

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

PEDRO BERTELLI

**SISTEMA DE MONITORAMENTO PARA DETERMINAÇÃO DE DESEMPENHO
EM UMA LINHA SERIADA COM BASE NA DETECÇÃO DE CONTORNOS E
VÉRTICES COM OPENCV**

CAMPO MOURÃO

2021

PEDRO BERTELLI

**SISTEMA DE MONITORAMENTO PARA DETERMINAÇÃO DE DESEMPENHO
EM UMA LINHA SERIADA COM BASE NA DETECÇÃO DE CONTORNOS E
VÉRTICES COM OPENCV**

**Monitoring system for determining performance in a serial line based on
contour and vertex detection with OpenCV**

Trabalho de conclusão de curso de graduação
apresentado como requisito parcial para obtenção do
título de Bacharel em Engenharia Eletrônica da
Universidade Tecnológica Federal do Paraná
(UTFPR).

Orientador: Prof. Me. Lucas Ricken Garcia

CAMPO MOURÃO

2021



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos.

Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

PEDRO BERTELLI

**SISTEMA DE MONITORAMENTO PARA DETERMINAÇÃO DE DESEMPENHO
EM UMA LINHA SERIADA COM BASE NA DETECÇÃO DE CONTORNOS E
VÉRTICES COM OPENCV**

Trabalho de conclusão de curso de graduação
apresentado como requisito parcial para obtenção do
título de Bacharel em Engenharia Eletrônica da
Universidade Tecnológica Federal do Paraná
(UTFPR).

Data de aprovação: 29 de novembro de 2021

Lucas Ricken Garcia
Mestre
Universidade Tecnológica Federal do Paraná

Leandro Castilho Brolin
Doutor
Universidade Tecnológica Federal do Paraná

Osmar Tormena Junior
Doutor
Universidade Tecnológica Federal do Paraná

CAMPO MOURÃO

2021

AGRADECIMENTOS

Com toda certeza, neste momento não conseguirei atender a todos que estiveram presentes durante o desenvolvimento deste trabalho. Peço desculpas aos que não estão nessas palavras, mas agradeço a cada uma delas que esteve comigo neste importante momento.

Agradeço ao meu orientador Prof. Me. Lucas Ricken Garcia, pela paciência, dedicação e integridade quanto ao meu projeto, não medindo esforços, independentemente do horário.

Aos meus pais e amigos próximos, que entenderam quando estive ausente e me deram o aporte necessário para continuar progredindo.

A minha namorada Bruna, pela paciência e dedicação durante todo o processo de desenvolvimento, certamente sem ela não seria possível manter a vida pessoal, o emprego e ainda conseguir trabalhar neste trabalho.

Ao meu grande amigo Mateus Moran, peça chave nos primeiros passos da escolha do tema e os meus amigos do grupo do ensino médio, que me deram um grande suporte quando a linguagem de programação falhava, principalmente a meu amigo Danilo, fundamental na fase final do desenvolvimento.

Enfim, a todos que participaram dessa etapa da minha vida, fica meu agradecimento.

RESUMO

As frequentes evoluções no mercado requerem tecnologias além das ferramentas simples de produção. Uma das principais aliadas na velocidade de armazenamento, transporte e produção nas plantas industriais são as esteiras. Grande parte dos produtos desenvolvidos de forma seriada são passados por uma esteira no início, meio ou final do processo. Devido a isso, faz-se necessário um monitoramento eficaz da matéria-prima e das peças finais, para checagem de conformidade do produto confeccionado, aumentando a qualidade do produto final, potencializando a produtividade, ao mesmo tempo reduzindo os custos por perdas. Neste trabalho, foi implementado um projeto de suporte à esteira, onde se utiliza sistemas embarcados e visão computacional para realizar o monitoramento dos produtos detectados com o intuito de reduzir acidentes de trabalho e aumentar a produção com o controle de peças analisadas diariamente. Para o desenvolvimento foi utilizada uma *Raspberry Pi* como placa de desenvolvimento e aporte da biblioteca OpenCV, utilizada para o estudo e programação da visão computacional. Com o desenvolvimento deste projeto, foi possível verificar a existência de peças inconformes, medir a qualidade da produção quantificando as perdas e apresentar ao operador números que podem ser úteis para melhoria do processo. Os experimentos foram simulados em um contexto simulado e controlado, mas demonstraram que o sistema é capaz de realizar a verificação de conformidade a partir das características geométricas de determinadas peças com um erro médio de 9,17 %. Este resultado indica um funcionamento satisfatório para a primeira versão, porém, é importante destacar que o sistema necessita de melhorias, com o objetivo de reduzir o erro e viabilizar sua implementação.

Palavras-chave: monitoramento; visão computacional; sistemas embarcados; OpenCV.

ABSTRACT

The frequent evolutions in the market require technologies beyond simple production tools. One of the main allies in the speed of storage, transport and production in industrial plants are conveyors. Most of the serially developed products are passed through a conveyor belt at the beginning, middle or end of the process. Because of this, it is necessary to effectively monitor the raw material and final parts, to check the conformity of the manufactured product, increasing the quality of the final product, enhancing productivity and reducing costs per loss. In this work, a treadmill support project was implemented, which uses embedded systems and computer vision to monitor the products detected in order to reduce work accidents and increase production by controlling the parts analyzed daily. For the development, a Raspberry Pi was used as a development board and contribution of the OpenCV library, used for the study and programming of computer vision. With the development of this project, it was possible to verify the existence of non-conforming parts, measure the quality of production counting losses and present to the operator numbers that can be useful to improve the process, if applied to a real context. The experiments were carried out in a simulated and controlled context, but they demonstrated that the system is able to perform a compliance check based on the geometric characteristics of certain parts with an average error of 9.17 % in. This result indicates the satisfactory operation for the first version, however, it is important to highlight that the system needs improvement, in order to reduce the error and make its implementation viable.

Keywords: monitoring; computer vision; embedded systems; OpenCV.

LISTA DE ILUSTRAÇÕES

Figura 1 - Esteira Transportadora de Lona - Geometric 2933	13
Figura 2 - (a) Veículo em imagem escura (b) Veículo com uma imagem mais clara e nítida pronta para ser analisada	14
Figura 3 - Visão geral de um sistema embarcado	17
Figura 4 - Display LCD 20x1	18
Figura 5 – Diagrama em blocos do sistema de monitoramento para uma esteira industrial	21
Figura 6 – Concepção do protótipo de monitoramento	22
Figura 7 – Esquemático desenvolvido para os periféricos utilizados no projeto	23
Figura 8 – Raspicam conectada à Raspberry	24
Figura 9 – Etapas do processamento e tratamento da imagem capturada	25
Figura 10 – Fluxograma do funcionamento do software	27
Figura 11 – Banco de dados utilizado	28
Figura 12 – Tela de relatório	30
Figura 13 - Tela de monitoramento apresentando os dados compilados	31
Figura 14 - Alerta mostrado na página web ao encontrar um defeito na peça	32
Figura 15 - (a) Peça superior com a lâmpada estilo <i>ring light</i> (b) visão frontal do equipamento feito com peças de isopor	33
Figura 16 - Peças utilizadas como exemplo para os testes	34
Figura 17 - Projeto em funcionamento com todos os aparatos apresentados no desenvolvimento	36
Figura 18 – Tela de Relatório a partir de uma pesquisa de data	37
Figura 19 – Comparativo entre a imagem recebida e a imagem pós processamento	38
Figura 20 - Apresentação do teste de detecção com peça em 3D	39
Figura 21 - Resultado do teste de peças inconformes	40
Figura 22 - Resultado do teste final apresentado com todos os dados na tela de relatório	42

LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

AGC	<i>Apollo Guidance Computer</i>
CLT	Consolidação das Leis do Trabalho
DAELN	Departamento Acadêmico de Engenharia Eletrônica
FTP	<i>File Transfer Protocol</i>
GPIO	<i>General Purpose Input/Output</i>
HTML	<i>Hypertext Markup Language</i>
LCD	<i>Liquid Cristal Display</i>
NASA	<i>National Aeronautics and Space Administration</i>
OEE	<i>Overall equipment effectiveness</i>
PHP	<i>Hypertext Preprocessor</i>
SoC	<i>System on a chip</i>
TI	Tecnologia da Informação
UTFPR	Universidade Tecnológica Federal do Paraná
OpenCV	<i>Open-Source Computer Vision Library</i>

SUMÁRIO

1	INTRODUÇÃO	8
1.1	OBJETIVOS	10
1.2	OBJETIVOS ESPECÍFICOS	10
1.3	JUSTIFICATIVA	10
2	FUNDAMENTAÇÃO TEÓRICA	12
2.1	AUTOMAÇÃO E LINHAS DE PRODUÇÃO SERIADA	12
2.2	VISÃO COMPUTACIONAL E SUA APLICAÇÃO EM LINHAS DE PRODUÇÃO SERIADA	13
2.2.1	Visão Computacional	14
2.2.2	Aplicação da Visão Computacional em Linhas de Produção Seriada	15
2.2.3	Biblioteca OpenCV	16
2.3	DESENVOLVIMENTO DE SISTEMAS EMBARCADOS E INTERFACEAMENTO COM O USUÁRIO	16
2.3.1	Interfaceamento com o usuário	18
2.3.2	Raspberry PI® e sistema operacional Raspbian	19
3	METODOLOGIA	21
3.1	VISÃO GERAL DO SISTEMA	21
3.2	CONFIGURAÇÕES DA RASPBERRY E RASPICAM	24
3.3	DESENVOLVIMENTO DO SOFTWARE E UTILIZAÇÃO DA BIBLIOTECA OPENCV	25
3.4	BANCO DE DADOS	27
3.5	INTERFACEAMENTO COM O USUÁRIO	29
3.6	ESTRUTURA FÍSICA	32
3.7	METODOLOGIA DE TESTES	33
4	RESULTADOS OBTIDOS	36
4.1	RESULTADOS PRELIMINARES DO SISTEMA DE MONITORAMENTO	37
4.2	FUNCIONALIDADES E DESEMPENHO DO SISTEMA	40
5	CONCLUSÃO	44
	REFERÊNCIAS	45

1 INTRODUÇÃO

Ao trabalhar com fabricação de produtos, logo se imagina a produção em larga escala. Trata-se da produção em série, conceito idealizado por Adam Smith e implantado pela primeira vez por Henry Ford em sua empresa de automóveis, no início do séc. XX (MASS..., 2009). De acordo com Bastos (2015), produção neste conceito baseia-se na especialização e divisão do trabalho, pois, um colaborador que tem total domínio de determinada função pode ter melhor desempenho realizando somente aquela atividade. Esta prática é diferente dos modelos industriais antigos, em que um único funcionário realizava todas as funções na fabricação do automóvel, por exemplo, no caso em que Ford aplicou.

A produção em série passou a ser conhecida como Fordismo, em homenagem à Henry Ford e foi instalada a primeira linha de produção semiautomatizada em 1914, nos EUA (GRAMSCI, 1950). Como a quantidade de carros desenvolvidos aumentou, o preço ficou mais acessível para a população, que antes só era possível para grandes empresas e famílias ricas. O Fordismo provocou um crescimento econômico muito grande e alcançou outras áreas da produção, com ênfase aos setores siderúrgicos e têxteis (BEZERRA, 2015).

O Fordismo foi a grande influência da Segunda Revolução Industrial (CASTELLS, 1999), tendo como diferencial a eletricidade e suas formas de atuar. Com a evolução da tecnologia e demais facilidades que o mundo oferece, atualmente trabalha-se com o conceito de “Indústria 4.0”, ou Quarta Revolução Industrial (ALVAROVELHO, 2017). Esta nova revolução tem como base o uso de robôs integrados, *big data*, computação em nuvem e sistemas ciberfísicos aplicados às linhas de produção. Klaus Schwab, em seu livro *A Quarta Revolução Industrial* diz que “em sua escala, alcance e complexidade, a transformação será diferente de qualquer coisa que o ser humano tenha experimentado antes” (SCHWAB, 2016, p. 45).

Com essas considerações, é possível afirmar que o projeto desenvolvido está conectado aos processos de automatização da terceira revolução industrial, e que o tema proposto está conectado ao contexto da Indústria 4.0, uma vez que tem potencial para aderir a novas tecnologias como a Internet das Coisas (IoT, do inglês *internet of things*) que, de acordo com Kim, Ramos e Mohammed (2017) é uma tecnologia capaz de monitorar, gerenciar e controlar dispositivos remotamente e de criar novas ideias e informações a partir de dados gerados em tempo real .

Neste contexto, a eletrônica é principal protagonista do desenvolvimento de soluções que se enquadram ao conceito de indústria 4.0, e um dos principais motivos para o crescimento da indústria eletrônica é a incorporação de sistemas nos mais diversos produtos, tais como automóveis, eletrodomésticos e dispositivos de comunicação (BARROS; CAVALCANTE, 2010). Nesta área, aplicam-se diversos conceitos de sistemas embarcados, baseados em sensores e atuadores inteligentes, possibilitando a comunicação e controle de operação, incluindo Sistemas Físico-Cibernéticos e aumentando a comunicação entre máquinas e os seres humanos (MELO, 2018).

Um sistema embarcado pode ser definido como a ação de inserir uma capacidade computacional em um circuito integrado ou sistema; não executa várias tarefas tal como um computador, porém, é específico e exclusivo para determinada tarefa. Tal como um aparelho de microondas ou uma lavadora de roupas moderna (CUNHA, 2007). De maneira simples, um sistema embarcado é composto por três componentes principais: (i) *firmware*: um código escrito para elaborar os passos que o processamento realizará; (ii) unidade de processamento, microcontrolador ou microprocessador que fará o processamento do sistema; e (iii) sistema final, a união dos sistemas já com uma interface para a utilização do usuário (CHASE, 2007).

Os sistemas embarcados são grandes influenciadores da Quarta Revolução Industrial e aliados à linha de produção seriada da Segunda Revolução, levam importantes inovações tecnológicas e criação de novas tecnologias (MELO, 2018). Sistemas de visão computacional estão sendo vistos nas mais diversas aplicações industriais, tal como empresas de transportes e cargas para armazenamento e controle de volume (MARQUES *et al.*, 1999). Estes processos têm sido aplicados ainda a vários projetos da área agrícola, tal como contagem de sementes de soja (WATANABE, 2019).

Nesse sentido, o presente trabalho apresenta o desenvolvimento de um sistema de monitoramento que utiliza o conceito de visão computacional para contagem e detecção de inconformidades de peças que seriam confeccionadas em uma planta industrial fictícia que utiliza esteiras transportadoras. O intuito foi exemplificar a integração de diferentes tecnologias e conceitos necessários para desenvolver um sistema de monitoramento simples, mas útil para detectar, registrar e indicar inconformidades, podendo ser aplicado no monitoramento automático do desempenho e/ou recurso para redução de riscos aos operadores.

1.1 OBJETIVOS

O objetivo deste trabalho é o desenvolvimento de um sistema para monitoramento e contagem de produtos em uma linha de produção seriada, permitindo a detecção, registro e indicação de inconformidades no processo de produção. Para isso, objetiva-se integrar conceitos de visão computacional, sistemas embarcados, e aplicação web para monitoramento e apresentação de dados.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos do projeto são:

- Aplicar o conceito de detecção de vértices via biblioteca OpenCV para realizar a contagem de produtos em uma esteira;
- Usar o mesmo conceito para a detecção de produtos inconformes ao padrão adotado pelo projeto, indicando uma falha no processo de fabricação ou presença de objeto estranho ao especificado na planta industrial;
- Utilizar banco de dados para armazenar as informações de produção, tais como quantidade de produtos confeccionados por hora, quantidade de produtos inconformes;
- Desenvolver uma aplicação web para disponibilizar os dados e permitir o acompanhamento dos resultados de qualquer local;
- Emitir alertas sonoros e visuais ao detectar produtos inconformes, tanto na aplicação web quanto in loco.

1.3 JUSTIFICATIVA

Segurança e análise de desempenho são fatores importantes e com grande procura nas indústrias, tanto de pequeno, médio e grande porte, visto que a falta dos mesmos pode gerar prejuízos financeiros e danos aos colaboradores (IMPORTÂNCIA..., 2018).

O primeiro é fundamental, um dever da empresa de acordo com o art. 6º do Título II da CLT (Consolidação das Leis do Trabalho), imposta pelo Senado Federal, em que impõe que a segurança faz parte do direito social do trabalhador (BRASIL, 2017). Já o segundo fator, remete ao aspecto financeiro e de eficiência no processo produtivo; com uma verificação de desempenho, pode-se descobrir quais os melhores horários e situações em que se obtém melhores resultados, abrindo a possibilidade de otimização do processo industrial, além de permitir o acompanhamento do processo produtivo de forma remota.

Neste contexto, o presente trabalho busca atender estes dois fatores: mensurar possíveis lacunas no trabalho seriado, mostrando a quantidade de objetos inconformes e quantidades totais desenvolvidas durante todo o processo e o tempo necessário para o desenvolvimento de cada peça, possibilitando alterações e melhorias no sistema produtivo; além de buscar oferecer uma condição mais segura para o colaborador, com um sistema capaz de detectar possíveis acidentes antes que eles ocorram.

2 FUNDAMENTAÇÃO TEÓRICA

Para se compreender os procedimentos metodológicos utilizados na elaboração do projeto são necessários alguns embasamentos teóricos das partes que o compõem. Para isso, apresenta-se uma breve revisão bibliográfica dos temas base necessários para o desenvolvimento.

2.1 AUTOMAÇÃO E LINHAS DE PRODUÇÃO SERIADA

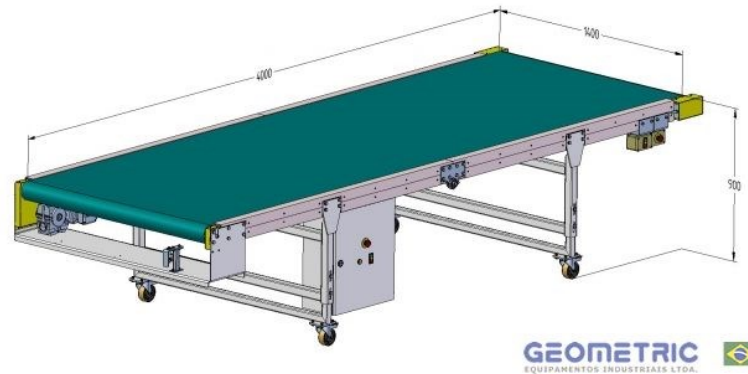
Os processos de fabricação de produtos manufaturados são amplamente utilizados na linha de produção em massa e suas definições podem ser subdivididas em: sistemas de produção e sistemas de manufatura. Um sistema de produção consiste em um conjunto de pessoas, equipamentos e procedimentos de manufatura, já um sistema de manufatura consiste na organização de facilidades de uma maneira eficiente para a realização de determinada missão (KALPAKJIAN; SCHMID, 2013).

Lopes (2011) aborda que algumas razões justificam a automação estar presente no cotidiano das indústrias e ser tão importante para a manufatura, tal como o aumento da produtividade com qualidade nos produtos, reduzir custos, melhorar segurança do trabalhador e reduzir o trabalho manual ou burocrático.

Sendo assim, estima-se que o desenvolvimento de um sistema otimizado seja capaz de proporcionar a redução de custos, aumento da produção e a redução do tempo gasto no desenvolvimento (ALBUQUERQUE; SILVA, 2011).

Uma das principais ferramentas da produção seriada é a esteira. Presente nos mais diversos segmentos e processos industriais, além de ser um dos equipamentos mais antigos da indústria, a esteira transportadora iniciou sua utilização somente para transporte de grãos e material a granel para pequenas distâncias (BORMAX, 2010). O modelo de esteira da Geometric pode ser visto na Figura 1. Este modelo é utilizado nas áreas de caldeiraria, indústria plástica, estação de trabalho e, principalmente, para transporte de cargas gerais.

Figura 1 - Esteira Transportadora de Lona - Geometric 2933



Fonte: Geometric (2018).

Atualmente, o material utilizado para produzir uma esteira, depende da sua aplicação (DONINI, et. al, 2013). Existem as mais diversas esteiras, porém, as principais partes são: rolete de carga, correia transportadora, conjunto de acionamento e conjunto de tração (DONINI, et. al, 2013). Como dito anteriormente, o Fordismo foi fundamental para a revolução industrial, e as esteiras obtiveram sua fama quando aplicada dentro da empresa Ford (MANSUR, 2012).

Uma desvantagem que a esteira apresenta é algo que se pretende tratar neste trabalho: a segurança. Há casos em que as esteiras não possuem um sistema de segurança para o controle de peças as quais transporta, sendo assim possível de acidentes com operário caindo na esteira ou o objeto causar danos à própria esteira, tal como apresentado por Dias (2019).

2.2 VISÃO COMPUTACIONAL E SUA APLICAÇÃO EM LINHAS DE PRODUÇÃO SERIADA

De acordo com Merengoni e Stringhini (2009), a fronteira entre processamento de imagens e visão computacional não é muito clara, na qual o processamento de imagens é um processo em que se inicia com uma imagem e o objeto de saída é um conjunto de valores numéricos, podendo ou não ser outra imagem. Já a visão computacional busca emular a própria visão humana, sendo assim, a entrada é igual ao processamento de imagens, porém, a saída é uma interpretação do que a entrada oferece. Neste momento é falado sobre visão computacional utilizando o processamento de imagens como ferramenta de auxílio.

2.2.1 Visão Computacional

Visão computacional é “a ciência responsável pela visão de uma máquina, pela forma como um computador enxerga o meio à sua volta, extraindo informações significativas a partir de imagens capturadas por câmeras de vídeo” (MILANO; HONORATO, 2014, p. 1). As informações captadas são apresentadas ao usuário de forma que possa trabalhar com quantidades, cores e características do que é apresentado na imagem.

Para se trabalhar com visão computacional, é necessário que a imagem esteja nítida e propícia para que os resultados sejam melhores, neste momento entra o processamento de imagens. Por exemplo, na Figura 2, à esquerda tem-se uma imagem de um automóvel em que a leitura da placa é dificultada pelas sombras. Após a imagem ser processada por filtros e equalização, é possível realizar a detecção e encontrar os dados do automóvel, função da visão computacional.

Figura 2 - (a) Veículo em imagem escura (b) Veículo com uma imagem mais clara e nítida pronta para ser analisada.



Fonte: Marengoni e Stringhini (2009, p. 126)

Desta forma, fica clara a diferença entre as áreas. O processamento de imagens faz com que uma imagem escura e sem foco passe a ter uma clareza maior para que a visão computacional capte a escrita na placa e leve ao operador para que a detecção do veículo seja possível.

De acordo com Milano e Honorato (2014), a visão computacional é aplicada nas mais diversas áreas, como física, biologia, forças armadas, entre outras. Como ilustração pode ser citado o reconhecimento facial em aeroportos para detecção de terroristas, detecção de unidades inimigas ou mísseis teleguiados, além de análises industriais tais como as aplicadas em montadoras de veículos.

2.2.2 Aplicação da Visão Computacional em Linhas de Produção Seriada

Como a visão computacional é amplamente utilizada na indústria, pode ser vista em várias etapas da produção, desde a detecção de peças iniciais, como na análise de qualidade do produto final (IMPORTÂNCIA..., 2018).

Como exemplo, uma empresa de refrigerante deve oferecer todas as garrafas com o mesmo formato, a tampa deve estar bem colocada e o líquido deve ter sempre a mesma quantidade. Este controle ajuda não somente à padronização, como pode levar a empresa a economizar, mantendo sempre a mesma quantidade.

Outra possibilidade é a de verificação na confecção de peças, para que todas as desenvolvidas sejam exatamente iguais, desde tamanho, peso ou cor. Uma peça com rebarba ou qualquer pequeno erro pode ser grave na utilização final dela, danificando componentes, maquinários e em casos extremos, podendo gerar acidentes (SILVA; WEIZENMANN; MUTZENBERG, 2019).

Rudek *et al.* (2001) apresenta sobre as aplicações de visão computacional no processo produtivo, citando algumas áreas principais para atuação, tal como orientação de deslocamento de robôs, automatização de processos de manufatura e controle de qualidade.

Em outros trabalhos que se utilizam da mesma tecnologia, se destacam o desenvolvimento de um sistema de visão computacional aplicado ao controle microbiano em processos de produção de etanol (QUINTA, 2009) e o trabalho sobre a análise de movimento de corpos deformáveis usando visão computacional (TAVARES, 2000), que se assemelha ao produzido nesse trabalho de acordo com a leitura dos dados, usando outra linguagem e outras abordagens, porém, são notadas semelhanças a partir das técnicas de contornos e matrizes de proximidades em objetos 2D e 3D.

2.2.3 Biblioteca OpenCV

Para se trabalhar com visão computacional são necessárias algumas ferramentas de trabalho e, para isso, uma das ferramentas disponíveis é a biblioteca OpenCV (*Open Source Computer Vision Library*, do inglês “Biblioteca Código Aberto de Visão Computacional”) que, segundo Coelho e Delai (2012, p. 1), trata-se de “um conjunto de ferramentas de programação para desenvolvimento de aplicações com visão computacional”.

De acordo com Marengoni e Stringhini (2009), a biblioteca possui mais de 500 funções e foi idealizada para que usuários e programadores tenham uma facilidade de acesso à visão computacional e tenha comunicação em tempo real. Dessa forma, a biblioteca OpenCV é ideal para o projeto desenvolvido e foi amplamente utilizado em todas as funções de detecção.

Ainda de acordo com Merengoni e Stringhini (2009), a biblioteca é aplicada tanto em visão computacional como em processamento de imagens, e é dividida em cinco grupos de funções:

- Processamento de imagens;
- Análise estrutural;
- Análise de movimento e rastreamento de objetos;
- Reconhecimento de padrões;
- Calibração de câmera e reconstrução 3D.

Esta ferramenta possui mais de 500 algoritmos das mais diversas áreas aplicadas a visão computacional e processamento de imagens, e a partir dela é possível realizar as detecções de bordas, vértices, filtros, calibração e segmentações (BRADSKI, 2000).

2.3 DESENVOLVIMENTO DE SISTEMAS EMBARCADOS E INTERFACEAMENTO COM O USUÁRIO

Como pode ser visto nas aplicações de visão computacional, é necessário implementar o sistema por meio de um sistema computacional dedicado a esta

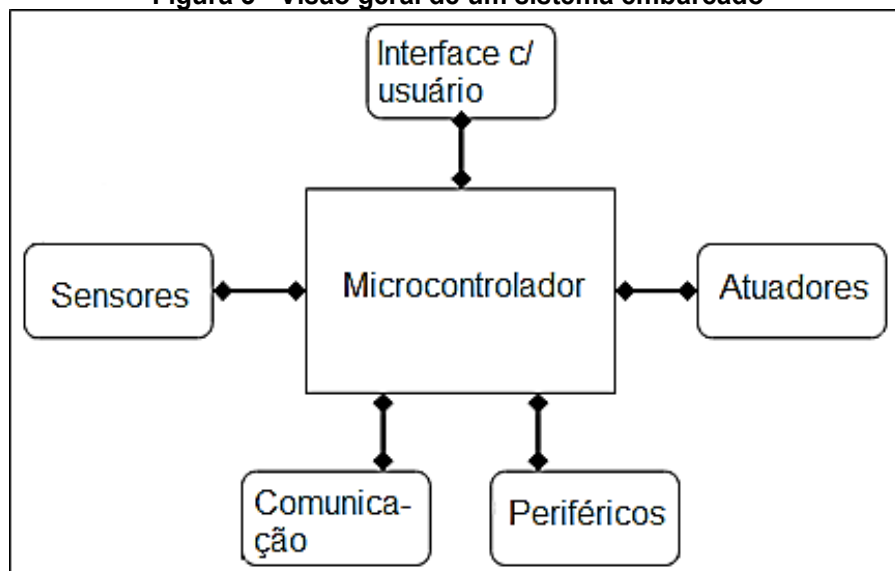
finalidade, com os devidos recursos e periféricos. Neste instante, é importante apresentar os principais conceitos de sistemas embarcados.

Um conceito inicial acerca de sistemas embarcados é a de que são sistemas microprocessados e o computador é totalmente dedicado ao dispositivo ou sistema utilizado. Avaliando o aspecto histórico, os computadores das décadas de 30 e 40 ainda possuíam funções específicas para o seu funcionamento, já nas décadas de 60 e 70 fora criado o *Apollo Guidance Computer (AGC)*, sendo assim considerado o primeiro sistema embarcado moderno (CHASE, 2007)

Nas décadas de 70 e 80, houve o surgimento e expansão dos primeiros microcontroladores e microprocessadores, com isso, houve uma grande arrancada nos sistemas embarcados. Entre os anos 80 e 90, a Intel e a AMD ofereceram algumas dezenas de variantes de microprocessadores, com diversas combinações de periféricos integrados (STADZISZ; RENAUX, 2016).

Um sistema embarcado tem como seu “cérebro” o microcontrolador, que controla as demais classes: sensores, atuadores, periféricos, comunicação e interfaceamento (GARCIA, 2018). Na Figura 4 pode-se ter uma visão geral sobre os sistemas embarcados com microcontrolador. Em que as ramificações são os periféricos, tal como memórias, interfaceamento, sistema de alimentação, sensores e atuadores.

Figura 3 - Visão geral de um sistema embarcado



Fonte: Adaptado de Garcia (2018, p. 1).

Criados na década de 50, os sensores tornaram-se peças fundamentais para a automação ao longo dos anos. São dispositivos capazes de detectar qualquer

movimentação em um determinado espaço, seja ele para contagem, direção ou monitoramento (TIPOS..., 2017).

De acordo com Sousa e Lopes (2011, p. 43), “os modernos sensores são dispositivos de baixo custo, inteligentes e equipados com múltiplos elementos de sensoriamento embarcados”. Portanto, é evidente que a utilização de sensores é fundamental para uma boa elaboração do projeto.

2.3.1 Interfaceamento com o usuário

Há várias possibilidades de comunicação para a realização do interfaceamento com o usuário, tal como uma apresentação local, com um display para consulta de dados, LEDs para informações binárias e sinais sonoros para alertas; também é possível uma apresentação remota com uma página web, em que o usuário pode acessar de qualquer lugar.

Dentre estas possibilidades, talvez o uso de um display LCD (*Liquid cristal display*, do inglês, visor de cristal líquido) seja uma das soluções mais simples. Um display de cristal líquido - LCD é uma poderosa ferramenta dos meios eletrônicos; é um dispositivo de interfaceamento e/ou comunicação visual utilizado na maior parte dos equipamentos eletrônicos (PUHLMANN, 2015). Os modelos mais utilizados são os 16x2, 20x1 e 40x4. Esses valores remetem a quantidade de dígitos e a quantidade de linhas. Por exemplo: Um módulo 20x1 tem 20 dígitos e predispõe de apenas 1 linha, como pode ser visto na Figura 5. O controle destes displays é realizado a partir da programação do microcontrolador. Há funções específicas para qualquer que seja a disposição dos valores.

Figura 4 - Display LCD 20x1



Fonte: PUHLMANN, 2015.

Neste projeto, foram utilizados como ferramentas de interfaceamento o display LCD e o *buzzer* como ferramentas de acompanhamento à interface principal,

que é a apresentação web, desenvolvido em PHP com ligação ao banco de dados utilizado e sendo acessado pelo microcontrolador escolhido: a Raspberry. Em outras aplicações, é possível a utilização de demais formas de interfaceamento, tal como displays maiores, aplicativo *mobile* além de demais saídas de áudio.

2.3.2 Raspberry PI® e sistema operacional Raspbian

Dentre os mais diversos microcontroladores e microprocessadores, destacam-se na área de desenvolvimento de projetos mais simples e de fácil acesso dois principais: Arduino® e Raspberry PI®. Ambos são muito abrangentes, se tratando de aplicações, porém, há particularidades entre eles que faz com que os usuários escolham sua preferência (SUEIRO, 2014).

Euler (2015) diz que ambas as placas de desenvolvimento são importantes e que aceitam a ligação com componentes externos a partir de suas GPIOs (General Purpose Input/Output) – Portas programáveis de entrada e saída. Arduino® é melhor para tarefas repetitivas, tal como abrir e fechar portas, leitura de sensores, etc. Raspberry PI® é melhor para tarefas mais complexas, tais como controlar um robô, realizar múltiplas tarefas e cálculos intensos.

De acordo com Almeida (2013), a Raspberry PI® é uma placa de desenvolvimento baseado em SoC (System on a Chip, do inglês, sistema em um chip), ou seja, um sistema integrado em um único dispositivo. A ideia da Raspberry PI® surgiu em 2006 na Universidade de Cambridge (UK) com a preocupação do ensino de TI (tecnologia da informação) para crianças. Em 2008, houve uma busca por redução dos custos e melhoria do processamento para utilização em dispositivos móveis. Em 2011, foi lançado o modelo Raspberry PI Modelo B.

O mínimo de periféricos para utilização da placa de desenvolvimento são: fonte de 5 V e ao menos 700 mA, cartão SD de ao menos 4 GB (preferencialmente 8 GB), cabo de conexão HDMI para com um monitor ou televisor, cabo de rede com conector RJ-46, teclado e mouse.

Para a utilização da Raspberry PI®, dentro do quadro de “Microcontrolador” da Figura 4, necessita-se de um sistema operacional, portanto, o mais acessível e simples de se trabalhar é o chamado Raspbian. De acordo com o site do fabricante

(RASPBerry, 2018), consiste em uma união entre Rasp – Raspberry e Bian – Debian.

O Debian, é um sistema operacional inteiramente composto por software livre (DEBIAN, 2018). O Raspbian segue do mesmo princípio: software livre. Porém, otimizado diretamente para uso nas placas Raspberry PI®. Diferente do que se imagina, o Raspbian não tem afiliação com a Fundação Raspberry PI, é composto por uma pequena equipe de desenvolvedores fãs do hardware disponível pela fundação.

3 METODOLOGIA

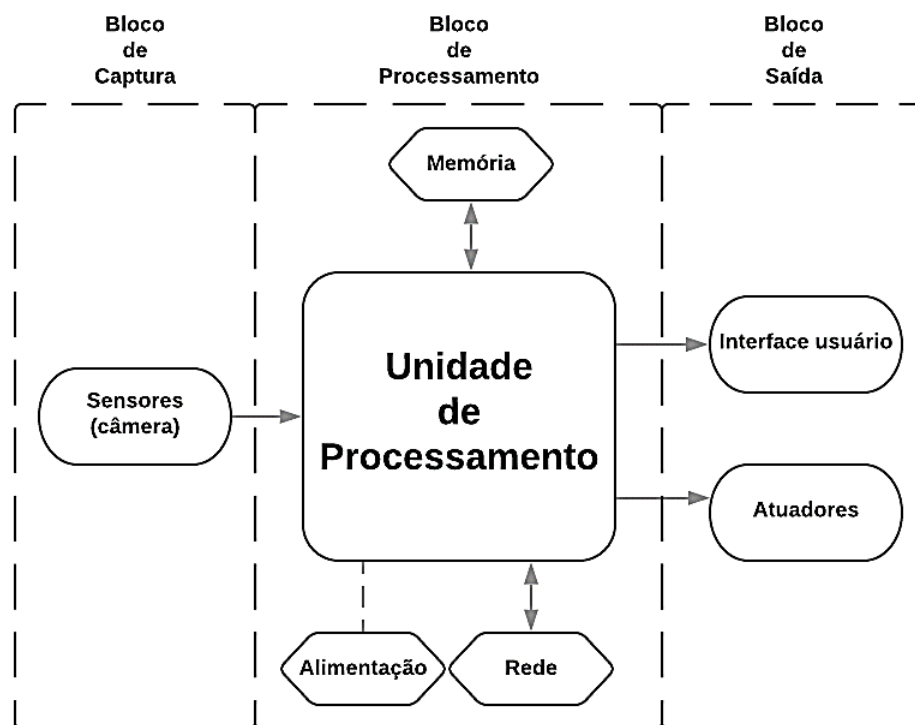
Neste capítulo são apresentadas as etapas utilizadas para confecção e desenvolvimento do projeto, tal como os testes realizados para averiguar sua eficácia.

O presente projeto foi desenvolvido em diversas etapas, baseadas em: pesquisa, configuração da placa de desenvolvimento, instalação de pacotes, recebimento das informações do dispositivo de entrada, programação do processamento das imagens obtidas, análise da visão computacional, além da configuração dos dispositivos de saída e criação da página web.

3.1 VISÃO GERAL DO SISTEMA

A partir da concepção do projeto comentado nas etapas anteriores, o sistema de monitoramento foi projetado seguindo o diagrama em blocos apresentado na Figura 5.

Figura 5 – Diagrama em blocos do sistema de monitoramento para uma esteira industrial.

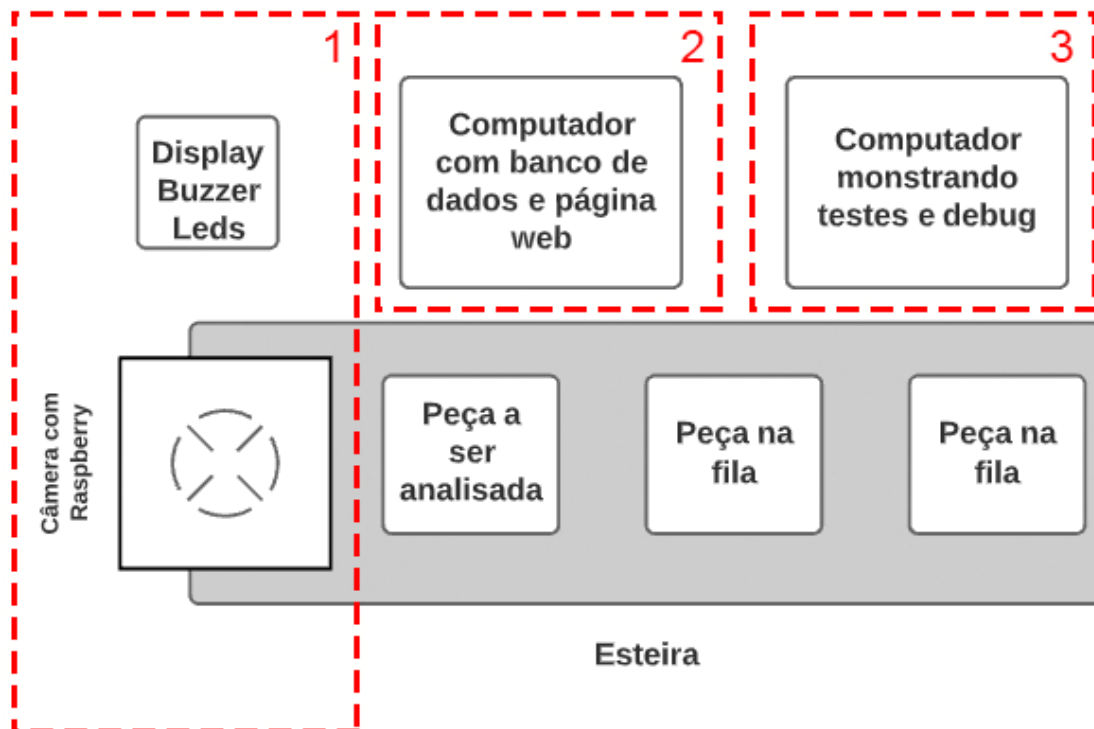


Fonte: Autoria Própria (2021)

O bloco de captura foi composto por uma *raspicam*, que foi conectada diretamente na unidade de processamento. O bloco de processamento é o “cérebro” do projeto, este foi composto por uma Raspberry PI® 3 – modelo B+. Nesse bloco são realizados todos os cálculos, contagem de peças obtidas, processos e ativa o bloco de saída. O último bloco é o de dispositivos de saída, composto pelo display 16x2, que mostra continuamente a quantidade de peças obtidas no processo, um *buzzer* que é ativado caso um objeto inconforme é identificado, além de leds para indicação do funcionamento do sistema. Todas essas informações também são apresentadas em uma página web de atualização constante dos dados, permitindo um monitoramento de qualquer local, dependendo apenas de conexão com a internet.

Na Figura 6 pode-se observar a visão geral do projeto, com as conexões e monitores de testes.

Figura 6 – Concepção do protótipo de monitoramento



Fonte: Autoria própria (2021)

As subdivisões apresentadas na Figura 6 são as seguintes:

1- Protótipo para iluminação da esteira utilizando uma lâmpada iluminadora “*ring light*”. Foi uma sugestão de apresentação utilizada devido à *Raspicam* ter uma definição em que há a necessidade de retirar todas as sombras para reduzir a

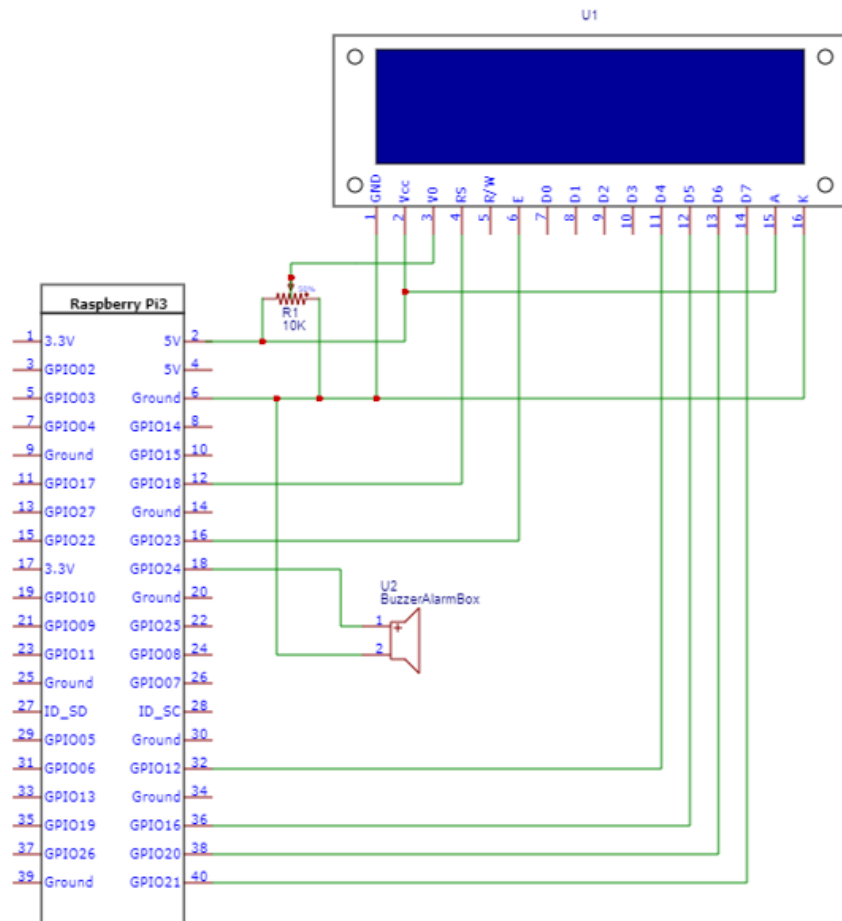
interferência. Acoplada à câmera, está a placa de desenvolvimento Raspberry PI conectado à *proto board* e os periféricos: *display* LCD e *buzzer*.

2- Notebook conectado à página web, ligada à rede local pelo WiFi, desta forma, o operador conseguirá acompanhar em tempo real o que se passa na esteira e os dados obtidos a partir dela. A página pode ser visualizada na Figura 11.

3- Monitor conectado à placa de desenvolvimento para acompanhar o *debug* do sistema. Esta etapa só é utilizada no período de testes do sistema. O funcionamento pode ser realizado sem a visualização em tempo real utilizando somente a página web.

O esquemático dos periféricos do sistema de monitoramento para aplicação em esteira transportadora pode ser visto na Figura 7. Na imagem pode ser observado que o *display* 16x2 foi nomeado como “U1”, o potenciômetro foi chamado de “R1” e o *buzzer* foi chamado de “U2”.

Figura 7 – Esquemático desenvolvido para os periféricos utilizados no projeto



Fonte: Autoria Própria (2021).

3.2 CONFIGURAÇÕES DA RASPBERRY E RASPICAM.

A primeira etapa do projeto consistiu na configuração geral para a utilização da Raspberry PI®, visto que é necessária a instalação do Sistema Operacional e de pacotes adicionais. O sistema escolhido foi o Raspbian, uma variante do sistema Debian® desenvolvida exclusivamente para utilização em Raspberry PI®. A escolha deste sistema foi feita pela praticidade e segurança fornecida pelo fabricante. Os pacotes adicionais foram instalados conforme a necessidade, tal como o banco de dados MySQL e MariaDB.

Para iniciar os procedimentos dentro da placa de captura, foi preciso o desenvolvimento da detecção de objetos e, para isso, utilizou-se o Módulo Raspberry Pi Câmera, mais conhecido como Raspicam, uma câmera desenvolvida para pesquisas e projetos utilizando a Raspberry. Esta câmera possui uma resolução para imagens de até 5MP (2592x1994) e uma resolução para vídeo de até 1080p.

O dispositivo foi conectado por um cabo flat e tem sua compatibilidade com a placa de desenvolvimento como principal diferencial. Na Figura 8 é apresentada a câmera utilizada conectada à placa:

Figura 8 – Raspicam conectada à Raspberry



Fonte: Autoria Própria (2021)

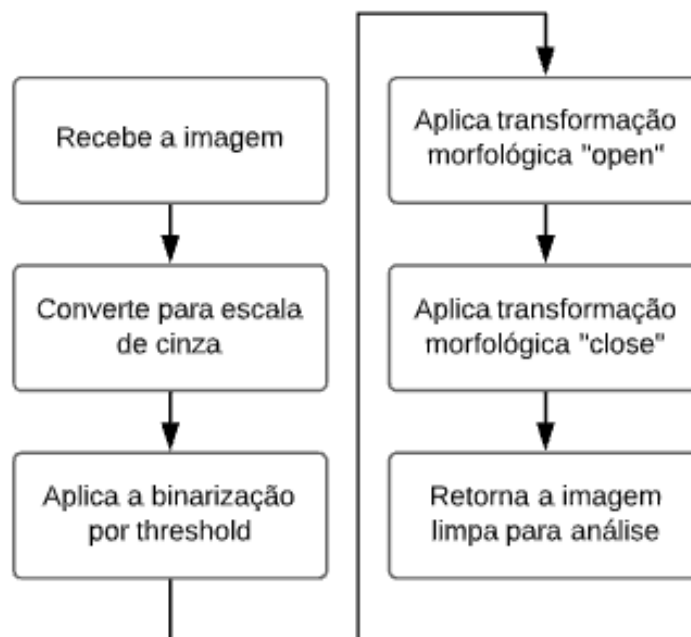
3.3 DESENVOLVIMENTO DO SOFTWARE E UTILIZAÇÃO DA BIBLIOTECA OPENCV

Para chegar ao projeto final de detecção de objetos semelhantes a um padrão, é necessário passar por alguns processos antes. O processo de detecção de figuras é realizado com a linguagem *python* e a biblioteca *OpenCV*.

Com base nos estudos de Rosebrock (2016) e Passarelli (2017), e dos exemplos de software utilizando o OpenCV disponibilizados por Rosebrock, foi desenvolvido um procedimento para detecção de objetos e pessoas em imagens estáticas primeiramente.

Antes de iniciar o processo de detecção, é necessário tratar a imagem recebida pela *raspicam*, visto que a mesma possui ruídos e também para um melhor resultado das funções pré-estabelecidas no OpenCV. Esse processo é realizado de forma contínua passando pelas etapas descritas na Figura 9.

Figura 9 – Etapas do processamento e tratamento da imagem capturada



Fonte: Autoria própria (2021)

Após a captura da imagem, a primeira etapa consiste na conversão da imagem em escala de cinza e binarizada por *threshold* para segmentar a imagem, a dividindo em objeto e fundo. O objeto identificado nesta etapa é passado para a função

de “*cleanup*”, em que são aplicadas as transformações morfológicas capazes de efetuar uma limpeza na imagem para retirar possíveis ruídos tanto fora do objeto (transformação *open*), quanto dentro do objeto (transformação *close*). Após a limiarização e limpeza, a imagem está pronta para ser analisada.

Ao receber a imagem binarizada e limpa de ruídos, é passado para o processo de inspeção, em que são detectados os contornos obtidos com a função *cv2.findContours* e analisada a sua área com a função *cv2.contourArea*. Para evitar possíveis ruídos restantes, nesta etapa são considerados somente contornos com área superiores à 2000 px, valor mensurado a partir dos testes em que se captam possíveis ruídos. Ao passar dessa condição, inicia-se o processo de detecção.

No processo de detecção, é calculado o perímetro do contorno apresentado na inspeção, para isso utiliza-se a função *cv2.arcLength*, e a partir do mesmo, foi utilizado o algoritmo conhecido como Ramer-Douglas-Peucker, que é a base da função *cv2.approxPolyDP* disponível na biblioteca OpenCV, e que se utiliza da ideologia de divisão e fusão (ROSEBROCK, 2016). É detectado a partir da aproximação de contorno, na qual se baseia na suposição de que uma curva pode ser aproximada por uma série de segmentos de linha curta, “isso leva a uma curva aproximada resultante que consiste em um subconjunto de pontos que foram definidos pela curva original” (ROSEBROCK, 2016, p. 1, tradução nossa). Isto é, a partir do centro de massa encontrado, é possível identificar os vértices de acordo com a curva aproximada da divisão entre as cores.

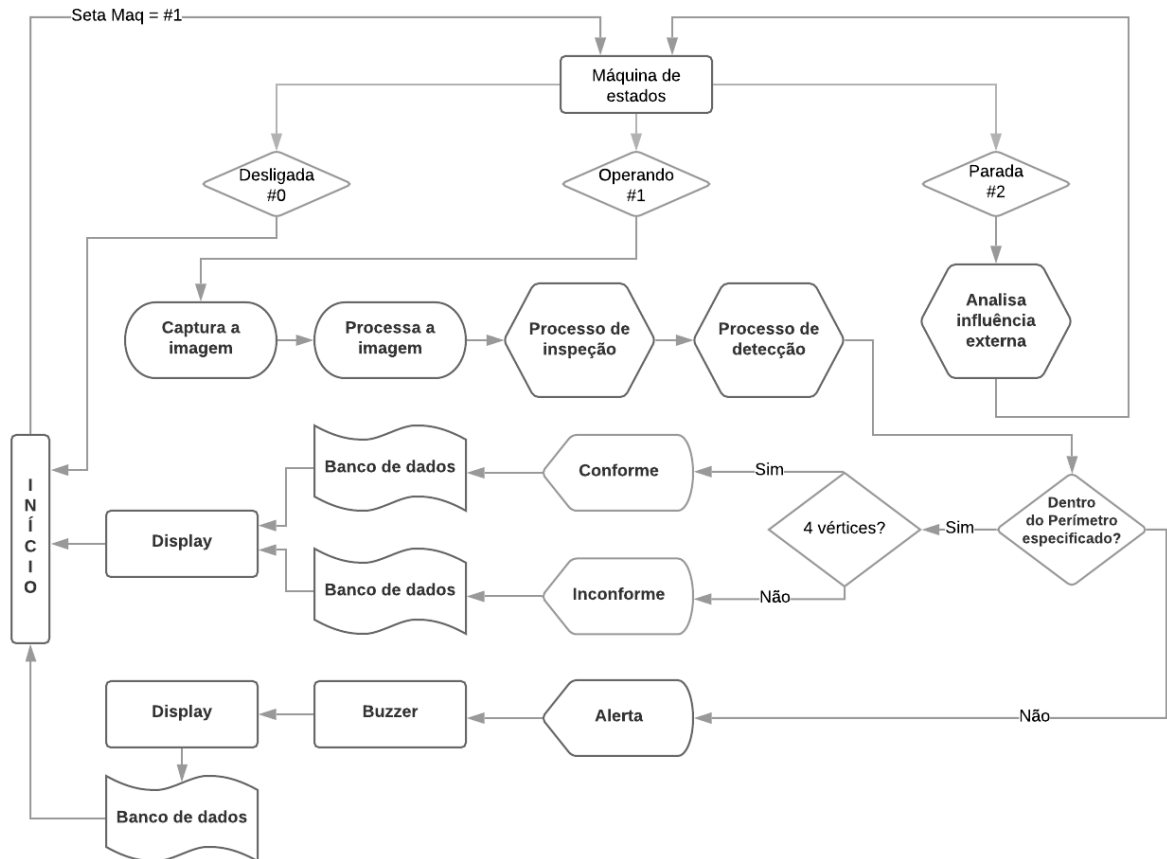
A partir da análise de perímetro e da contagem de vértices, há três situações:

- a. Se o perímetro está dentro dos parâmetros estabelecidos e possui 4 vértices, é considerado uma peça conforme;
- b. Se o perímetro está dentro dos parâmetros estabelecidos e não possui 4 vértices, é considerado uma peça inconforme;
- c. Se o perímetro não está dentro dos parâmetros estabelecidos, é considerado um alerta de perigo, tal como um trabalhador ter caído na máquina ou uma peça que pode danificar o equipamento.

A partir do estudo em imagens estáticas, é incluída a detecção a partir da Raspicam utilizada. Parte do mesmo princípio de detectar contornos e, a partir disso, verificar qual a figura geométrica mais se assemelha.

Todo o processo é desenvolvido a partir do fluxograma apresentado na Figura 11, que apresenta desde a captação da imagem a partir da Raspicam, até os dispositivos de saída.

Figura 10 – Fluxograma do funcionamento do software



Fonte: Autoria própria (2021)

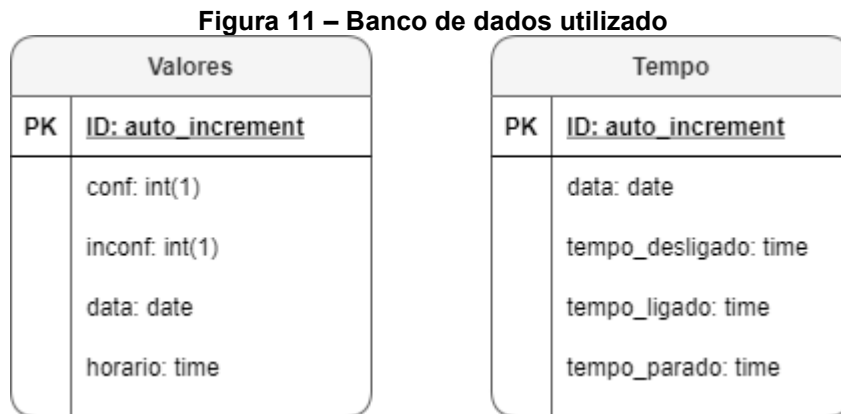
3.4 BANCO DE DADOS

Finalizando a detecção e classificação das formas, chega-se à parte de registro dos resultados de monitoramento e, para isso, utilizou-se um banco de dados. Como explicado por Campos (2008), um banco de dados é uma coleção organizada de dados. Os dados são fatos que podem ser gravados e que possuem um significado implícito, por exemplo números de telefone, nomes, número de documentos. Na maior parte das aplicações dos sistemas embarcados é necessária a utilização de um banco de dados.

O sistema de gerenciamento de banco de dados escolhido para a elaboração deste projeto foi o MariaDB (MARIADB, 2018) pela praticidade de utilização.

Portanto, fazendo uso do MariaDB como ferramenta de criação do banco de dados, as informações captadas no processo anterior são expostas na tabela a ser acessada pela página web. Este processo é fundamental para o projeto, pois, a partir dele é possível o acionamento de atuadores e a página *web* consegue manter seus dados atualizados em tempo real.

Os resultados obtidos são armazenados em duas tabelas que são vistas na Figura 11, e são alterados em tempo real à detecção, ou seja, a informação chega mais rápido ao usuário.



Fonte: Autoria própria (2021)

Na tabela “Valores” estão subdivididos da seguinte forma:

- ID: identificação e contagem de objetos;
- conf: se a peça estiver de acordo com a detecção, marca-se com o número “1”, se não, “0”;
- inconf: inverso ao anterior, se a peça estiver de acordo com a detecção, marca-se com o número “0”, se não, “1”;
- data: dia, mês e ano da detecção;
- horario: hora, minutos e segundos da detecção

Já na tabela “Tempo” as subdivisões são as seguintes:

- ID: identificação e contagem de linhas;
- tempo_desligado: hora exata em que o sistema é desligado;
- tempo_ligado: hora exata em que o sistema é ligado;

- tempo_desligado: hora exata em que o sistema parado devido a algum alerta emitido.

O banco de dados é alimentado diretamente pela *Raspberry* via *WiFi* e função *mysql.connector*. O banco é acessado pela página *web* para apresentação dos dados para o usuário. A coluna “ID” é feita com incrementação automática, as colunas “conf” e “inconf” são números inteiros e as colunas “data” e “horario” estão no formato de datas, para melhor tratamento dos dados.

3.5 INTERFACEAMENTO COM O USUÁRIO

Para que o usuário tenha uma melhor visualização do que está sendo processado e para que as informações sejam mais claras e objetivas, são necessárias algumas aplicações de saída. Para isso foram utilizados: *display* LCD, *buzzer* e página *web*.

O *buzzer* consiste em um componente semelhante a uma sirene, em que alerta de forma sonora quando uma peça detectada está fora das especificações. Para possibilitar um monitoramento de *status* da máquina *in loco* foi implementado um *display* LCD que indica o número de peças conformes e inconformes, bem como pode ser útil para indicação de mensagens de erro ou alerta no sistema.

Como sistema de saída principal foi desenvolvida uma interface *web* para apresentação dos dados em tempo real a partir do desenvolvimento do banco de dados. A opção pela página é a de que possa ser acessada de qualquer lugar, além de que, pode ser adaptada à um aplicativo, por exemplo, em futuras modificações do projeto.

A linguagem escolhida foi o PHP com HTML, devido a facilidade de comunicação com o banco de dados e a familiaridade do autor com a linguagem.

Para se utilizar o interfaceamento PHP, é necessária a utilização de algumas ferramentas, tal como o banco de dados. Para facilitar no processo de instalação, existem alguns pacotes que fazem a instalação de todas as ferramentas, tal como o XAMPP.

Foram criadas duas páginas para a apresentação dos dados: Monitoramento e Relatório. A tela de relatório é uma apresentação do monitoramento em um dia

específico e nela são mostradas todas as detecções do banco de dados de forma mais simples e prática, apresentando ao usuário uma análise peça a peça, seja ela conforme ou inconforme. Na Figura 12 é possível observar a Tela de Relatório aguardando a inserção de uma data pelo usuário.

Figura 12 – Tela de relatório

UTFPR - Universidade Tecnológica Federal do Paraná - Campus Campo Mourão

PROJETO DE UM SISTEMA EMBARCADO PARA MONITORAMENTO DE UMA ESTEIRA INDUSTRIAL UTILIZANDO VISÃO COMPUTACIONAL

Acadêmico: Pedro Bertelli
Orientador: Prof. Ms. Lucas Ricken Garcia

ESCOLHA UM DIA PARA GERAR O RELATÓRIO

Data:

[MONITORAMENTO | TELA DE RELATÓRIO](#)

Fonte: Autoria própria (2021)

A página de monitoramento trata-se de uma página simples, como pode-se ver na Figura 13, tendo como função somente a apresentação dos dados conforme a detecção é realizada, além de servir como unidade de controle para quando o sistema é parado devido a detecção de objetos indesejados.

As informações apresentadas nesta tela são insumos para uma posterior análise mais profunda, porém, para efeito de ilustração, a tela apresenta algumas informações de possíveis recursos que podem ser úteis para avaliação de um sistema seriado de produção, tais como: (i) a quantidade total de peças no período de monitoramento da máquina, (ii) quantidade de produtos conformes e inconformes, (iii) a qualidade obtida a partir das detecções que pode ser definida como o número de produtos conformes dividido pelo total de produtos analisados, (iv) a utilização deste fator de qualidade para definição de um indicador de eficiência OEE (*Overall Equipment Effectiveness*, do inglês, eficiência geral do equipamento) e, por fim, (v) o tempo de funcionamento da máquina naquele período de monitoramento.

Vale salientar que este indicador *OEE* é uma ferramenta importante no contexto produtivo, uma vez que o mesmo apresenta um valor medido em percentual

que fornece uma medida para o acompanhamento da produtividade. O indicador tem seu cálculo realizado a partir do produto das taxas de disponibilidade do equipamento, desempenho (ou performance) e qualidade (ZATTAR, RUDEK, TURQUINO, 2010).

O valor de qualidade foi medido no terceiro quadrado de apresentação da tela de monitoramento; a disponibilidade do equipamento é medida a partir do percentual do tempo ligado sobre o tempo total de operação; e o desempenho é medido a partir do percentual do período produzindo sobre o tempo ligado.

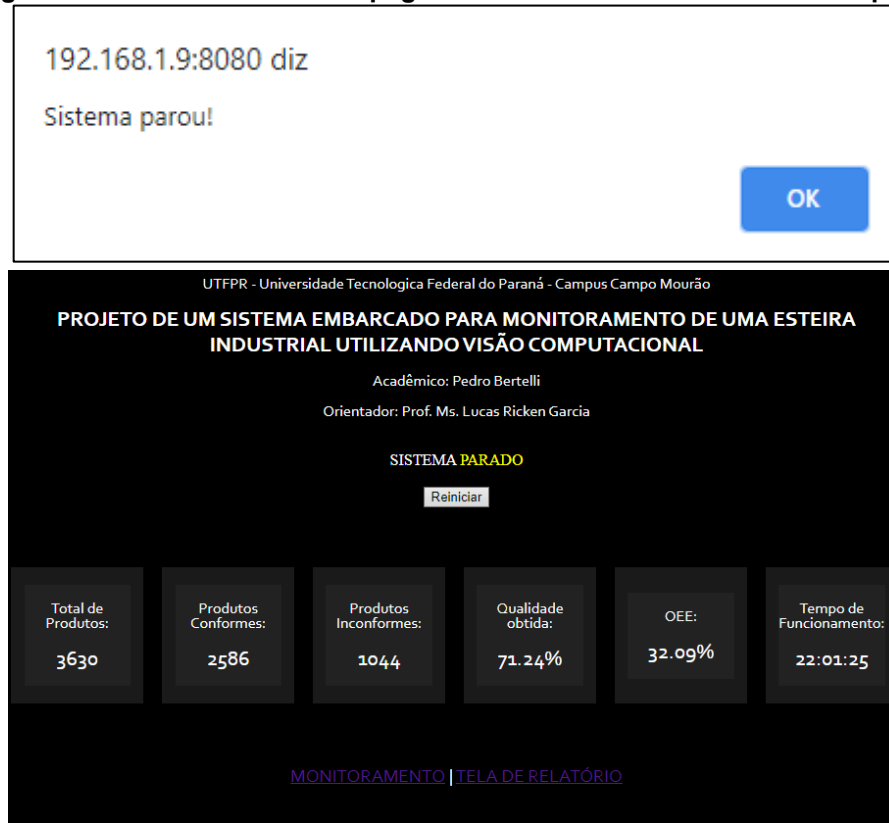
Figura 13 - Tela de monitoramento apresentando os dados compilados



Fonte: Autoria própria (2021)

Quando há algum alerta recebido, seja ele por uma peça com perímetro grande demais ou pequeno demais, a página web emite um alerta na tela, avisando ao operador e atualizando o *status* do banco de dados para que o sistema pare. O sistema ficará parado até que haja uma interferência externa do operador, reiniciando o sistema a partir do botão “reiniciar” da tela de parada. O alerta e o botão são apresentados na Figura 14.

Figura 14 - Alerta mostrado na página web ao encontrar um defeito na peça



Fonte: Autoria própria (2021)

3.6 ESTRUTURA FÍSICA

Para uma melhor detecção dos objetos, é necessária uma boa iluminação e um local apropriado para eliminar ou reduzir ruídos e sombras. Dado isso, a estrutura física do projeto é essencial. Para isso utiliza-se os seguintes materiais:

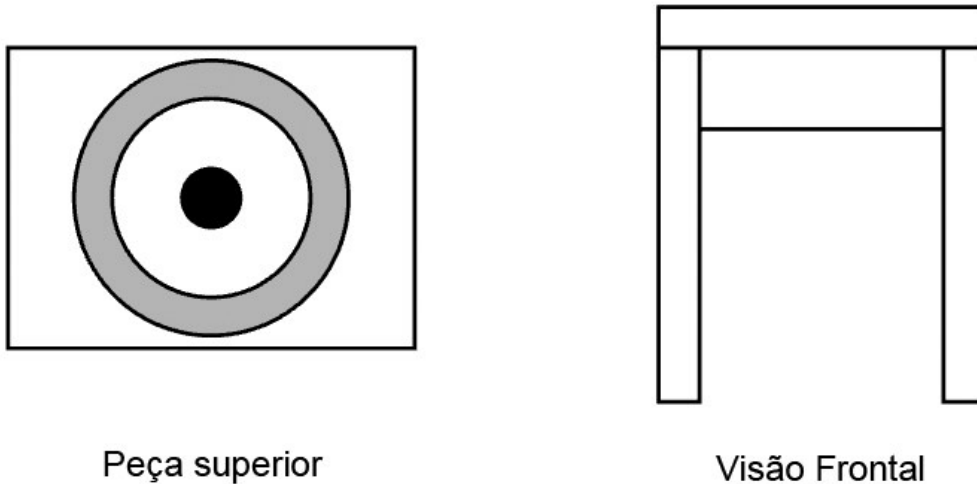
- 2 placas de isopor tamanho 25x40 cm para as laterais;
- 1 placa de isopor tamanho 25x35 cm para o topo;
- 2 placas de isopor tamanho 7x30 cm para sustentação;
- 1 lâmpada estilo *ring light* para iluminação.

Foram feitos furos na placa do topo para anexar a Raspicam e a lâmpada. Após isso, a lâmpada foi posicionada no topo desta estrutura suportada pelas placas laterais. Foi utilizado hastes de sustentação para que não houvesse movimento, atrapalhando a detecção.

Na Figura 15 (a) é possível observar a peça superior com a lâmpada acoplada e com um furo no meio para inserção da Raspicam, e na Figura 15 (b) é possível ver

como é o projeto da estrutura, em que no meio das placas laterais irão passar as peças a serem detectadas.

Figura 15 - (a) Peça superior com a lâmpada estilo *ring light* (b) visão frontal do equipamento feito com peças de isopor



Fonte: Autoria própria (2021)

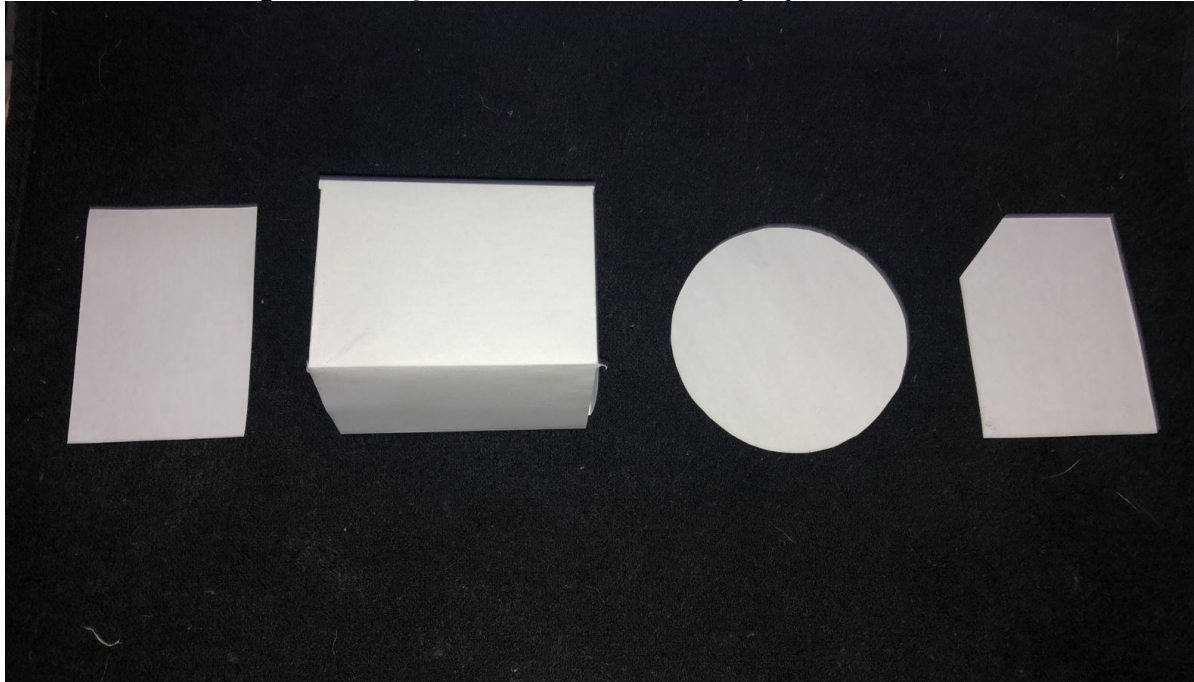
3.7 METODOLOGIA DE TESTES

Para a realização dos testes foram desenvolvidos três conjuntos de peças:

- as consideradas *em conformidade* com as especificações programadas. Estas peças devem possuir um perímetro entre 300 e 1000 px, formado por quatro vértices. Dentro deste grupo de peças foram confeccionadas peças em 2D e 3D para observar a influência da sombra sobre o processo de detecção;
- as consideradas *não conforme*, que possuem especificações similares, porém, apresentam indícios de avarias. Estas peças são classificadas desta forma quando o perímetro está entre 300 e 1000px e foram detectados menos ou mais que 4 vértices;
- e, por fim, peças classificadas como *potencialmente perigosas*. Neste caso, estas são classificadas desta forma quando possuem um perímetro superior a 1000px ou inferior a 300px, indicando a entrada de um objeto estranho ao processo produtivo.

Todas essas peças (Figura 16) foram desenvolvidas com papel A4, que foram colocadas em uma superfície escura e móvel, simulando o andamento da esteira. Infelizmente, para os testes do projeto não foi possível implementar uma esteira devido à falta de equipamentos e tempo hábil para o desenvolvimento, ficando assim, uma dificuldade de padronizar velocidade e frequência das peças analisadas.

Figura 16 - Peças utilizadas como exemplo para os testes



Fonte: Autoria própria (2021)

Em uma primeira etapa, com o intuito de verificar o desempenho do sistema e a sensibilidade em detectar diferentes tipos de formas, observou-se a resposta do sistema para cada tipo de peça individualmente alterando tamanho e características, no caso das peças inconformes, alterando o tamanho e quantidade de seções (e, conseqüentemente, número de vértices).

Nestes testes preliminares foram realizadas 20 detecções em sequência para cada tipo de peça. Os testes foram realizados sempre da mesma maneira: uma fila de peças foi posicionada em sequência no aparato e movimentando o fundo escuro simulando o comportamento de uma esteira. Cada detecção realizada foi observada tanto manualmente (observando as telas auxiliares no Raspberry), quanto pela tela de monitoramento desenvolvida.

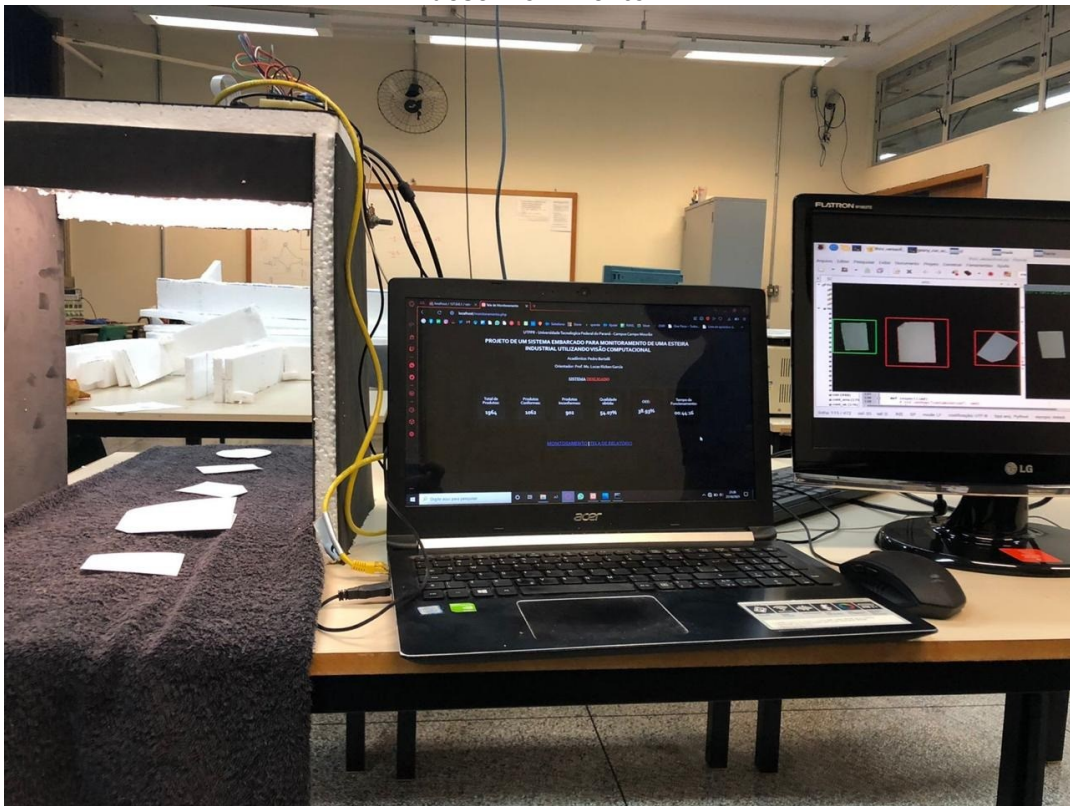
Em seguida, foram realizados testes mais abrangentes, considerando a entrada de peças diversas, 30 consideradas inconformes, 30 consideradas conformes

e 2 exemplos de peças que podem indicar um objeto estranho à planta e que, portanto, deve exigir uma parada do sistema para conferência. Desta forma, o intuito deste teste foi verificar tanto as funcionalidades do sistema, registrando corretamente ou não as diferentes situações que podem ocorrer em uma produção seriada via esteira industrial, bem como avaliar o desempenho e assertividade do sistema de monitoramento desenvolvido.

4 RESULTADOS OBTIDOS

Neste capítulo são apresentados os principais resultados obtidos a partir da implementação descrita no Capítulo 3. O ambiente para testes desenvolvido, incluindo aparato de monitoramento, iluminação e suporte para posicionamento da câmera, ambiente de monitoramento da Raspberry e computador para execução da página *web* são apresentados na Figura 17.

Figura 17 - Projeto em funcionamento com todos os aparatos apresentados no desenvolvimento

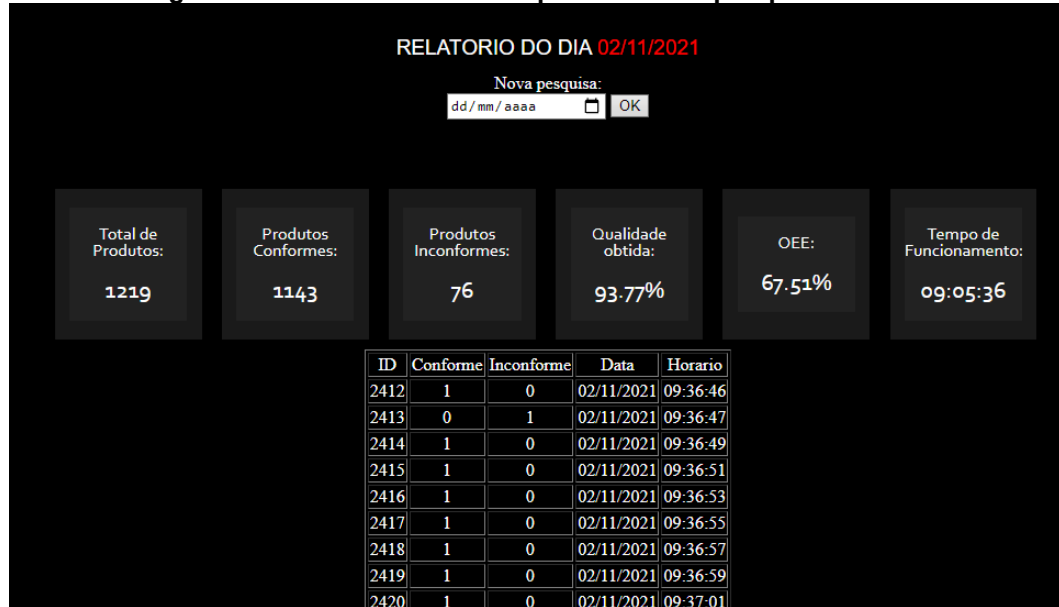


Fonte: A autoria própria (2021)

A página *web* desenvolvida é mostrada nas Figuras 12 e 13, sendo a primeira a Tela de Relatório, em que é possível observar os dados obtidos e apresentados em formato de tabela no dia escolhido. Na segunda a Tela de Monitoramento, em que é atualizada em tempo real como uma visualização do funcionamento do sistema e demais dados apresentados. Quando se pesquisa uma data na tela de relatório, ela apresenta todos os dados apresentados na tela de monitoramento filtradas pela data pesquisada seguido por todo o histórico, como visto na Figura 18. Esses dados podem

ser utilizados para comparação dia a dia ou descobrir possíveis falhas no sistema produtivo ao longo de um período de interesse.

Figura 18 – Tela de Relatório a partir de uma pesquisa de data

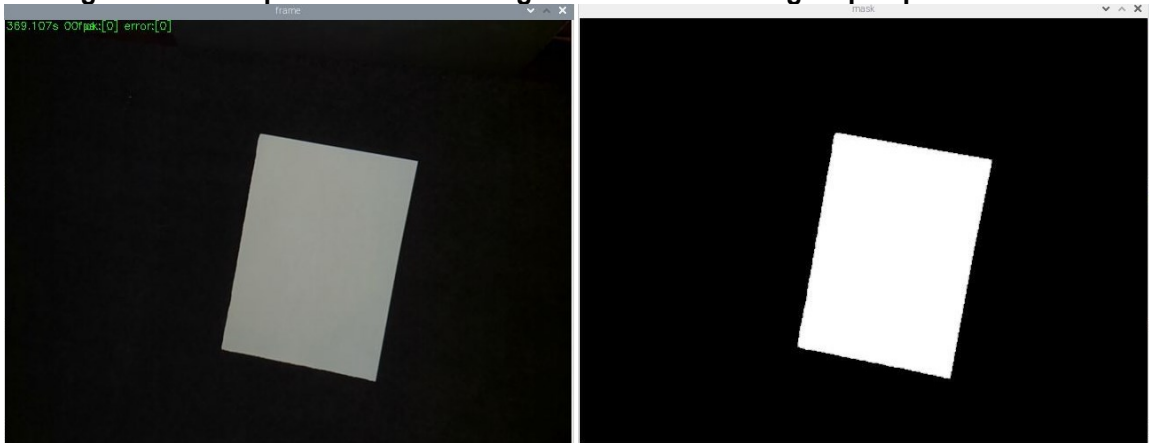


Fonte: Autoria própria (2021)

4.1 RESULTADOS PRELIMINARES DO SISTEMA DE MONITORAMENTO

A partir dos testes preliminares executados, foi possível testar a assertividade do sistema para cada tipo de peça.

Na Figura 19 é apresentado um exemplo de detecção de uma peça ideal, e, portanto, em conformidade com as especificações programadas. Nota-se que na tela à esquerda há a imagem original obtida a partir da Raspicam e à direita a imagem após o processamento, binarização e aplicação de técnicas de morfologia.

Figura 19 – Comparativo entre a imagem recebida e a imagem pós processamento

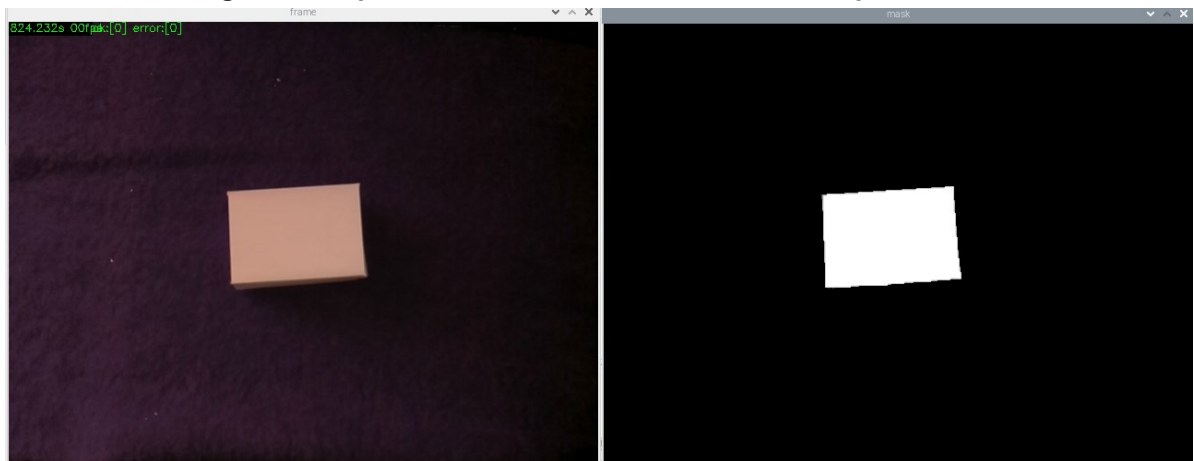
Fonte: Autoria própria (2021)

No primeiro teste, foram escolhidas somente peças ideais em 2D e com 4 vértices, com o perímetro dentro da faixa estabelecida. Ou seja, o resultado que deveria apresentar seria de 100% de peças conformes, com a contagem de 30 peças.

O resultado final foi de 100% de peças conformes, porém, com a contagem de 27 peças. Isso quer dizer que o sistema não detectou 10% das peças que passaram por ele, devido à velocidade de captura ser inferior à velocidade que as peças foram passadas pelo sistema.

No teste 2, foram utilizadas 30 peças em 3D, como visto na Figura 20. Assim como no primeiro, este teste deveria apresentar 100% de peças conformes e com a contagem de 30 peças. O resultado final foi de 26 peças conformes, 2 consideradas inconformes e 2 não foram captadas. As não captadas se deve ao mesmo motivo do teste 1 e as que apresentaram o resultado errado foi devido as sombras projetadas pelas peças.

Figura 20 - Apresentação do teste de detecção com peça em 3D

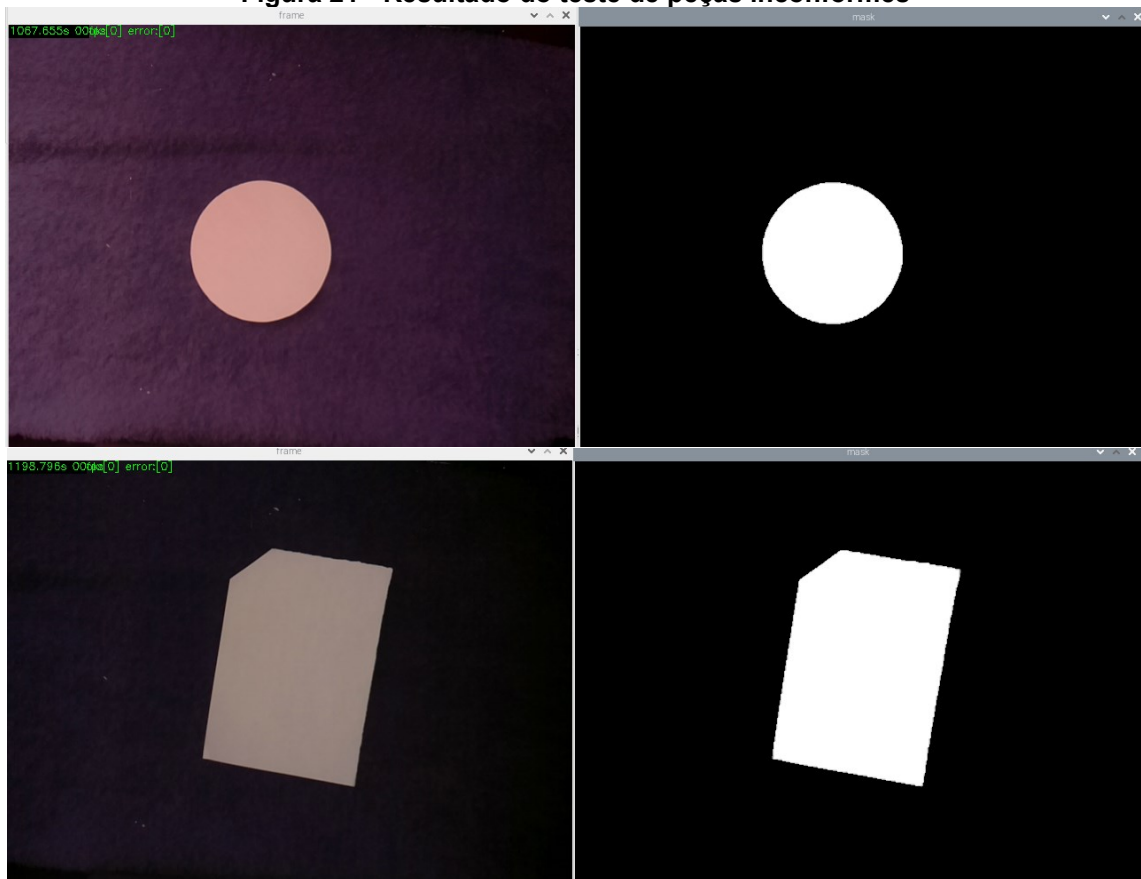


Fonte: Autoria própria (2021)

O terceiro teste foi realizado com 30 peças inconformes, ou seja, que tem um perímetro dentro do especificado, mas não possui 4 vértices, tal como círculos e peças com os vértices cortados, em que o exemplo pode ser visto na Figura 21. A permissividade do corte do vértice pode ser alterada na etapa “*detect*” nos parâmetros da função *cv2.approxPolyDP*, em que foi usado 2% do perímetro para uma permissividade pouco maior devido a algumas peças terem defeitos no corte por terem sido feitas manualmente.

Neste teste, o sistema teve 100% de acerto nas 30 tentativas, ou seja, em todas elas o sistema acusou como um objeto inconforme ao padrão estabelecido. Um dos motivos de 100% de acerto é porque os objetos realizados são feitos de maneira que é ideal para o sistema reconhecer.

Figura 21 - Resultado do teste de peças inconformes



Fonte: Autoria própria (2021)

Com a finalização dos resultados preliminares, foi necessário a calibração e análise da frequência de peças que passam na detecção, pois algumas peças não foram detectadas devido a velocidade que passou pela câmera. Em uma esteira industrial essa velocidade é constante, então facilita no momento da calibração. Como nos testes foram usados aparatos manuais, foi necessário um *delay* um pouco maior e um estudo de acordo com a média dos tempos para que não houvesse duplicidade na captação da imagem e nem que as peças passassem sem a captação.

4.2 FUNCIONALIDADES E DESEMPENHO DO SISTEMA

Após os testes preliminares serem encerrados e as correções serem realizadas para os testes finais, foram divididos novamente peças conformes, inconformes e exemplos de alerta. O *delay* aplicado para cada verificação foi de 1,4 segundos, considerado o mais próximo do ideal de acordo com a velocidade que as peças foram

captadas neste exemplo. Porém, é visto que esse *delay* em vias industriais deve ser abaixo de 1 segundo, adaptado de acordo com a velocidade da esteira aplicada.

Diferente dos testes preliminares em que as peças eram colocadas divididas entre os grupos conformes, inconformes e alertas, neste momento foram colocadas as peças em posições aleatórias, sempre em fila. Os testes foram realizados em 4 baterias de 30 peças cada, com as peças aleatoriamente colocadas para contagem e com peças de alerta entre elas.

As peças de alerta tiveram 100% de assertividade, portanto, pode-se afirmar que o sistema está garantido contra peças fora das especificações de segurança e o mesmo só volta à detecção se houver influência externa e reiniciar a captação pela página *web*.

Dentre as 120 peças captadas, houveram 11 erros, gerando uma assertividade de 90,83%. Dentre eles, o principal erro foram 8 peças que não foram captadas, e as outras 3 foram peças conformes que foram consideradas inconformes, provavelmente ocasionado devido à sombras presentes na imagem.

Neste período, a contagem de tempo realizada na tabela “Tempo”, mostrada na Figura 11, auxilia no cálculo de OEE do teste, devido a contagem de tempo em que fica operando, parado ou desligado. Os resultados do teste podem ser vistos na Figura 23, em que é mostrado todos os dados de acordo com a data. Neste teste, o tempo total foi de 23 min e 07 segundos, com um tempo de parada de 03 min e 48 segundos e não houve tempo desligado. Devido a isso, o cálculo de disponibilidade foi de 100%, porém, devido ao tempo em que o sistema ficou parado para a retirada de peças potencialmente perigosas, a performance foi de 83,56% baixando o resultado do OEE para 75,90% realizando a fórmula:

$$\text{OEE} = 90,83\% * 100 \% * 83,56\% \quad (1)$$

Figura 22 - Resultado do teste final apresentado com todos os dados na tela de relatório



Fonte: Autoria própria (2021)

O principal problema encontrado na captação dos testes foram as peças que passaram sem serem analisadas. Peças essas que poderiam ser alertas e gerar problemas à máquina e aos operadores. Este problema pode ser amenizado ao utilizar-se de uma esteira com velocidade constante, em que as peças estarão sempre na mesma velocidade, facilitando a calibração. Outra ação que pode ser realizada é a redução de processos sendo executados no sistema operacional da placa de desenvolvimento *Raspberry*, visto que a mesma foi utilizada com interfaceamento gráfico para análise dos testes, porém, na versão final pode ser utilizada sem a necessidade dessa ferramenta, que consome mais memória no comparativo.

Com a finalização do projeto, as teorias apresentadas passam a ficar mais claras e também verifica-se a possibilidade de adaptação para vários produtos diferentes. Pode-se notar que ainda existem grandes margens a serem ajustadas para uma possível versão futura do projeto, visto que há espaço para melhorias com um conhecimento mais apurado acerca de visão computacional e outras ferramentas de detecção.

Cada parte do projeto tem sua dificuldade e suas funções, por esse motivo, pode-se usar os códigos e esquemáticos para o desenvolvimento de outras ferramentas úteis para as mais diversas áreas, visto que a análise de visão computacional pode ser utilizada desde a agricultura à desenvolvimentos da área robótica.

As maiores dificuldades encontradas na elaboração e do projeto foi esbarrar em limitações de processamento da placa de desenvolvimento e limitações do próprio autor, pois foram necessários estudos de uma nova linguagem e de ferramentas ainda não vistas pelo mesmo. A própria ferramenta XAMPP utilizada para a ligação do MySQL e Apache, fundamentais na criação do banco de dados e comunicação entre os sistemas, foi um problema nas atualizações que o sistema operacional efetuava, visto que existem funções que passam a ser diferentes de acordo com a versão e diversas vezes foram percalços encontrados.

O sistema possui grandes limitações se tratando de detecção, visto que o mesmo possui maior capacidade de detecção de peças inconformes de acordo com os vértices, porém, essa abordagem por vértices não tem bons resultados quando se procura detecção de defeitos em peças ou manter padrão de qualidade, visto que se houver qualquer alteração de relevo, o sistema atual irá aprovar.

Com uma melhor ferramenta de detecção e qualidade de captura, o sistema pode ser utilizado em larga escala e comercializado como ferramenta de segurança para os trabalhadores e como controle de produção, como foi apresentado neste trabalho.

5 CONCLUSÃO

Neste trabalho, foi apresentado um protótipo, utilizando *Raspberry* e visão computacional, para a realização de monitoramento de produtos em uma esteira industrial, com principais objetivos de aumentar a produtividade no desenvolvimento de produtos e garantir a qualidade das peças, com relatórios em tempo real para o controle de produção seriada.

O projeto consistiu em, primeiramente, estudar sobre a funcionalidade das esteiras, quais os indicadores de desempenho necessários para um bom controle de qualidade, além de estudar quais seriam os melhores métodos para a aplicação. O projeto foi realizado a partir da placa de desenvolvimento *Raspberry*, com o auxílio da *Raspicam* como ferramenta de captura e linguagem *Python* para a programação. A partir da captura e análise das peças, todos os dados foram armazenados em um banco de dados para interligar o sistema à uma interface *web* e aos dispositivos de saída, que apresentam dados e alertas ao operador.

O sistema foi capaz de realizar a detecção de vértices para análise de formatos e tamanhos, via biblioteca OpenCV, possibilitando a contagem e conformidade das peças. Os testes realizados permitiram a identificação de peças potencialmente perigosas, tal como uma mão humana e formatos diferente do esperado, reduzindo a possibilidade de acidentes. Além de ser possível a detecção do tamanho da peça, permitindo a calibração de acordo com altura da câmera e material utilizado no funcionamento.

Como possibilidade de melhorias, é possível identificar a partir dos problemas encontrados, tal como o processamento lento da placa de desenvolvimento, em que pode ser melhorado a partir de uma instalação mais limpa do sistema operacional e as dificuldades de detecção encontradas, em que podem ser identificados os problemas de qualidade de imagem captada, de acordo com a luminosidade instável de acordo com o horário do dia e a resolução de captura da *Raspicam*.

Ao final de todo o projeto, com todas as etapas em mãos, pode-se afirmar que o resultado foi satisfatório, com ressalvas e possibilidades de evolução. O ramo de atividade pode ser explorado nas mais diversas áreas, e este projeto pode ser utilizado como base ou ferramenta de desenvolvimento para outros trabalhos.

REFERÊNCIAS

- ALBUQUERQUE, M.; ALBUQUERQUE, M. **Processamento de imagens: métodos e análises**. CBPF. Disponível em: <http://www.cbpf.br/cat/pdsi/pdf/ProcessamentoImagens.PDF>. Acesso em: 22 out. 2018.
- ALGERI, T.; ABREU, P. G.; PAULA FILHO, P. Rastreamento da movimentação em aviários através de processamento digital de imagens. *In: XVII SEMINÁRIO TÉCNICO CIENTÍFICO DE AVES E SUÍNOS - AVESUI*, 2018, Medianeira. **Anais[...]** Medianeira: UTFPR, 2018.
- ARAUJO, A. P. **Sensor de imagem CMOS**. Infoescola. Disponível em: <https://www.infoescola.com/fotografia/sensor-de-imagens-cmos/>. Acesso em: 22 out. 2018.
- BARROS, E.; CAVALCANTE, S. **Introdução aos sistemas embarcados**. Universidade Federal de Pernambuco-UFPE, 2010. Não publicado.
- BASTOS, E. **O que é a produção em série?** Portal Gestão. 2015. Disponível em: <https://www.portal-gestao.com/artigos/7673-o-que-%C3%A9-a-produ%C3%A7%C3%A3o-em-s%C3%A9rie.html>. Acesso em: 16 out. 2018.
- BRASIL. Senado Federal. **Consolidação das leis do trabalho – CLT e normas correlatadas**. Brasília: Senado Federal. 2017.
- BRADSKI, G.; KAEHLER, A. **OpenCV**. *Dr. Dobb's journal of software tools*, v. 3, 2000. Disponível em: [http://roswiki.autolabor.com.cn/attachments/Events\(2f\)ICRA2010Tutorial/ICRA_2010_OpenCV_Tutorial.pdf](http://roswiki.autolabor.com.cn/attachments/Events(2f)ICRA2010Tutorial/ICRA_2010_OpenCV_Tutorial.pdf). Acesso em 09 out. 2021.
- CAMPOS, L. **Banco de Dados**. 2008. Disponível em: http://www.univasf.edu.br/~leonardo.campos/Arquivos/Disciplinas/POO_2007_2/Aula_09.pdf. Acesso em: 26 out. 2018.
- CASTELLS, M. **A sociedade em rede: a era da informação**. 2 ed. São Paulo. Editora Paz e Terra. 1992.
- CHASE, O. **Sistemas embarcados**. 2007. Notas de aula. Disponível em: http://www.maxpezzin.com.br/aulas/6_EAC_Sistemas_Embarcados/1_SE_Introducao.pdf. Acesso em 09 out. 2021.
- CUNHA, A. F. **O que são sistemas embarcados**. *Saber Eletrônica*. v. 43, n. 414, p. 1-6, 2007.
- DEBIAN: the universal operating system. DEBIAN.ORG. Disponível em: <https://www.debian.org/index.pt.html>. Acesso em: 26 out. 2018.

DELAI, R. L.; COELHO, A. D. **Visão computacional com a OpenCV**: material apostilado e veículo seguidor autônomo. 2012. Não publicado.

DIAS, A. J. R. *et al.* **Caracterização de acidentes radiológicos industriais**. 2019. Disponível em: <http://repositorio.ipen.br/bitstream/handle/123456789/29870/25650.pdf>. Acesso em: 21 out. 2021.

DONINI, C. *et al.* **Esteira transportadora automatizada**. Trabalho de Conclusão de Curso - Centro Paula Souza. p. 44. 2014.

TIPOS e aplicações de sensores na indústria. 2017. Disponível em: <http://www.engerey.com.br/blog/tipos-e-aplicacoes-de-sensores-na-industria>. Acesso em: 25 out. 2018.

EULER, G. **Diferenças e semelhanças entre arduino e raspberry pi**. 2015. Disponível em: <https://www.infodicas.com.br/sistemas-embarcados/diferencas-e-semelhanças-entre-arduino-e-raspberry-pi>. Acesso em: 26 out. 2018.

GARCIA, F. **Introdução aos sistemas embarcados e microcontroladores**. 2018. Disponível em: <https://www.embarcados.com.br/sistemas-embarcados-e-microcontroladores/>. Acesso em: 25 out. 2018.

GRAMSCI, A. **Americanismo e fordismo**. Universale economica. 1950. Editora Hedra. 2008.

GONZALES, R.; WOODS, R. **Digital image processing**. 2 ed. Prentice Hall. Nova Jérsei. 2002.

HALL, E. C. **Journey to the moon**: the history of the apollo guidance computer. Reston, Virginia, USA: American Institute of Aeronautics and Astronautics. 1996. p. 196.

IMPORTÂNCIA do monitoramento de produção na indústria 4.0. 2018. Disponível em: <http://vedois.com.br/site/importancia-do-monitoramento-de-producao-na-industria-4-0/>. Acesso em: 09 out. 2021.

KIM, T.; RAMOS, C.; MOHAMMED, S. **Smart city and IoT**. 2017. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0167739X17305253>. Acesso em: 09 out. 2021.

LOPES. **Sistemas de manufatura e estratégias de produção**. 2011. Notas de Aula. Disponível em: http://sites.poli.usp.br/d/pmr2202/arquivos/aulas/PMR2202_Aula_L2.pdf. Acesso em: 23 out. 2018.

MANSUR, A. **O Fordismo e o Modelo T**. Disponível em: <https://meuartigo.brasilecola.uol.com.br/historia-geral/o-fordismo-modelo-t.htm>. Acesso em: 24 out. 2018.

- MARENGONI, M.; STRINGHINI, D. Tutorial: introdução à visão computacional usando opencv. **Revista de informática teórica e aplicada**, v. 16, n. 1, p. 125-160, 2009.
- MARIADB products and tools for development and production. MARIADB. Disponível em: <https://mariadb.com/downloads/> Acesso em: 25 out. 2018.
- MARQUES, D. C. *et al.* Um sistema de medição de volume por visão computacional *In: Simpósio Brasileiro de Automação Inteligente*. São Bernardo do Campo. **Anais[...]** São Bernardo do Campo: Centro Universitário FEI, 1999.
- MASS Production. 2009. Disponível em: <https://www.economist.com/news/2009/10/20/mass-production>. Acesso em: 16 out. 2018.
- MELO, M. R. *et al.* Indústria 4.0 inovando com qualidade. *In: 3º Congresso Internacional de Pesquisa, Ensino e Extensão*. Anápolis. **Anais[...]** Anápolis: UniEvangélica, 2018.
- MILANO, D.; HONORATO, L. B. **Visão computacional**. 2014. Não publicado.
- OLIVEIRA, A. S.; ANDRADE, F. S.. **Sistemas embarcados: hardware e firmware na prática**. Editora Érica Ltda, 2006.
- PASSARELLI, L. **Aplicação de visão computacional com OpenCV**. Embarcados. 2017. Disponível em: <https://www.embarcados.com.br/aplicacao-de-visao-computacional-com-opencv/>. Acesso em: 28 abr. 2021.
- QUINTA, L. N. B. **Desenvolvimento de um sistema de visão computacional para o controle microbiano em processos de produção de etanol**. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação) - Universidade Católica Dom Bosco, Campo Grande, 2009.
- ROSEBROCK, A. **OpenCV shape detection**. Pyimagesearch. 2016. Disponível em: <https://www.pyimagesearch.com/2016/02/08/opencv-shape-detection/>. Acesso em: 28 abr. 2021.
- RUDEK, M.; COELHO, L. S.; CANGIOLIERI JUNIOR, O. Visão computacional aplicada a sistemas produtivos: fundamentos e estudo de caso. *In: XXI Encontro Nacional de Engenharia de Produção*. Salvador. **Anais[...]** Salvador: Pontíficia Universidade Católica do Paraná, 2001.
- SCHWAB, K. **A Quarta Revolução Industrial**. 1 ed. Edipro. 2016.
- SILVA, H. R. Z. da; WEIZENMAN, G. S.; MUTZENBERG, R. Algoritmo para reconhecimento automatizado de defeitos em frangos de linha de produção utilizando visão computacional. *In: XXII Congresso Brasileiro de Automática*, João Pessoa. 2018. **Anais[...]** João Pessoa, 2018.

SOUSA, M. P.; LOPES, W. T. A. Desafios em Redes de Sensores sem Fio. **Revista de Tecnologia da Informação e Comunicação**, v. 1, n. 1, p. 41-47, 2011.

STADZISZ, P. C.; RENAUX, D. P. **Software Embarcado**. UTFPR. Disponível em: http://www.dainf.ct.utfpr.edu.br/~douglas/Stad_Renaux_Software_Embarcado.pdf. Acesso em: 25 out. 2018.

SUEIRO, D. **Arduino vs Raspberry Pi: entenda as diferenças**. 2014. Disponível em: <https://www.embarcados.com.br/arduino-vs-raspberry-pi/>. Acesso em: 26 out. 2018.

TAVARES, J. M. R. S. **Análise de movimento de corpos deformáveis usando visão computacional**. Tese (Engenharia Electrotécnica e de Computadores) – Universidade do Porto, 2000.

WATANABE, D. T. **Sistema de visão computacional para a contagem de sementes de soja**. Monografia de especialização (Especialização em Automação Industrial) – Universidade Tecnológica Federal do Paraná. Curitiba, 2019.

ZATTAR, I. C.; RUDEK, S.; TURQUINO, G. S. **O uso do indicador OEE como ferramenta na tomada de decisões em uma indústria gráfica: um caso prático**. Iberoamerican Journal of Industrial Engineering, v. 2, n. 4, p. 113-132, 2010.