



**UNIVERSIDADE FEDERAL DO PARÁ  
CAMPUS UNIVERSITÁRIO DE TUCURUÍ  
FACULDADE DE ENGENHARIA DA COMPUTAÇÃO**

**VISÃO COMPUTACIONAL APLICADA: SISTEMA EMBARCADO PARA  
CONTROLE DE CONSUMO ENERGÉTICO DE EQUIPAMENTOS ELÉTRICOS**

**LEONARDO NUNES GONÇALVES**

**TUCURUÍ**

**2019**



**UNIVERSIDADE FEDERAL DO PARÁ  
CAMPUS UNIVERSITÁRIO DE TUCURUÍ  
FACULDADE DE ENGENHARIA DA COMPUTAÇÃO**

**LEONARDO NUNES GONÇALVES**

**VISÃO COMPUTACIONAL APLICADA: SISTEMA EMBARCADO PARA  
CONTROLE DE CONSUMO ENERGÉTICO DE EQUIPAMENTOS ELÉTRICOS**

Trabalho de Conclusão de Curso  
apresentado para obtenção de grau de  
engenheiro da Computação, Faculdade de  
Engenharia da Computação, do Campus  
Universitário de Tucuruí.

**TUCURUÍ**

**2019**

**VISÃO COMPUTACIONAL APLICADA: SISTEMA EMBARCADO PARA  
CONTROLE DE CONSUMO ENERGÉTICO DE EQUIPAMENTOS ELÉTRICOS**

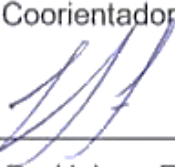
Este trabalho foi julgado adequado em 18/01/2019 para obtenção do grau de Engenheiro da Computação, aprovado em sua forma final pela banca examinadora que atribuiu o conceito EXC.



Prof. Dr. Bruno Merlin  
Orientador



Prof. Dr. Fabricio de Sousa Farias  
Coorientador



Prof. Dr. Heleno Fülber

Membro da Banca Examinadora



Prof. Dr. Marco José de Souza

Membro da Banca Examinadora



Prof. Dr. Bruno Merlin

Diretor da Faculdade de Engenharia da Computação

“In Memoriam” de Tereza de Melo, Benedito Evilásio e Maria da Luz.

## **AGRADECIMENTOS**

A Deus Su, por todas as proteções, orientações e concessões.

Meus pais, Jorge e Iolete e avô Eliésio, pois sem o suporte dos senhores, este e muitos dos meus sonhos não se realizariam. Meus tios Socorro e Arodinei pelo imenso apoio. Meu irmão Leandro que pela competitividade sempre me motivou.

Meus orientadores, Prof. Dr. Bruno Merlin e Prof. Dr. Fabrício de Sousa Farias, pela compreensão, confiança e incentivo para realização deste trabalho.

Meus amigos e membros do Laboratório de Programação Extrema, pela parceria e troca de conhecimentos, especialmente Eliezer Miranda que muito contribuiu no desenvolvimento deste projeto.

O prazer no trabalho aperfeiçoa a obra.  
(Aristóteles)

## RESUMO

A partir da constatação de que empresas e instituições consomem grande quantidade de energia elétrica e isto causa impacto direto em seus orçamentos, verificou-se a necessidade de tornar os ambientes corporativos empresariais e institucionais mais autônomos no que diz respeito a operação de seus equipamentos elétricos, para isso, neste trabalho será apresentado e testado um sistema embarcado, na forma de mínimo produto viável (MVP), que faz uso de visão computacional e redes neurais artificiais (RNA) para controle de consumo energético de equipamentos elétricos. Com esta finalidade realizou-se um levantamento técnico sobre as metodologias de desenvolvimento de sistemas embarcados, visão computacional e RNA. Para validação, testes foram conduzidos visando verificar a eficiência do algoritmo *backpropagation*, isto é, baseado em RNA, como alternativa para o processo de classificação. Os testes foram realizados durante as etapas de treinamento e após a instalação da RNA treinada no MVP proposto. A partir dos resultados é possível indicar que a solução se comporta de maneira satisfatória quando embarcada, além disso, os resultados também indicam o grande potencial desta ferramenta proposta como solução alternativa para o controle de consumo energético.

Palavras-chave: sistema embarcado; consumo energético; redes neurais artificiais; *backpropagation*; visão computacional.

## **ABSTRACT**

Based on the finding that companies and institutions consume a large amount of electric energy and this has a direct impact on their budgets, it was verified the need to make corporate and institutional environments more autonomous with respect to the operation of their electrical equipment, for this, in this work will be presented and tested an embedded system, in the form of minimum viable product (MVP), that makes use of computer vision and artificial neural networks (RNA) to control the energy consumption of electrical equipment. To this end, a technical survey was carried out on the methodologies of embedded systems development, computational vision and RNA. For validation, tests were conducted to verify the efficiency of the backpropagation algorithm, that is, based on RNA, as an alternative to the classification process. The tests were performed during the training stages and after the installation of the RNA trained in the proposed MVP. From the results it is possible to indicate that the solution behaves satisfactorily when shipped, in addition, the results also indicate the great potential of this tool proposed as alternative solution for the control of energy consumption.

Keywords: embedded system; energy consumption; Artificial neural networks; backpropagation; computer vision.



## LISTA DE FIGURAS

Figura 1 – Plataforma <i>Arduino</i> .....	23
Figura 2 – Plataforma <i>Intel Galileo</i> .....	24
Figura 3 – Plataforma <i>Raspberry Pi</i> .....	25
Figura 4 – Processo de filtragem da imagem por máscara de convolução .....	29
Figura 5 – Passos do processamento digital de uma imagem.....	31
Figura 6 – Exemplo de neurônio artificial .....	32
Figura 7 – Rede Neural Artificial de Múltiplas Camadas.....	35
Figura 8 – Processo recursivo da regra da cadeia .....	37
Figura 9 – Arquitetura geral do sistema.....	38
Figura 10 – Comunicação do sistema com ambiente de desenvolvimento.....	39
Figura 11 – Esquemático do <i>hardware</i> desenvolvido .....	40
Figura 12 – Esquemático final do sistema (MVP).....	41
Figura 13 – Planta baixa do LABEX .....	42
Figura 14 – Vetorização das amostras da base de dados .....	44
Figura 15 – Segmentação das amostras da base de dados .....	45
Figura 16 – Fluxograma do <i>software</i> embarcado .....	46
Figura 17 – Treinamento da RNA com 500 épocas.....	48
Figura 18 – Treinamento da RNA com 1000 épocas.....	49
Figura 19 – Treinamento da RNA com 1500 épocas.....	50
Figura 20 – Exemplos dos testes de validação .....	53

## LISTA DE TABELA

Tabela 1 - Tipos de eficiência energética .....	14
Tabela 2 - Estimativa de consumo médio de centrais de ar condicionado.....	15
Tabela 3 – Patentes de invenção .....	17
Tabela 4 – Relação de custo das plataformas de desenvolvimento .....	26
Tabela 5 – Melhores índices de treinamento.....	51
Tabela 6 – Percentual de acerto das RNA .....	51
Tabela 7 – Relação de entradas e status do MVP .....	52

## LISTA DE EQUAÇÕES

Equação (1) – Coeficientes de eficiência energética ( $CEE$ ) .....	15
Equação (2) – Custos mensais de operação ( $CM_{op}$ ) .....	16
Equação (3) – Convolução: Resultado do pixel central ( $R$ ) .....	28
Equação (4) – Convolução: Valor de máscara ( $S$ ) .....	29
Equação (5) – RNA: Calculo de propagação (rede) .....	36
Equação (6) – RNA: Valor de ativação ( $saída_{Calculada}$ ) .....	36
Equação (7) – RNA: Erro médio quadrático ( $E$ ) .....	36
Equação (8) – RNA: derivada parcial do erro em relação ao peso ( $\frac{\partial E}{\partial w}$ ) .....	37
Equação (9) – RNA: Fórmula geral para atualização dos pesos ( $w_{novo}$ ) .....	37

**LISTA DE SIGLAS E ABREVEATURAS**

APIs	Application Programming Interfaces
BMVA	Associação Britânica de Visão de Máquina
CSI	<i>Camera Serial Interface</i>
FIRJAN	Federação das Indústrias do Estado do Rio de Janeiro
GHz	Giga Hertz
INMETRO	Instituto Nacional de Metrologia, Qualidade e Tecnologia
INPI	Instituto Nacional da Propriedade Industrial
LABEX	Laboratório de Programação Extrema
MHz	Mega Hertz
MVP	Mínimo produto viável
NCO	Neurônios na camada oculta
PBE	Programa Brasileiro de Etiquetagem
PROCEL	Programa Nacional de Conservação de Energia Elétrica
RNA	Redes Neurais Artificiais
SSH	<i>Security Shell</i>
USB	<i>Universal serial bus</i>

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>14</b>
1.1 TRABALHOS RELACIONADOS.....	16
1.2 PATENTES RELACIONADAS .....	17
1.3 OBJETIVOS.....	19
1.4 METODOLOGIA .....	19
<b>2 FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>22</b>
2.1 SISTEMAS EMBARCADOS.....	22
<b>2.1.1 Plataformas padronizadas.....</b>	<b>23</b>
<b>2.1.2 Custo benefício de produção .....</b>	<b>25</b>
2.2 VISÃO COMPUTACIONAL .....	27
<b>2.2.1 Pré-processamento de imagem .....</b>	<b>27</b>
<b>2.2.2 Filtros espaciais.....</b>	<b>28</b>
2.3 REDES NEURAIS ARTIFICIAIS.....	31
<b>2.3.1 Aprendizado Artificial .....</b>	<b>32</b>
<b>2.3.2 Backpropagation .....</b>	<b>34</b>
<b>3 PROJETO ELETRIC CONTROL.....</b>	<b>38</b>
3.1 ARQUITETURA GERAL DO SISTEMA.....	38
3.2 PROJETO DE <i>HARDWARE</i> .....	39
3.3 METODOLOGIA DE DESENVOLVIMENTO DO <i>SOFTWARE</i> .....	41
<b>3.3.1 Instalação do sistema.....</b>	<b>42</b>
<b>3.3.2 Captura, organização e pré-processamento da base de dados.....</b>	<b>43</b>
<b>3.3.3 Treinamento da RNA.....</b>	<b>44</b>
3.4 FUNCIONAMENTO DO <i>SOFTWARE</i> EMBARCADO. ....	45
<b>4 AVALIAÇÃO DE DESEMPENHO DA RNA E ETAPA DE TESTE .....</b>	<b>48</b>
<b>5 CONCLUSÃO.....</b>	<b>54</b>
<b>6 REFERÊNCIAS.....</b>	<b>56</b>
<b>APÊNDICE A.....</b>	<b>63</b>
<b>APÊNDICE B.....</b>	<b>64</b>

## 1 INTRODUÇÃO

O consumo energético de corporações públicas ou privadas, impacta diretamente em seu orçamento. De acordo com o estudo realizado pela Federação das Indústrias do Estado do Rio de Janeiro – FIRJAN (2016) o consumo de energia elétrica pode representar mais de quarenta por cento (40%) dos custos de produção destas corporações. Em alguns casos, a despesa pode representar um valor significativo do lucro do negócio. Parte desta é advinda de desperdícios causados pela má gestão do acionamento, uso e desligamento de equipamentos elétricos.

A partir dessa problemática, Goldemberg e Lucon (2006) enfatizam que a consideração de parâmetros de eficiência energética nesses aparelhos é a maneira mais efetiva de reduzir os custos orçamentários e impactos ambientais.

Para regulamentação desses critérios de eficiência para cada aparelho, foi criado o Programa Brasileiro de Etiquetagem (PBE) que é coordenado pelo Instituto Nacional de Metrologia, Qualidade e Tecnologia (INMETRO) com parceria do Programa Nacional de Conservação de Energia Elétrica (PROCEL). A Tabela 1 apresenta a escala de eficiência energética proposta pelo PBE.

**Tabela 1** - Tipos de eficiência energética definidos pelo PBE

<b>Tipo</b>	<b>Coeficiente de eficiência energética (CEE)</b>
<b>A</b>	$3,23 < CEE$
<b>B</b>	$3,02 < CEE \leq 3,23$
<b>C</b>	$2,81 < CEE \leq 3,02$
<b>D</b>	$2,60 \leq CEE \leq 2,81$

**Fonte:** ENCE – Etiqueta Nacional de Conservação de Energia (2018).

Os índices estabelecidos na Tabela 1 enfatizam que eficiência energética significa obter o mesmo serviço (“realizar trabalho”) com menos energia (EPE, 2018). A escala é composta por quatro níveis, isto é, A, B, C e D, os quais indicam os coeficientes de eficiência energética (*CEE*) em quilo Joule por Watt hora (*kJ/Wh*) e são divididos em intervalos fechados. Dentre os intervalos, o nível A classifica os equipamentos com melhor eficiência energética, isto é, os que possuem um *CEE* maior a 3,23.

Os critérios de eficiência energética definidos pelo PBE são usados no Brasil para classificar todos os equipamentos elétricos comercializados. Dentre esses, destacam-se como os mais frequentes em uma empresa: os computadores, notebooks, celulares, projetores multimídia, impressoras e centrais de ar condicionado. Neste cenário, Weise e Hornburg (2007) destacam as centrais de ar como os aparelhos que mais tributam no gasto de energia de empresas e escritórios, pois são responsáveis por grandes desperdícios.

O cálculo do  $CEE$  para centrais de ar pode ser obtido a partir da divisão do valor da capacidade total de refrigeração ( $q_{rt}$ ) pela potência elétrica total ( $E$ ), conforme equação (1):

$$CEE = \frac{q_{rt}}{E} \text{ (kJ/Wh)} \quad (1)$$

Para exemplificar o  $CEE$ , a Tabela 2 apresenta uma estimativa do gasto energético de aparelhos em função do consumo médio mensal de diferentes centrais de ar condicionado. Para simplificação, é considerado determinado uso mensal do aparelho e uma quantidade fixa de uso diário, sendo respectivamente trinta (30) dias e oito (8) horas por dia.

**Tabela 2** - Estimativa de consumo médio de centrais de ar condicionado

ID	APARELHOS	CONSUMO MÉDIO MENSAL (CMM)
		(kWh)
1	Split menor ou igual a 10.000 BTU	142,288
2	Split de 10.001 a 15.000 BTU	193,760
3	Split de 15.001 a 20.000 BTU	293,680
4	Split de 20.001 a 30.000 BTU	439,200
5	Split maior que 30.000 BTU	679,200

Fonte: PROCEL Adaptado (2018).

A partir da consideração do menor e maior ( $C_{MM}$ ) descritos na Tabela 2, isto é, 1 e 5, assim como, considerando o custo do kWh na região Norte ( $C_{kWh}$ ) igual a R\$ 0,67098 (CELPA, 2018), é possível estimar os custos mensais de operação ( $CM_{op}$ ) dessas centrais de ar, conforme a equação (2):

$$CM_{op} = C_{MM} C_{kWh} \text{ (R\$)} \quad (2)$$

Os  $CM_{op}$  obtidos para os itens 1 e 5 são respectivamente R\$ 95,47 e R\$ 455,73. No entanto, o não desligamento do equipamento por esquecimento ou qualquer outra razão relacionada a fatores externos, isto é, deixando o equipamento operando durante 24 horas, pode aumentar o  $CM_{op}$  para respectivamente R\$ 286,42 e R\$ 1.367,19. Com isso, é possível perceber que a má gestão de somente um equipamento elétrico pode acarretar um aumento de duzentos por cento (200%) no custo individual de um aparelho.

Diante deste contexto, diversos trabalhos e patentes têm investigado soluções para minimizar o consumo energético e conscientizar os usuários sobre o consumo consciente de energia elétrica.

## 1.1 TRABALHOS RELACIONADOS

Davis e Gertler (2015) apresentam um estudo sobre o consumo energético de centrais de ar condicionado analisando os parâmetros de temperatura, renda e uso desses aparelhos em uma base de dados do México. Foram realizadas estimativas do aumento da renda populacional, o aumento da temperatura global até o final do século e o percentual de pessoas no mundo que farão o uso de aparelhos de ar condicionado em suas residências. Os autores, por meio do estudo realizado, concluíram que em torno de 90% da população possuirá aparelhos de ar condicionado até o final do século. A partir disso, observa-se a importância da busca por novas fontes de geração de energia com baixo teor de emissão de dióxido de carbono na atmosfera para fins de uma baixa emissão em relação ao alto consumo energético.

McNeil e Letschert (2008) realizaram um levantamento sobre o futuro do consumo energético dos aparelhos de ar condicionado em países subdesenvolvidos. O estudo é feito a partir de dados familiares como renda, clima e parâmetros demográficos. Como resultado, são obtidas informações detalhadas sobre o futuro cenário do consumo energético dos aparelhos de ar condicionado.

Gillingham et al. (2003) fornecem uma revisão sobre melhorias na eficiência da utilização de aparelhos de ar condicionado em edifícios de áreas tropicais considerando o conforto dos ocupantes. O processo de aplicação da melhoria na eficiência foi aplicado em escolas, universidades, escritórios e ambientes residenciais



por meio de ferramentas de simulação computacional para o ajuste dos termostatos dos aparelhos de ar condicionado. No final foi constatado uma redução significativa do consumo energético desses ambientes.

Chua et al. (2013) apresentam uma revisão sobre as tecnologias de resfriamento utilizadas atualmente e apresentam novas soluções que ainda estão sendo estudadas no ramo de eficiência energética para aparelhos de ar condicionado. A revisão é concluída defendendo que as soluções encontradas podem ser facilmente aplicadas para atender necessidades específicas do uso de centrais de ar condicionado e aparelhos de resfriamento nos ambientes em que forem instalados.

## 1.2 PATENTES RELACIONADAS

Pesquisas realizadas no site do Instituto Nacional da Propriedade Industrial (INPI), evidenciaram a procura por uma melhor gestão energética de diversos aparelhos eletrônicos, na Tabela 3 são apresentadas algumas das invenções relevantes para discussão do desenvolvimento técnico deste trabalho.

**Tabela 3 – Patentes de invenção**

<b>P#</b>	<b>Nome</b>	<b>Número</b>	<b>Autor</b>
<b>P1</b>	Kit inteligente para refrigeração, ar condicionado e aquecimento	BR 10 2012 018917 8	(PINNOLA et al., 2014)
<b>P2</b>	Método de gerenciamento de consumo energético para aparelhos eletrodomésticos	BR 10 2015 015639 1	(MURAD, 2017)
<b>P3</b>	Aproveitamento energético e estético utilizando sistema de ar condicionado tipo split com condensação à água	MU 8901228 3	(RODRIGUES, 2011)

P#	Nome	Número	Autor
P4	Processo e dispositivo de redução do consumo elétrico de uma interface ethernet	PI 1016169 4	(CAYE-DANGY, 2016)
P5	Aparelho para redução do consumo elétrico de chuveiros elétricos através do monitoramento da vazão de água e da temperatura de banho	BR 10 2012 012424 6	(GROTHER, 2013)

**Fonte:** elaborado pelo autor.

Em P1 é apresentado um dispositivo encapsulado composto por diversos componentes eletrônicos agrupados em um quadro elétrico. Seu funcionamento se dá a partir do controle de velocidade do motor do compressor, ou seja, sua velocidade é constante até que haja uma nova solicitação para aumento de velocidade; assim há uma redução da carga de operação do aparelho. P2 demonstra um método para ser aplicado no uso de máquinas de lavar roupas, que tem como objetivo reduzir significativamente a quantidade de energia consumida durante a progressão funcional do motor nas etapas de agitação do ciclo de lavagem. P3 propõe a utilização de aparelhos de ar condicionado do tipo Split ou Multi-Split com condensação à água, pois, estes aparelhos propiciam maior eficiência reduzindo o consumo de energia elétrica, e a água aquecida no condensador do aparelho pode ser reaproveitada para diferentes finalidades. P4 apresenta um processo para redução do consumo elétrico de uma interface Ethernet que oscila para o modo de baixo consumo quando uma inatividade é detectada (durante intervalos de tempo pré-determinados) na sua camada física ou na sua camada MAC. P5 oferece um sistema embarcado que gerencia componentes eletrônicos para gerar eficiência no ponto de vista do consumo de água e energia em banheiros equipados com chuveiros elétricos.

As invenções apresentadas são baseadas na utilização de sistemas embarcados que realizam o gerenciamento de consumo de diversos equipamentos elétricos como roteadores, máquinas de lavar roupas, chuveiro elétrico e centrais de

ar condicionado. Algumas dessas soluções conseguem certa redução do consumo a partir da alteração do modo de funcionamento do aparelho, ou seja, ocorre apenas a mudança do modo de operação padrão do aparelho para um modo de economia de energia, assim pode-se afirmar que estes sistemas não controlam o tempo em que o aparelho permanece ligado. Dessa forma, é possível perceber a importância de se desenvolver ferramentas e tecnologias computacionais para otimizar e garantir maior autonomia de consumo de equipamentos elétricos.

Com o avanço tecnológico, a utilização de técnicas de visão computacional se tornou extremamente favorável no desenvolvimento de sistemas capazes de perceber e reconhecer o ambiente em que este é instalado. Diversas aplicações podem ser desenvolvidas e em todas estas aplicações há a execução de teorias e métodos específicos para extração de informações úteis contidas em imagens.

### 1.3 OBJETIVOS

A partir das observações ressaltadas, este trabalho possui como objetivo geral:

Projetar, desenvolver e testar um sistema para realizar a gestão de acionamento, uso e desligamento de equipamentos elétricos, a partir de artifícios de visão computacional e RNA, podendo ser instalado em: salas de aulas, escritórios e até mesmo empresas;

E como objetivos específicos:

- Propor um circuito eletrônico capaz de coletar imagens, processar e responder acionando o desligamento do aparelho;
- Utilizar e avaliar técnicas e conceitos de visão computacional;
- Verificar a eficiência do algoritmo *backpropagation* como alternativa para controle do desligamento do equipamento elétrico;
- Apresentar um estudo de caso usando um produto mínimo viável (MVP);

### 1.4 METODOLOGIA

Para que os objetivos deste trabalho sejam atingidos são necessárias cinco (5) etapas que buscam expor os passos metodológicos que segundo Carvalho et al (2007) devem garantir generalidade, ou seja, que os dados apresentados neste trabalho possam ser alcançados em outras situações.

1. Revisão bibliográfica: Após a definição do ponto focal do trabalho, foi efetuado um levantamento técnico sobre os métodos de análise e classificação de imagens e sistemas embarcados; essa pesquisa foi realizada através de sítios especializados como *google* acadêmico e INPI, através das seguintes *strings* de busca: sistemas embarcados, redes neurais artificiais, visão computacional, métodos de classificação de imagens, processamento de imagem, sistemas computacionais embarcados e eficiência energética industrial e residencial.

Essa busca teve como objetivo principal fornecer embasamento teórico para então se analisar qual tipo de abordagem e plataforma de desenvolvimento seria mais eficiente para solução da problemática especificada anteriormente.

2. Análise de bibliotecas: Nesta etapa foram analisados alguns produtos que já estão disponíveis no mercado, isto é, que já foram validados por suas equipes de desenvolvimento, dentre estes destacam-se o IBM Watson e *OpenCV*.

Watson é uma plataforma de serviços cognitivos que disponibiliza soluções de negócios pré-treinadas a partir de diversas APIs, umas destas *Visual Recognition* realiza a classificação de determinada imagem tendo como resultado um JSON apresentando os valores e rótulos da classificação. Esta solução possui grande diversidade de dados, no entanto seu uso gratuito é limitado a um mil (1000) classificações por mês além do mais para se obter um treinamento personalizado e utilização ilimitada é necessário a contratação de planos do serviço.

*OpenCV* é uma biblioteca de processamento de imagens *open source*, que disponibiliza algumas rotinas de classificação, dentre estas, o reconhecimento de padrões como detecção de rostos e olhos, também possibilita ao desenvolvedor criar uma rotina de reconhecimento própria utilizando classificadores em cascata baseados em recursos Haar, um método de detecção proposto por Paul Viola e Michael Jones (2001) intitulado "Rapid Object Detection using a Boosted Cascade of Simple Features". Entretanto esta abordagem não consegue classificar um padrão de imagens não linear, já que a rotina de identificação de rostos e olhos da biblioteca não consegue realizar a detecção quando a imagem não apresenta uma estrutura facial em primeiro plano.

3. Modelagem da arquitetura: Utilizou-se fundamentos básicos de eletrônica analógica e circuitos elétricos, idealizando um circuito auxiliar de acionamento de cargas, bem como a modelagem da arquitetura do sistema, um

dispositivo físico (sistema embarcado) que controla o acionamento de equipamentos elétricos a partir da análise e classificação de imagens capturadas pelo próprio sistema.

4. Implementação de *hardware* e *software*: O *hardware* foi construído conforme planejado na etapa anterior, entretanto o *software* foi segmentado, por causa da limitação de processamento das plataformas de desenvolvimento de sistemas embarcados, com isso o treinamento da RNA responsável por realizar o processo de classificação de imagens foi realizado em uma máquina virtual com ambiente GNU/Linux e após esse processo o conhecimento da mesma (pesos sinápticos e estrutura) foram encapsulados na plataforma escolhida para realizar o controle do sistema.

Para se realizar o treinamento da RNA é necessário seguir os seguintes passos: instalar o sistema em determinado ambiente, capturar uma base dados diversificada, realizar processos pré-processamento de imagem em toda base de dados, dividir a base de dados em oitenta por cento (80%) para treinamento e vinte por cento (20%) para teste e pôr fim, realizar o processo de treinamento da rede, neste trabalho esse processo é repetido dez (10) vezes para cada configuração de rede.

5. Avaliação: Tem como objetivo analisar de forma quantitativa os resultados alcançados no treinamento da RNA; seleciona-se a configuração de rede que obter um melhor aprendizado, para isso cada treinamento deve ter seu erro médio aritmético calculado, a rede selecionada é utilizada num segundo teste que tem como objetivo estimar o percentual de acerto da mesma, nesta etapa a base de dados de testes deve ser apresentada para a RNA e seus resultados comparados com os resultados desejados de classificação. Porém isto não é o bastante para se afirmar que o sistema se comportará de maneira bem-sucedida, ainda é necessário analisar um possível comportamento do sistema. Para isso, alguns padrões de entrada devem ser escolhidos de maneira aleatória da base de testes e apresentados ao dispositivo físico.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados os principais conceitos sobre sistemas embarcados, visão computacional e redes neurais artificiais que darão suporte teórico para a construção da solução proposta.

### 2.1 SISTEMAS EMBARCADOS

Sistemas embarcados revolucionaram o mundo melhorando a qualidade de vida humana e impulsionando o desenvolvimento tecnológico de diversas áreas do conhecimento (EMBARCADOS, 2013).

Para Cunha (2007) colocar capacidade computacional dentro de um circuito integrado, equipamento ou sistema, tornando-o independente e preparado para realizar apenas uma tarefa específica é a definição ideal para um sistema embarcado. Dessa forma, Broy (1997) elucida que esses sistemas têm finalidade de controlar um dispositivo físico com o envio de sinais de controle para atuadores, a partir de sinais de entrada providos por usuários ou sensores.

Em sua tese Moraes (2009) discorre que os sistemas embarcados que são desenvolvidos atualmente não diferem dos sistemas desenvolvidos na década de oitenta (80), isso, muito por conta da arquitetura destes sistemas, constituídos pela existência de um programa, uma unidade de processamento e um ciclo de busca-decodificação-execução.

Para Cola (2017) grande parte destes sistemas são compostos por uma placa de circuito impresso e conectores para comunicação com outros sistemas, com isso, diversos fornecedores oferecem módulos que incluem camada de *software*, além de bibliotecas que configuram o acesso e comunicação com o *hardware*.

Ossada e Martins (2010) consideram que o “coração” destas placas é um microcontrolador, responsável pela interação do sistema com seus sensores, dessa forma o microcontrolador realiza o processamento necessário para funcionamento do sistema e envia o resultado para os atuadores, outrem Carro e Wagner (2003) consideram outras arquiteturas e discorrem que estes sistemas também podem ser controlados por microprocessadores, que apresentam maior poder computacional do que os microcontroladores, entretanto afirmam que o uso de microprocessadores não

é vantajoso para todos os projetos, principalmente devido ao custo e desperdício de processamento.

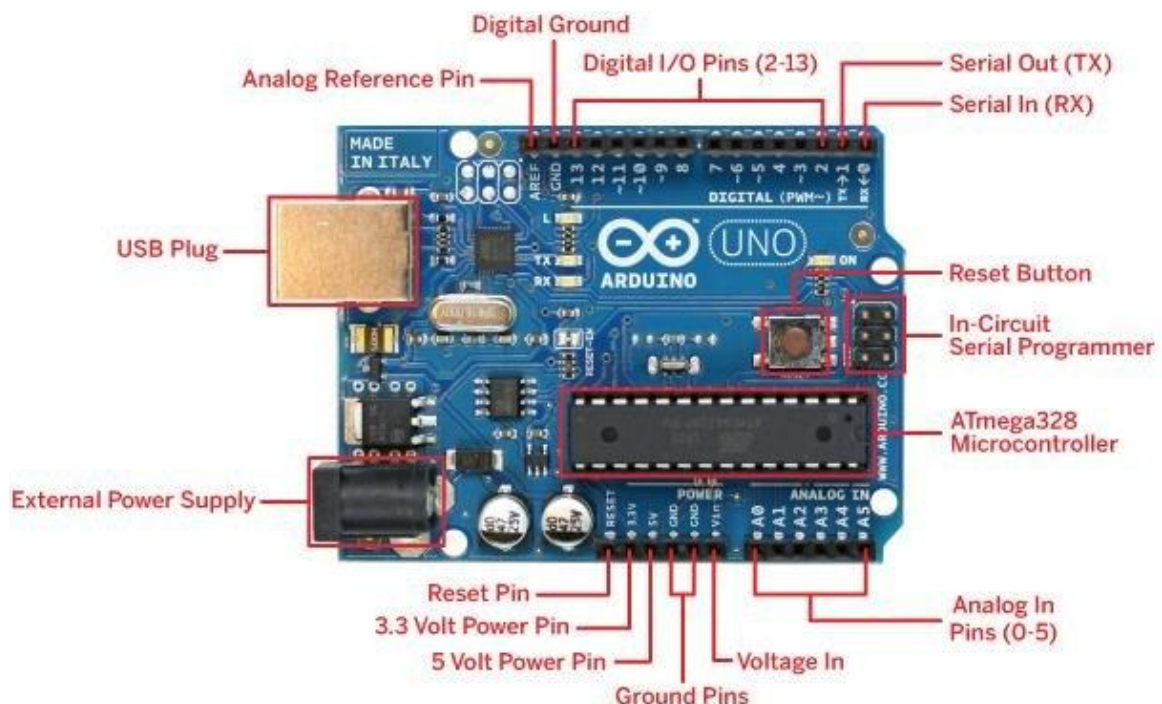
A construção destes sistemas não é uma tarefa fácil, pois há certos desafios a serem contornados, entretanto a adoção de plataformas de *hardware* e *software* padronizadas permite redução no tempo de projeto de um novo sistema.

### 2.1.1 Plataformas padronizadas

A adoção das plataformas de desenvolvimento foi democratizada, pois estas oferecem uma infraestrutura de recursos de *hardware* e ou *software* que possibilitam o desenvolvimento de sistemas com rapidez e baixos custos. Assim, segundo Ana (2016), o desenvolvimento de um sistema deve começar pela avaliação da arquitetura, para que esta atenda o objetivo final do projeto. Há no mercado uma grande variedade de plataformas de desenvolvimento, com isso, neste trabalho são apresentadas as plataformas: *Arduino*, *Intel Galileo* e *Raspberry*.

A Figura 1 apresenta uma das plataformas mais populares do mundo, isto é, a plataforma *Arduino* na versão UNO R3.

**Figura 1** – Plataforma Arduino



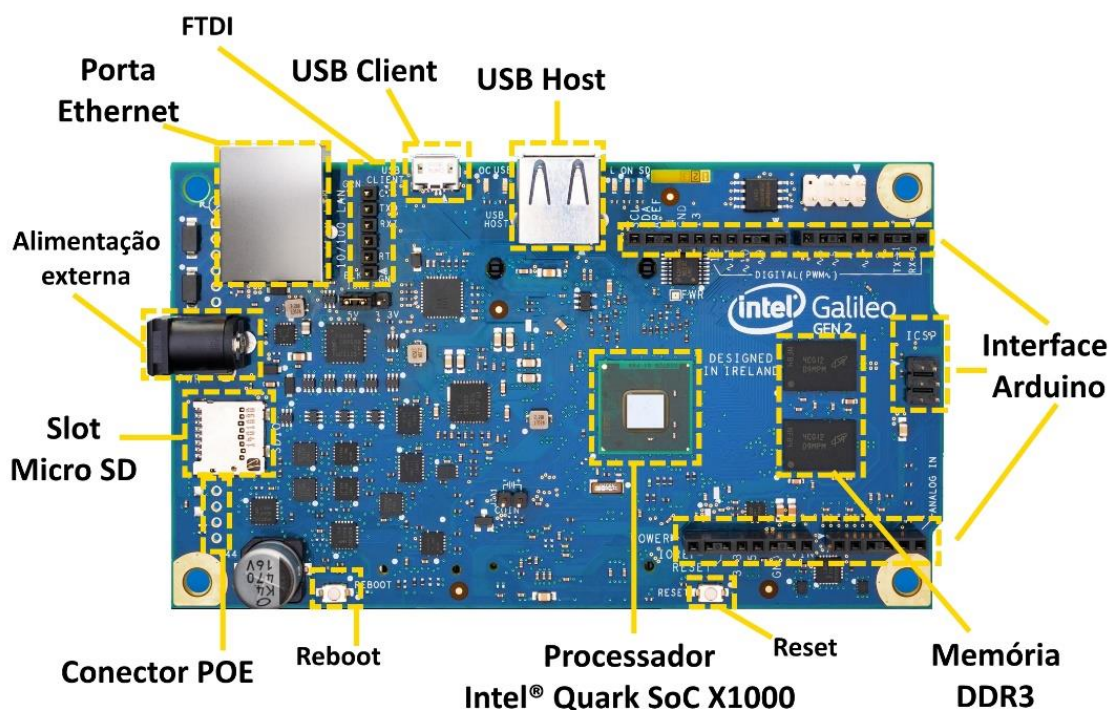
**Fonte:** Jesus (2015).

A plataforma *Arduino*, utilizada em diversos projetos e aplicações; foi criada em 2005 por um grupo de pesquisadores com o objetivo de ser um facilitador no ensino

de eletrônica; disponibiliza um microcontrolador ATMEL ATMEGA328 de oito (8) bits que pode operar em até vinte (20) MHz, vinte e oito (28) pinos dos quais, vinte e três (23) podem ser utilizados como entrada e saída (I/O), além de fornecer tensão de 3.3V e 5V para alimentação de circuitos externos (EMBARCADOS, 2013). Vale ressaltar também, a grande variedade de componentes periféricos disponíveis para esta plataforma.

A Figura 2 apresenta a plataforma baseada na arquitetura *Intel* na versão GALILEO GEN2.

**Figura 2 – Plataforma Intel Galileo**



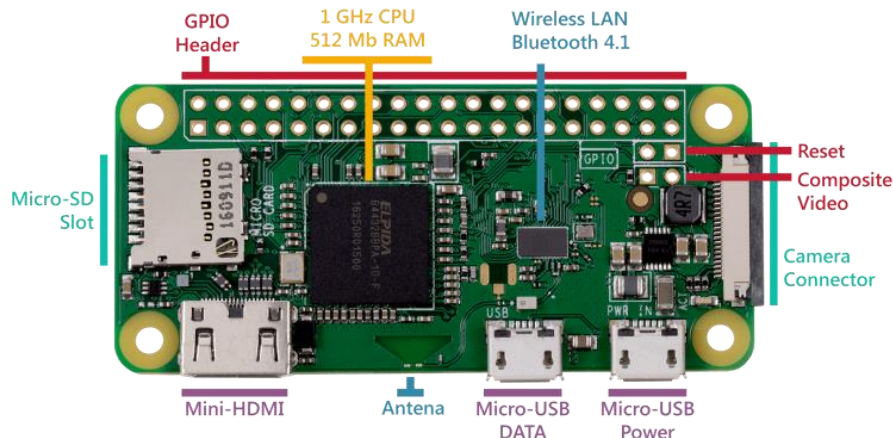
**Fonte:** FILIPEFLOP (2015).

A plataforma *Intel Galileo*, um projeto que visa integrar a simplicidade da plataforma *Arduino* com a performance dos processadores *Intel*; foi apresentada no evento Maker Faire 2013, a fim de emplacar os produtos da empresa no segmento de sistemas embarcados; esta plataforma disponibiliza um processador Quark SoC (System on chip) X1000 de quatrocentos (400) MHz e trinta e dois (32) bits, duzentos e cinquenta e seis (256) MB de memória com slot de cartão micro SD de até trinta e dois (32) GB, uma interface Ethernet e possibilidade de instalação de um sistema operacional na placa, ressalta-se que a tensão de alimentação desta plataforma é de 7V à 15V (FILIPEFLOP, 2015).



A Figura 3 apresenta a plataforma *Raspberry*, versão Pi Zero W.

**Figura 3 – Plataforma Raspberry Pi**



Fonte: Lanza (2017).

A plataforma *Raspberry*, oriunda de um projeto de uma fundação de mesmo nome, foi criada em 2005 como objetivo disseminar o ensino básico de computação nas escolas, disponibiliza um microcomputador com *Wifi* e *Bluetooth* integrados e quarenta (40) pinos de entrada e saída (I/O), slot de cartão micro SD de até trinta e dois (32) GB, driver gráfico, saída HDMI, conector de câmera, tendo como fonte de alimentação uma entrada micro USB de cinco (5) V, em sua versão mais básica conta com um processador dedicado Broadcom BCM2835 ARM11 de 1 GHz e quinhentos e doze (512) MB de memória (FILIPEFLOP, 2018).

### 2.1.2 Custo benefício de produção

As plataformas apresentadas na subseção 2.2.1 impulsionaram a prototipação e desenvolvimento de sistemas embarcados, entretanto não extinguiram dificuldades inerentes ao desenvolvimento desse tipo de projeto. Já que, segundo a equipe Embarcados (2015) nos ambientes de pesquisa e desenvolvimento das universidades brasileiras existem obstáculos oriundos dos valores de impostos e burocracia relacionados à aquisição de equipamentos.

Dessa forma, Santos (2005) destaca a importância de se avaliar a opção mais satisfatória para solução das necessidades da aplicação. Assim, além da avaliação da arquitetura a ser utilizada no projeto, se faz necessário também verificar o custo de aquisição destas plataformas para que no fim do desenvolvimento, o custo de produção do sistema seja rentável. Para isso um levantamento dos custos de

diferentes itens que compõem um sistema embarcado, assim como, um levantamento dos custos das plataformas foi realizado por meio de consultas em sítios de fornecedores especializados.

A Tabela 4 apresenta o custo<sup>1</sup> de aquisição das três plataformas (placas e fontes de alimentação), enquanto que o Apêndice A apresenta os componentes necessários para inicialização do desenvolvimento do sistema proposto.

**Tabela 4 –** Relação de custo das plataformas de desenvolvimento

<b>Plataforma</b>	<b>Custo (R\$)</b>
<b>Arduino</b>	186,57
<b>Raspberry</b>	275,38
<b>Intel Galileo</b>	472,93

**Fonte:** FILIPEFLOP (2018) e BAU DA ELETRÔNICA (2019).

Analisando as opções de plataformas é possível destacar que o *Arduino* é a melhor opção quando o projeto se trata de controle de *hardware*, essa função é perfeitamente solucionada pela gama de periféricos disponibilizadas para a plataforma; por outro lado o *Raspberry* pode se tornar a melhor opção quando o projeto trata de processamento de dados, aplicações de *software* e utilização de poucos periféricos; por fim, a plataforma *Intel Galileo* combina o desempenho de *hardware* e *software* das outras plataformas anteriores.

Com isso, outro parâmetro importante na escolha da plataforma a ser utilizada é a tarefa que se deseja realizar, ou seja, o requisito mínimo do projeto, por exemplo, para aplicações repetitivas que objetivam capturar dados de informação externa e a partir disso tomar uma decisão é de praxe que a plataforma *Arduino* seja escolhida; agora se objetivo final de do projeto for dependente da realização de múltiplas tarefas e cálculos intensos a plataforma *Raspberry* é a mais indicada. Entretanto, deve-se ressaltar que a combinação destas plataformas também é alternativa viável.

<sup>1</sup> Incluindo frete para região Norte

## 2.2 VISÃO COMPUTACIONAL

Segundo a Associação Britânica de Visão de Máquina - BMVA (2018), a visão computacional é a ciência que procura fazer com que uma máquina ou computador tenha capacidade de ver e interpretar o mundo ao seu redor. Para isso, conforme Ballard e Brown (1982) há a necessidade da extração de informação de uma imagem que Souza e Correia (2007) definem como uma matriz de pontos elementares, onde cada ponto representa um *pixel* e conforme a quantidade de *pixels* da imagem, melhor sua resolução e consequentemente seu tamanho.

Sousa (2013) destaca que a partir da entrada de uma imagem, para fazer com que um sistema “perceba” o ambiente é necessário antes de mais nada, aplicar técnicas de processamento de imagem a fim de eliminar quaisquer informações indesejadas e após esta etapa, pode-se executar os processos para reconhecimento de padrões. Dessa forma, Perelmutter et al. (1995) explana que um sistema de visão artificial otimizado, após a captura de imagem, deve possuir duas etapas: pré-processamento e classificação.

### 2.2.1 Pré-processamento de imagem

Segundo Marengoni e Stringhini (2009) após o processo de aquisição de imagem diversos ruídos são capturados, e oriundos a partir de diversos fatores, como: iluminação do ambiente, posição relativa entre o objeto de interesse e a câmera, entre outros; com isso há a necessidade de submissão da imagem a tratamentos que visam reduzir e ou eliminar os ruídos presentes na mesma.

Estes tratamentos são realizados a partir de técnicas de pré-processamento que conforme Souza e Correia (2007) a aplicação destas técnicas visa ressaltar as imagens a fim de que a imagem resultante seja mais pertinente para uso em uma aplicação específica. Para Esquef, Albuquerque e Albuquerque (2003) estas técnicas são subdivididas em duas categorias principais: as de domínio espacial que utilizam filtros que manipulam o plano da imagem e as de domínio da frequência que utilizam filtros que atuam no espectro da imagem.

Osorio (1991) enfatiza que as técnicas que operam no domínio espacial são mais proveitosas, pois são realizadas diretamente nos *pixels* da imagem, evitando assim, que a imagem sofra alterações prévias e posteriores ao seu processamento;

enquanto que, as técnicas realizadas no domínio da frequência necessitam que a imagem seja transformada do domínio espacial para o domínio da frequência e após ser processada, é então novamente transformada para o domínio do espaço. A partir desse exposto, este trabalho se concentra em discorrer sobre os filtros espaciais.

### 2.2.2 Filtros espaciais

Para Sanches et al. (2015) os filtros classificam-se em dois (2) grupos: Lineares e Não Lineares; sendo que, filtros Lineares têm a finalidade de eliminar frequências ou ruídos indesejáveis a partir da média entre a vizinhança de *pixels*, criando um efeito desfocado na imagem e, filtros Não Lineares têm como finalidade realçar ou minimizar os ruídos a partir da variância.

Segundo Niku (2013) processos no domínio espacial acessam e operam as informações de *pixels* individuais. Neste contexto, uma das técnicas mais utilizadas nesse domínio é a convolução. Esta consiste na sobreposição de uma máscara à imagem, podendo assim alterá-la *pixel* por *pixel*; dessa forma, há a multiplicação dos valores das células da máscara pelos valores dos *pixels* correspondentes da imagem, estes números são somados e seus resultados são normalizados para posterior substituição no *pixel* que está presente no centro da área de interesse; essa operação é repetida até que a imagem seja completamente processada. A Figura 4 ilustra uma demonstração da técnica.

Assim, na Figura 4a apresenta-se de forma matricial certa imagem de tamanho (4x4); na Figura 4b é realizado o processo de convolução da imagem com uma máscara de convolução de tamanho (3x3), nesse passo, a máscara de convolução é alocada no canto superior esquerdo da imagem e inicia-se o processo, com isso, o resultado do pixel central é gerado a partir da equação (3), em que o valor de S é expresso pela equação (4), a máscara é então movida um pixel para direita e o mesmo processo é repetido em forma de varredura até o final da imagem; e finalmente na Figura 4c o resultado da operação é apresentado.

Niku (2013) ressalta que “as primeiras e as últimas linhas e colunas não são afetadas por essa operação e, portanto, são geralmente ignoradas”, dessa forma estas linhas e colunas podem ser mantidas ou assumem valores iguais a zero.

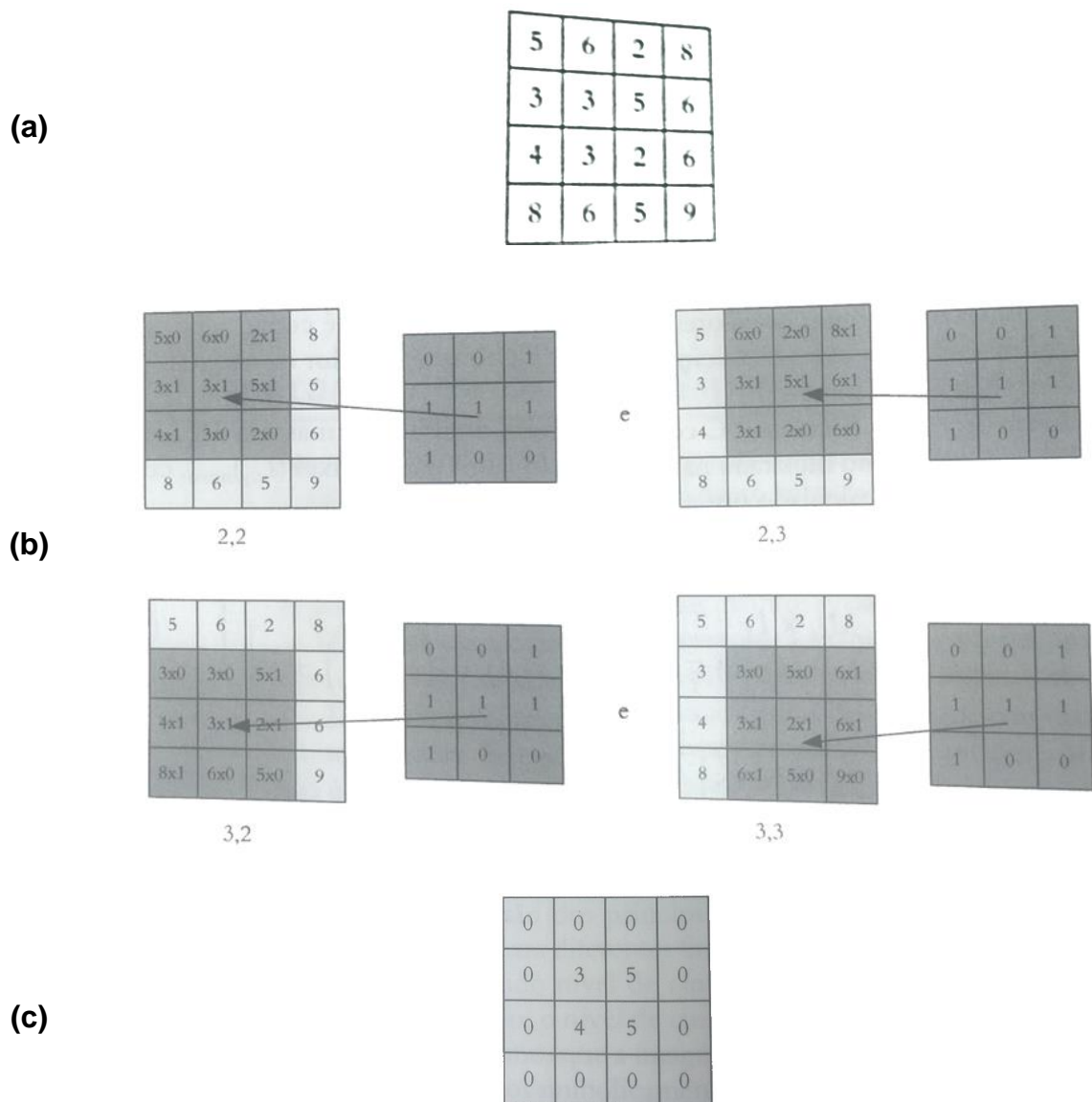
$$R = (I_1.M_1 + I_2.M_2 + I_3.M_3 + \dots + I_m.M_n) / S \quad (3)$$

$$S = |M_1 + M_2 + M_3 + \dots + M_n| \quad (4)$$

Nas equações 3 e 4, os termos  $I$  e  $M$  representam elementos da imagem e da máscara de convolução, respectivamente, enquanto que, os índices  $m$  e  $n$  representam concomitantemente, o número de pixels total da imagem e o número de pixels total da máscara de convolução.

**Figura 4** – Processo de filtragem de imagem por máscara de convolução

(a) Células da imagem. (b) Convolução da máscara com as células da imagem. (c) resultado da operação



**Fonte:** Adaptado de Niku (2013).

O procedimento realizado e ilustrado na Figura 4 opera calculando a intensidade de determinado *pixel* em função da intensidade de seus vizinhos (Estrela, 2005). Além disso, muitos filtros espaciais baseiam-se nesta técnica, dentre eles

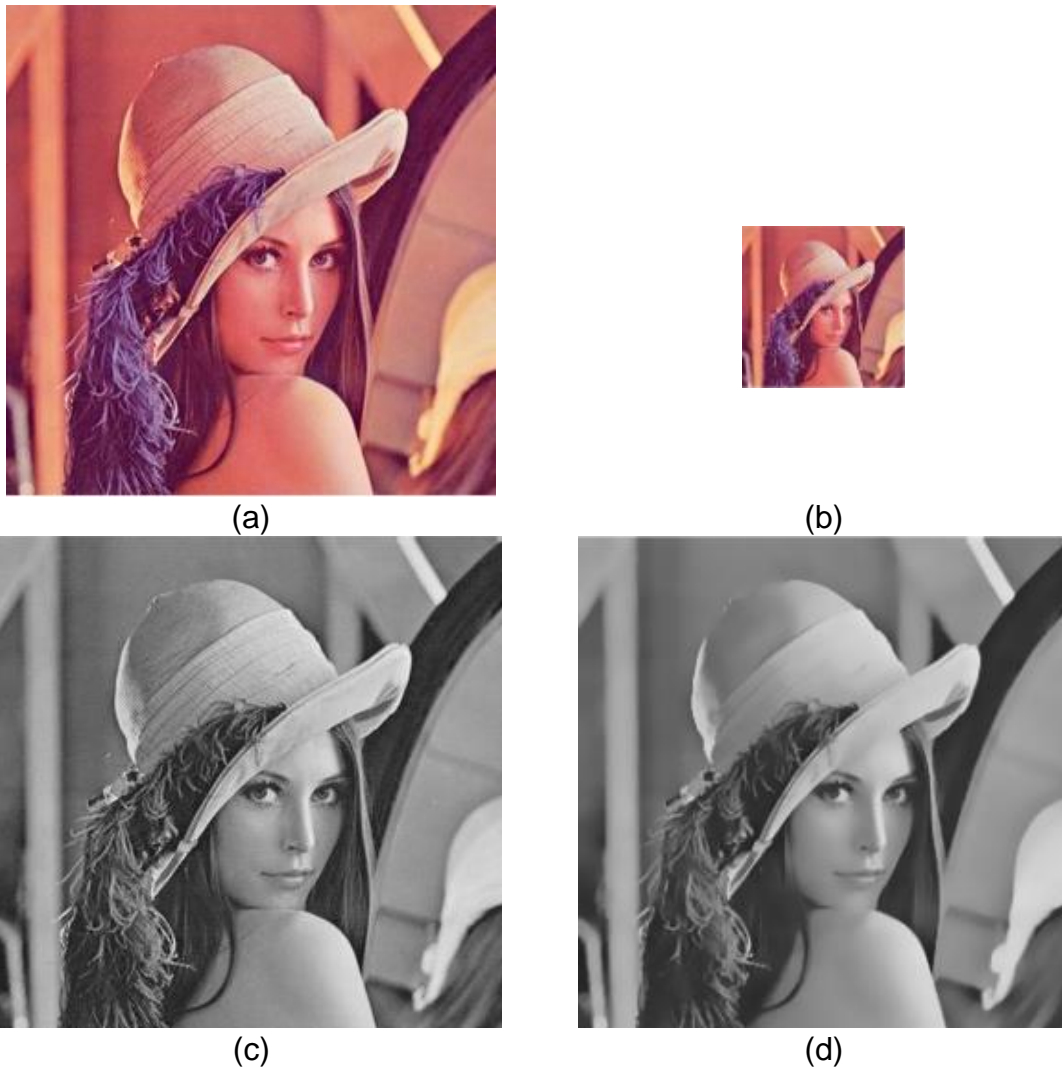
destacam-se os filtros passa baixa e filtros passa alta. Segundo Marengoni e Stringhini (2009) filtros passa baixa realizam a suavização da imagem minimizando ruídos, entretanto ocasionam a perda de nitidez; já, filtros passa alta tendem a realçar bordas ou regiões de interesse com transições abruptas de intensidade dos *pixels*, no entanto, acabam realçando os ruídos advindos do processo de captura da imagem.

Os principais filtros passa baixa são os filtros de média e gaussiano; Azevedo, Conci e Leta (2018) definem o filtro de média como uma máscara de convolução ( $n \times n$ ) com todos os coeficientes iguais a um (1) e divididos pelo número de pixels da máscara ( $n^2$ ), a partir dessa máscara obtém-se a média entre os tons de *pixel*/vizinhos, com isso é possível estabelecer uma relação de proporcionalidade entre a máscara e a área de redução de contraste, assim como uma relação com a perda de definição da imagem; já o filtro gaussiano, um dos mais importantes, é baseado na aproximação digital da função gaussiana e apresenta-se nas formas *1D* e *2D*, este filtro não possui todos os coeficientes iguais, por conta disso gera uma imagem mais suave, entretanto a forma mais prática para sua utilização é a aplicação da convolução de máscara *1D* na direção *x* e em seguida a aplicação da mesma máscara *1D* agora na direção *y* da imagem.

Um filtro passa alta básico descrito por Marques Filho e Neto (1999) deve apresentar uma máscara com coeficientes positivos nas proximidades de seu centro e negativos longe dele, isso implica em um resultado igual a zero (0) quando realizada a soma algébrica destes coeficientes, assim quando aplicado em uma região homogênea da imagem resulte um valor muito baixo ou igual a zero (0), enfatizando o princípio da filtragem passa alta.

A partir destas técnicas, muitas empresas e estudiosos dedicam-se no desenvolvimento e publicação de novas ferramentas de visão computacional, por exemplo, Marengoni e Stringhini (2009) destacam a biblioteca *OpenCV* desenvolvida pela Intel e que possui mais de quinhentas (500) funções divididas em cinco (5) grupos; dentre os grupos destaca-se o de “processamento de imagens” que disponibiliza inúmeras funções dentre as quais podemos destacar: ***resize*** que redimensiona a imagem para o tamanho especificado em seus parâmetros; ***cvtColor*** que converte a imagem de um padrão de cor para outro; e ***bilateralFilter*** que suaviza a imagem de entrada mantendo as bordas nítidas. A Figura 5 apresenta o resultado da utilização destas funções.

**Figura 5** – Passos do processamento digital de uma imagem  
(a) Imagem original. (b) Resize. (c) CvtColor. (d) BilateralFilter.



**Fonte:** elaborado pelo autor.

A Figura 5a apresenta a imagem original, enquanto que a Figura 5b apresenta a imagem após aplicação da função de redimensionamento com parâmetros de altura e largura iguais a cem (100), na Figura 5c foi aplicado o filtro para alteração do padrão de cor da imagem e na Figura 5d foi aplicado o filtro para suavização de imagem, dessa forma, ressalta-se a importância da implementação de uma etapa de pré-processamento a fim de se maximizar os índices obtidos na etapa de classificação.

### 2.3 REDES NEURAIS ARTIFICIAIS

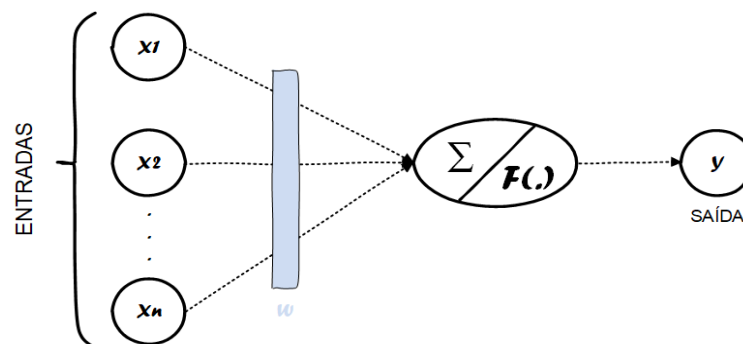
Redes Neurais Artificiais (RNA) é um ramo da Inteligência Artificial. A inteligência artificial é um método científico que visa encontrar metodologias genéricas para automatizar atividades perceptivas, cognitivas e manipulativas através da capacidade de processamento de um computador (Perreira, 1988).

As RNA tiveram sua origem a partir de três (3) importantes publicações: Warren McCulloch e Walter Pitts (1943), Donald Hebb (1949) e Frank Rosenblatt (1958).

McCulloch e Pitts desenvolveram o primeiro modelo computacional para redes neurais, que conforme Barreto (2002) “tem inspiração nos neurônios biológicos e nos sistemas nervosos”. Hebb apresentou a teoria de que as conexões neurais são fortalecidas cada vez que são utilizadas, assim explicando a maneira de como se dá o aprendizado humano. Rosenblatt apresentou a mais antiga rede neural ainda em uso, baseada no perceptron.

O neurônio artificial proposto por McCulloch e Pitts implementa de forma simples um neurônio biológico. A Figura 6 apresenta o modelo de um neurônio artificial.

**Figura 6 – Exemplo de neurônio artificial**



**Fonte:** Adaptado de Felix e Possan (2018).

Ferneda (2006) elucida que um neurônio artificial é composto por três elementos básicos: (I) um conjunto de  $n$  conexões de entrada ( $x$ ) caracterizadas por pesos ( $w$ ), (II) um somador ( $\Sigma$ ) para acumular os sinais de entrada e (III) uma função de ativação ( $F(.)$ ) que limita o intervalo permissível de amplitude do sinal de saída a um valor fixo. Além disso, o autor define que a combinação de diversos neurônios compõe uma RNA que visa simular o processamento de informação do cérebro humano. Dessa forma, “a implementação de um sistema de visão artificial baseado em redes neurais faz uso de suas características intrínsecas, isto é, da sua capacidade de extrair padrões de conjunto de dados” (PERELMUTER, et al., 1995).

### 2.3.1 Aprendizado Artificial

Segundo Flores e Peres (2001), a primeira regra de aprendizado para as RNA definida por Hebb argumentava que se dois neurônios interconectados tivessem uma



ativação simultânea a força da conexão entre eles era então aumentada, outrem se estes neurônios fossem ativados de forma assíncrona a força de conexão entre eles era então enfraquecida ou eliminada. Dessa forma, conclui-se que o processo de aprendizagem se dá, quando há uma estimulação repetida e persistente na conexão entre neurônios.

Para Barreto (1999), “a propriedade mais importante das redes neurais é a habilidade de aprender seu ambiente e com isso melhorar seu desempenho”, para isso é realizado o processo de treinamento, no qual um número de épocas é estipulado para que os pesos da RNA sejam ajustados iterativamente, a partir da estimulação de conexões entre seus neurônios; com isso, destacam-se como metodologias de aprendizado: o modelo supervisionado e o não supervisionado.

Segundo Osório, Bittencourt e Osório (2000) RNA são ferramentas de inteligência artificial com capacidade de adaptação e aprendizagem para realização de determinada aplicação, essa característica é implementada para tarefas de reconhecimento de padrões através do uso de algoritmos de aprendizagem supervisionada, tendo como ponto de partida um conjunto de amostras previamente “classificadas” e utilizadas no treinamento da rede neural.

No modelo de aprendizagem supervisionada, o treinamento da RNA é realizado com o auxílio de um supervisor, que neste caso, é a disposição de um parâmetro de classificação<sup>2</sup>, ou seja, uma referência que objetiva ensinar a RNA a classificar determinado padrão de entrada. Para classificação das amostras de entrada o modelo de aprendizagem supervisionada utiliza um intervalo de zero (0) a um (1). Este intervalo estabelece um limiar de classificação que é comparado com os resultados esperados pelo sistema. Dessa forma, Barreto (1999) elucida que a RNA deverá possuir um conjunto com as saídas desejadas para cada entrada; Osório, Bittencourt e Osório (2000) ressaltam que a partir desse conjunto, a RNA deve ser capaz de medir a diferença entre o comportamento da saída da RNA e o comportamento de referência, sendo capaz assim, de corrigir seus pesos.

No modelo de aprendizagem não supervisionada, não há a existência de um parâmetro de classificação indicando a resposta desejada para os padrões de entrada; Osório, Bittencourt e Osório (2000) explanam que os pesos da RNA são

---

<sup>2</sup> O parâmetro de classificação pode ser denominado como “saída desejada”, “classe” ou “classe de saída”.

ajustados em função de critérios internos, como a repetição de padrões de ativação em paralelo de vários neurônios.

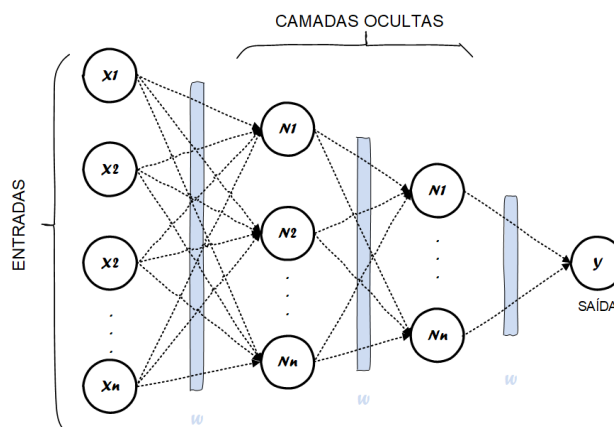
A forma tradicional de apresentação dos dados para determinada RNA é através da utilização de matrizes, onde cada linha é definida como um registro que possui um parâmetro de classificação correspondente, enquanto que, cada coluna é definida como um atributo que possui determinada informação específica, por exemplo, o valor de um pixel ou na medicina um parâmetro médico como a temperatura do paciente. Segundo Gama et al. (2015) é necessário que cada objeto do conjunto de treino possua atributos de entrada e saída, para que quando submetida uma nova amostra à rede neural, esta possa ser utilizada para prever um valor de classificação.

No processo de treinamento, a taxa de aprendizado e o número de épocas são parâmetros bastante importantes para que a RNA possa obter resultados satisfatórios. A taxa de aprendizado é uma constante que visa ajustar os pesos da RNA de forma suave e lenta, enquanto que o número de épocas é um critério de parada que define a quantidade máxima de iterações a serem realizadas.

### **2.3.2 Backpropagation**

Rauber (2005) destaca que o potencial das RNA é advindo justamente da criação de um conjunto de neurônios interligados que recebem um “estímulo” nas suas entradas, isto é um padrão de entrada, que é processado e então emitido como um novo sinal de saída, que é recebido por outros elementos da RNA. Assim, segundo Felix e Possan (2018), as RNA de múltiplas camadas possibilitam um aumento na capacidade de processamento de informações na rede.

A Figura 7 apresenta a estrutura de uma RNA de múltiplas camadas, normalmente composta por uma camada de entrada conectada a um conjunto de neurônios que compõe a intitulada camada oculta, onde esse conjunto pode ser  $n$  vezes conectado, até ser conectado a uma camada de saída que apresenta o resultado da rede.

**Figura 7 – Rede Neural Artificial de Múltiplas Camadas**

**Fonte:** Adaptado de Felix e Possan (2018).

Essa estrutura e muitas outras configurações de RNA são propostas e utilizadas com variações: no número de neurônios por camada, tipo de função de ativação, número de camadas ocultas, taxa de aprendizado e entre outros. Segundo Binoti, da Silva Binoti e Leite (2014) “O número de neurônios na camada oculta é definido principalmente de forma empírica pela experiência do usuário [...]”; bem como, a taxa de aprendizado deve considerar a convergência do algoritmo de treinamento, ou seja, a adoção de um valor muito baixo pode gerar um processo de aprendizagem muito lento, entretanto com a adoção de um valor muito alto a aprendizagem da rede pode perder uma gama de soluções.

Existem inúmeros algoritmos para reconhecimento de padrões, cada um possui suas propriedades e especificidades, entre eles destaca-se neste trabalho o *backpropagation* que opera com múltiplas camadas e resolve problemas “não linearmente separáveis”.

Uma RNA que utiliza este algoritmo é baseada no aprendizado supervisionado por correção e erro, que de acordo com Farias et al. (2012) a partir das etapas de propagação (*forward*) e retropropagação (*backward*), estipula o erro comparando o valor calculado na saída da rede (etapa de propagação), com a saída desejada apresentada, a correção do erro é fundamentada no método do gradiente descendente<sup>3</sup> (etapa de retropropagação), dessa forma, os pesos são iterativamente

<sup>3</sup> O gradiente descendente faz uso de derivadas parciais objetivando obter a minimização da função erro através da atualização dos pesos sinápticos.

calculados e atualizados objetivando minimizar o erro até a ocorrência de um erro mínimo global.

Para exemplificar a aprendizagem realizada pelo algoritmo *backpropagation* são apresentados os modelos matemáticos que se referem ao treinamento de uma RNA de uma única camada. O processo de treinamento da rede é inicializado a partir da alocação de um registro na entrada da mesma e com isso, realiza-se o passo de propagação que conforme Haykin (2007) é realizado de acordo com a equação (6):

$$rede = \sum_{i=1}^n x_i * w_i \quad (5)$$

onde  $x$ ,  $w$  e  $n$  representam respectivamente o vetor de elementos de entrada, os pesos sinápticos e o tamanho do vetor de entrada a ser percorrido.

Para cada neurônio da RNA, o valor obtido desta equação é submetido a uma função de ativação que é a transformação não linear realizada ao longo do sinal de entrada, definindo se a informação recebida pelo neurônio é relevante para o processo de classificação, ou seja, basicamente estas funções indicam se o neurônio deve ser ou não ativado, conseqüentemente este resultado é enviado para a próxima camada de neurônios como entrada (DEEP LEARNING BOOK, 2019).

Para este trabalho foi adotado a função Sigmoid, a qual é apresentada na equação (7):

$$saídaCalculada = \frac{1}{1 + e^{-rede}} \quad (6)$$

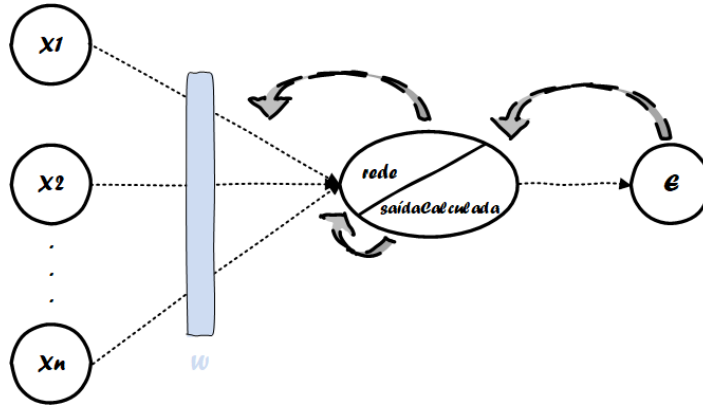
Com o resultado é possível estimar o erro usando a função de erro médio quadrático, conforme equação (7):

$$E = \sum \frac{1}{2} (saídaEsperada - saídaCalculada)^2 \quad (7)$$

onde *saídaEsperada* representa o parâmetro de classificação desejado.

Com os resultados obtidos pelas equações (5), (6) e (7), o próximo passo para que os pesos da RNA sejam atualizados é a realização da etapa de retropropagação, neste, é calculado o gradiente do erro estipulado na camada de saída e com o mesmo aplica-se de forma recursiva a regra da cadeia, visualmente esse processo é ilustrado na Figura 8.

**Figura 8 –** Processo recursivo da regra da cadeia



**Fonte:** Adaptado de Mazur (2019).

Para o exemplo de uma única camada, a derivada parcial do erro em relação a determinado peso é definida pela equação (8):

$$\frac{\partial E}{\partial w_{i,j}} = \frac{\partial E}{\partial \text{saídaCalculada}_j} \frac{\partial \text{saídaCalculada}_j}{\partial \text{rede}_j} \frac{\partial \text{rede}_j}{\partial w_{i,j}} \quad (8)$$

onde  $\frac{\partial E}{\partial \text{saídaCalculada}_j}$ ,  $\frac{\partial \text{saídaCalculada}_j}{\partial \text{rede}_j}$  e  $\frac{\partial \text{rede}_j}{\partial w}$  representam respectivamente a derivada parcial do erro em relação a saída calculada, a derivada parcial da saída calculada em relação a rede e a derivada parcial da rede em relação aos pesos sinápticos.

Após o cálculo do gradiente é possível atualizar os pesos sinápticos usando a fórmula geral de atualização dos pesos que foi introduzida por Leite (2018), conforme equação (9):

$$w_{\text{novo}} \leftarrow w_{\text{atual}} - \alpha \frac{\partial E}{\partial w} \quad (9)$$

onde  $w_{\text{novo}}$ ,  $w_{\text{anterior}}$ ,  $\alpha$  e  $\frac{\partial E}{\partial w}$  representam respectivamente o novo peso sináptico após a atualização, o peso sináptico atual que deve ser recalculado, a taxa de aprendizagem, e a derivada parcial do erro em função ao peso sináptico. Com isso, o valor do peso na iteração atual será seu valor na iteração anterior corrigido de forma proporcional ao negativo do gradiente em relação ao peso.

Por fim, o processo iterativo é repetido até que uma das seguintes condições seja alcançada, isto é, alcance o número máximo de épocas pré-estabelecidos ou o erro médio quadrático seja minimizado e alcance o valor mínimo previsto.

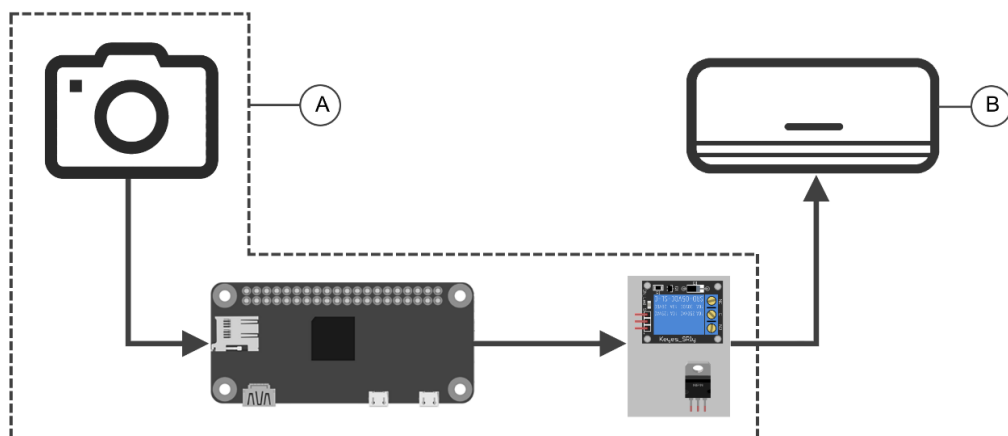
### 3 PROJETO ELETRIC CONTROL

O projeto de controle de equipamentos elétricos, denominado “Projeto Eletric Control” será apresentado conforme planejado. Primeiramente será apresentado a arquitetura geral do sistema, em seguida, o projeto de *hardware*, por fim, o projeto de *software* e testes realizados.

#### 3.1 ARQUITETURA GERAL DO SISTEMA

A Figura 9 apresenta a arquitetura geral do sistema, onde o elemento (A) representa o sistema embarcado encapsulado, enquanto que (B) é um exemplo da instalação em determinado aparelho que se deseja controlar, neste caso a figura ilustra a possível instalação em uma central de ar condicionado. O elemento (A) é composto pela câmera, placa controladora e circuito integrado. A câmera é responsável em coletar as imagens que serão processadas pelo sistema. Já a placa controladora é responsável em armazenar e processar as imagens coletadas via a câmera. Durante a etapa de processamento, as imagens são convertidas em uma matriz de dados para posterior apresentação em uma RNA treinada para classificar a presença ou ausência de indivíduos no ambiente. O circuito integrado é responsável em realizar o acionamento do equipamento (B) de acordo com a classificação da RNA.

**Figura 9 –** Arquitetura geral do sistema



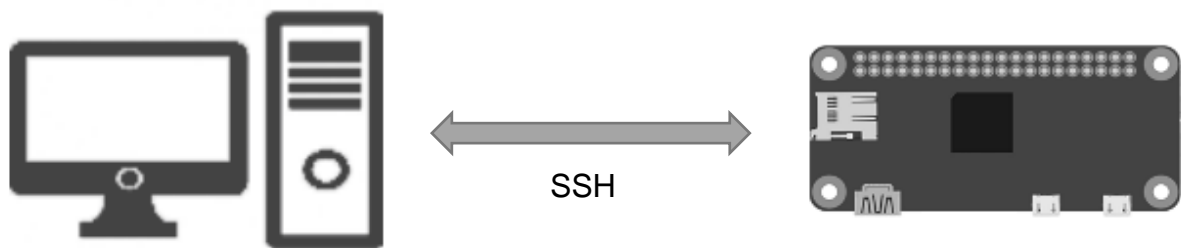
**Fonte:** elaborado pelo autor.

Para que este projeto fosse desenvolvido, um sistema operacional *GNU/Linux* (GNU/LINUX, 2018) foi virtualizado para disponibilizar o ambiente que faz execução do treinamento da RNA e para instalação do ambiente de desenvolvimento do *software* que foi embarcado na placa controladora. O *Spyder* foi escolhido como

ambiente de desenvolvimento dos *scripts* (ANACONDA, 2018). Além disso, foi utilizado a linguagem de programação *python* (PYTHON, 2018) e as bibliotecas para processamento de imagens e RNA, respectivamente, *OpenCV* (OPENCV, 2018) e *PyBrain* (PYBRAIN,2018).

A Figura 10 apresenta um esquemático que representa a comunicação do sistema embarcado com o ambiente de desenvolvimento responsável pelo treinamento da RNA que será embarcada na placa controladora do sistema. Para comunicação foi utilizado o protocolo de rede *Security Shell* (SSH).

**Figura 10** – Comunicação do sistema com ambiente de desenvolvimento



**Fonte:** Elaborado pelo autor.

A escolha do protocolo SSH visa tornar mais simples o acesso ao sistema operacional instalado na placa de controle do sistema. A Figura 10 também engloba o canal de comunicação fornecido pelo SSH que segundo Peterson (2004) disponibiliza ao usuário uma ferramenta de conexão remota segura a outro computador, já que os dados utilizados na conexão são criptografados. A partir desse canal foram ativados todos *scripts* implementados (no ambiente virtualizado) e salvos em um dispositivo USB que foi conectado a placa controladora.

### 3.2 PROJETO DE *HARDWARE*

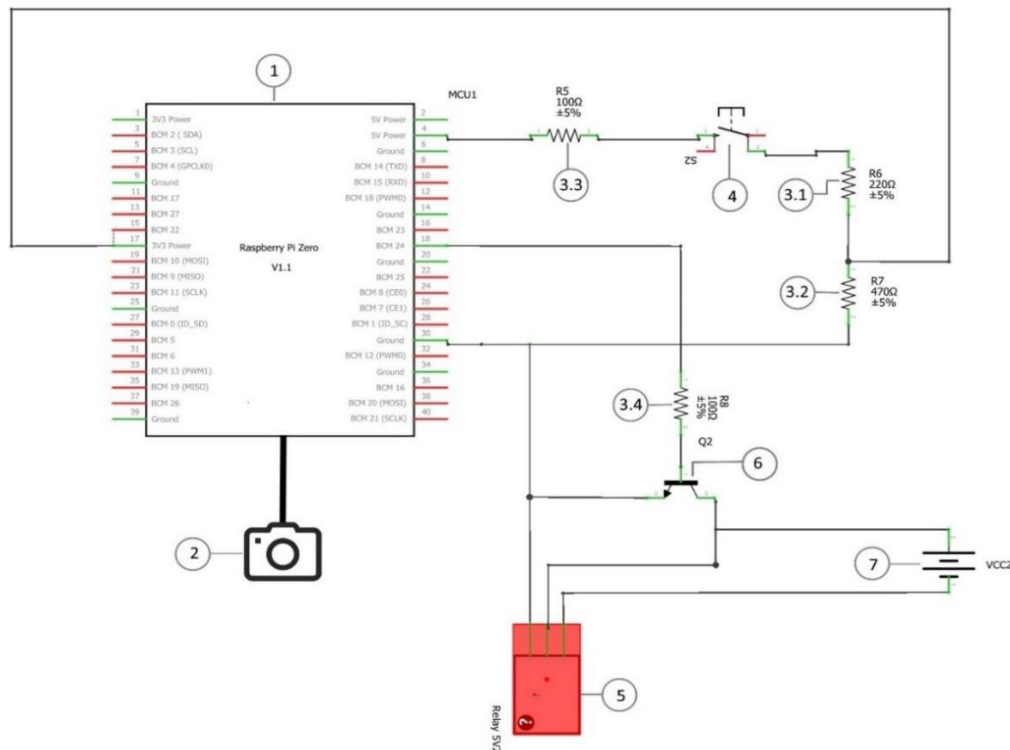
Nesta fase do projeto, foi realizada uma pesquisa de mercado visando selecionar uma plataforma de desenvolvimento que suprisse as necessidades de custo e processamento necessários para operação do sistema, assim foram analisadas *Arduino*, *Intel Galileo* e *Raspberry*.

Realizando uma análise de custo benefício em relação às plataformas supracitadas e a aquisição de periféricos para montar a arquitetura necessária para o funcionamento do sistema, optou-se pelo desenvolvimento na plataforma *Raspberry*, que fornece computadores de baixo custo e alto desempenho (RASPERRY, 2018).

Para este trabalho foi selecionada a placa *Raspberry Pi Zero W* (FILIPEFLOP, 2017), esta é responsável por controlar a aplicação a partir do sistema operacional livre *Raspbian* (RASPBIAN, 2018). A placa utiliza a conexão *camera serial interface* (CSI) e tecnologia *plug and play* no módulo de câmera para *Raspberry Pi V2* 8MB (FILIPEFLOP, 2016).

Para modelagem e prototipação do projeto utilizou-se o *Fritzing* (ARDUINO, 2015), que foi adotado por ser uma ferramenta de *software* livre que permite a geração de imagens esquemáticas, e acima de tudo disponibilizar a estrutura da maioria das plataformas destacadas neste trabalho. A Figura 11 apresenta a disposição esquemática dos componentes utilizados para construção do *hardware* sendo: (1) *Raspberry*, (2) módulo de câmera, (3) resistores, (4) *push button*, (5) módulo relé, (6) transistor NPN e (7) fonte de alimentação externa.

**Figura 11** – Esquemático do *hardware* desenvolvido



**Fonte:** elaborado pelo autor.

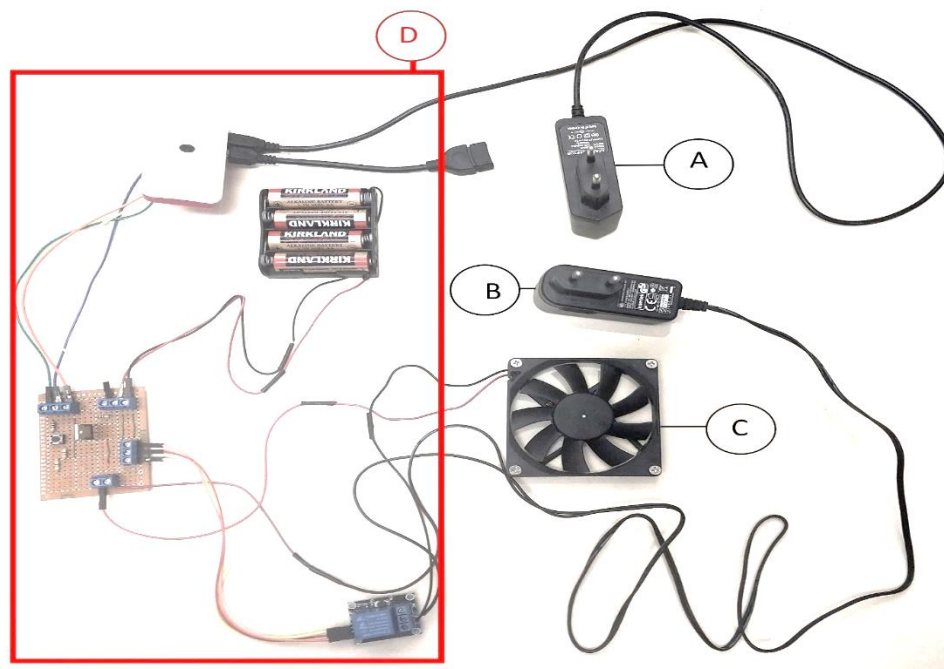
A fonte de alimentação externa é utilizada para fornecer carga para o módulo relé, já que a amperagem de operação fornecida pelo *Raspberry* é insuficiente para o funcionamento do mesmo e do módulo relé simultaneamente; a fonte também é conectada ao pino *coletor* do transistor para que, quando o pino 16 do *Raspberry* que é conectado ao pino *base* do transistor for acionado, a tensão da fonte seja transmitida



pelo pino *emissor* do transistor para acionar o módulo relé; o *push button* representa um acionamento do sistema realizado pelo usuário, onde (3.1) e (3.2) são resistores de 220 e 470 ohms, respectivamente, que são responsáveis por limitar a corrente que flui no circuito.

Após a elaboração do projeto, foi realizado o processo de montagem e soldagem dos componentes, resultando no esquema apresentado na Figura 12.

**Figura 12 – Esquemático final do sistema (MVP)**



**Fonte:** Elaborado pelo autor.

A Figura 12 apresenta os componentes utilizados na construção do sistema sendo (A) uma fonte de alimentação 5V, (B) fonte de alimentação 12V, (C) *cooler* de ventilação 12V (representando o equipamento elétrico que deverá ser acionado) e (D) representa o sistema embarcado encapsulado.

Assim (A) é responsável em fornecer carga para (D); (B) é responsável em fornecer carga para (C), representando respectivamente a rede elétrica predial e o aparelho elétrico. A partir dessa composição o conceito de funcionamento do sistema será validado.

### 3.3 METODOLOGIA DE DESENVOLVIMENTO DO SOFTWARE

O *software* do sistema embarcado é responsável por realizar a aquisição de imagens para processamento, classificação e acionamento do circuito proposto. Este



foi fixado numa aresta da sala a uma altura de dois metros e cinquenta e cinco centímetros (2,55 m).

### 3.3.2 Captura, organização e pré-processamento da base de dados

Com o sistema instalado foi dado seguimento a sua produção, nesta etapa o foco do desenvolvimento concentrou-se nos processos de captura, organização e pré-processamento da base de dados.

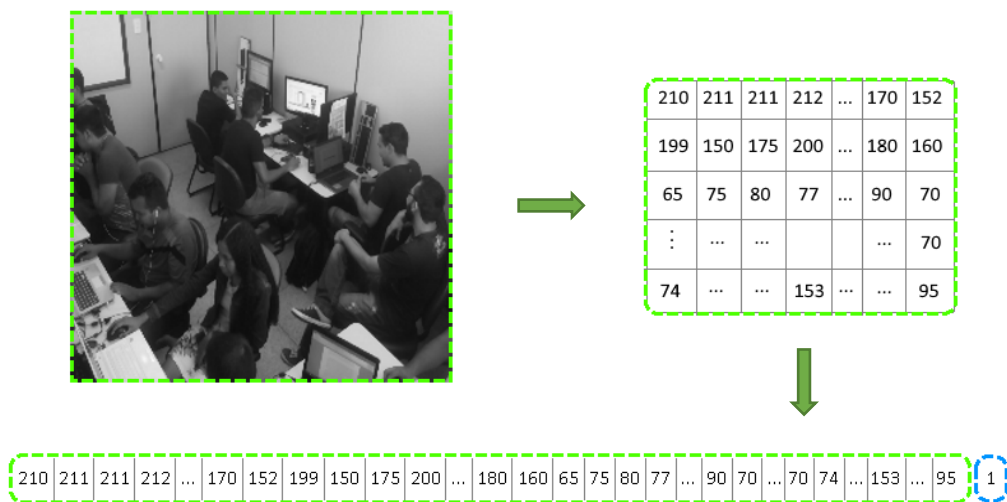
Para Gama et al. (2015) “a análise das características presentes num conjunto de dados permite a descoberta de padrões e tendências”. Por esse motivo criar uma base de dados diversificada significa garantir eficiência do sistema. Sendo assim, foram coletadas 330 imagens (amostras) do ambiente num período de sete (7) dias corridos, abrangendo horários de expedientes e intervalos de funcionamento do LABEX, para esta tarefa foi escrito um *script* que realiza a aquisição de uma nova imagem numa frequência de tempo de sete (7) minutos, essas imagens foram salvas diretamente num dispositivo de armazenamento externo USB plugado ao *Raspberry*.

A organização da base de dados foi uma etapa de suma importância para o desenvolvimento do sistema, já que a adoção do algoritmo *backpropagation* segue um modelo de aprendizagem supervisionada; assim a realização desta tarefa foi exclusivamente analítica, ou seja, cada amostra de imagem era classificada por análise humana e então redirecionada para um diretório específico resultando então em dois (2) diretórios distintos, de classes negativas (sem presença) e positivas (com presença), com 52 e 278 amostras, respectivamente.

Nesta etapa, outro *script* foi escrito utilizando métodos da biblioteca de processamento de imagens *OpenCV*. A partir da aplicação do *script* toda base de dados foi então reescrita para o tamanho cem por cem (100x100) e a coloração foi convertida para escala de cinza; porém segundo Silva (2009) alterar a proporção de uma imagem, pode causar uma distorção dos objetos presentes nela. Para atenuar as possíveis distorções (ruídos) oriundos do redimensionamento da base de dados, aplicou-se um filtro também presente na biblioteca, *bilateral filter*, que é altamente eficaz na redução de ruídos e mantém nítidas as bordas da imagem e ressaltam os contornos do objeto.

Após a criação e pré-processamento, foi modificada a representação das amostras, a qual foi convertida da forma matricial para uma estrutura vetorizada, onde cada vetor disposto em uma linha de uma matriz passou a representar determinada imagem coletada. A Figura 14 ilustra o procedimento de conversão da imagem matricial para vetorizada.

**Figura 14** – Vetorização das amostras da base de dados



**Fonte:** elaborado pelo autor.

Assim, foram finalizados os procedimentos referentes à preparação da base de dados para treinamento da RNA, que será discutido na próxima seção.

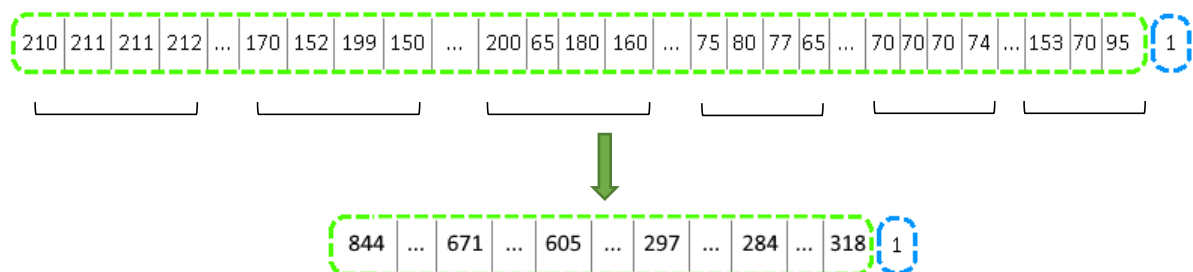
### 3.3.3 Treinamento da RNA

Nesta etapa definiu-se a arquitetura da RNA, no caso, uma rede de três (3) camadas, a primeira (entrada) recebendo os atributos de entrada, ou seja, as amostras de imagem, a segunda (camada escondida), que teve seu número de neurônios variados para avaliação de desempenho e a última (saída) que deverá apresentar o resultado de classificação da RNA.

A execução do treinamento da rede neural utilizou a biblioteca *PyBrain*, que reúne algoritmos de aprendizado supervisionado, não supervisionado e evolutivo. Desse modo, foi implementado um *script* que recebe como variáveis: a base de dados vetorizada, taxa de aprendizagem, o número de épocas e neurônios da camada oculta.

Para reduzir o tempo de treinamento da RNA, um *script* foi implementado visando diminuir o número de atributos de entrada. Inicialmente a base de dados possuía um padrão matricial de cem por cem (100x100), assim representando um vetor com um número de atributos de entrada equivalente a dez mil (10.000); este número acarreta em maior custo computacional durante etapa de treinamento e resulta em maior tempo de execução do *software* embarcado. Para reduzir o tempo de processamento, foi proposto o controle do número de posições em que se deve percorrer a amostra vetorizada, desta forma, os valores de *pixel* dessas posições foram somados e acumulados em um novo vetor, isto é, um vetor com menos elementos. Este procedimento foi realizado de forma iterativa em todas as imagens armazenadas no banco de dados. A Figura 15 ilustra esse procedimento para uma amostra da base de dados.

**Figura 15** – Segmentação das amostras da base de dados



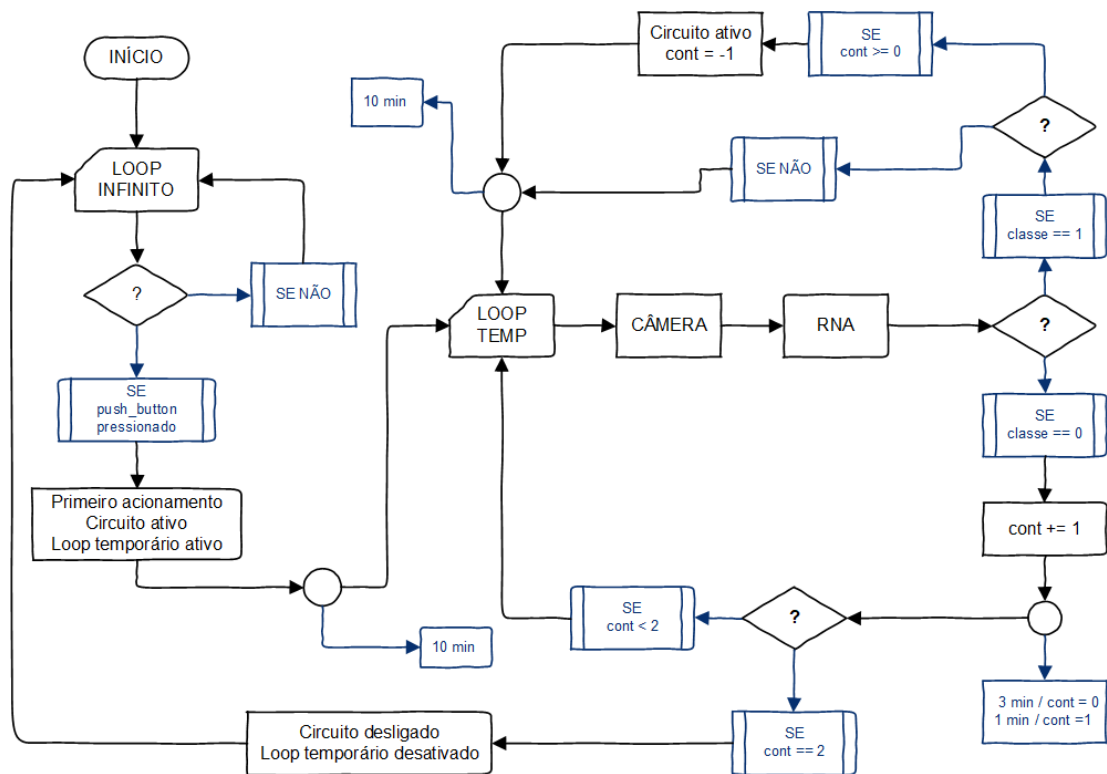
**Fonte:** elaborado pelo autor.

A partir da aplicação da redução do tamanho dos vetores foram criadas duas (2) bases de dados novas contendo quinhentos (500) e um mil (1000) atributos de entrada; todas as três bases de dados foram submetidas à etapa de treinamento da RNA e seus resultados serão apresentados na seção 4.

### 3.4 FUNCIONAMENTO DO SOFTWARE EMBARCADO.

A última etapa de desenvolvimento do projeto teve como objetivo modelar uma diretriz de funcionamento para o sistema. Esta levou em consideração o acionamento realizado por um usuário do sistema e se preocupou em considerar a capacidade computacional das RNA para o controle dos equipamentos elétricos. Para isso, um *script* para gestão do sistema foi modelado na forma de fluxograma e codificado. A Figura 16 apresenta o fluxograma do *software* embarcado e no Apêndice B um pseudocódigo deste fluxograma é apresentado.

**Figura 16** – Fluxograma do *software* embarcado



**Fonte:** elaborado pelo autor.

Assim que é ativado, o sistema entra em um *loop* infinito que monitora o acionamento do botão que indica uma ação do usuário com o pedido de inicialização de operação do aparelho, em seguida, o sistema gera um *flag* que aciona o relé para que seja ligado o aparelho e aguarda um período de dez (10) minutos para então entrar em um *loop* temporário que é responsável pela captura de uma amostra, essa amostra é submetida às etapas de pré-processamento e classificação por meio da RNA. Nesta etapa, a RNA é responsável em classificar a amostra como positiva (com presença humana) ou negativa (sem presença humana).

Quando classificado como positiva, a *flag* de controle do relé permanece ativa e é aguardado novamente dez (10) minutos para que uma nova amostra seja capturada; caso seja classificado como negativa, um contador (variável do sistema) é acrescido de um (1) e então o tempo de espera para que seja coletada uma nova amostra é reduzido para três (3) minutos, após três minutos uma nova imagem é capturada e apresentada para classificação da RNA. Caso a amostra seja novamente de classe negativa, o contador é outra vez acrescido de um (1) e o tempo de para captura de uma nova amostra é reduzido para um (1) minuto, por fim, se a nova amostra também for de classe negativa o contador é acrescido de um (1), assim

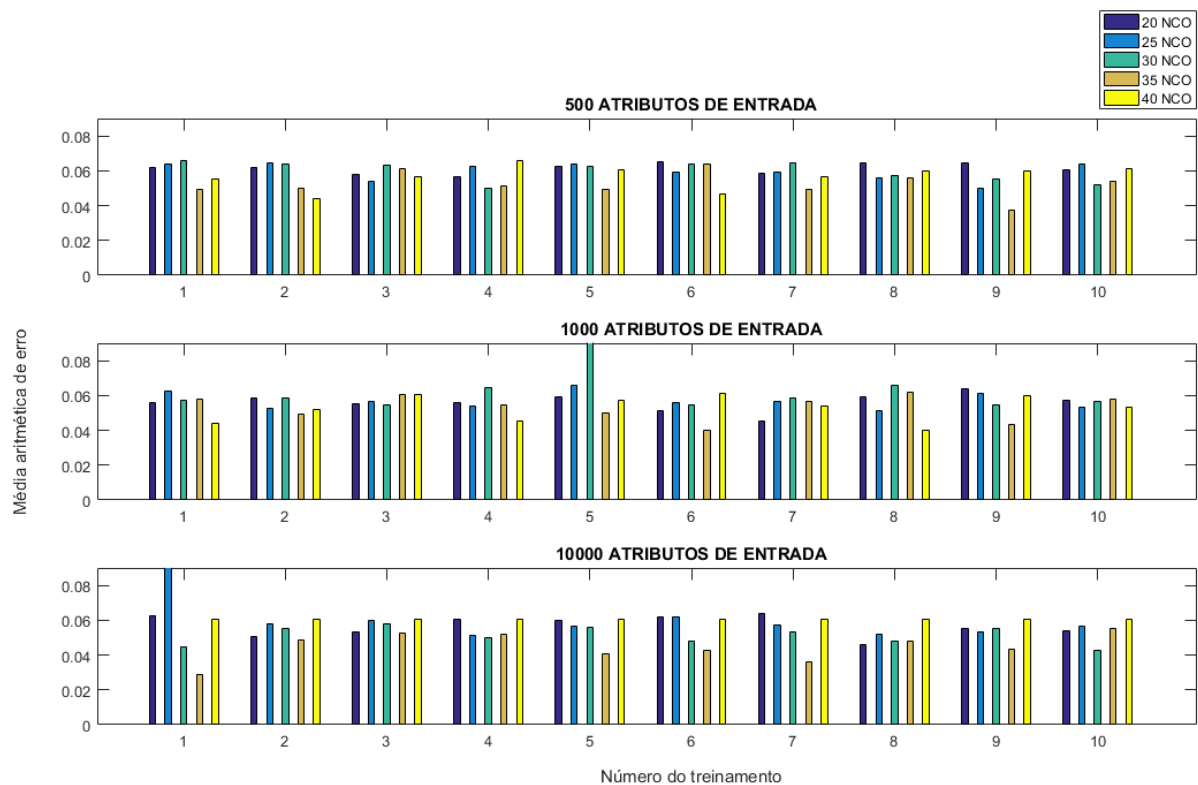
alcançando o valor de dois (2), o qual é o valor estipulado para desligamento do aparelho. Com este resultado, a *flag* que acionava o relé é então desabilitada, assim fazendo com que o aparelho seja desligado devido ao encerramento do *loop*. Para que o aparelho seja ligado novamente é necessário que o usuário pressione o botão de inicialização do sistema.

#### 4 AVALIAÇÃO DE DESEMPENHO DA RNA E ETAPA DE TESTE

Nesta etapa o *script* de treinamento da RNA foi executado inúmeras vezes variando-se o número de neurônios da camada oculta, número de épocas de treinamento e atributos de entrada. Além disso, a cada execução, a taxa de erro médio aritmético do treinamento e os pesos sinápticos da rede foram salvos. Para avaliação do desempenho dos treinamentos, cada variação da rede de treinamento foi executada dez (10) vezes, assim pode-se garantir coerência na análise e escolha da melhor RNA. Vale a pena reforçar que no treinamento das RNA utilizou-se oitenta por cento (80%) das amostras da base de dados, e vinte por cento (20%) foram preparadas para realização dos testes de classificação.

A Figura 17 apresenta os resultados obtidos no treinamento com número de épocas igual a quinhentos (500), com variação no número de atributos de entrada e neurônios na camada oculta (NCO).

**Figura 17 – Treinamento da RNA com 500 épocas**



**Fonte:** elaborado pelo autor.

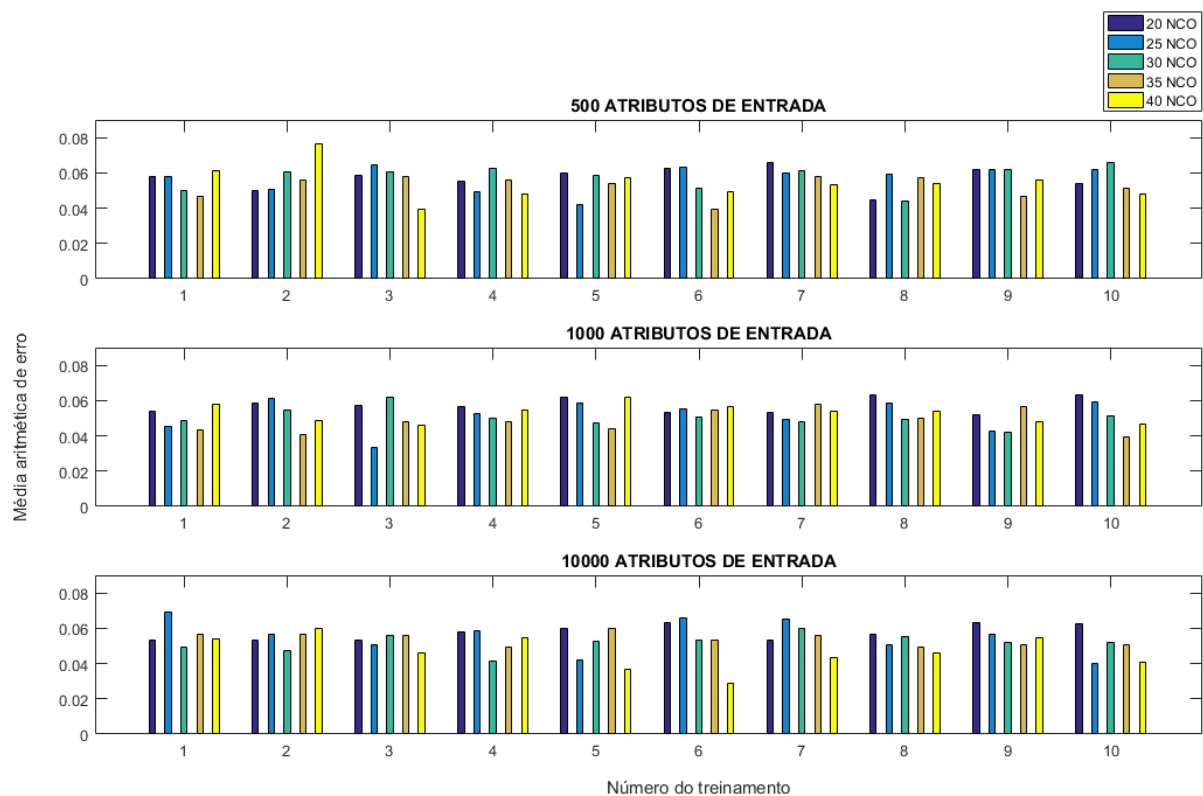
Com estes resultados, é possível perceber que nos treinamentos com: quinhentos (500) e um mil (1000) atributos de entrada, os melhores e piores índices foram esparsos, ou seja, ocorreram em quase todas as variações de neurônios da



camada oculta, porém ressalta-se que maiores ocorrências de eficiência foram observadas nas RNA com trinta e cinco (35) NCO para quinhentos atributos de entrada; e para os casos de trinta e cinco (35) e quarenta (40) NCO para um mil (1000) atributos de entrada. Em contrapartida para o treinamento com dez mil (10000) atributos de entrada, os melhores índices foram concentrados nas RNA com trinta (30) e trinta e cinco (35) NCO e os piores índices foram dispersos nas RNA com vinte (20), vinte e cinco (25) e quarenta (40) NCO.

A Figura 18 apresenta os resultados obtidos no treinamento com número de épocas igual a um mil (1000), com variação no número de atributos de entrada e NCO.

**Figura 18 – Treinamento da RNA com 1000 épocas**



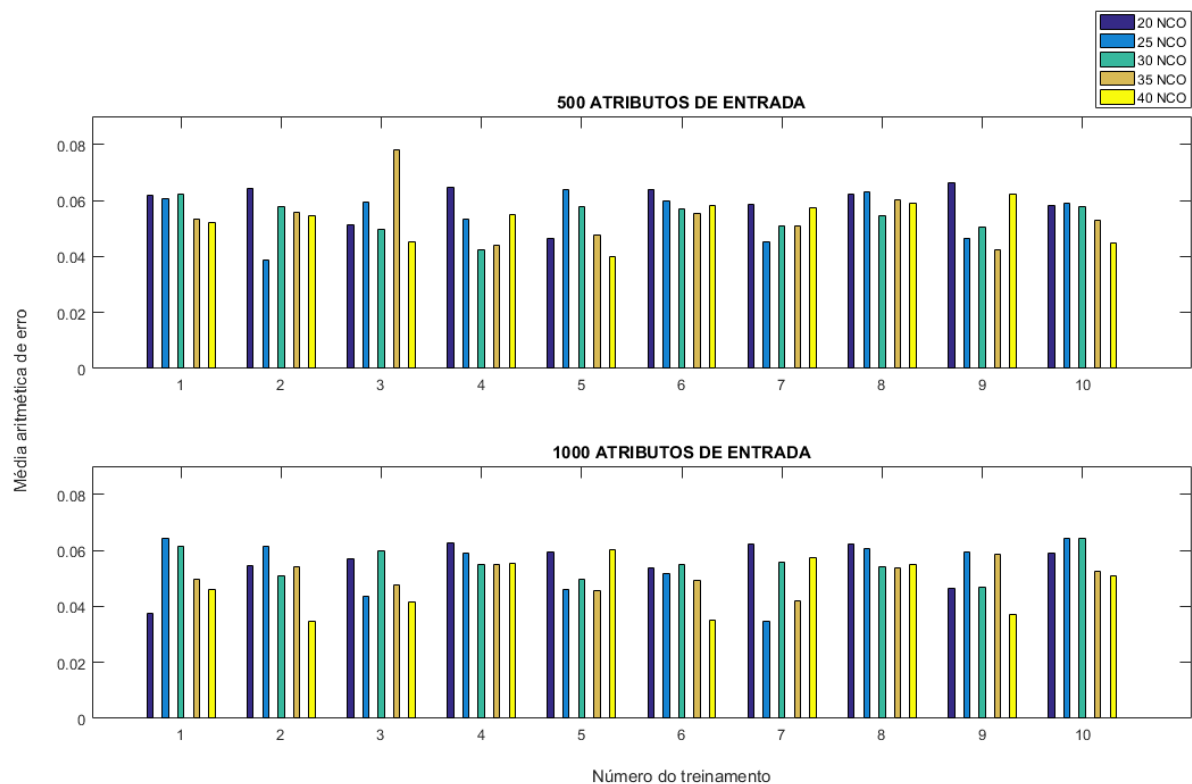
**Fonte:** elaborado pelo autor.

Nesta análise, observou-se uma dispersão dos melhores e piores índices para todos os atributos de entrada, entretanto na maioria dos casos os melhores índices convergiram para as RNA com os maiores números de neurônios na camada oculta e os piores índices se comportaram de maneira oposta, evidenciando uma tendência de melhor treinamento para RNA com alto número de neurônios na camada oculta. Assim para quinhentos (500) atributos de entrada foram registradas quatro (4) ocorrências de melhor índice para RNA com NCO igual a quarenta (40); para um mil (1000)

atributos de entrada foram registradas cinco (5) ocorrências de melhor índice para RNA com NCO igual a trinta e cinco (35); e para dez mil (10000) atributos de entrada foram registradas cinco (5) ocorrências de melhor índice para RNA com NCO igual a quarenta (40).

A Figura 19 apresenta os resultados obtidos no treinamento com número de épocas igual a um mil e quinhentos (1500), com variação no número de atributos de entrada (ressalta-se que o treinamento com dez mil (10000) atributos de entrada não foi realizado devido seu alto custo computacional) e NCO.

**Figura 19 – Treinamento da RNA com 1500 épocas**



**Fonte:** elaborado pelo autor.

Nesta análise, mais uma vez foi perceptível a tendência de eficiência no treinamento para RNA com alto número de neurônios na camada oculta, nestes casos, para quinhentos atributos de entrada e para um mil (1000) atributos de entrada observou-se, respectivamente, quatro (4) e cinco (5) ocorrências de melhor índice para RNA com quarenta (40) NCO.

A partir destas constatações a Tabela 5 apresenta, de acordo com o número de épocas do treinamento, as arquiteturas das RNA mais eficientes e suas respectivas taxas de erro aritméticas.

**Tabela 5 – Melhores índices de treinamento**

<i>500 ÉPOCAS</i>		
Número de atributos de entrada	Número de neurônios na camada oculta	Taxa de erro aritmético
500	35	0,03766
1000	40	0,03979
10000	35	<b>0,02903</b>
<i>1000 ÉPOCAS</i>		
Número de atributos de entrada	Número de neurônios na camada oculta	Taxa de erro aritmético
500	40	0,03952
1000	25	0,03309
10000	40	<b>0,02896</b>
<i>1500 ÉPOCAS</i>		
Número de atributos de entrada	Número de neurônios na camada oculta	Taxa de erro aritmético
500	25	0,03864
1000	40	<b>0,03527</b>

**Fonte:** elaborado pelo autor.

A Tabela 6 apresenta o percentual de acerto das melhores RNA. Esses percentuais foram calculados a partir da relação de quantidade proporcional obtida por meio de uma regra de três simples. Assim constatou-se a eficiência das RNA como método de classificação e controle do sistema; isso representa que o sistema contornará os erros que podem comprometer seu desempenho, que no caso, seriam indicar o desligamento do equipamento conectado quando na verdade ainda há presença de pessoas no ambiente ou mantê-lo ligado quando não houver necessidade.

**Tabela 6 – Percentual de acerto das RNA**

Número de atributos de entrada	Percentual de acerto (%)
500	85,1
1000	<b>92,5</b>
10000	83,6

**Fonte:** elaborado pelo autor.

Além da validação apresentada na Tabela 6, neste trabalho também foi realizado um teste controlado utilizando o MVP apresentado na Figura 12. Para isso, foram utilizados os seguintes dados: melhor RNA treinada, isto é, RNA com 1000

atributos de entrada e 40 NCO (apresentado na Tabela 5), e uma base de dados de teste composta por seis (6) testes compostos por seis (6) imagens cada (que será apresentado na Tabela 7).

Para a realização de todos os testes, o sistema foi iniciado via o botão (4) *push button*, da Figura 11. Em seguida, as imagens foram apresentadas ao sistema seguindo os passos definidos no Fluxograma apresentado na Figura 16. Tabela 7 apresenta a relação dos padrões de entrada apresentados e o *status* final do MVP. Nesta tabela (P) simboliza a presença de pessoas na amostra e (A) simboliza a ausência de pessoas no ambiente.

**Tabela 7 – Relação de entradas e status do MVP**

<b>Teste</b>	<b>Padrão de entrada</b>						<b>Status do MVP</b>
<b>1</b>	A	A	A	-	-	-	Desligado
<b>2</b>	P	A	A	A	-	-	Desligado
<b>3</b>	P	A	A	P	A	A	Desligado
<b>4</b>	P	A	A	P	P	A	Ligado
<b>5</b>	P	P	A	A	P	A	Ligado
<b>6</b>	A	P	P	P	P	P	Ligado

**Fonte:** elaborado pelo autor.

No primeiro teste três imagens A foram apresentadas. Após o reconhecimento correto o sistema foi desligado pelo algoritmo proposto. No segundo teste a primeira imagem indicava a presença de pessoas na sala, em seguida, três imagens foram apresentadas sem a presença de pessoas, logo o sistema foi desligado após o cumprimento das etapas do fluxograma. No terceiro teste, a primeira imagem apresentada indicava a presença de pessoas na sala, em seguida duas imagens não indicavam a presença de pessoas, em seguida uma imagem P foi apresentada, após houve a apresentação de duas imagens A e então o sistema foi desligado. No quarto teste repetiu-se o padrão de identificação do terceiro teste até a quarta amostra de imagem, na quinta amostra houve a apresentação de uma imagem P, em seguida foi apresentada uma imagem A, dessa forma o sistema permaneceu ligado. No quinto teste, foram apresentadas duas amostras que indicavam presença de pessoas em seguida duas amostras que indicavam a ausência de pessoas, a quinta amostra apresentada era uma imagem P e a sexta amostra uma imagem A, neste caso sistema permaneceu ligado. No sexto teste, a primeira imagem apresentada não possuía a

presença de pessoas e as outras cinco (5) amostras seguintes possuíam a presença de pessoas, nesta configuração o sistema permaneceu ligado.

Como esperado, o sistema proposto se comportou de maneira satisfatória nos seis (6) testes realizados sempre obedecendo ao fluxograma proposto. A Figura 20 apresenta as telas do terminal de comando utilizado para realizar os testes dois (2) e seis (6).

**Figura 20 – Exemplos dos testes de validação**

```

gnleo@developer:~$ ssh pi@192.168.137.196
pi@192.168.137.196's password:
Linux gnleo 4.14.34+ #11118 Mon Apr 16 14:51:42 BST 2018 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Jan 6 19:27:23 2019
pi@gnleo:~$ cd /media/pi/LE0#2/cod_fontes/ && python3 software_embarcado_simulacao.py
Sistema offline - ligar sistema

Sistema online
primeiro acionamento do sistema - feito por presença humana
delay representativo 10 min

realizando classificacao RNA | amostra = 0
SAIDA RNA = [ 0.97426347]
CLASSE POSITIVA
contador de controle = -1
delay representativo 10 min

realizando classificacao RNA | amostra = 1
SAIDA RNA = [ 0.19592016]
CLASSE NEGATIVA
contador de controle = 0
delay representativo 3 min

realizando classificacao RNA | amostra = 2
SAIDA RNA = [ 0.43160130]
CLASSE NEGATIVA
contador de controle = 1
delay representativo 1 min

realizando classificacao RNA | amostra = 3
SAIDA RNA = [ 0.19592016]
CLASSE NEGATIVA
contador de controle = -1
Sistema offline - ligar sistema

```

```

pi@gnleo:/media/pi/LE0#2/cod_fontes$ python3 software_embarcado_simulacao.py
Sistema offline - ligar sistema

Sistema online
primeiro acionamento do sistema - feito por presença humana
delay representativo 10 min

realizando classificacao RNA | amostra = 0
SAIDA RNA = [ 0.97426347]
CLASSE POSITIVA
contador de controle = -1
delay representativo 10 min

realizando classificacao RNA | amostra = 1
SAIDA RNA = [ 0.9076453]
CLASSE POSITIVA
contador de controle = -1
delay representativo 10 min

realizando classificacao RNA | amostra = 2
SAIDA RNA = [ 0.96490809]
CLASSE POSITIVA
contador de controle = -1
delay representativo 10 min

realizando classificacao RNA | amostra = 3
SAIDA RNA = [ 0.99969225]
CLASSE POSITIVA
contador de controle = -1
delay representativo 10 min

realizando classificacao RNA | amostra = 4
SAIDA RNA = [ 0.9076453]
CLASSE POSITIVA
contador de controle = -1
delay representativo 10 min

realizando classificacao RNA | amostra = 5
SAIDA RNA = [ 0.96490809]
CLASSE POSITIVA
contador de controle = -1
delay representativo 10 min

Numero de amostras atingido - saindo do loop de controle

```

(a)
(b)

**Fonte:** elaborado pelo autor.

A Figura 20(a) é um *print screen* das saídas do teste dois (2) e ilustra um funcionamento com 100% de acerto por parte do sistema, onde a RNA classificou corretamente todas as entradas; enquanto que na Figura 20(b) um *print screen* do teste seis (6) e demonstra um (1) erro de classificação da RNA, o qual foi contornado pela variável de controle do sistema, que como especificado, quando esta variável atingir o valor dois (2) a mesma volta a obter o valor de inicialização e então o equipamento conectado é desligado.

Com isso a partir dos testes executados ressalta-se o potencial deste sistema para atuar no controle de consumo energético de residências, empresas públicas, privadas e até mesmo ambientes acadêmicos.

## 5 CONCLUSÃO

Este trabalho apresentou um sistema embarcado que executa o gerenciamento inteligente de aparelhos elétricos, controlando o tempo de operação dos mesmos; possibilitando com que haja uma redução no consumo de eletricidade dos aparelhos. Consoante aos estudos e pesquisas elucidados, destaca-se o potencial deste produto para solução do alto consumo energético de instituições públicas ou privadas.

A partir deste trabalho é possível afirmar que a metodologia empregada para solução da problemática é altamente eficiente; no que diz respeito a utilização do algoritmo *backproagation*, como ferramenta de identificação do padrão estabelecido (presença ou ausência de indivíduos no ambiente controlado), mesmo quando utilizado como *software* embarcado. Além disso, os resultados demonstram o poder de generalização do algoritmo *backproagation*, quando apresentado a diferentes tamanhos de base de dados, pois para diferentes quantidades de neurônios na camada oculta o algoritmo se adaptou e respondeu com erros médios inferiores a 0,08.

Os resultados demonstram que é possível monitorar e gerenciar o consumo energético de equipamentos elétricos usando sistema embarcado, algoritmos de inteligência artificial e fluxograma de gerenciamento. Desta forma, sendo possível considerar o uso de um sistema embarcado quanto ao processo de acionamento, uso e desligamento do dispositivo controlado (*cooler*, representando determinado equipamento elétrico genérico). Além disso, os resultados comprovam que a implantação deste sistema pode ser um caminho para a futura redução do consumo elétrico, assim gerando economia energética para as instituições.

A partir dos resultados é possível concluir que a utilização de imagens coletadas por determinada câmera pode ser um caminho eficiente e de baixo custo para controle e redução do consumo energético, pois a partir das mesmas em conjunto com o sistema embarcado proposto neste trabalho, torna-se possível gerenciar o desligamento automático do sistema elétrico que alimenta o dispositivo eletrônico. Desta forma, gerando a possibilidade de economia financeira, assim como, reduzindo a emissão desnecessária de gás carbônico para a atmosfera.

Para trabalhos futuros espera-se escrever a proposta de patente deste sistema, assim como, realizar testes integrando o sistema proposto com uma placa real de determinado aparelho, desenvolver uma central de controle, a partir da implementação de uma rede intranet, em que várias unidades deste sistema possam ser instaladas em um edifício, por exemplo, e serem acionadas por um painel central de comando; com isso há a possibilidade de integração deste produto com sistemas de segurança, apresentando um indicador de presença mais confiável que monitore o ambiente a partir de padrões de comportamento das pessoas que fazem uso do espaço em que este sistema esteja instalado. Com isso, pode-se afirmar que a apresentação deste sistema é nada mais do que uma proposta de empreendimento que pode e deverá receber *upgrades*.

## 6 REFERÊNCIAS

**A Step by Step Backpropagation Example.** MAZZUR, Matt, 2019. Disponível em <<https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/>>.

Acesso em 1 jan. 2019.

**About.** OPENCV, 2018. Disponível em <<https://opencv.org/about.html>>. Acesso em 26 set. 2018.

**About.** PYTHON, 2018. Disponível em <<https://www.python.org/about/>>. Acesso em 26 set. 2018.

**About us.** RASPBERRY, 2018. Disponível em <<https://www.raspberrypi.org/about/>>. Acesso em 24 de setembro de 2018.

**Arduino UNO.** EMBARCADOS, 2013. Disponível em <<https://www.embarcados.com.br/arduino-uno/>>. Acesso em 13 nov. 2018.

AZEVEDO, Eduardo; CONCI, Aura; LETA, Fabiana R. Computação gráfica-volume 2: Teoria e prática. Elsevier Brasil, 2018.

Ballard, Dana H. and Brown, Christopher M. (1982). "Computer Vision". Prentice Hall. ISBN 0131653164.

BARRETO, Jorge Muniz. Inteligência Artificial no limiar do século XXI. Florianópolis: PPP edições, v. 97, 1999.

BARRETO, Jorge M. Introdução as redes neurais artificiais. **V Escola Regional de Informática. Sociedade Brasileira de Computação, Regional Sul, Santa Maria, Florianópolis, Maringá**, p. 5-10, 2002.

BINOTI, Daniel Henrique Breda; DA SILVA BINOTI, Mayra Luiza Marques; LEITE, Helio Garcia. Configuração de redes neurais artificiais para estimação do volume de árvores. **Revista Ciência da Madeira (Brazilian Journal of Wood Science)**, v. 5, n. 1, p. 10-12953/2177-6830. v05n01a06, 2014.

Broy M. "Requirements Engineering for Embedded Systems", *Proceedings of FemSys*: 1997.

CARRO, Luigi; WAGNER, Flávio Rech. Sistemas computacionais embarcados. **Jornadas de atualização em informática. Campinas: UNICAMP**, 2003.



CARVALHO, Alex et al. O que é metodologia científica. **Marília: Fundação Unimed**, 2007.

CHUA, K. J.; CHOU, S. K.; YANG, W. M.; YAN, J. **Achieving better energy-efficient air conditioning—a review of technologies and strategies**. *Applied Energy*, 104, 87-104. 2013.

**Cobrança de tarifas.** CELPA, 2018. Disponível em <<http://www.celpe.com.br/residencial/informacoes/cobranca-de-tarifas>>. Acesso em 15 out. 2018.

CUNHA, Alessandro F. O que são sistemas embarcados. **Saber Eletrônica**, v. 43, n. 414, p. 1-6, 2007.

DANGY-CAYE, Nicolas. **Processo e dispositivo de redução do consumo elétrico de uma interface ethernet**. Int H04L 12/12. PI 1016169-4 A2. 16 abr. 2010, 19 abr. 2016. Instituto Nacional da Propriedade Industrial.

DAVIS, L. W.; GERTLER, P. J. **Contribution of air conditioning adoption to future energy use under global warming**. *Proceedings of the National Academy of Sciences*. p. 201423558. 2015.

**Editorial: Por que é tão difícil desenvolver sistemas embarcados no Brasil?** EMBARCADOS, 2015. Disponível em: <<https://www.embarcados.com.br/editorial-desenvolver-sistemas-embarcados-no-brasil/>>. Acesso em: 04 dez. 2018.

**Eficiência energética.** EPE, 2018. Disponível em <<http://www.epe.gov.br/pt/abcdenergia/eficiencia-energetica>>. Acesso em 28 set. 2018.

**Equipe Raspberry Pi lançou Raspberry Pi Zero Wireless.** LANZA, Nancy, 2017. Disponível em: <<https://www.techzoogle.com/computing/raspberry-pi-team-has-released-raspberry-pi-zero-wireless>>. Acesso em: 04 dez. 2018.

ESQUEF, Israel Andrade; ALBUQUERQUE, Márcio Portes de; ALBUQUERQUE, Marcelo Portes de. Processamento digital de imagens. CENTRO BRASILEIRO DE PESQUISAS FÍSICAS—CBPF, 2003.

ESTRELA, Carlos. Metodologia científica: ciência, ensino, pesquisa. Artes Médicas, 2005.

**Estratégias de Desenvolvimento de Hardware para Sistemas Embarcados – Arquitetura Modular.** COLA, Fernando Luiz, 2017. Disponível em <<https://www.embarcados.com.br/estrategias-de-desenvolvimento-arquitetura-modular/>>. Acesso em 21 nov. 2018.

FARIAS, F. S. et al. Método de Identificação de Ruído em Redes DSL usando Rede Neural Artificial. **XXX Simpósio Brasileiro de Telecomunicações–SBrT**, p. 978-85. 2012.

FELIX, Emerson Felipe; POSSAN, Edna. Modeling the Carbonation Front of Concrete Structures in the Marine Environment through ANN. **IEEE Latin America Transactions**, v. 16, n. 6, 2018.

FERNEDA, Edberto. Redes neurais e sua aplicação em sistemas de recuperação de informação. **Ciência da Informação**, v. 35, n. 1, 2006.

**FLORES**, Franklin César; **PERES**, Sarajane Marques. Revisão Bibliográfica: Análise de Componentes Principais Linear e Não Linear X Redes Neurais. 2001.

**Fritzing – Modelagem de Circuitos.** ARDUDINO, 2015. Disponível em <<https://ardudino.wordpress.com/2015/12/13/fritzing-modelagem-de-circuitos/>>. Acesso em 28 set. 2018.

**Função de ativação.** DEEP LEARNING BOOK, 2019. Disponível em <<http://deeplearningbook.com.br/funcao-de-ativacao/>>. Acesso em 21 de jan. 2019.

GAMA, João et al. **Extração de conhecimento de dados: data mining.** 2015.

GILLINGHAM, K.; RAPSON, D.; WAGNER, G. **The rebound effect and energy efficiency policy.** *Review of Environmental Economics and Policy*. v. 10, n. 1, p. 68-88, 2016.

GROTHER, Everaldo Luiz. **Aparelho para redução do consumo elétrico de chuveiros elétricos através do monitoramento da vazão de água e da temperatura de banho.** Int A47K 3/28, F24D 17/00, E03C 1/044. BR 10 2012 012424-6 A2. 24 mai. 2012, 14 mai. 2013. Instituto Nacional da Propriedade Industrial.

HAYKIN, Simon. **Redes neurais: princípios e prática**. Bookman Editora, 2007.

**Intel Galileo Gen2**. BAU DA ELETRONICA, 2019. Disponível em <<http://www.baudaeletronica.com.br/>>. Acesso em 21 de jan. 2019.

MARENGONI, Maurício; STRINGHINI, Stringhini. Tutorial: Introdução à visão computacional usando opencv. Revista de Informática Teórica e Aplicada, v. 16, n. 1, p. 125-160, 2009.

MARQUES FILHO, Ogê; NETO, Hugo Vieira. Processamento digital de imagens. Brasport, 1999.

MCNEIL, M. A.; LETSCHERT, V. E. **Future air conditioning energy consumption in developing countries and what can be done about it: the potential of efficiency in the residential sector**. 2008.

MORAIS, Antonio Higor Freire de. **Eosi: um modelo para desenvolvimento de sistemas embarcados tolerantes a falhas**. 2009. Dissertação de Mestrado. Universidade Federal do Rio Grande do Norte.

MURAD, Omar R; ANTUNES, Luiz Fernando; DANIEL, Denis Allan. **Método de gerenciamento de consumo energético para aparelhos eletrodomésticos**. Int G05F 1/66; G05F 1/56. BR 10 2015 015639-1 A8. 30 jun. 2015, 03 jan. 2017. Instituto Nacional da Propriedade Industrial.

NIKU, Saeed B. Introdução à robótica: análise, controle, aplicações. LTC, 2013.

**Novidade! Câmera Raspberry Pi 8 Megapixels**. FILIPEFLOP, 2016. Disponível em <<https://www.filipeflop.com/blog/nova-camera-raspberry-pi-8-megapixels/>>. Acesso em 24 set. 2018.

**Novo Raspberry Pi Zero W com Wifi e Bluetooth**. FILIPEFLOP, 2017. Disponível em <<https://www.filipeflop.com/blog/raspberry-pi-zero-w-com-wifi-e-bluetooth/>>. Acesso em: 24 set. 2018.

OSORIO, Fernando Santos. Um estudo sobre reconhecimento visual de caracteres através de Redes Neurais. 1991

OSÓRIO, Fernando S.; BITTENCOURT, João Ricardo; OSÓRIO, Fernando Santos. Sistemas Inteligentes baseados em redes neurais artificiais aplicados ao processamento de imagens. In: **I Workshop de inteligência artificial**. 2000.

OSSADA, Jaime Cazuhiro; MARTINS, Luiz Eduardo Galvão. Um estudo de campo sobre o estado da prática da elicitación de requisitos em sistemas embarcados. 2010.

PERELMUTER, Guy et al. Reconhecimento de imagens bidimensionais utilizando Redes Neurais Artificiais. **Anais do VIII SIBGRAPI**, p. 197-203, 1995.

PEREIRA, Luís Moniz. Inteligência Artificial: mito e ciência. **Revista Colóquio-Ciências**, v. 3, p. 1-13, 1988.

PINNOLA, Carlos Fernando; VARGAS, José Viriato Coelho; PARISE, José Alberto dos Reis; ORDONEZ, Juan Carlos; BUZELIN, Luiz Otávio Soranço; CAMPOS, Marcos Carvalho. **Kit inteligente para refrigeração, ar condicionado e aquecimento**. Int F24F 11/00; G055D 23/19. BR 10 2012 018917-8 A2. 30 jul. 2012, 30 dez. 2014. Instituto Nacional da Propriedade Industrial.

**Primeiros passos com Intel Galileo Gen 2**. FILIPEFLOP, 2015. Disponível em <<https://www.filipeflop.com/blog/primeiros-passos-intel-galileo-gen-2/>>. Acesso em 13 nov. 2018.

**Primeiros passos com Raspberry Pi e Linux**. FILIPEFLOP, 2018. Disponível em <<https://www.filipeflop.com/blog/primeiros-passos-raspberry-pi-e-linux/>>. Acesso em 13 nov. 2018.

**Projeto de Hardware, como escolher qual plataforma utilizar**. ANA, Arles Sant, 2016. Disponível em: <<https://www.embarcados.com.br/projeto-de-hardware-como-escolher-qual-plataforma-utilizar/>>. Acesso em: 04 dez. 2018.

RAUBER, Thomas Walter. Redes neurais artificiais. Universidade Federal do Espírito Santo, 2005.

**Redes neurais, perceptron multicamadas e o algoritmo backpropagation**. LEITE, Tiago M, 2018. Disponível em <<https://medium.com/ensina-ai/redes-neurais-perceptron-multicamadas-e-o-algoritmo-backpropagation-eaf89778f5b8>>. Acesso em: 1 jan. 2019.

**Representação do Arduino UNO (ATMEL ATMEGA328).** JESUS, Hudson Azevedo de, 2015. Disponível em: <[https://www.researchgate.net/figure/Figura-2-Representacao-do-Arduino-UNO-ATMEL-ATMEGA328\\_fig9\\_309919381](https://www.researchgate.net/figure/Figura-2-Representacao-do-Arduino-UNO-ATMEL-ATMEGA328_fig9_309919381)>. Acesso em: 04 dez. 2018.

RIOS, Luiz Romário Santana. Visão computacional. **Departamento de Ciência da computação — Universidade Federal da Bahia, Salvador, BA**, 2011.

RODRIGUES, Afonso Celso Simões. **Aproveitamento energético e estético utilizando sistema de ar condicionado tipo Split com condensação à água.** Int F24F 5/00. MU 8901228-3 U2. 04 jun. 2009, 22 fev. 2011. Instituto Nacional da Propriedade Industrial.

SANCHES, Carlos H. et al. Técnicas de Suavização de Imagens e Eliminação de Ruídos. Anais do EATI-Encontro Anual de Tecnologia da Informação, p. 21-30. 2015.

SANTOS, Danillo Moura. Projeto de sistemas embarcados: Um estudo de caso baseado em microcontrolador e seguindo AOSD. 2005.

SILVA, Oscar Paesi da. **Redimensionamento de imagens preservando a proporção dos objetos.** 2009.

**Sistema Embarcado – O que é? Qual a sua importância?** EMBARCADOS, 2013. Disponível em <<https://www.embarcados.com.br/sistema-embarcado/>>. Acesso em 16 nov. 2018.

SOUZA, Taciana; CORREIA, Suzete. Estudo de técnicas de realce de imagens digitais e suas aplicações. **João Pessoa. Paraíba**, 2007.

**The most popular Python data Science platform.** ANACONDA, 2018. Disponível em <<https://www.anaconda.com/what-is-anaconda/>>. Acesso em 26 set. 2018.

WEISE, Andreas D.; HORNBURG, Ricardo André. Gestão de energia em edifícios. **XXVII Encontro Nacional de Engenharia de Produção, Foz de Iguaçu, Proceedings... Foz de Iguaçu: ENEGEP**, 2007.

**Welcome to PyBrain.** PYBRAIN, 2018. Disponível em <<http://pybrain.org/>>. Acesso em 26 set. 2018.

**Welcome to raspbian.** RASPBIAN, 2018. Disponível em <<https://www.raspbian.org/>>. Acesso em 27 set. 2018.

**What is GNU/Linux.** (GNU/LINUX, 2018). Disponível em <<https://help.ubuntu.com/lts/installation-guide/armhf/ch01s03.html>>. Acesso em 27 set. 2018.

## APÊNDICE A

COMPONENTES	Plataformas – Custo (R\$)		
	ARDUINO	RASPBERRY	INTEL GALILEO
Resistores <sup>1</sup>	4,8	4,8	4,8
Push button	3,4	3,4	3,4
Transistor NPN	2,9	2,9	2,9
Cartão de memória Micro SD	-	74,9	74,9
Módulo câmera	104,78	170	104,78
Módulo relé	8,9	8,9	8,9
sub Total	124,78	264,9	199,38

<sup>1</sup> Conjunto de 20 unidades

## APÊNDICE B

```

var
  on = 0
  start = 1
  cont_controle = -1
  pino_entrada = 15
  pino_saida = 16

# ROTINA DO SISTEMA
Enquanto start == 1 Faça

  Se pino_entrada == HIGH Entao
    on = 1
    primeiro_acionamento = 1

  Enquanto on == 1 Faça

    Se primeiro_acionamento == 1 Entao
      primeiro_acionamento = 0
      acionar(pino_saida)
      delay(600) # 10 min = 600 s

    classe = RNA()

    Se classe == 0 Entao

      cont_controle += 1

      Se cont_controle == 0 Entao
        delay(180) # 3 min = 180 s

      Se cont_controle == 1 Entao
        delay(60) # 1 min = 60 s

      Se cont_controle == 2 Entao
        desacionar(pino_saida)
        on = 0
        cont_controle = -1

    Senao

      Se cont_controle >= 0 Faça
        cont_controle -= 1

      delay(600) # 10 min = 600 s

  fimEnquanto
fimEnquanto

```

```

function capturar_img()
  return img

function pre_processamento(img)
  img_resize = resize(img, 100, 100)
  img_gray = gray(img_resize)
  img_filtrada = bilateralFilter(img_gray)
  return img_filtrada

function forward(img, neural)
  rede = neural
  return resultado = img.dot(rede)

function RNA()
  constant arq_neural_treinada
  img = capturar_img()
  img_pre = pre_processamento(img)
  res = forward(img_pre, neural_treinada)

  Se res < 0.5 Entao
    classe = 0
  Senao
    classe = 1

```