: “O que eu sou, sou uma máquina ou um ser

humano? Sou um herói de guerra ou um criminoso?”. O policial responde que não pode julgá-lo, e Turing diz que então o policial não pode ajudá-lo em nada.

Turing era professor na Universidade de Cambridge quando foi condenado por atentado ao pudor – o homossexualismo era proibido na Inglaterra – e teve de escolher entre a prisão e a castração química. Optou pela castração química para poder continuar trabalhando na máquina Christopher, que, então, segundo Turing, estava muito mais inteligente. De acordo com o filme, os efeitos da terapia hormonal afetaram sua capacidade de trabalhar, seu raciocínio, até mesmo sua habilidade para resolver palavras-cruzadas, jogo em que ele sempre fora excelente.

Após um ano de terapia hormonal, o filme descreve que ele se suicidou aos 41 anos, em 7 de junho de 1954. Todo o trabalho dele e do grupo foi mantido em segredo pelo governo por 50 anos. Após a guerra acabar, tudo foi destruído, queimado, por ordem do governo britânico.

Em linhas gerais, o filme pode ser considerado a memória do reconhecimento da participação de Turing na II Guerra Mundial. Por ser homossexual declarado, a importância de Turing foi renegada pelo governo britânico. Essa injustiça foi corrigida em 2013, quando a rainha Elizabeth concedeu a ele um perdão póstumo real, e ele foi reconhecido pelo governo britânico como um homem excepcional, com uma mente brilhante (JORNAL DO BRASIL, 2013). Apesar disso, Turing é considerado uma espécie de “Pai da Computação” pelos profissionais da área desde 1975, tendo em vista seus estudos, os quais começaram a delinear a ideia de computabilidade, e a máquina teórica de Turing, que inspirou as gerações seguintes, chamadas de Máquinas de Turing. A máquina teórica de Turing tornou possível o processamento de símbolos, ligando a abstração de sistemas cognitivos e a realidade concreta dos números: base da inteligência artificial (LEMOS, 2014).

Voltando ao filme, cabe ressaltar que se trata de uma obra de ficção, que não tem compromisso único com a verdade. Uma obra cinematográfica pode ser baseada em fatos reais, mas os produtores têm de considerar o que o público quer ver, o que vai agradar, emocionar e garantir o retorno financeiro. A abordagem feita no filme é apenas uma dentre tantas outras possíveis, e seria ingenuidade tomá-la como retrato da verdade. Assim, é necessário fazer uma pesquisa sobre a vida de Alan Turing para cotejar com o filme e se ter uma ideia mais próxima da realidade, no que diz respeito aos fatos envolvidos. Nesse sentido, destacam-se alguns pontos, a seguir.

O filme foi criticado por usar licenças históricas em demasia, a fim de tornar-se mais atrativo ao público. Como resultado disso, deixa a desejar na precisão histórica e distancia-se da realidade para se aproximar da ficção. Conforme destaca Lessa (2017, s.p.), o diretor e o roteirista disseram que o filme “não buscava ser um relato preciso da história de Turing e a quebra do Enigma”, mas “apenas ser uma versão que retratasse a história de forma emocionante e cinematográfica”. Por conta disso, Turing é mostrado como “um sujeito solitário, difícil de conviver e questionado pelos superiores”, quando na verdade ele era “comprovadamente sociável, respeitado em Bletchley Park e parte de um esforço coletivo de criptografia”. Ainda, omite-se a “contribuição polonesa na decodificação do Enigma”, há personagens sobre os quais “não há provas de que tenham convivido com Turing” e há alteração de eventos, como a prisão de Turing ocorrer “um ano antes da vida real”. Também, o filme mostra um “viés de espionagem inexistente e a criação de dois detetives fictícios para a investigação”.

Quanto à contribuição polonesa, conforme Barczack (2014, p. 18),

Há muitas descrições sobre a participação de outros matemáticos poloneses e de como, por muitos anos antes da segunda guerra, na Agência de Cifras denominada “Biuro Szyfrów”, os criptanalistas [liderados pelo matemático Marian Rejewski] conseguiram decifrar mensagens secretas da Alemanha nazista cifradas pela [Enigma]. [...] *O sucesso de Rejewski na decifração da complexa máquina Enigma é um dos maiores feitos na criptanálise, mostra o seu imenso poder intelectual* [Simon Singh].

Os poloneses sabiam das alterações feitas na Enigma, como a inserção de rotores, e se adaptavam a elas, desenvolvendo novos métodos de “quebra” da máquina – como as folhas perfuradas de Zigalski e as Bomby, réplicas da Enigma que eles chegaram a criar, conseguindo sempre decifrá-la (BARCZACK, 2014).

Assim, os poloneses souberam antes que a Polônia seria atacada pela Alemanha nazista e decidiram fornecer cópias da Enigma à França e à Gâ-Bretanha. Antes, os criptoanalistas queimaram todo o trabalho, para que os alemães não soubessem da decifração (BARCZACK, 2014). Então, com a chegada dessas descobertas a Bletchley Park, Turing, a partir das estratégias idealizadas por Rejewski e Zigalski, as Bomby e as folhas perfuradas, que os britânicos reconstruíram, conseguiu, com sua equipe, descobrir padrões rígidos nas mensagens e decifrá-las, apesar da nova codificação que apresentavam. Houve, inclusive, encontro entre Turing e os criptoanalistas poloneses. Fica claro que muito mais tempo, fatos, pessoas e conhecimentos estiveram envolvidos no processo de decifração da Enigma por Turing e sua equipe do que faz crer o recorte, bastante reduzido, apresentado pelo filme.

Por fim, apesar dos efeitos da terapia à base de estrogênio – como o desenvolvimento dos seios –, há relatos de que, antes de sua morte, contrariamente ao que mostra o filme, o trabalho transcorria bem, Turing estava alegre e interagia socialmente, com normalidade e boa disposição (ROCHA, 2017). Essa contradição não causa surpresa, dado que se trata de uma obra de ficção, como dito, cujo objetivo principal é cativar o espectador.

A partir de tudo isso, percebe-se que o filme é fiel a seu gênero discursivo – obra de ficção – pois não se preocupa com a história em si, em ser leal aos fatos, mas apenas com a vida de Turing, falhando ao atribuir a ele uma personalidade questionável e a decifração total e quase solitária da Enigma. Isso mostra que se cometeu outra injustiça, agora com os poloneses, que foram totalmente negligenciados, mas cuja importância foi tão grande quanto a de Turing no trabalho com a Enigma. Assim, fica aqui a recomendação de leitura do livro *A indecifrável Enigma* (BARCZACK, 2014), que conta a história da Enigma e inclusive traz os métodos matemáticos usados pelos poloneses para decifrá-la por anos a fio.

De qualquer modo, pode-se dizer que parte da temática do filme, sobretudo no que diz respeito ao trabalho de Turing para o governo na decodificação da Enigma, torna-o uma obra à qual se deve assistir, especialmente por interessados em Ciência da Computação, matemática, história – essa última com ressalvas. Esse episódio da II Guerra Mundial definitivamente merecia ser registrado, a injustiça de 50 anos com Turing precisava ser corrigida e divulgada. Cabe ao público que lhe assiste, contudo, o discernimento e a interpretação apurada, menos emocional e mais plausível e desconfiada, próxima, portanto, da realidade, do grande profissional da computação que Turing foi, da grande contribuição que legou à sociedade, e especialmente, no que tange a seu desempenho na II Guerra Mundial, das milhões de vidas que – juntamente com uma equipe de grandes cientistas, dentre os quais estão os grandes criptanalistas poloneses – ajudou a salvar.

Vera Lúcia V. S. Araújo

Graduanda em Ciência da Computação da Universidade Tecnológica Federal do Paraná (UTFPR), câmpus Santa Helena  
*vera.2015@alunos.utfpr.edu.br*

## REFERÊNCIAS

- BARCZACK, L. C. **A indecifrável Enigma**: História da máquina e dos matemáticos que a decifraram. 2.ed. São Paulo: Clube dos Autores, MMXIV (2014).
- HISTÓRIA DA COMPUTAÇÃO. Disponível em: <<http://computerstory.weebly.com/preacute---histoacuteria.html>>. Acesso em: abr. 2017.
- JORNAL DO BRASIL. Rainha concede perdão a Alan Turing após 59 anos. 24/12/2013. Disponível em: <<http://www.jb.com.br/ciencia-e-tecnologia/noticias/2013/12/24/rainha-concede-perdao-a-alan-turing-apos-59-anos/>>. Acesso em: set. 2017.
- LEMOS, Leonardo. Alan Turing. **Enciclopédia Maxwell**. 2014. Disponível em: <<https://enciclopediamaxwell.wordpress.com/2014/07/28/alan-turing/>>. Acesso em: set. 2017.
- LESSA, V. O jogo da imitação. **Crítica**. Disponível em: <<http://www.proximoparagrafo.com.br/2017/01/critica-o-jogo-da-imitacao.html>>. Acesso em: set. 2017.
- ROCHA, Phelipe. **Tributo a Alan Turing**. Disponível em: <<https://codepen.io/pheliperocha/pen/WwJmPp>>. Acesso em: set. 2017.

---

### Referência:

O JOGO da imitação. Direção: Morten Tyldum, Produção: Teddy Schwarzman e Grahan Moore. Estados Unidos e Reino Unido: Warner Bros., Black Bear Pictures, Bristol Automotive, 2014. 1h54min. Resenha de: ARAÚJO, V. L. V. S. O jogo da imitação: o reconhecimento tardio. In: SEMANA TECNOLÓGICA ACADÉMICA DE CIÊNCIA DA COMPUTAÇÃO DA UTFPR-SH, 4., 2017, Santa Helena, PR. Anais..., Santa Helena: UTFPR, 2017. p.149-152. Disponível em: <<http://sh.utfpr.edu.br/setac/Anais2017.pdf>>.

---



Semana Tecnológica Acadêmica de Ciência da Computação da UTFPR-SH  
**Santa Helena, Paraná**  
<http://sh.utfpr.edu.br/setac/>