

SISTEMA DE SOFTWARE PARA DETECÇÃO, RASTREAMENTO E CONTAGEM DE PEIXES EM VÍDEOS DE PASSAGEM EM CANALETA

**SOFTWARE SYSTEM FOR DETECTION, TRACKING AND COUNTING OF FISH
IN PASSAGE-CHANNEL VIDEOS**

Engenharias • 09/08/2026

REGISTRO DOI: [10.70773/revistatopicos/786245352](https://doi.org/10.70773/revistatopicos/786245352)

Eduardo Mizael Clemente¹

Geraldo Andrade de Oliveira²

Leonardo Garcia Marques³

RESUMO

A contagem automatizada de peixes em vídeo exige integrar detecção, associação temporal e uma regra operacional de contagem. Este capítulo apresenta um pipeline com região de interesse (ROI) ajustável, detecção pela família YOLO, rastreamento com ByteTrack, gerenciamento de identificadores (IDs) e contagem por cruzamento de linha virtual. Na avaliação integrada em oito vídeos completos de *Astyanax lacustris*, a configuração principal contabilizou 2.404 peixes para uma referência de 2.400 eventos, atingiu erro médio absoluto (MAE) de 1,25 peixe por vídeo e obteve contagem exata em três vídeos. O detector alcançou precisão média média (mAP, do inglês mean Average Precision) de 0,9687 em mAP@0,50, e o pipeline apresentou média ponderada superior a 60 quadros por segundo em GPU dedicada. A configuração principal produziu 2.531 IDs, evidenciando fragmentação de trajetórias apesar do baixo erro final. Os 2.400 eventos em 8 min 10 s correspondem a aproximadamente 294 eventos por minuto, cadência compatível com o requisito operacional levantado. Como os vídeos avaliados também originaram parte dos frames usados no desenvolvimento do detector, esses resultados caracterizam o desempenho no domínio estudado, e não uma validação independente.

Palavras-chave: visão computacional; YOLO; ByteTrack; contagem de peixes; rastreamento de objetos; aquicultura.

ABSTRACT

Automated fish counting in video requires the integration of detection, temporal association, and an operational counting rule. This chapter presents a pipeline with an adjustable region of interest (ROI), YOLO-family detection, ByteTrack tracking, identifier (ID) management, and virtual line-crossing counting. In the integrated evaluation on eight complete *Astyanax lacustris* videos, the primary

configuration counted 2,404 fish against a reference of 2,400 events, achieved a Mean Absolute Error (MAE) of 1.25 fish per video, and produced exact counts in three videos. The detector reached a mean Average Precision (mAP) of 0.9687 at mAP@0.50, while the pipeline achieved a weighted mean above 60 frames per second on a dedicated GPU. The primary configuration produced 2,531 IDs, evidencing trajectory fragmentation despite the low final counting error. The 2,400 events recorded over 8 min 10 s correspond to approximately 294 events per minute, a rate compatible with the stated operational requirement. Because the evaluated videos also provided part of the frames used during detector development, these results characterize performance within the studied domain rather than independent validation.

Keywords: computer vision; YOLO; ByteTrack; fish counting; object tracking; aquaculture.

1. INTRODUÇÃO

A contagem de peixes na piscicultura é tradicionalmente realizada de forma manual e aproximada durante as etapas de manejo. No procedimento convencional, os peixes são retirados dos tanques com auxílio de peneiras ou redes, agrupados em pequenas porções e estimados visualmente pelos trabalhadores antes de serem transferidos ou comercializados. Embora seja uma prática simples e de baixo custo direto, ela apresenta alta susceptibilidade a erros de estimativa, exige esforço físico, consome tempo operacional relevante e impõe estresse aos animais em momentos críticos da produção. A automação dessa etapa por visão computacional e aprendizado profundo tem sido amplamente investigada na aquicultura como alternativa não invasiva e mais precisa (Barbedo, 2022; Saleh; Sheaves; Rahimi Azghadi, 2022). Iniciativas nessa direção

já foram conduzidas na piscicultura brasileira, inclusive com dados do próprio IF Goiano, demonstrando redução de tempo operacional e de estresse frente à contagem manual (Cabral; Martins, 2023).

Para mitigar essas limitações, a proposta desta dissertação desloca a contagem do interior dos tanques — onde a água turva, a variação de profundidade e a desorganização espacial inviabilizam a visão computacional — para uma etapa de passagem monitorada. O equipamento físico descrito no Capítulo II conduz os peixes por uma canaleta inclinada com lâmina de água e iluminação controlada. No entanto, a existência de um dispositivo físico de passagem não resolve a contagem por si só; é necessário um sistema de software capaz de processar o fluxo de vídeo e converter os movimentos dos peixes em eventos de contagem confiáveis.

O software apresentado neste capítulo combina detecção por modelos da família YOLO (Redmon et al., 2016), rastreamento temporal com o algoritmo ByteTrack (Zhang et al., 2022), controle de região de interesse (ROI), gerenciamento e filtragem de identificadores (IDs), contagem por cruzamento de linha virtual, interface web para operação remota e gravação auditável de dados. A contribuição central do componente computacional está tanto na integração dessas etapas quanto no ciclo de melhoria que conecta os erros observados em vídeo à seleção de novos frames, à curadoria dos dados, ao treinamento e à reavaliação da contagem. A tarefa não se limita a localizar peixes em quadros isolados nem a comparar modelos como finalidade autônoma, mas busca obter sucessivamente uma configuração integrada mais adequada ao objetivo operacional. Abordagens que integram detecção e rastreamento para contagem de peixes em vídeo têm sido propostas recentemente (Kandimalla et al., 2022; Lin et al., 2024;

Dawkins et al., 2024), o que reforça a pertinência do pipeline aqui adotado. Mais do que um detector isolado, o que essa literatura tem convergido a adotar é a combinação de detecção, rastreamento por trilhas e regra de contagem por linha; este trabalho instancia esse arranjo e o submete ao ciclo iterativo de melhoria descrito adiante.

Este capítulo descreve o desenvolvimento do componente computacional do sistema, desde a formação e evolução dos conjuntos de dados até a integração entre detecção, rastreamento e contagem. A avaliação foi organizada em três níveis complementares: desempenho do detector em imagens anotadas, erro de contagem em vídeos completos e estabilidade dos identificadores gerados pelo rastreador. Essa separação é necessária porque uma alta capacidade de localizar peixes em quadros isolados não garante, por si só, que cada indivíduo seja contado uma única vez ao longo do vídeo.

1.1. Problemas Práticos e Papéis dos Componentes

A Tabela 1 resume a relação entre os problemas práticos observados na contagem, o papel desempenhado pelo equipamento físico de captura e a resposta computacional implementada no software.

Tabela 1. Relação entre problemas práticos do manejo, papel do hardware e papel do software

Problema observado	Papel do hardware (Capítulo II)	Papel do software (Capítulo III)
Contagem manual aproximada	Conduzir peixes em passagem monitorada por canaleta	Registrar eventos de passagem por cruzamento de linha

Turbidez da água do tanque	Usar fluxo de água limpa na canaleta	Lidar com ruído visual residual e reflexos secundários
Aglomerção e sobreposição	Espalhar os peixes pela largura da canaleta	Rastrear trajetórias e manter consistência de IDs
Oclusão entre indivíduos	Fornecer lâmina de água rasa para evitar empilhamento	Manter histórico temporal com ByteTrack e tolerar falhas de detecção
Reflexos na superfície	Utilizar PVC branco e iluminação por LED superior	Definir ROI interna, treinar com imagens negativas e filtrar bordas
Velocidade de passagem	Ajustar inclinação da canaleta e vazão da bomba	Manter alta taxa de quadros, expressa em quadros por segundo (FPS), e linha virtual com margem temporal
Instabilidade de enquadramento	Fixar estrutura de madeira e suporte de câmera	Permitir configuração visual de ROI e linha virtual

Fonte: elaborada pelo autor (2026).

2. CONTEXTO E REQUISITOS DO SOFTWARE

2.1. Requisitos Funcionais

Os requisitos funcionais do software derivam do objetivo operacional de registrar a passagem dos peixes pela canaleta com o menor erro possível. Esses requisitos foram divididos em três grupos funcionais: núcleo da contagem, apoio ao desenvolvimento experimental e operação/auditoria.

O núcleo da contagem reúne aquisição de vídeo, detecção, rastreamento, cruzamento de linha e controle básico da operação. O apoio experimental compreende seleção de frames, pré-anotação, divisão rastreável dos dados e orquestração dos treinamentos, enquanto a operação e a auditoria abrangem a configuração da ROI e da linha de contagem e a persistência dos vídeos e metadados. O detalhamento de cada requisito e de seu papel na solução foi deslocado para a Tabela 2, no Apêndice A.

2.2. Requisitos Não Funcionais e Ambiente de Execução

Os requisitos não funcionais especificam os critérios de qualidade, robustez, reprodutibilidade e desempenho exigidos pelo sistema.

Os critérios adotados abrangem acurácia de contagem, robustez visual, eficiência computacional, reprodutibilidade, auditabilidade e configurabilidade. Eles são avaliados, conforme o caso, pelo erro de contagem nos vídeos completos, pelo desempenho computacional, pela preservação de configurações e manifestos e pela geração de vídeos e metadados de auditoria. A especificação detalhada encontra-se na Tabela 3, no Apêndice A.

Um requisito operacional importante, levantado junto ao produtor e derivado dos ensaios na empresa parceira, é a capacidade de processar lotes com **aproximadamente 300 indivíduos por minuto**. Essa taxa corresponde ao ritmo habitual de liberação dos peixes durante a contagem para comercialização. Ela implica que o pipeline precisa manter rastreamento e contagem contínuos em alta cadência de passagens, sem que o custo computacional cause perda de quadros ou acúmulo de latência que comprometa o

registro de cruzamentos. Os resultados de FPS descritos na Seção de resultados são interpretados em relação a esse requisito.

A avaliação final do software nos oito vídeos experimentais foi executada em um computador equipado com processador AMD Ryzen 5 7600X (6 núcleos, 12 threads), 32 GB de memória RAM e placa de vídeo NVIDIA GeForce RTX 3070 com 8 GB de memória dedicada. O sistema operacional utilizado foi o Windows 11 Pro de 64 bits (build 26200). O ambiente computacional empregou Python 3.12.13, Ultralytics 8.4.37 (Ultralytics, 2025) e PyTorch 2.8.0 com suporte a CUDA 12.6, operando no dispositivo cuda:0.

2.3. Escopo e Delimitação Computacional

O escopo deste trabalho concentra-se no processamento de vídeo, detecção, rastreamento temporal, contagem por linha virtual e avaliação integrada do pipeline. Ficam explicitamente fora do escopo funcional:

- A estimativa de peso, biomassa ou comprimento dos peixes a partir das caixas delimitadoras (hipótese descartada no escopo atual devido à variação de pose e flexão dos animais);
- A validação operacional de implantação em dispositivo de borda (Raspberry Pi), cujos testes preliminares demonstraram taxa de quadros insatisfatória e são tratados como trabalho futuro;
- A classificação automática de espécies ou a calibração fotométrica em tempo real.

3. ARQUITETURA DO PIPELINE DE SOFTWARE

3.1. Visão Geral da Arquitetura

O software foi construído segundo uma arquitetura modular em camadas, desacoplando a leitura de dados, o processamento computacional, a interface de usuário e a persistência para auditoria. O fluxo de dados segue a sequência:

O fluxo inicia-se na leitura do quadro (VideoSource, baseada na biblioteca OpenCV (Bradski, 2000)), passa pelo recorte da Região de Interesse (ROIManager), segue para a inferência pelo modelo YOLO (YOLODetector), sofre associação temporal pelo algoritmo ByteTrack (ByteTracker), é filtrado pelo gerenciador de identificadores (IDManager), tem seus cruzamentos checados pelo contador de linha virtual (LineCounter) e gera o resultado consolidado (FrameResult) que alimenta o streaming visual da interface web (implementada com FastAPI (Ramírez, 2018)) e a gravação de metadados.

A Figura 1 ilustra um quadro processado pelo pipeline, com detecções, identificadores e a linha de contagem; a Figura 2 apresenta a interface web de operação.

Figura 1. Quadro 700 do vídeo 2026-07-17 15-26-33.mp4, processado com o modelo promovido da execução d13f3a, mostrando caixas de detecção, confiança, identificadores de rastreamento, região de interesse e linha virtual de cruzamento. Fonte: elaborado pelo autor (2026).



Fonte: elaborada pelo autor (2026).

Figura 2. Interface web de operação do sistema, com o vídeo processado, as detecções e os identificadores sobrepostos, a contagem acumulada e os controles operacionais. Fonte: elaborado pelo autor (2026).



Fonte: elaborada pelo autor (2026).

3.2. Síntese dos Módulos do Pipeline

A Tabela 4 detalha a responsabilidade de cada módulo componente do sistema computacional.

O pipeline é composto por fonte de vídeo, gerenciador de ROI, detector de objetos, rastreador temporal, gerenciador de IDs, contador por linha, interface web, gravador de auditoria e camada de configuração. A responsabilidade e o elemento de implementação associados a cada módulo estão documentados na Tabela 4, no Apêndice A.

3.3. Distinção Entre Detector, Rastreador e Contador

Uma definição operacional clara é fundamental para a análise dos resultados. Em cada quadro, o **detector** produz caixas delimitadoras, classes e valores de confiança. O **rastreador** associa essas detecções ao longo do tempo e atribui um identificador (ID) a cada trajetória. A **contagem** é realizada somente quando uma trajetória válida cruza a linha virtual dentro da região de interesse e atende aos critérios temporais definidos. Portanto, o número de detecções, o número de IDs únicos e o número de cruzamentos contabilizados são grandezas essencialmente diferentes.

Um mesmo peixe pode receber mais de um ID se sua trajetória for interrompida por oclusão, saída parcial do quadro ou perda momentânea da detecção. Por outro lado, nem todo ID criado pelo rastreador atravessa a linha de contagem. Isso explica por que o sistema pode apresentar contagem final muito próxima da referência real e, simultaneamente, registrar um número de IDs únicos superior à quantidade física de animais.

4. DADOS, ANOTAÇÃO E TREINAMENTO DOS MODELOS

4.1. Evolução do Fluxo Experimental e Dataset Studio

Nas primeiras iterações do projeto, as rotinas de extração de frames, preparação de tarefas para anotação e treinamento dos detectores estavam integradas ao código do sistema de contagem. Essa organização permitiu executar rapidamente os experimentos iniciais, mas passou a dificultar a identificação da origem de cada imagem, da revisão de anotações utilizada, da divisão aplicada ao conjunto de dados e do checkpoint resultante à medida que novos vídeos e treinamentos eram incorporados.

Como consequência, a rastreabilidade não é uniforme em toda a trajetória experimental. Para as bases e os treinamentos anteriores à autonomização da ferramenta, foi possível reconstruir tamanhos, composição geral e linhagem dos pesos a partir dos artefatos preservados, mas não existe, em todos os casos históricos, um manifesto padronizado que relacione cada frame ao vídeo de origem e à divisão aplicada. Nas iterações recentes, especialmente na campanha de 17 de julho de 2026 e no treinamento executado pelo Dataset Studio, essa relação passou a ser registrada diretamente pelos manifestos da ferramenta. Assim, a rastreabilidade completa deve ser atribuída às versões recentes, sem ser retroativamente presumida para os experimentos anteriores.

A necessidade de repetir o ciclo experimental sem perder a relação entre suas etapas motivou o desenvolvimento do Dataset Studio como ferramenta autônoma. A separação não alterou o objetivo do estudo: a ferramenta passou a organizar a preparação dos dados e o treinamento, enquanto o sistema de contagem permaneceu

responsável pela inferência em vídeo, pelo rastreamento e pela contabilização dos cruzamentos. Essa delimitação permite aperfeiçoar o detector em sucessivas iterações sem acoplar a gestão do conjunto de dados à aplicação operacional.

No Dataset Studio, cada campanha de aquisição é registrada como uma origem que relaciona os vídeos, seus identificadores, as unidades de captura e a configuração usada para extrair os frames. A extração pode empregar amostragem uniforme ou seleção assistida por um detector, preservando um manifesto dos quadros gerados. A anotação é realizada com integração ao Label Studio, incluindo importação controlada das tarefas e recuperação das revisões concluídas. O emprego de pré-anotação assistida por modelos de detecção e rastreamento para reduzir o esforço manual de rotulagem é estratégia consolidada na literatura (Vijiyakumar; Govindasamy; Akila, 2024), aqui incorporada ao fluxo de extração e revisão. A partir dessas revisões, uma versão materializada reúne imagens e rótulos, aplica a divisão por vídeo ou unidade de captura e registra os arquivos resultantes em manifesto.

Essa divisão agrupada é particularmente importante no presente estudo. Frames visualmente próximos de um mesmo vídeo não devem ser distribuídos entre treinamento, validação e teste de imagens, pois essa distribuição produziria conjuntos artificialmente semelhantes. Ao vincular cada frame à sua mídia de origem, o fluxo permite manter as unidades de captura inteiras em uma única divisão. Essa propriedade evita sobreposição de arquivos e reduz o vazamento entre os conjuntos de imagens, embora não torne independente a avaliação posterior nos vídeos completos, conforme discutido na Seção de limitações.

A Tabela 5 resume as etapas organizadas pela ferramenta e as evidências persistidas para permitir auditoria e repetição do processo.

As etapas de origem, extração, anotação, versionamento, treinamento e avaliação formam uma cadeia auditável, na qual cada operação produz evidências de proveniência. A correspondência detalhada entre etapa, operação e artefato preservado está na Tabela 5, no Apêndice A.

O Dataset Studio não é avaliado nesta dissertação como produto independente quanto à usabilidade, ao desempenho de sua interface ou à comparação com outras plataformas de gestão de dados. Seu papel no trabalho é metodológico: tornar explícito e auditável o caminho entre as capturas, os frames anotados, a versão empregada no treinamento e o checkpoint avaliado. A promoção desse checkpoint para o sistema de contagem permanece uma etapa deliberada, pois somente modelos selecionados após a avaliação devem integrar o diretório de modelos disponíveis para inferência.

4.2. Evolução dos Conjuntos de Dados

O conjunto de dados foi ampliado de forma iterativa à medida que novos modos de falha eram identificados. A Tabela 2 apresenta a evolução das bases de dados preparadas ao longo do projeto.

Tabela 2. Evolução dos conjuntos de dados anotados para treinamento do detector

Etapas / Versão	Imagens	Caixas	Neg.	Característica principal
----------------------------	----------------	---------------	-------------	-------------------------------------

Base inicial	128	222	0	Primeira adaptação à canaleta metálica
Base intermediária	605	901	228	Primeira inclusão de quadros vazios
Base ampliada	1.599	3.323	473	Diversificação de densidades e fundos
PVC parcial	581	2.374	189	Liberação parcial da canaleta de PVC
PVC completa válida	776	5.649	191	Base de referência utilizada nos treinos recentes

Fonte: elaborada pelo autor (2026).

A ampliação da base não consistiu apenas em aumentar o número total de arquivos. Foram incorporados novos cenários, variações de densidade, entradas e saídas parciais pelas bordas, reflexos, ondulações na água, detritos e quadros vazios.

4.3. Composição e Divisão da Liberação Completa de PVC

A liberação completa do dataset de PVC contém 776 imagens e 5.649 caixas delimitadoras anotadas. A Tabela 3 detalha a distribuição das imagens entre os conjuntos de treinamento, validação, teste normal e teste de estresse.

Tabela 3. Composição e divisão do dataset de PVC completo (776 imagens)

Divisão	Total de imagens	Com peixes	Imagens negativas	% Negativas
Treinamento	466	363	103	22,1%
Validação	104	83	21	20,2%
Teste normal	153	102	51	33,3%
Teste de estresse	53	37	16	30,2%
Total	776	585	191	24,6%

Fonte: elaborada pelo autor (2026).

4.4. Imagens Negativas e Representação do Fundo

Na anotação de objetos, a atenção tende a se concentrar nas imagens que contêm a classe de interesse. Para o detector, entretanto, também é necessário representar situações nas quais nenhum peixe está presente. Uma imagem negativa é um quadro válido sem caixas anotadas; ela informa ao modelo que elementos visuais presentes na cena não devem ser classificados como peixes.

No ambiente estudado, esses quadros contêm ondulações da água, pequenos detritos, reflexos luminosos, sombras, bordas da canaleta e variações de exposição. A inclusão de imagens negativas buscou reduzir respostas indevidas a esses elementos. Na liberação completa da canaleta de PVC, 191 das 776 imagens eram negativas, correspondendo a 24,6% do conjunto.

As observações qualitativas indicam que a representação do fundo foi relevante para estabilizar o detector, reduzindo falsos positivos na ausência de peixes. Contudo, como tamanho da base, composição e

hiperparâmetros mudaram conjuntamente, a contribuição causal exata das imagens negativas deve ser tratada como hipótese metodologicamente fundamentada, e não como ganho percentual isolado.

4.5. Linhagem e Treinamento dos Modelos

Os treinamentos foram organizados como uma linhagem contínua na qual pesos adaptados serviram de inicialização para etapas posteriores. Os primeiros experimentos utilizaram pesos genéricos COCO da arquitetura YOLO26n. Em seguida, o modelo foi ajustado nos conjuntos intermediários e deu origem ao checkpoint `yolo26n_seed_43.pt`.

Os treinamentos da liberação completa de 776 imagens (50 e 150 épocas), bem como o treinamento mais recente (`t_20260724T140336_d13f3a`), foram inicializados a partir desse checkpoint já adaptado ao domínio visual da canaleta, e não a partir de pesos genéricos. Essa linhagem garante que os modelos incorporem o aprendizado prévio do fundo e da geometria da passagem.

O treinamento recente `t_20260724T140336_d13f3a` foi executado no fluxo Dataset Studio com as 776 imagens (incorporando as 53 imagens do conjunto de estresse ao treino), inicializado com `yolo26n_seed_43.pt`, prevendo 150 épocas e atingindo seu melhor ponto na época 122. Após a avaliação integrada, o checkpoint foi promovido no Dataset Studio, que registrou seu alias e criou um bundle imutável de implantação. Em seguida, o arquivo promovido foi copiado para o caminho canônico `models/yolo26n_canaleta_pvc_776_d13f3a_best.pt` do sistema de

contagem.

○

hash

SHA-256

F845258882D05F3A8B0D2434531B09024D5B5F4956ADE916AAD117

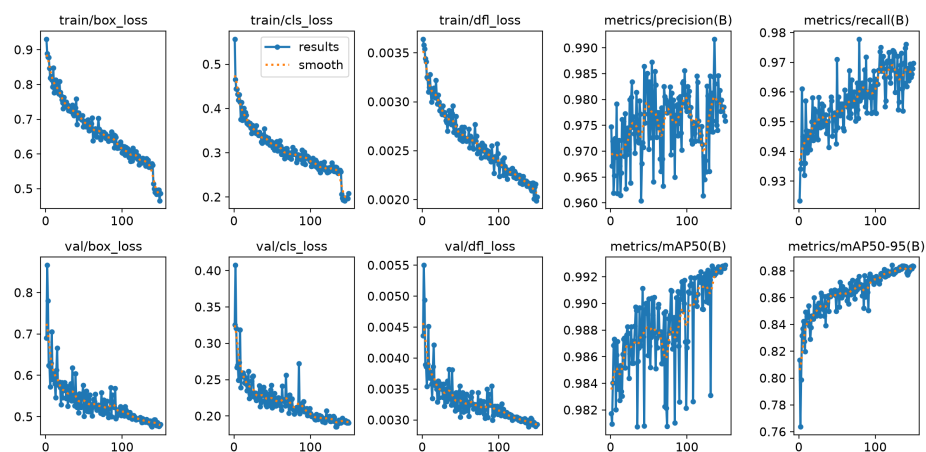
8B4F684371 vincula as cópias ao artefato original do treinamento.

5. RESULTADOS E DISCUSSÃO

5.1. Desempenho do Detector por Imagem

A Tabela 8 apresenta as métricas obtidas pelos modelos de detecção nos conjuntos de validação e teste de imagens. A Figura 3 apresenta a evolução das métricas ao longo do treinamento do detector.

Figura 3. Evolução das perdas de treinamento e validação e das métricas de precisão, revocação e mAP ao longo das 150 épocas da execução t_20260724T140336_d13f3a. Fonte: artefato results.png gerado pela Ultralytics durante o treinamento (2026).



Fonte: elaborada pelo autor (2026).

Tabela 4. Métricas de desempenho dos modelos de detecção

Modelo / Treiname nto	Conjunto	Precisão	Revocaçã o	mAP@0,5 0	mAP@ 0:0,95
-----------------------------	----------	----------	---------------	--------------	----------------

PVC completa - 50 épocas	Validação	0,9781	0,9386	0,9887	0,8793
PVC completa - 50 épocas	Teste normal	-	-		0,8487

⚠ Esta tabela possui muitas colunas e foi cortada para impressão. Para visualizá-la completa, acesse o artigo original em: <https://revistatopicos.com.br/artigos/sistema-de-software-para-deteccao-rastreamento-e-contagem-de-peixes-em-videos-de-passagem-em-canaleta?noblockage>

Fonte: elaborada pelo autor (2026).

Os dados confirmam que os detectores treinados alcançaram elevada capacidade de localização espacial dos peixes em quadros isolados, com mAP@0,50 superior a 0,96 e mAP@0,50:0,95 acima de 0,85 no conjunto de teste.

5.2. Resultados da Contagem nos Oito Vídeos Completos

A avaliação principal do sistema integrado foi realizada na campanha de oito vídeos gravados durante os testes na empresa parceira Lambari Itumbiara. Cada vídeo possui a referência real de 300 peixes liberados via peneira, acumulando um alvo total de 2.400 eventos de passagem. Como o mesmo lote de 300 peixes foi reutilizado nas oito sessões, o total acumulado mede o comportamento integrado do sistema em 2.400 passagens, mas não em 2.400 indivíduos distintos.

A configuração principal do modelo promovido está preservada em capture_20260724_d13f3a_count.yaml: resolução de inferência de 960 pixels, ROI interna lateral (x=200 a x=1720), linha virtual em

x=1350 e dispositivo cuda:0. A Tabela 4 consolida os parâmetros de detecção, rastreamento, contagem e região de interesse necessários à reprodução. No registro histórico das demais configurações variaram os pesos, o limiar de confiança (conf) e o limiar alto de rastreamento (track_high_thresh); essas variações são indicadas explicitamente.

Tabela 5. Parâmetros do pipeline empregados na avaliação de contagem

Etapas	Parâmetro	Valor
Detecção	Resolução de inferência (imgsz)	960 px
	Limiar de confiança (conf)	0,10 (principal); 0,22 (comparado)
	Limiar de IoU	0,50
	Máximo de detecções por quadro	100
	Meia precisão (half)	sim
Rastreamento	Algoritmo	ByteTrack
(ByteTrack)	track_high_thresh	0,50 (principal); 0,35 (comparado)
	new_track_thresh	0,70
	track_low_thresh	0,10
	match_thresh	0,85
	track_buffer	120 quadros
Gerência de IDs	Confirmação temporal (n_init)	2 quadros

	Paciência de perda (lost_patience)	120 quadros
	Histórico de posições	30 quadros
Contagem	Linha virtual (vertical)	x=1350
	Intervalo de proteção (cross_cooldown)	25 quadros
Região de interesse	Modo	máscara
	Faixa horizontal	x=200 a x=1720

Fonte: elaborada pelo autor (2026).

A Tabela 6 apresenta o resultado comparativo dos modelos avaliados na campanha dos oito vídeos completos.

Tabela 6. Comparação de desempenho dos modelos nos oito vídeos completos (alvo acumulado: 2.400 peixes; 300 por vídeo)

Modelo / Experimento	conf	c@Contagem em total	c@Saldo líquido	c@MAE contagem	c@IDs totais
Último treino (latest_run _conf10)	0,10	2.404	+4	1,25	2.531
Último treino (latest_run	0,22	2.404	+4	1,50	2.542

⚠ Esta tabela possui muitas colunas e foi cortada para impressão. Para visualizá-la completa, acesse o artigo original em: <https://revistatopicos.com.br/artigos/sistema->

Fonte: elaborada pelo autor (2026).

A Tabela 7 apresenta o detalhamento de contagem e IDs únicos por vídeo para a configuração principal do último treino (latest_run_conf10).

Tabela 7. Detalhamento de contagem e IDs por vídeo no modelo mais recente (latest_run_conf10)

Vídeo	Alvo real	Contagem obtida	Erro absoluto	IDs únicos
2026-07-17 15-26-33.mp4	300	302	+2	315
2026-07-17 15-31-41.mp4	300	300	0	325
2026-07-17 15-33-35.mp4	300	300	0	309
2026-07-17 15-36-50.mp4	300	299	1	312
2026-07-17 15-39-23.mp4	300	304	+4	317
2026-07-17 15-42-47.mp4	300	301	+1	311
2026-07-17 15-50-35.mp4	300	300	0	320
2026-07-17 15-56-45.mp4	300	298	2	322
Total / Média	2.400	2.404	MAE: 1,25	Total: 2.531

Fonte: elaborada pelo autor (2026).

No domínio experimental avaliado, a configuração integrada latest_run_conf10 apresentou resultado promissor: erro médio absoluto (MAE) de 1,25 peixe por vídeo (inferior a 0,5% em relação ao lote de 300 peixes) e contagem exata em três dos oito vídeos. Esse patamar ficou abaixo das metas operacionais reportadas na literatura para contagem automatizada em ambiente produtivo, próximas de 5% (Bukas et al., 2024). A comparação serve como registro da trajetória iterativa de desenvolvimento e da configuração obtida, não como ensaio estatístico independente de superioridade entre modelos.

Considerando a duração acumulada de 8 min 10 s documentada para os oito vídeos e a referência de 2.400 eventos de passagem, a campanha apresentou cadência média de aproximadamente 294 eventos por minuto. Esse valor é compatível com o requisito operacional de aproximadamente 300 indivíduos por minuto. Em paralelo, as médias ponderadas de processamento entre 63,40 e 71,15 FPS igualaram ou superaram a taxa de aquisição de 60 FPS, sem indicar formação de fila computacional no ensaio de processamento. Essas evidências não demonstram, isoladamente, ausência de perda de quadros durante a gravação original, pois não foi produzido um registro específico de descarte de frames na captura.latest_run_conf10, correspondendo a um MAE de IDs de 16,38 por vídeo).

5.3. Estabilidade do Rastreamento e Fragmentação de IDs

Embora o erro de contagem final tenha sido reduzido a níveis mínimos, a Tabela 10 revela uma discrepância importante entre

contagem e rastreamento: todos os modelos geraram mais IDs únicos do que o número real de peixes (excesso de 131 IDs no modelo latest_run_conf10, correspondendo a um MAE de IDs de 16,38 por vídeo).

As métricas formais de rastreamento HOTA, MOTA e IDF1 não foram calculadas neste estudo, uma vez que o conjunto de dados não dispõe de anotações temporais com a identidade individual dos peixes ao longo dos quadros. A quantidade excedente de identificadores gerados pelo rastreador foi utilizada apenas como indicador operacional de fragmentação das trajetórias, não devendo ser interpretada como substituta dessas métricas.

Essa diferença é compatível com a **fragmentação de trajetórias**. Quando um peixe passa por uma região de aglomeração intensa, sofre oclusão temporária por outro indivíduo ou entra parcialmente pela borda da imagem, o rastreador pode perder a associação da caixa e atribuir um novo ID ao mesmo peixe quando ele reaparece. Métodos de rastreamento que exploram a confiança das detecções ao longo de toda a trajetória têm sido propostos para reduzir essas trocas de identidade (Mandel et al., 2023). Em cenários de aglomeração extrema, nos quais a detecção por caixas delimitadoras perde eficácia, a literatura oferece rotas complementares que este trabalho não adotou: detectores adaptativos à oclusão (Li et al., 2022) e a regressão por mapas de densidade (Li et al., 2024), esta voltada à estimativa de fluxo quando a contagem individual se torna inviável. A estratégia aqui foi conter a fragmentação com a ROI interna, a linha virtual e o intervalo de proteção temporal, preservando a contagem individual.

A proximidade entre a contagem obtida (2.404) e a referência (2.400), apesar do excesso de IDs (2.531), é compatível com o papel conjunto da ROI interna, da linha virtual e do intervalo de proteção temporal. Parte dos IDs fragmentados pode ter sido criada antes ou depois da linha ou não ter atendido aos critérios vetoriais completos de cruzamento, sem que cada novo ID fosse convertido em contagem duplicada. Como esses componentes não foram avaliados em uma ablação específica mantendo constantes os demais parâmetros, o resultado não permite atribuir causalmente a nenhum deles, de forma isolada, a compensação da fragmentação.

5.4. Efeito das Bordas e da ROI Interna

As extremidades laterais do enquadramento concentram detecções parciais de peixes que estão entrando ou saindo do campo de visão da câmera. Nessas regiões, apenas uma fração do corpo do animal fica visível, gerando caixas delimitadoras instáveis e quebras de rastreamento.

A adoção da ROI interna ($x=200$ a $x=1720$) excluiu essas faixas laterais da análise de inferência e foi associada a menor surgimento de IDs espúrios. No modelo parcial, a configuração interna coincidiu com a redução do MAE de IDs de 15,25 para 10,00 e do MAE de contagem de 1,63 para 1,25. Como essa comparação também envolveu ajustes na resolução e nos limiares de confiança, o ganho deve ser compreendido como resultado do conjunto de otimizações da configuração, sem atribuição causal isolada à ROI.

5.5. Problemas Visuais Residuais: Aglomeração, Oclusão e Respingos

Apesar dos avanços obtidos no controle físico e computacional, a análise qualitativa das sessões revelou limitações visuais remanescentes. A Figura 4 documenta uma passagem com alta proximidade entre indivíduos no vídeo que apresentou o maior erro absoluto de contagem. O quadro ilustra a dificuldade visual, mas não permite identificar isoladamente qual evento ou componente do pipeline produziu a diferença final:

Figura 4. Quadro 3.000 do vídeo 2026-07-17 15-39-23.mp4, com aglomeração, sobreposição de caixas e oclusão parcial entre peixes. O vídeo apresentou contagem de 304 peixes para uma referência de 300; entretanto, o quadro isolado apenas documenta uma condição de maior dificuldade e não identifica causalmente o evento responsável pela diferença final. Fonte: elaborado pelo autor a partir dos vídeos experimentais (2026).



Fonte: elaborada pelo autor (2026).

1. **Aglomerção e oclusão:** Em momentos de vazamento rápido da peneira, peixes deslizam sobrepostos. A oclusão pode ocultar contornos corporais, dificultar a separação das caixas e interromper temporariamente trajetórias durante o cruzamento da linha.

2. **Respingos sobre a lente:** A movimentação dos peixes na lâmina de água pode projetar gotas em direção à câmera, criando manchas circulares de refração que persistem até a evaporação ou limpeza da lente.

3. **Reflexos e detritos:** Pequenas bolhas de ar e ondulações formadas pela bomba de recirculação geram reflexos especulares dinâmicos na água, mitigados parcialmente pelas 191 imagens negativas do dataset.

5.6. Desempenho Computacional em Tempo Real

A Tabela 12 apresenta os resultados de desempenho computacional medidos na máquina de desenvolvimento durante o processamento dos vídeos na resolução de 960 pixels.

Tabela 8. Desempenho computacional dos modelos no ambiente de desenvolvimento

Modelo / Configuração	Resolução	Dispositivo	Faixa de FPS	FPS médio ponderado
PVC completa – 50 épocas	960 px	cuda:0	62,17 – 66,63	64,57
PVC completa – 150 épocas	960 px	cuda:0	65,17 – 68,10	66,75
Último treino (d13f3a conf=0,10)	960 px	cuda:0	56,50 – 67,40	63,40
Último treino (d13f3a conf=0,22)	960 px	cuda:0	66,00 – 73,10	71,15

Fonte: elaborada pelo autor (2026).

Em hardware dotado de GPU de alto desempenho (RTX 3070), as médias ponderadas medidas ficaram acima de 60 FPS. A menor taxa instantânea registrada foi 56,50 FPS; por isso, os resultados sustentam a viabilidade média de processamento na cadência da câmera, mas não uma garantia de que cada instante permaneceu acima de 60 FPS.

Por outro lado, ensaios preliminares em dispositivos de borda Raspberry Pi 3B+ e Raspberry Pi 4 indicaram desempenho insatisfatório para a configuração atual, com taxas de quadros muito abaixo das obtidas em GPU e insuficientes para manter a integridade do rastreamento ByteTrack. Esses ensaios tiveram caráter exploratório e não foram registrados em relatório sistemático de benchmark; um levantamento controlado em dispositivo de borda permanece como trabalho futuro. Assim, a implantação em borda permanece como objetivo futuro condicionado à otimização TensorRT/ONNX ou ao uso de aceleradores dedicados (como NVIDIA Jetson). Nessa direção, arquiteturas leves projetadas para dispositivos de borda têm sido propostas para detecção em ambientes aquícolas (Jiang et al., 2025; Jiang et al., 2024).

6. LIMITAÇÕES DO ESTUDO

Este estudo apresenta seis limitações principais que devem orientar a interpretação dos resultados:

1. **Reutilização do lote animal:** Os oito vídeos foram gravados reutilizando o mesmo lote de 300 peixes de *Astyanax lacustris*. O total de 2.400 eventos avalia a repetibilidade da contagem

mecânica, mas não a variabilidade morfológica de 2.400 indivíduos distintos.

2. **Dependência entre desenvolvimento e avaliação:** Os conjuntos de treinamento, validação e teste de imagens foram formados com divisões estáveis e sem sobreposição de arquivos entre esses splits. Entretanto, os oito vídeos completos usados na contagem também originaram parte dos frames empregados no desenvolvimento do detector. Como a avaliação percorre os vídeos completos, ela inclui esses instantes e compartilha a mesma canaleta, o mesmo equipamento e o mesmo lote. O MAE de 1,25 caracteriza, portanto, o desempenho in-domain da configuração obtida e não uma validação independente nem uma estimativa de generalização para outros lotes, espécies ou ambientes.

3. **Ausência de ablação isolada de negativos:** A inclusão de 24,6% de imagens negativas foi uma das mudanças do processo iterativo, mas não foi realizada uma ablação estrita mantendo todos os demais hiperparâmetros constantes para isolar seu ganho percentual exato.

4. **Ausência de ground truth temporal:** Não foram produzidas anotações persistentes da identidade de cada peixe quadro a quadro; portanto, HOTA, MOTA e IDF1 não puderam ser calculadas. O excesso de IDs é apenas um indicador operacional de fragmentação.

5. **Fragmentação de trajetórias:** O surgimento de 131 IDs excedentes na configuração principal indica que o

rastreamento ainda sofre perdas temporárias de identidade em aglomerações e oclusões.

6. **Restrição ao ambiente GPU:** A viabilidade computacional foi comprovada exclusivamente em desktop com GPU dedicada, sem validação funcional em dispositivos de borda de baixo custo.

7. CONCLUSÃO

Este capítulo apresentou a concepção, implementação e avaliação do componente de software para detecção, rastreamento e contagem automatizada de peixes em vídeos de passagem por canaleta. A solução integrou a família YOLO de detectores, o algoritmo ByteTrack para associação temporal, ROI interna ajustável, filtragem de IDs e regra vetorial de cruzamento de linha virtual.

Na avaliação in-domain sobre oito vídeos completos (2.400 eventos de passagem), a configuração integrada latest_run_conf10 alcançou erro médio absoluto (MAE) de contagem de 1,25 peixe por vídeo, contabilizando 2.404 peixes no total acumulado (saldo de +4 em relação à referência). O detector atingiu mAP@0,50 de 0,9687 e o pipeline operou, em média, acima de 60 quadros por segundo em GPU dedicada. A cadência média de aproximadamente 294 eventos por minuto foi compatível com o requisito operacional de aproximadamente 300 indivíduos por minuto, sem que esse cálculo seja interpretado como comprovação de ausência de perda de quadros na captura. Esses indicadores são específicos da campanha e devem ser interpretados como demonstração do processo no cenário estudado.

Os resultados mostram que, nesta configuração e nestes vídeos, a fragmentação de trajetórias não foi acompanhada por aumento proporcional da contagem final. Esse comportamento é compatível com a atuação conjunta da ROI interna e das regras de cruzamento com intervalo de proteção, mas não permite estabelecer causalidade sem uma ablação controlada.

A contribuição metodológica do capítulo é o ciclo reproduzível de melhoria que conecta aquisição controlada, extração e curadoria de frames, divisão rastreável dos dados, treinamento, execução em vídeos completos e análise dos erros. A aplicação a *Astyanax lacustris* demonstra que esse processo pode produzir resultados promissores e ser iterado para aperfeiçoar a contagem; sua aplicação a outros domínios requer dados representativos e nova avaliação.

Como trabalhos futuros, recomenda-se:

- Avaliar o pipeline em vídeos gravados com lotes inéditos de peixes e diferentes espécies/tamanhos;
- Otimizar e quantizar o modelo (via TensorRT ou OpenVINO) para execução em hardware de borda;
- Desenvolver mecanismos de separação física ou pré-filtragem para mitigar a aglomeração intensa na canaleta.

REFERÊNCIAS BIBLIOGRÁFICAS

BARBEDO, Jayme Garcia Arnal. A Review on the Use of Computer Vision and Artificial Intelligence for Fish Recognition, Monitoring, and Management. *Fishes*, v. 7, n. 6, p. 335, 2022. DOI: 10.3390/fishes7060335.

BRADSKI, Gary. The OpenCV Library. Dr. Dobb's Journal of Software Tools, v. 25, n. 11, p. 120–125, 2000.

BUKAS, C. et al. Robust deep learning based shrimp counting in an industrial farm setting. Journal of Cleaner Production, v. 468, p. 143024, 2024. DOI: 10.1016/j.jclepro.2024.143024.

CABRAL, I. T.; MARTINS, Y. S. Detecção de peixes utilizando redes neurais convolucionais. Rio Verde: [s. n.], 2023. Trabalho de Conclusão de Curso.

DAWKINS, M. et al. FishTrack23: An ensemble underwater dataset for multi-object tracking. In: IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). [S. l.: s. n.], 2024.

JIANG, Hanyu et al. Mobile-YOLO: A Lightweight Object Detection Algorithm for Four Categories of Aquatic Organisms. Fishes, v. 10, n. 7, p. 348, 2025. DOI: 10.3390/fishes10070348.

JIANG, Linhua et al. Improved YOLOv7-TINY Network for Sea Bream Detection. Journal of Computing and Electronic Information Management, 2024. DOI: 10.54097/c6kqnn36.

KANDIMALLA, V. et al. Automated detection, classification and counting of fish in fish passages with deep learning. Frontiers in Marine Science, v. 8, 2022. DOI: 10.3389/fmars.2021.823173.

LI, H. et al. Robust detection of farmed fish by fusing YOLOv5 with DCM and ATM. Aquacultural Engineering, v. 99, p. 102301, 2022. DOI: 10.1016/j.aquaeng.2022.102301.

LI, Ximing et al. LDNet: High Accuracy Fish Counting Framework using Limited Training Samples with Density Map Generation Network. Journal of King Saud University – Computer and Information Sciences, v. 36, n. 7, p. 102143, 2024. DOI: 10.1016/j.jksuci.2024.102143.

LIN, F. et al. Machine vision approach for monitoring and quantifying fish school using DVE-YOLO. Ecological Indicators, 2024. DOI: 10.1016/j.ecolind.2024.112769.

MANDEL, T. et al. Detection confidence driven multi-object tracking to recover reliable tracks from unreliable detections. Pattern Recognition, v. 135, p. 109107, 2023. DOI: 10.1016/j.patcog.2022.109107.

RAMÍREZ, Sebastián. FastAPI. Documentação oficial do software. 2018. Disponível em: <https://fastapi.tiangolo.com/>. Acesso em: 27 jul. 2026.

REDMON, Joseph et al. You Only Look Once: Unified, Real-Time Object Detection. [S. l.: s. n.], 2016. arXiv: 1506.02640.

SALEH, Alzayat; SHEAVES, Marcus; RAHIMI AZGHADI, Mostafa. Computer vision and deep learning for fish classification in underwater habitats: A survey. Fish and Fisheries, v. 23, n. 4, p. 977–999, 2022. DOI: 10.1111/faf.12666.

ULTRALYTICS. Ultralytics YOLO Documentation. [S. l.: s. n.], 2025. Disponível em: <https://docs.ultralytics.com/>.

VIJIYAKUMAR, K.; GOVINDASAMY, V.; AKILA, V. An effective object detection and tracking using automated image annotation with Inception-based Faster R-CNN model. International Journal of

ZHANG, Yifu et al. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. In: EUROPEAN Conference on Computer Vision (ECCV). [S. l.: s. n.], 2022.

APÊNDICE A – DOCUMENTAÇÃO DETALHADA DE ENGENHARIA

Este apêndice reúne as especificações detalhadas deslocadas do corpo do artigo para preservar a continuidade da argumentação.

As tabelas mantêm a numeração original para conservar as remissões feitas no texto.

Tabela 1. Requisitos funcionais do sistema de software de contagem

Grupo	Requisito funcional	Papel na solução
Núcleo da contagem	Aquisição e processamento de vídeo	Ler fluxos contínuos de câmera ou arquivos gravados
Núcleo da contagem	Deteção por quadro	Localizar peixes e fornecer caixas delimitadoras com confiança
Núcleo da contagem	Rastreamento temporal	Associar detecções entre quadros e manter IDs persistentes
Núcleo da contagem	Contagem por linha virtual	Registrar eventos quando uma trajetória válida cruza a linha
Núcleo da contagem	Interface e controle básico	Exibir contagem em tempo real e permitir início/parada

Apoio experimental	Extração inteligente de frames	Selecionar quadros representativos para anotação
Apoio experimental	Pré-anotação assistida	Integrar servidor de inferência com Label Studio
Apoio experimental	Preparação e estratificação	Organizar conjuntos de treino/validação/teste com splits fixos
Apoio experimental	Orquestração de treinos	Executar treinamentos por sementes e gerar relatórios
Operação e auditoria	Região de Interesse (ROI)	Restringir o processamento à área útil da canaleta
Operação e auditoria	Ajuste de linha de contagem	Configurar posição e sentido de cruzamento na cena
Operação e auditoria	Gravação e metadados	Salvar vídeos brutos e registros em JSONL por quadro

Fonte: elaborada pelo autor (2026).

Tabela 2. Requisitos não funcionais e critérios de avaliação

Requisito	Descrição técnica	Forma de avaliação
Acurácia de contagem	Minimizar o erro absoluto em relação à referência real	MAE por vídeo e saldo acumulado
Robustez visual	Tolerar reflexos, respingos, sombras e oclusões parciais	Desempenho nos 8 vídeos experimentais
Eficiência computacional	Executar inferência e tracking em tempo real	FPS médio e tempo de processamento por quadro
Reprodutibilidade	Permitir repetição idêntica de treinos e avaliações	Splits com semente fixa e relatórios CSV

Auditabilidade	Garantir verificação posterior de cada evento contabilizado	Metadados JSONL e vídeos brutos gravados
Configurabilidade	Permitir alteração de ROI, linha e limiares sem alterar código	Arquivos de configuração YAML

Fonte: elaborada pelo autor (2026).

Tabela 3. Módulos integrantes da arquitetura de software

Módulo	Responsabilidade principal	Classe / Elemento base
Fonte de Vídeo	Leitura sequencial de frames da câmera ou de arquivo MP4	VideoSource / cv2.VideoCapture
Gerenciador de ROI	Aplicação de polígono e recorte (crop/mask) para inferência	ROIManager
Detector de Objetos	Inferência de caixas delimitadoras e níveis de confiança	YOLODetector / Ultralytics
Rastreador Temporal	Associação de detecções entre quadros consecutivos	ByteTracker
Gerenciador de IDs	Confirmação temporal, filtragem por tamanho e controle de expiração	IDManager
Contador por Linha	Deteção de cruzamento vetorial em linha virtual de chegada	LineCounter
Interface Web	Servidor REST e streaming MJPEG para monitoramento remoto	FastAPI / uvicorn
Gravador e Auditoria	Persistência de vídeo bruto e metadados por quadro em JSONL	SessionRecorder

Configuração	Centralização de parâmetros em arquivos de especificação YAML	AppConfig
--------------	---	-----------

Fonte: elaborada pelo autor (2026).

Tabela 4. Etapas do fluxo experimental organizadas pelo Dataset Studio

Etapa	Operação principal	Evidência preservada
Origem	Cadastro dos vídeos e das unidades de captura	Identidade das mídias, hashes e configuração de extração
Extração	Amostragem uniforme ou assistida e seleção de frames	Manifesto de frames e relação com o vídeo de origem
Anotação	Integração com o Label Studio e recuperação das revisões	Tarefas, exportações e revisão selecionada
Versionamento	Materialização das imagens e rótulos e divisão agrupada	Manifesto, configuração, relatório de construção e hashes
Treinamento	Fixação da versão, do peso inicial e dos hiperparâmetros	Receita, log, métricas, pesos e cópia da proveniência do dataset
Avaliação	Aplicação do melhor checkpoint ao conjunto de teste de imagens	Métricas e artefatos associados à execução de treinamento

Fonte: elaborada pelo autor (2026).

¹ Mestrando no Programa de Pós-Graduação em Engenharia Aplicada e Sustentabilidade, Instituto Federal Goiano – Campus Rio Verde. Servidor do IFG – Campus Itumbiara (Tecnologia da Informação).

² Orientador. Prof. Dr., Programa de Pós-Graduação em Engenharia Aplicada e Sustentabilidade, Instituto Federal Goiano – Campus Rio Verde.

³ Coorientador. Prof. Dr., Programa de Pós-Graduação em Engenharia Aplicada e Sustentabilidade, Instituto Federal Goiano – Campus Rio Verde. E-mail: [acesse o artigo original para visualizar o e-mail](#).