



Faculdades Integradas de Taquara - Faccat  
Av. Oscar Martins Rangel, 4.500  
Taquara, RS, CEP 95600-000

## **Curso de Sistemas de Informação**

### **CCI: CRÉDITO DE CAFÉ INTELIGENTE<sup>1</sup>**

Gabriel Krummenauer de Moraes<sup>2</sup>

Rodrigo Henrich<sup>3</sup>

#### **RESUMO**

O projeto consiste no desenvolvimento de um classificador de imagens com modelo supervisionado, atuando em classificação de imagens capturadas em tempo real em um sistema já existente, em uso no laboratório de TI da FACCAT com um Raspberry. O classificador é utilizado para que os alunos tenham a opção de receber um crédito diário na máquina de café disponibilizada pelos cursos de TI. Alunos que levam sua xícara de café são beneficiados com um crédito de café bônus, dessa forma os alunos têm um incentivo em abandonar o uso de copos descartáveis que trazem problemas ao meio ambiente e à sua saúde a longo prazo. O objetivo geral é o desenvolvimento do classificador e sua respectiva inclusão no sistema, sendo o desenvolvimento organizado de forma geral em levantamento de requisitos, criação do código, implementação no ambiente e teste. O projeto gerou a funcionalidade com o algoritmo em pleno funcionamento e taxas aceitáveis de acerto na classificação para a liberação do crédito extra, o desenvolvimento do classificador se tornou o maior desafio no decorrer do desenvolvimento porém foi superado após diversas iterações e testes.

**Palavras-chave:** Aprendizado de Máquina; OpenCV; Raspberry; Viola-Jones.

#### **ABSTRACT**

The project consists of developing an image classifier with a supervised model, working to classify images captured in real time in an existing system, in use in the FACCAT IT laboratory with a Raspberry. The classifier is used so that college students have the option of receiving a daily credit at the coffee machine provided by IT courses. Students who bring their cup of coffee benefit from a bonus coffee credit, this way students have an incentive to abandon the use of disposable cups that cause problems for the environment and their health in the long term. The general objective is the development of the classifier and its increment in the system, with development generally organized into requirements gathering, code creation, implementation in the environment and testing. The project generated functionality

<sup>1</sup> Trabalho de Conclusão de Curso. Data da submissão e aprovação: A definir.

<sup>2</sup> Acadêmica do curso de Sistemas de Informação das Faculdades Integradas de Taquara – Faccat/RS. *E-mail:* gabrieldemorais@sou.faccat.br.

<sup>3</sup> Professor orientador das Faculdades Integradas de Taquara – Faccat/RS. *E-mail:* rodrigohenrich@faccat.br

with the algorithm fully functioning and acceptable classification success rates for releasing extra credit. The development of the classifier became the biggest challenge during development, but it was overcome after several iterations and tests.

**Keywords:** Machine Learning; OpenCV; Raspberry; Viola-Jones.

## 1 INTRODUÇÃO

Segundo a organização para desenvolvimento econômico, OECD (2022), atualmente o objetivo internacional de eliminar a poluição plástica está muito distante de ser atingido, com projeções de triplicar o consumo constatado até 2060 se as práticas atuais de consumo se mantiverem.

A poluição plástica é pautada em discussões ao redor do mundo a décadas e se mantém cada vez mais ativa conforme novas campanhas são criadas, porém o consumo de plástico está longe de ser abolido. Os cursos de Tecnologia da Informação da FACCAT disponibilizam para os alunos um sistema de créditos de café, onde os alunos recebem créditos em determinadas situações, principalmente em aula, cedidos pelos professores. Os créditos são utilizados em uma máquina localizada no laboratório dos cursos de TI e, geralmente, os alunos utilizam copos descartáveis para resgatarem seus créditos.

O classificador implementado terá um funcionamento muito simples, o aluno utiliza um QR code para resgatar seus créditos, caso o aluno não tenha nenhum crédito na conta e faça a leitura dele na máquina, ele terá a oportunidade de receber um crédito extra com o auxílio da câmera do próprio sistema que faz o reconhecimento da xícara do aluno e libera um crédito para ele. Atualmente o sistema do crédito extra já existe, porém de forma manual, o aluno deve solicitar ao monitor do laboratório o seu crédito extra, o que por vezes é ineficiente. Dessa forma esse processo será automatizado com o uso de inteligência artificial por meio de modelos de reconhecimento de imagem.

A proposta desse projeto é contribuir principalmente para a diminuição do uso de copos plásticos, pois a implementação criará a possibilidade de o aluno levar consigo uma xícara e resgatar um crédito extra à noite. O desenvolvimento do projeto foi dividido em seis etapas principais, sendo elas a ideação, pesquisa e testes, adaptação e validação, desenvolvimento, ajustes e testes no ambiente e, por fim, a implementação final. Não houve um cronograma fixo, mas sim datas definidas

para o fim das etapas seguindo um modelo de desenvolvimento ágil adaptado ao projeto. Dessa maneira o objetivo geral foi o desenvolvimento da nova funcionalidade e implementação da mesma sem interferir no restante do sistema, já os objetivos específicos seriam a criação do classificador a partir do método Viola-Jones com uma boa acurácia e a revisão do código-fonte do sistema para implementação da nova funcionalidade.

O impacto gerado pela implementação não se limita a diminuir o consumo de plástico mas também a diminuir a exposição dos alunos a substâncias prejudiciais que são liberadas pelo plástico em contato com líquidos quentes. Segundo Zenilda (UFMG) os materiais dos quais os copos plásticos geralmente são compostos são formados por substâncias que, ao estarem expostas a líquidos quentes, liberam toxinas prejudiciais a saúde a curto e longo prazo uma das substâncias liberadas que também a mais prejudicial é o estireno que pode causar até câncer com exposições mais longas.

## **2 FUNDAMENTAÇÃO TEÓRICA**

O sistema original, disponível para os alunos dos cursos de TI da FACCAT, foi desenvolvido por um aluno como seu projeto de conclusão de curso. Ele permite que os alunos recebam créditos em uma conta vinculada a seu email da faculdade e utilizem esses créditos com um QR code na máquina disponível no laboratório dos cursos. A ideia de levar uma xícara e ser recompensado com um crédito já existia, porém ela dependia do aluno encontrar alguém encarregado do laboratório para que o crédito fosse liberado manualmente.

As tecnologias utilizadas foram Raspberry em conjunto da linguagem de programação Python, que é uma linguagem nativa do hardware de acordo com a empresa que o produz em sua página “What is a Raspberry Pi?” (Raspberry, 2022). Já a biblioteca OpenCV será a responsável, dentro da ferramenta de aprendizado de máquina voltada para classificação de imagens, por classificar as imagens capturadas pela câmera presente no Raspberry e definir se o aluno tem um xícara consigo, para isso foi desenvolvido um classificador de imagens em cascata baseado no método Viola Jones (2001).

O desenvolvimento do código fonte foi por meio do VSCode<sup>4</sup> com extensões para Python e para Raspberry, a implementação e teste foi feita diretamente no ambiente onde o projeto funciona pois a diferença entre o ambiente de desenvolvimento e o de implementação poderiam acarretar em falhas.

O uso da aplicação acontece da mesma forma que os alunos já estavam acostumados, o QR code é apontado para a câmera e o Raspberry faz a liberação do café e caso o aluno não tenha créditos a nova função permite que ele apresente uma xícara e receba um crédito extra, uma vez por noite.

## 2.1 Raspberry

O hardware principal utilizado pela aplicação em comunicação entre a interface do usuário e a máquina de café. Este hardware é considerado um microcomputador pois representa o funcionamento de um computador e permite a implementação de diversas funcionalidades com um placa relativamente pequena em comparação aos computadores convencionais, sendo também modular então permite a adição de recursos necessários.

O Raspberry é algo semelhante a um computador porque permite a execução de tarefas de forma muito similar, para isso utiliza de vários pinos de interface para a adição de outros componentes e dispositivos, a partir desses pinos é possível desenvolver uma grande variedade de sistemas de acordo com o hardware adicionado, inclusive acionar os dispositivos acoplados (Raspberry, 2023). Isso permite ao usuário explorar uma ampla gama de projetos com poucas barreiras para o desenvolvimento dos mesmos.

## 2.2 Python

É a linguagem de programação de alto nível, interpretada e orientada a objetos, sua tipagem e sintaxe são simples. Utilizada para o desenvolvimento do projeto, tem uma vasta comunidade e permite implementações em diversas áreas,

---

<sup>4</sup> VSCode é uma ferramenta de edição de texto muito simples e com alto desempenho que possibilita a instalação de plugins e gerenciamento de código em alto nível devido aos recursos que podem ser adicionados conforme a necessidade do projeto.

sendo a área de machine learning<sup>5</sup> uma das principais áreas onde a linguagem é explorada possibilitando estudos mais aprofundados de acordo com a necessidade do projeto com livros como Python Machine Learning: Machine Learning and Deep Learning with Python de Sebastian Raschka e Vahid Mirjalili, publicada terceira edição em 9 de dez. de 2019.

Atualmente a linguagem tem grande uso por parte de empresas e desenvolvedores principalmente pela simplicidade da mesma, estando na primeira posição de linguagens utilizadas de acordo com a pesquisa feita pela Hostinger em 2022, os códigos permitem desenvolver praticamente qualquer tipo de software, principalmente com o uso de bibliotecas que são responsáveis por ampliar o alcance do programa sendo desenvolvido adicionando inúmeras funções com algoritmos complexos de forma simples e facilitada, permitindo que o desenvolvedor tire o melhor proveito possível com um tempo de desenvolvimento consideravelmente menor e aumentando a qualidade do código produzido com o uso de algoritmos bem trabalhados como por exemplo a biblioteca utilizada na elaboração do projeto OpenCV que disponibiliza modelos muito avançados de machine learning para uso acadêmico.

### **2.3 OpenCV**

Segundo o website oficial da ferramenta (OpenCV, 2023), OpenCV é uma biblioteca desenvolvida para transformar projetos que utilizem aprendizado de máquina e permitir um acesso a algoritmos de alto nível de forma simples e gratuita para incentivar avanços acadêmicos em estudos e projetos que envolvam essa área.

Responsável por fazer o tratamento das imagens e representá-las em dados com algoritmos mais sofisticados e de fácil implementação, essa tecnologia foi desenvolvida principalmente para uso acadêmico e atualmente é responsável por possibilitar diversos projetos de maneira facilitada. A biblioteca também conta com diversas ferramentas para outros fins dentro do aprendizado de máquina, sendo a classificação de imagem a mais importante para esse projeto.

---

<sup>5</sup> O Machine Learning (Aprendizado de máquina) consiste em um método para análise de dados para a construção de modelos analíticos, sendo um ramo muito utilizado da inteligência artificial para a criação de sistemas menos dependentes da intervenção humana pois tomam decisões a partir de dados e padrões que são identificados.

## 2.4 Algoritmo Classificador

O algoritmo de classificação é completamente baseado em um framework, uma estratégia de classificação desenvolvida e proposta por Viola, P. e Jones, M.(2001). Onde, de forma simplificada, a imagem capturada passa por uma série de testes em cascata cada vez mais complexos que se complementam e, se chegar ao fim dos testes, o classificador retorna um positivo para a imagem testada, este classificador foi muito importante na época de seu desenvolvimento pois tem uma taxa aceitável de acerto com um custo pequeno de processamento, assim se tornou a proposta perfeita para esse projeto devido às limitações físicas do hardware utilizado no sistema. O custo computacional para o desenvolvimento do algoritmo é relativamente alto porém o custo para uso do mesmo é uma pequena fração, assim o classificador permite sua implementação sem impactar o sistema de forma negativa. Trata-se de um algoritmo de classificação supervisionado, onde a construção do classificador se faz com a indicação de um conjunto positivo de imagens contendo o objeto esperado e um conjunto negativo contendo imagens que não contenham esse objeto.

Para que a imagem passe pelo classificador com um retorno positivo, ela é classificada por meio de uma técnica baseada nas características haar<sup>6</sup>, a partir dos parâmetros de classificação é feita uma varredura na imagem em busca de características específicas que compõem o que se está buscando identificar.

O algoritmo é muito indicado para detecção de faces pelas altas taxas de acerto mas também é adaptado em diversas situações, como na seção de trabalhos similares, onde ele pode ser utilizado para reconhecimento de objetos, figuras específicas, entre outros. Se torna uma opção muito interessante em sistemas de baixo custo ou de âmbito acadêmico.

O funcionamento do classificador está correlacionado com quatro conceitos muito importantes em sua composição (Macedo, Gomes, 2020). As características de Haar, onde a imagem passa por uma série de transformações com formas geométricas resultando em um valor único. As imagens integrais são imagens geradas a partir de um modelo matemático onde a soma dos pixels se torna muito mais eficiente para o classificador, assim o desempenho dele se mantém alto

---

<sup>6</sup> Características Haar se referem a métodos matemáticos para processamento e análise de sinais em dados propostos em 1909 por Alfred Haar, algo utilizado na era digital para a criação de métodos de aprendizado de máquina.

mesmo com imagens maiores. O algoritmo Adaboost é um método de classificação onde são gerados vários classificadores mais simples que trabalham como filtros e combinados resultam em um classificador com taxas de acerto muito superiores. Os classificadores em cascata formam a última etapa do algoritmo, onde classificadores são utilizados em sequência de forma cascata para transformar a imagem de entrada removendo partes irrelevantes à classificação.

### **2.3 Trabalhos Relacionados**

Um dos principais estudos relacionados a este projeto também é a base dele. Sendo ele o estudo publicado por Viola, P. e Jones, M.(2001) com o título “Rapid object detection using a boosted cascade of simple features”. O trabalho resume-se ao desenvolvimento de um modelo de classificação que se tornou muito útil para a área de aprendizado de máquina na época, principalmente na detecção facial e posteriormente utilizado para outros tipos de detecção.

O modelo Viola-Jones também foi utilizado para replicar algumas habilidades humanas, como identificar a direção de setas, conforme descrito por Elisiane (2018), na entrega de seu trabalho de conclusão de curso. Nesse caso o algoritmo foi treinado exclusivamente com a finalidade de identificar essas setas nas imagens capturadas e obteve uma taxa superior a 98% de acertos em suas detecções.

Em outro trabalho o mesmo modelo foi utilizado para testes de detecção de faces e relógios com o uso de um hardware Raspberry e um módulo de câmera, sendo o trabalho desenvolvido e publicado como artigo no XXXVI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SOUSA OLIVEIRA et al., 2018).

Já para o artigo de Elhoseny (2020) um novo método chamado MODT é apresentado para detecção de objetos em vídeo em tempo real, sendo uma proposta interessante principalmente para sistemas de vigilância. Além de existirem outros métodos de classificação no mercado.

Por mais que o método Viola Jones já existe há mais de duas décadas no mercado ele ainda se mostra uma opção muito promissora para projetos de menor escala pois apresenta resultados consistentes, uma taxa de acerto aceitável em sua detecção e não cria impactos no desempenho do sistema onde é implementado. Existem outros métodos com características semelhantes porém a vasta

documentação e diversos materiais disponíveis sobre o método Viola Jones se tornam um grande diferencial para o desenvolvimento do classificador.

### **3 PLANEJAMENTO DO DESENVOLVIMENTO**

O desenvolvimento foi composto por três etapas fundamentais. Metodologia a ser utilizada, Análise do sistema existente e o Projeto de desenvolvimento da nova funcionalidade.

#### **3.1 Metodologia**

O projeto foi desenvolvido baseado na metodologia de desenvolvimento ágil, seguindo o manifesto para desenvolvimento ágil e os 12 princípios propostos em 2001 pela colaboração de 17 autores (Jeff Sutherland et al., 2001). Seguindo mais especificamente algumas etapas presentes no método Scrum<sup>7</sup> de maneira solo, sendo elas:

Planejamento e levantamento de requisitos do sistema, definindo uma lista de tarefas a serem desenvolvidas durante o desenvolvimento do projeto. Cronograma de entregas com prazos para o desenvolvimento de cada tarefa e prazos para a realização dos testes de garantia de qualidade necessários. As tarefas foram categorizadas conforme sua prioridade e seu tempo baseado em sua complexidade. Durante o desenvolvimento de cada tarefa e do projeto em si a ajuda do orientador foi essencial para a discussão do que será produzido e entregue. Revisão da tarefa após sua conclusão para eventuais correções e como forma de separar o que agregou ao projeto e o que não agregou.

A partir desses passos o desenvolvimento das tarefas ocorreu em sequência até que o projeto estivesse pronto. Após o desenvolvimento do classificador foram necessários testes manuais com as imagens utilizadas no treinamento do mesmo para levantar dados e métricas sobre seu desempenho.

---

<sup>7</sup> Scrum trata-se de uma metodologia ágil amplamente utilizada no mercado de trabalho atualmente e segundo o 16th State of Agile Report (2022) está em 87% dos projetos que utilizam metodologias ágeis. Já o solo se refere a uma equipe de desenvolvimento composta de uma única pessoa.

### 3.2 Análise

Houve duas análises de grande impacto para o desenvolvimento, a primeira análise consistiu em definir quais as características do sistema existente seriam aproveitadas para a implementação da nova funcionalidade, sendo elas físicas como hardware e virtuais como o ciclo de funcionamento do sistema.

Em um primeiro momento o foco do desenvolvimento se deu pela criação de códigos com o funcionamento limitado a nova funcionalidade para sua validação e posterior integração com o sistema existente.

Com o código-fonte da nova funcionalidade pronto para ser integrado e passar pelos testes necessários no ambiente de funcionamento do sistema, houve uma segunda análise com o foco na inclusão desse novo trecho desenvolvido no sistema.

Como o projeto foi desenvolvido a anos, houve grandes alterações nas tecnologias utilizadas causando conflitos e falhas no código novo, portanto foram necessárias novas interações no ciclo de desenvolvimento focadas em refatoração do código fonte original e atualização das dependências dele.

Com o sistema original funcionando já com seu código refatorado e dependências atualizadas a implementação da nova funcionalidade ocorreu sem problemas, sendo introduzida ao ciclo já existente do sistema.

### 3.3 Projeto

A partir da primeira análise o foco do desenvolvimento foi na criação do código fonte da funcionalidade, como ela é modular, pode ser introduzida no sistema existente sem muito esforço então o esforço maior se deu em preparar o ambiente de desenvolvimento, atualizar todas as dependências necessárias e desenvolver o código com o auxílio da documentação das ferramentas e comunidade.

Devido aos conflitos que ocorreram ao integrar o novo módulo ao já existente, foi necessário revisar todo o sistema original, dessa forma houve também, de forma inesperada, um grande esforço para atualizar o código já existente, para o tornar compatível com as tecnologias mais atuais utilizadas na funcionalidade, a refatoração e atualização também criou um impacto positivo no funcionamento do sistema pois ele passou a funcionar com maior desempenho.

## 4 DESENVOLVIMENTO, IMPLEMENTAÇÃO E TESTES

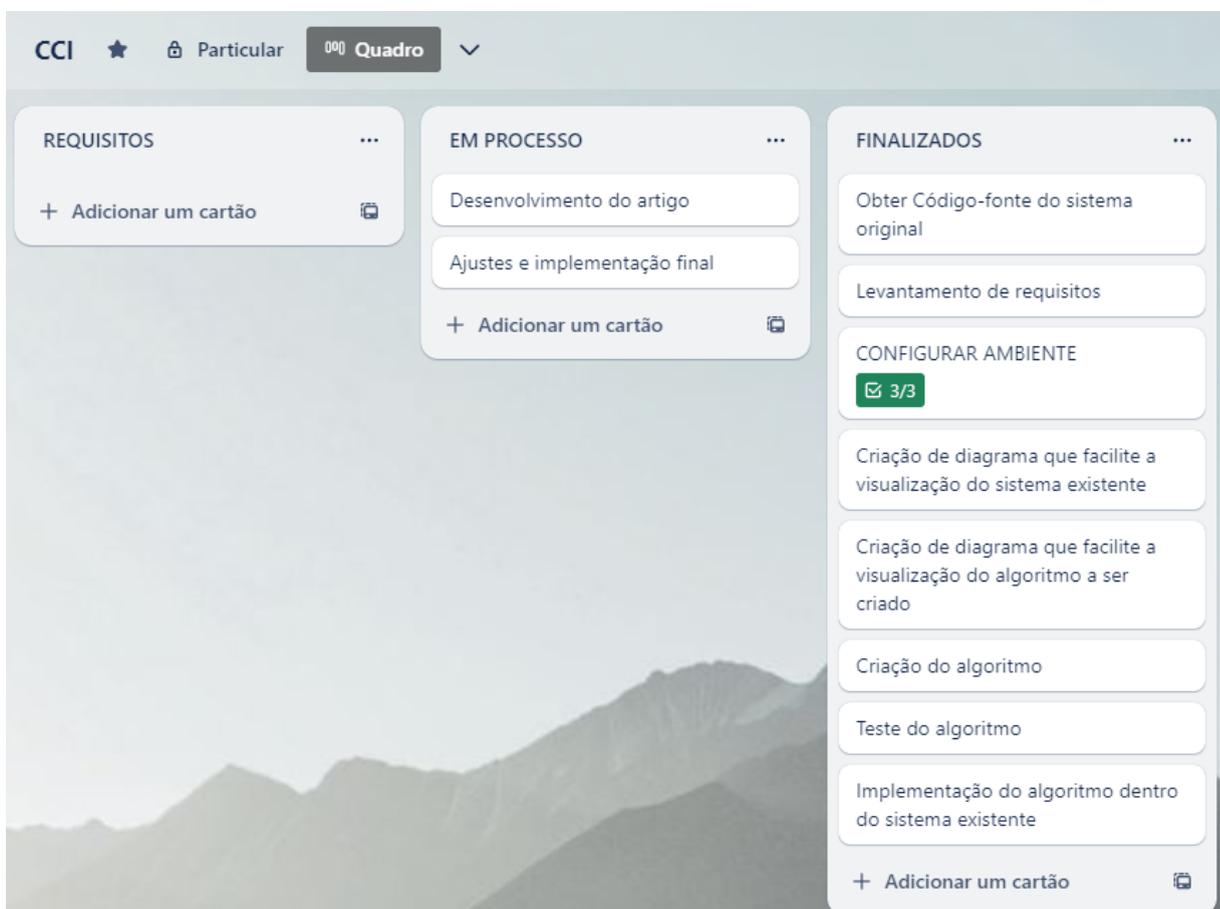
### 4.1 Desenvolvimento e Implementação

O software foi desenvolvido em etapas a partir das análises realizadas, ao definir os objetivos de desenvolvimento foram realizadas pesquisas e testes das linguagens e ferramentas a fim de garantir que o produto gerado a partir delas estaria de acordo com o objetivo principal proposto. Baseado no manifesto ágil de desenvolvimento de software, publicado originalmente em 2001, o desenvolvimento foi realizado seguindo algumas características mas se adaptando a necessidade do projeto.

Para auxílio durante o desenvolvimento houve um quadro montado no Trello identificando as etapas de forma mais geral.

A figura a seguir apresenta o quadro montado para servir de guia durante o desenvolvimento do projeto, com etapas simples mas bem definidas.

**Figura 1 – Quadro Trello**



Fonte: Imagem gerada pelo autor (2023).

Com o auxílio do Trello foi possível mensurar, até certo ponto, o andamento do projeto, sendo muito importante também para identificar os avanços durante seu desenvolvimento e, principalmente, para montar o roteiro de desenvolvimento seguido.

O ponto principal foi o equipamento utilizado, Raspberry, pois o código desenvolvido deveria ser compatível com o mesmo, criando a necessidade de adaptar cada trecho gerado ao momento da implementação.

Como a linguagem Python é uma linguagem nativa para o Raspberry, também de acordo com a empresa que o desenvolveu, além de ser a linguagem utilizada no projeto de créditos de café que já estava funcionando, foi a linguagem definida para o desenvolvimento desse projeto. Como o OpenCV também tem grande compatibilidade com desenvolvimento em Python (OpenCV, 2023) ambos se tornaram uma combinação muito importante no projeto, assim desde o hardware até o software as tecnologias são bastante compatíveis.

A implementação a ser adicionada foi desenvolvida primariamente em um computador com o interpretador VS Code e extensões para Python, assim como a biblioteca OpenCV instalada porém conforme deu-se o desenvolvimento de grande parte do código foi necessário a adição da mesma no ambiente de funcionamento, dessa forma foi possível identificar falhas e inconsistências que posteriormente foram corrigidos e até melhorados em relação ao código inicial.

Algumas partes do código desenvolvido foram corrigidas e concluídas dentro do ambiente do sistema operacional do Raspberry.

Para a criação do classificador foram testados diversos métodos disponíveis e documentados tanto pela empresa responsável pelo OpenCV como métodos disponibilizados por estudiosos da área. O método com melhor desempenho e maior simplicidade se tratou do uso de uma ferramenta criada exclusivamente para a criação desse tipo de algoritmo, a ferramenta Cascade Trainer GUI, na versão 3.3.1. A ferramenta já existe há algum tempo e é utilizada em projetos acadêmicos para facilitar o processo de geração do algoritmo, com uma interface muito amigável e um funcionamento muito simples de se entender ela entrega o classificador pronto para implementação.

Mesmo com o uso da ferramenta foi necessário fazer o levantamento das imagens positivas e negativas e gerar o classificador a partir delas, podendo ser

alterados parâmetros na geração dele conforme a necessidade de detecção do projeto.

Para o levantamento das imagens foram utilizados alguns algoritmos, sendo o de maior destaque atualmente o algoritmo criado a partir da ferramenta de pesquisa Bing (pypi, 2023), que possibilitou a busca de imagens a partir de termos com um algoritmo relativamente simples de se implementar. Houveram outras formas de se levantar as imagens porém se mostraram ineficientes devido a demanda do projeto pois não possibilitam de forma semi autônoma o levantamento de múltiplas imagens.

Depois de vários experimentos a conclusão do projeto contou com a criação do algoritmo utilizando 241 imagens positivas e 5896 imagens negativas. Com uma taxa de acerto geral superior a 82%. O algoritmo final desenvolvido levou em torno de 1 hora e 24 minutos para ser treinado, sendo ele utilizado no formato xml.

Foram gerados diversos classificadores, alguns com mais imagens positivas, outros com apenas uma imagem positiva. Então foram feitos comparativos em testes básicos em busca de aperfeiçoar o classificador para obter uma taxa aceitável de acerto. Um grande empecilho durante o desenvolvimento do projeto foi em razão do custo de processamento para geração do classificador que, no sentido oposto do classificador em si, necessita de uma parcela significativa de tempo e processamento computacional para ser criado, para computadores convencionais pode ser até certo ponto uma barreira porém o classificador resultante é utilizado com uma fração desse processamento e milésimos de segundos.

A tabela a seguir representa o teste do algoritmo de forma individual em todas as imagens utilizadas para seu desenvolvimento.

**Figura 2 – Teste do algoritmo**

<b>Matriz de Confusão</b>			
		<b>Detectada</b>	
		<b>Sim</b>	<b>Não</b>
<b>Real</b>	<b>Sim</b>	204	37
	<b>Não</b>	763	5133

Fonte: Imagem gerada pelo autor (2023).

A figura 2 representa uma matriz de confusão, muito utilizada dentro do aprendizado de máquina para identificar métricas e, nesse caso, medir o desempenho do classificador.

## 4.2 Testes

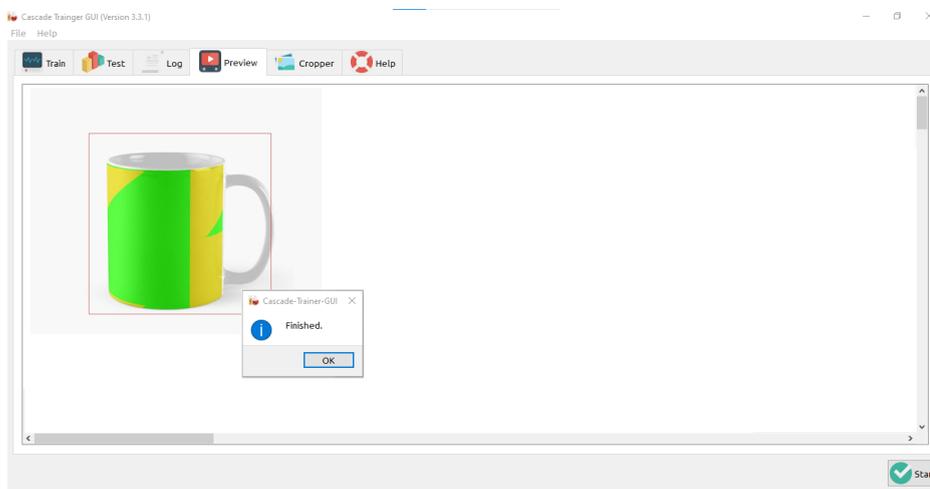
O programa como um todo pode ser testado quase que imediatamente após a primeira iteração, com a refatoração dele todo o sistema passou a funcionar de forma mais fluida e consistente. Os testes ocorreram conforme o andamento do período letivo, onde os alunos do curso puderam utilizar seus códigos para resgatar seus créditos sem problemas de desempenho do sistema.

O classificador foi testado primariamente utilizando as imagens das quais o mesmo foi construído diretamente na mesma ferramenta utilizada para produzir o algoritmo, sendo elas 241 positivas e 5896 negativas , assim gerando uma taxa aproximada de 86% de acurácia.

Os testes ocorreram de duas formas, a primeira gerando os dados para formação da matriz de confusão, sendo feito diretamente pela ferramenta que gerou o classificador e posteriormente a revisão manual dos resultados. A segunda feita com o auxílio de um código específico para testar o classificador com uma câmera pelo computador de desenvolvimento, sendo esse um teste de validação manual e simples de se realizar.

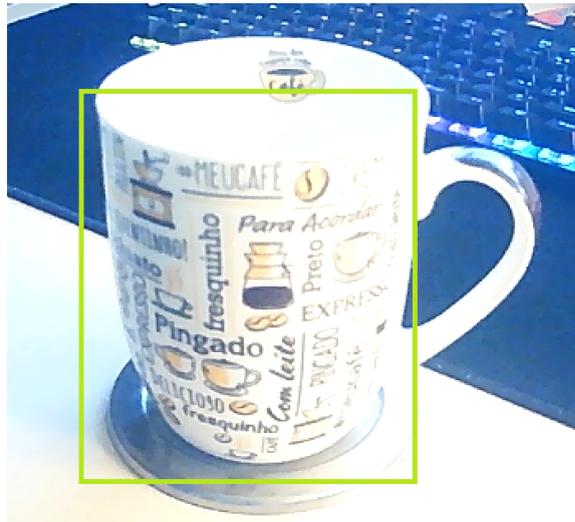
As duas figuras a seguir mostram as duas formas de testes, sendo o teste pela ferramenta e o teste manual, respectivamente.

### Figura 3 – Testes pela ferramenta GUI



Fonte: Imagem gerada pelo autor (2023).

**Figura 4 – Teste manual**



Fonte: Imagem gerada pelo autor (2023).

Ocasionalmente o classificador não identifica a xícara na imagem, sendo necessário um ajuste no ângulo dela para que identifique, pode ocorrer por diversos motivos então no sistema ocorre a coleta pela câmera do Raspberry múltiplas vezes, assim conforme o aluno manuseia a sua xícara as chances de ela não ser identificada são menores.

A seguir uma imagem do teste onde a xícara não é identificada e, posteriormente, uma imagem do código-fonte identificando a coleta das imagens pelo sistema.

**Figura 5 – Teste do algoritmo, falso negativo**



Fonte: Imagem gerada pelo autor (2023).

**Figura 6 – Código-Fonte: Coleta de imagem no sistema**

```
def creditoDiario(qrLido):
    if not (textoCreditoDiario(qrLido)):
        return
    global creditoLiberado
    print('qr liberado')
    for x in range(10):
        sucess, img = cam.read()
        img = cv2.resize(img, (300,300))
        imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        objeto = haarCascade.detectMultiScale(imgGray, 1.3, 2)
        if (objeto != ()):
            liberaCafe()
            creditoLiberado.append(qrLido)
            return
        time.sleep(1)
    lcd_string("  SUA XICARA  ",LCD_LINE_1)
    lcd_string("  NAO ENCONTRADA  ",LCD_LINE_2)
    acendeled(16)
    time.sleep(2)
    apagaled(16)
    return
```

Fonte: Imagem gerada pelo autor (2023).

A figura 6 também representa a função principal por onde ocorre o uso do algoritmo dentro do sistema, sendo responsável por identificar o código QR lido, definir se já recebeu ou não seu crédito diário e disponibilizar o crédito.

## 5 RESULTADOS ALCANÇADOS

A refatoração do código-fonte, foi uma etapa crucial para ser possível implementar a nova funcionalidade pois o código-fonte que funcionava no sistema utilizava de versões desatualizadas das tecnologias e já estava causando problemas no funcionamento geral do sistema, algo notado após a refatoração foi uma melhora no desempenho e tempo de resposta do sistema, funções simples como mostrar no display a quantidade de créditos atual do aluno voltaram a funcionar de forma plena e a máquina melhorou como um todo.

A figura a seguir mostra um comparativo entre um trecho do código-fonte original e um do código refatorado, sendo a esquerda o original e a direita o refatorado.

**Figura 7 – Comparativo entre Código original e o refatorado**

<pre>def decodifica():     try:         cam2 = Camera()         img2 = cam2.getImage()         barcode = img2.findBarcode()         if(barcode is not None):             barcode = barcode[0]             result = str(barcode.data)             cam2 = ()             img2 = ()             barcode = []             time.sleep(0)             consutaCafe (result)         return ('LENDO 2')     except:         print ('ERRO 1455')         acendeled(16)         time.sleep(0.5)         apagaled(16)         time.sleep(0.1)         acendeled(16)         time.sleep(0.5)         apagaled(16)         return 1</pre>	<pre>def decodifica():     try:         sucess, img = cam.read()         if (sucess):             img = cv2.resize(img, (300,300))             qrlido = pyzbar.decode(img)             img = ()             for qr in qrlido:                 result = qr.data.decode()                 time.sleep(0)                 print(result)                 consultaCafe (result)             qrlido = None             return print('LENDO QR')     except:         lcd_string(" *** ERRO *** ",LCD_LINE_1)         lcd_string(" QR INVALIDO ",LCD_LINE_2)         acendeled(16)         time.sleep(2)         apagaled(16)         return print('NAO LEU')</pre>
--	--

Fonte: Imagem gerada pelo autor (2023).

Outro ponto muito importante na refatoração foi um foco maior em aumentar a manutenibilidade do código, tornando mais fácil identificar cada parte do mesmo.

Uma das mudanças feita no código-fonte original foi relacionada a uma função chamada “baixaCafe”, ela tinha duas responsabilidades, primeiro, identificar o saldo do usuário e solicitar a baixa pelo banco de dados. Segundo, liberar o café para o usuário. Seguindo os princípios de SOLID<sup>8</sup> e boas práticas de desenvolvimento a função foi dividida em duas funções menores, cada uma com sua respectiva funcionalidade, sendo elas “baixaCafe” e “liberaCafe”.

A figura a seguir mostra a função antiga e ambas novas, sendo a esquerda o código original e a direita o refatorado.

<sup>8</sup> Os princípios SOLID são práticas definidas no desenvolvimento de um código-fonte com a finalidade de tornar o código gerado mais limpo e organizado, assim melhorando a vida útil do sistema em questões de manutenção, etc. O princípio utilizado nesse trecho é o de Responsabilidade Única.

Figura 8 – Funções de baixa e liberação do crédito

```
#Funcao baixa cafe
def baixaCafe (saldo, qrLido, tomou):
    desconto = 1
    saldoNovo = int(saldo) - int(desconto)
    saldoNovo2 = str(saldoNovo)
    tomouNovo = int(tomou) + int(desconto)
    tomouNovo2 = str(tomouNovo)
    #print saldoNovo
    #print qrLido
    #print type(qrLido)
    #return 1

    #sql2="UPDATE qrcores_geral SET quantidade=" + saldoNovo
    #sql2=("""UPDATE qrcores_geral SET quantidade=%s WHERE
    # criando uma conexao mysql
    #conexao=MySQLdb.connect( 'mysql.qrcafe.com.br' , 'qr
    # selecionando o banco 'teste'
    #conexao.select_db( 'qrcafe' )
    sql2="UPDATE qrcores_geral SET quantidade=" + saldoNovo
    try:
        cursor=conexao.cursor()
        cursor.execute(sql2)
        conexao.commit()
        cursor.fetchall()
        #connect.close()
        acendeled(12)
        #print 'LIGA LED'
        GPIO.setup(11, GPIO.OUT)
        #print("RELE LIGADO\n")
        GPIO.output(11, GPIO.LOW)

    # Exibe no display
    lcd_string("ESCOLHA SEU CAFE",LCD_LINE_1)
    lcd_string("      8      ",LCD_LINE_2)
    time.sleep(1)
    # Exibe no display

def baixaCafe (saldo, qrLido, tomou):
    desconto = 1
    saldoNovo = int(saldo) - int(desconto)
    saldoNovo2 = str(saldoNovo)
    tomouNovo = int(tomou) + int(desconto)
    tomouNovo2 = str(tomouNovo)

    sql2="UPDATE qrcores_geral SET quantidade=" + saldoNovo
    try:
        cursor=conexao.cursor()
        cursor.execute(sql2)
        conexao.commit()
        cursor.fetchall()
        liberaCafe()
        lcd_string(" SALDO ATUAL  ",LCD_LINE_1)
        lcd_string("      " + saldoNovo2 ,LCD_LINE_2)
        time.sleep(1)
    except MySQLdb.Warning as w:
        print (w)
    except MySQLdb.Error as e:
        print ("Erro executando baixa de cafe \n ", sql2)
        print (e)

def liberaCafe():
    acendeled(12)
    GPIO.setup(11, GPIO.OUT)
    GPIO.output(11, GPIO.LOW)
    # Exibe no display
    lcd_string("ESCOLHA SEU CAFE",LCD_LINE_1)
    lcd_string("      8      ",LCD_LINE_2)
    time.sleep(1)
    # Exibe no display
    lcd_string("ESCOLHA SEU CAFE",LCD_LINE_1)
    lcd_string("      7      ",LCD_LINE_2)
    time.sleep(1)
    # Exibe no display
    lcd_string("ESCOLHA SEU CAFE",LCD_LINE_1)
    lcd_string("      6      ",LCD_LINE_2)
    time.sleep(1)
    # Exibe no display
    lcd_string("ESCOLHA SEU CAFE",LCD_LINE_1)
    lcd_string("      5      ",LCD_LINE_2)
```

Fonte: Imagem gerada pelo autor (2023).

Ainda dentro das boas práticas de código, as funções foram identificadas de maneira simétrica, facilitando a identificação visual ao ler o código, os comentários foram realocados de acordo com sua descrição, identificando as funções utilizadas.

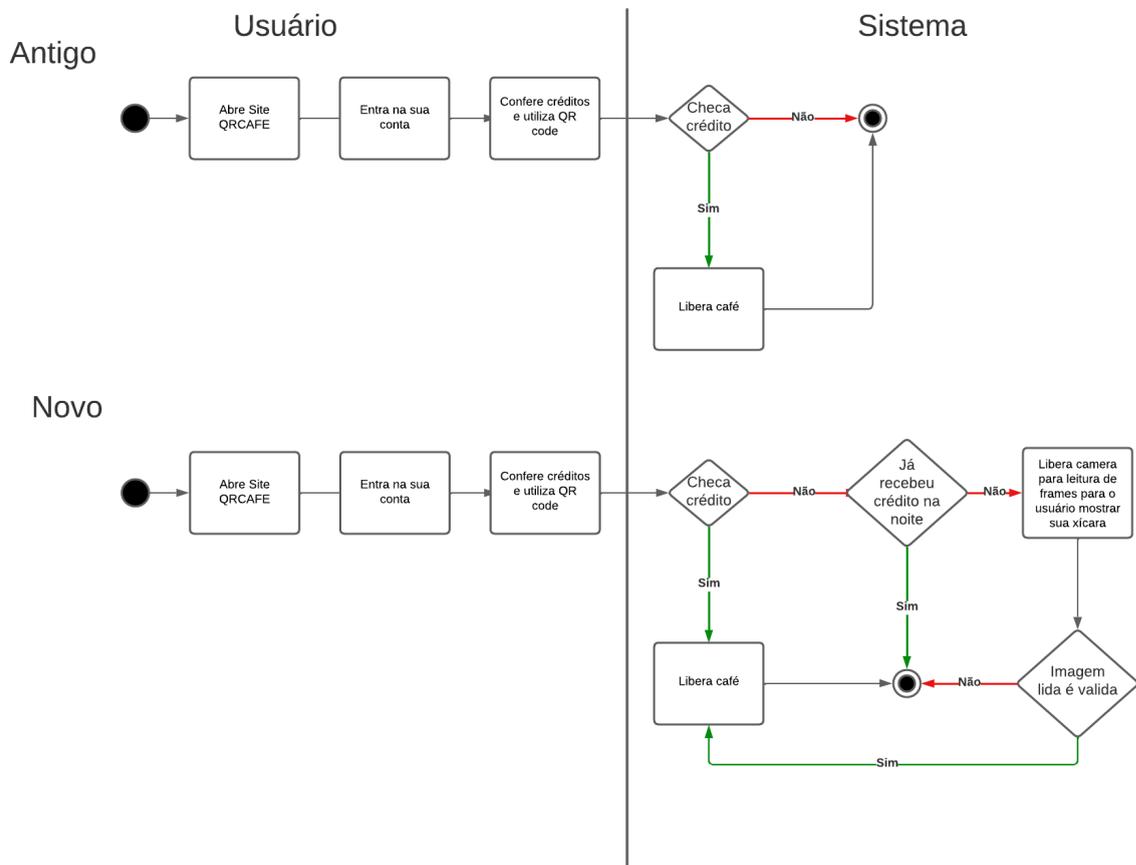
Algumas redundâncias foram reduzidas do código-fonte e as funções foram otimizadas conforme foi possível simplificá-las.

Outro recurso muito importante quando utilizamos a sintaxe de Python está relacionado ao espaçamento em cada linha de código, ele existe para que o compilador<sup>9</sup> identifique os blocos de código porém se feito de forma desorganizada tornam o código confuso e causam um atraso considerável na manutenção devido ao maior tempo identificando cada parte do código.

A figura a seguir mostra a diferença gerada pela nova implementação no sistema.

<sup>9</sup> O compilador é um software que traduz o código escrito pelo desenvolvedor em um código que é interpretado pelo computador, em muitos casos é o que transforma um programa em algo executável.

**Figura 9 – Comparação do sistema com a implementação**

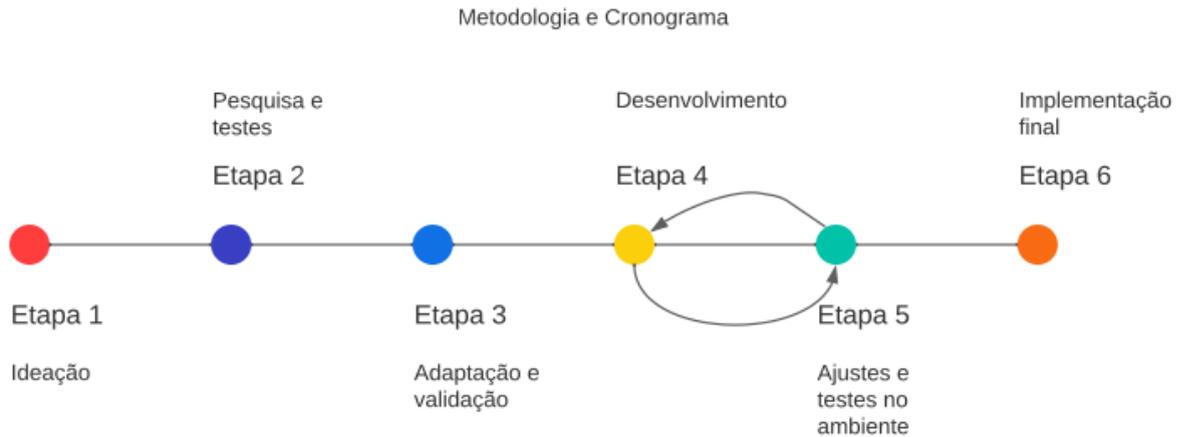


Fonte: Imagem gerada pelo autor (2023).

Como demonstrado na figura 9, a implementação foi ramificada a partir de uma função já existente no sistema, não afetando de forma alguma o funcionamento anterior mas abrindo uma nova linha de execuções para que a funcionalidade seja adicionada.

A seguir a figura 10 representa a metodologia seguida durante o desenvolvimento do projeto.

**Figura 10 – Metodologia de desenvolvimento**



Fonte: Imagem gerada pelo autor (2023).

O desenvolvimento seguiu as etapas da seguinte forma. A primeira etapa, de ideação, foi essencial para definir o objetivo principal do projeto e quais as formas de desenvolvimento possíveis, desde ferramentas, tecnologias, técnicas, etc. A segunda etapa definiu as tecnologias de forma mais concreta, a partir dela o desenvolvimento poderia ser voltado para a criação do código com base nas tecnologias definidas. A terceira etapa serviu como uma última validação das tecnologias, assim garantindo que não haveria problemas no andamento e, principalmente, nas etapas finais do projeto, assim garantindo que as tecnologias teriam as características necessárias para o projeto. A quarta e quinta etapa foram as etapas onde o código foi escrito e testado, sendo necessário algumas adaptações e testes durante o desenvolvimento para que o sistema executasse corretamente no ambiente de funcionamento. Por último, a sexta etapa é onde ocorreu a implementação final do sistema, a partir dela o sistema não recebe mais atualizações ou correções, ficando disponível para outros alunos com interesse para atualizar ou implantar algo novamente.

Por fim, foi possível entregar o sistema com a nova funcionalidade em pleno funcionamento e disponível para todos os alunos dos cursos de TI.

## 6 CONSIDERAÇÕES FINAIS

Por muitas vezes a manutenção de um código é vista como uma tarefa muito simples, sendo o maior foco dos desenvolvedores em funcionalidades novas para os

sistemas porém a manutenção de um sistema que já está em funcionamento a bastante tempo muitas vezes é mais importante do que a implementação de novas opções, a refatoração e atualização das tecnologias feitas no código já existentes são um exemplo perfeito disso pois acarretaram em grandes melhorias de desempenho geral no sistema, algo que reflete diretamente ao usuário final, algo que foi relatado pelos próprios alunos que utilizam a máquina é que foi possível notar que a mesma passou a apresentar um melhor funcionamento, tanto na leitura do código do aluno quando na liberação de seu crédito. Algo que à primeira vista pode não parecer importar pois o código funcionou por anos e não requer necessariamente manutenções para continuar funcionando porém se tornou um diferencial positivo para o usuário.

Em questão do desenvolvimento, as plataformas utilizadas foram de grande ajuda na leitura e criação do código-fonte, facilitando muito na refatoração e organização do código do projeto, identificadores coloridos e até mesmo a disposição do texto na forma em que ocorre no VS Code acabam se tornando uma ferramenta poderosa no desenvolvimento de códigos.

O maior empecilho para a entrega do projeto foi em especial o desenvolvimento do classificador, sendo necessário a busca de informações em diversos artigos, fóruns e documentações. Até chegar em um ponto onde o algoritmo teve um funcionamento aceitável levaram muitas horas de criações e testes.

O método classificatório escolhido teve uma compatibilidade muito boa com a proposta do projeto, não causou a lentidão do funcionamento do sistema e entregou um funcionamento aceitável da funcionalidade, mesmo sendo um modelo desenvolvido em 2001. Existem diversos outros algoritmos atualmente, então seria interessante em futuras implementações modificar a forma de classificação, melhorando o desempenho ainda mais e evitando possíveis falsos positivos, mas ainda sim é de grande importância pensar no ambiente onde o algoritmo funciona pois ele tem suas limitações.

Sobre as limitações do classificador, existem maneiras de enganar o classificador pois ele basicamente define se existe ou não o objeto na imagem capturada, não é levado em consideração se a imagem é uma impressão ou até mesmo uma imagem que o usuário está mostrando a partir de seu celular, então a funcionalidade se torna muito dependente do pressuposto de que os usuários terão ética em seu uso, dessa forma fica aberto um ponto muito interessante para um

trabalho futuro, na geração de algum algoritmo ou método de além de identificar o objeto também definir a qualidade da imagem lida, não permitindo o uso de métodos para burlar o sistema. Outro projeto que poderia ser implementado seria para ampliar o tipo de xícaras aceitas pelo classificador, no modelo atual são aceitas apenas xícaras mais padrões então copos e xícaras com outros formatos que estão se tornando comuns acabam não sendo aceitas pelo sistema.

## REFERÊNCIAS

Beck, Kent. Beedle, Mike. Bennekum, Arie van. Cockburn, Alistair. Cunningham, Ward. Fowler, Martin. Grenning, James. Highsmith, Jim. Hunt, Andrew. Jeffries, Ron. Kern, Jon. Marick, Brian. Martin, Robert C. Mellor, Steve. Schwaber, Ken. Sutherland, Jeff. Thomas, Dave. **Manifesto para Desenvolvimento Ágil de Software**. Disponível em: <<https://agilemanifesto.org/iso/ptbr/manifesto.html>>. Acesso em: 13 nov. 2023.

Cardeal, Zenilda (2011). **COPO PLÁSTICO: DESCARTE!**. Disponível em: <<https://www.ufmg.br/cienciaparatodos/wp-content/uploads/2011/11/25-copoplasticodescarte.pdf>>. Acesso em 13 nov. 2023.

HOSTINGER. **As 10 Linguagens de Programação Mais Usadas em 2022:** Aprimore suas Habilidades em Desenvolvimento Web. Disponível em: <<https://www.hostinger.com.br/tutoriais/linguagens-de-programacao-mais-usadas>>. Acesso em 15 de dez. 2022.

OECD (2022). **Global Plastics Outlook: Policy Scenarios to 2060**. Disponível em: <<https://www.oecd-ilibrary.org/sites/aa1edf33-en/index.html?itemId=/content/publication/aa1edf33-en>>. Acesso em 13 nov. 2023.

OpenCV. **About**. Disponível em: <<https://opencv.org/about/>>. Acesso em 13 nov. 2023.

PAIXAO, Elisiane Pelke. **APLICAÇÃO DO ALGORITMO VIOLA-JONES NA DETECÇÃO DE OBJETO**. Disponível em: <<https://bibliodigital.unijui.edu.br:8443/xmlui/bitstream/handle/123456789/5832/Elisiane%20Pelke%20Paix%C3%A3o.pdf?sequence=1&isAllowed=y>>. Acesso em: 13 nov. 2023.

RASPBERRY. **What is a Raspberry Pi?**. Disponível em: <<https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>>. Acesso em 13 nov. 2023.

RASCHKA, Sebastian; MIRJALILI, Vahid. **Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2**. 3. ed. Birmingham: Packt Publishing, 2019. 772 p. v. 1. ISBN 1789955750. Acesso em: 13 nov. 2023.

SOUSA OLIVEIRA, Kaique Rijkaard; SOUSA FARIAS, Ricardo; SILVA QUEIFER, Hoffmann; DOMINGUES BARROS FILGUEIRA, Jannayna; ARAÚJO FILGUEIRA, Geam Carlos. **Deteção e classificação de objetos utilizando computador de baixo custo Raspberry PI**. XXXVI SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES E PROCESSAMENTO DE SINAIS, CAMPINA GRANDE, PB, p. 1-5, 19 set. 2018. Disponível em: <<https://biblioteca.sbrt.org.br/articlefile/908.pdf>>. Acesso em: 20 out. 2023.

VIOLA, Paul; JONES, Michael. **Rapid object detection using a boosted cascade of simple features**. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, HI, USA, 14 dez. 2001.

DIGITAL.AI. **16th State of Agile Report**. Disponível em: <<https://digital.ai/resource-center/analyst-reports/state-of-agile-report/>>. Acesso em 14 de dez. 2022.

PYPI. **bing-image-downloader 1.1.2**: Disponível em: <<https://pypi.org/project/bing-image-downloader/>>. Acesso em 20 de jul. 2023.

Elhoseny, M. **Multi-object Detection and Tracking (MODT) Machine Learning Model for Real-Time Video Surveillance Systems**. Circuits Syst Signal Process 39, 611–630 (2020). <https://doi.org/10.1007/s00034-019-01234-7>