

# ESTUDO DE SISTEMA BASEADO EM SENSOR INFRAVERMELHO APLICADO EM PROTÓTIPO DE MÓDULO DA EQUAÇÃO DA CONTINUIDADE

Palavras-Chave: IOT, SMART CITIES, MONITORAMENTO

Autores(as):

VINÍCIUS GABRIEL BARBOSA, FT – UNICAMP

Prof<sup>a</sup>. Dr<sup>a</sup>. LUBIENSKA C. L. JAQUIÊ RIBEIRO (orientadora), FT – UNICAMP

Prof<sup>a</sup>. Dr<sup>a</sup>. TALÍA SIMÕES DOS SANTOS XIMENES (coorientadora), FT - UNICAMP

## INTRODUÇÃO:

A hidráulica é o campo de estudo que se dedica às leis que regem o equilíbrio e movimento dos líquidos, bem como à aplicação dessas leis para a resolução de problemas reais [1]. Sob essa perspectiva, a gestão e controle dos sistemas de abastecimento de água são desafios recorrentes, e os modelos hidráulicos computacionais têm se mostrado essenciais para resolver esses impasses sócio urbanos [2].

As redes de água inteligentes são soluções eficazes para problemas de distribuição, combinando tecnologias de coleta de dados, computação, controle e comunicação para melhorar o desempenho e a resiliência dos sistemas [3]. Esses sistemas interconectados permitem a atenuação de falhas e a otimização dos recursos hídricos [4]. Os modelos hidráulicos computacionais oferecem representações virtuais precisas de sistemas reais, permitindo simulações detalhadas do comportamento desses sistemas sob diversas condições [5].

Para essa pesquisa, além do uso de sensores ópticos IR no Arduino-Uno conforme a Figura 1; para fins de uma análise de hidráulica moderna, foi incorporado também, por último, a utilização de linguagens de programação como Python para a modelagem e simulação dos sistemas, com base em visão computacional. Dessa forma permitindo, uma análise comparativa mais precisa e a procura por diferentes tipos de implementação de soluções otimizadas para diferentes recursos hídricos [6].

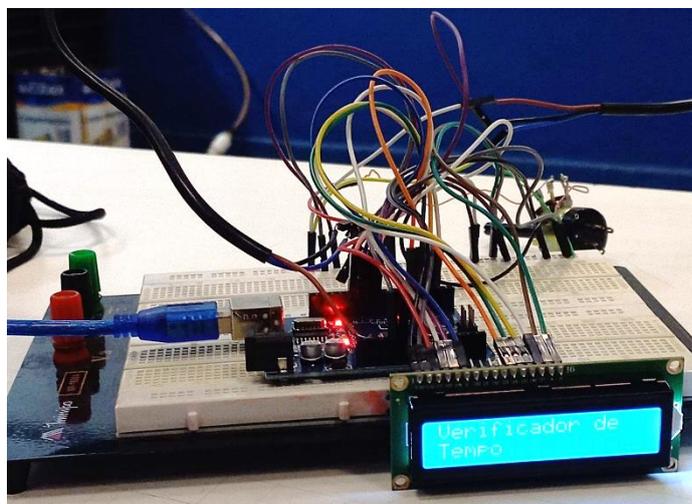


Figura 1 – Protótipo para aplicação dos sensores –

fonte: Coletânea do autor

## **METODOLOGIA:**

O presente estudo iniciou com a assimilação de conceitos fundamentais e do módulo hidráulico, onde a pesquisa e suas análises seriam aplicadas. Paralelamente, foi realizada uma familiarização com diversos tipos de microcontroladores e sensores passíveis de aplicação no projeto.

Como procedimento inicial, foram realizadas simulações utilizando o software Tinkercad, proporcionando um primeiro contato com alguns dos dispositivos planejados para integração no projeto. Subsequentemente, no Laboratório de Sistemas Embarcados, foram conduzidos testes preliminares de diferentes tipos de sensores Infra-Red (IR) para determinar a melhor adequação ao projeto.

Após as simulações virtuais, definiu-se o uso do sensor infravermelho óptico reflexivo de proximidade da linha E18-D80NK PNP e NPN, escolhido por suas características, como sensibilidade reduzida à luz visível, garantindo um funcionamento confiável em ambientes iluminados, e operação simplificada, ideal para projetos envolvendo detecção de obstáculos e barreiras.

Posteriormente, determinou-se que o microcontrolador Arduino-UNO seria utilizado na pesquisa e montagem do protótipo. Os sensores foram integrados ao sistema de teste e a lógica programacional foi implementada utilizando as linguagens C/C++ na IDE do Arduino.

O programa foi projetado de forma a permitir a comunicação entre os sensores e a placa microcontroladora Arduino. O código inicia um display LCD e dois sensores, que funcionam como um cronômetro. Quando o sensor inicial é ativado no ponto A, o cronômetro começa a contagem e exibe continuamente o tempo decorrido no LCD. Ao ser acionado o segundo sensor no ponto B, o cronômetro para e mostra por alguns segundos o tempo final no LCD, antes de retornar ao estado inicial para aguardar a detecção de um novo objeto pelo primeiro sensor.

Para uma análise mais detalhada e uma comparação, foi criado um programa em Python para rastreamento do tempo de objetos, com foco em cronometrar e identificar objetos em imagens. Utilizando a biblioteca de visão computacional OpenCV, o programa rastreia vários objetos em um vídeo. Inicialmente, o vídeo é carregado e os objetos (ROI) são manualmente selecionados, com suas coordenadas, cores e tempos de início armazenados. Em seguida, um rastreador CSRT múltiplo é iniciado e atualizado em cada quadro do vídeo, desenhando retângulos ao redor dos objetos e exibindo informações sobre o tempo de rastreamento. A taxa de quadros do vídeo é ajustada manualmente para garantir reprodução em tempo real. Por fim, o tempo total de processamento é calculado e exibido juntamente com a liberação dos recursos.

Os testes com sensores e visão computacional foram realizados para analisar suas respectivas efetividades de aplicação e resultados. Os sensores foram posicionados em diferentes momentos em dois módulos hidráulicos, para identificação de objetos e contagem de tempo de captação em tubulações. Foram também verificadas possíveis falhas, variações de luz, problemas no ambiente e versatilidade, tanto em sistemas abertos quanto fechados, contribuindo para uma análise detalhada dos fatores que poderiam afetar a operação dos sensores e do microcontrolador em condições reais de uso.

Os testes com Python foram realizados de forma simplificada, utilizando vídeos gravados com a câmera de um smartphone comum, que foram importados para o diretório do programa e executados três vezes

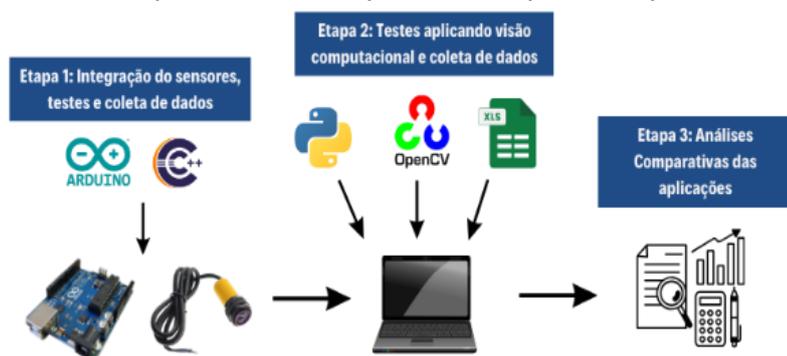


Figura 2 – Diagrama do sistema implementado – fonte: Coletânea do autor

## RESULTADOS E DISCUSSÃO:

Durante a concepção da aplicação, diversas modificações foram realizadas, desde a implementação de um sistema hidráulico adicional, que não estava previsto para testes, até a inclusão de um método comparativo usando visão computacional. Essas mudanças, juntamente com as otimizações do modelo, evidenciaram a complexidade do problema. As simulações virtuais no TinkerCad, utilizadas para definir o sensor, apresentaram várias limitações, especialmente em relação às condições ambientais, como variações de iluminação que afetam o desempenho dos componentes eletrônicos e a identificação de objetos pelos sensores.

O software TinkerCad não foi muito eficaz na definição do melhor sensor para o protótipo original, principalmente porque os sensores E18-D80NK têm ajustes de sensibilidade limitados, feitos manualmente por meio de um potenciômetro embutido para distâncias e angulações. No laboratório, o protótipo do módulo da equação da continuidade enfrentou problemas devido à sujeira que opacificou a tubulação, acionando os sensores prematuramente antes da passagem da bolinha no tubo. Mesmo com alguns testes realizados com objetos transparentes e uma mangueira sem coloração externa, o sensor ainda identificava erroneamente a mangueira e outros objetos, dependendo da angulação, posicionamento e iluminação, tornando-o inadequado para esse sistema hidráulico.

A visão computacional enfrentou desafios semelhantes, com dificuldades para identificar e acompanhar objetos de forma consistente, embora tenha realizado algumas identificações, como mostrado na figura 3. O que impossibilitou a comparação no primeiro protótipo hidráulico. Uma possível alternativa para o primeiro protótipo seria substituir os sensores por um *reed-switch*, utilizando a esfera interna para estender o campo magnético e fechar o circuito, ativando a cronometragem. Na visão computacional, uma abordagem viável seria usar *machine learning* para treinar o programa a identificar objetos dentro da tubulação.



Figura 3 – Teste Python no primeiro protótipo do módulo da equação da continuidade – fonte: Coletânea do Autor

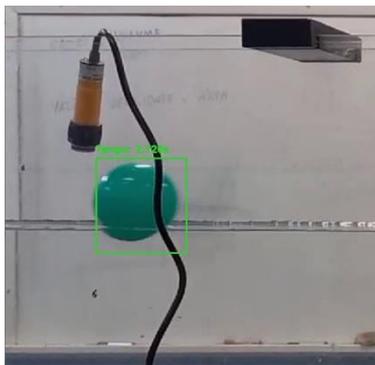


Figura 4 – Teste Python no segundo protótipo do módulo da equação da continuidade – fonte: Coletânea do Autor

Como alternativa, para que fosse possível realizar as análises comparativas, foi feita uma mudança para um protótipo aberto, onde os sensores e o sistema em Python operavam em condições mais favoráveis para a identificação de objetos, conforme ilustrado na figura 4. Nesse novo equipamento, os sensores se desempenharam melhores pois não faziam identificação da água e só eram ativados, quando passado a bolinha na superfície da água. Ademais, o Python se demonstrou muito efetivo, não necessitando utilizar mais ferramentas e tornar o programa com um grau de complexidade maior, e por último foi possível realizar os testes necessários para coleta de dados e fazer cálculos para os dois tipos de aplicação, conforme ilustra a tabela 1 a seguir.

Sensor E18-D80NK	Tempo Total (s)	Python	Tempo Total (s)			Média (s)
			Teste 1	Teste 2	Teste 3	
Tentativa 1	2.028	Tentativa 1	2.103	2.104	2.102	2.103
Tentativa 2	2.463	Tentativa 2	2.469	2.470	2.469	2.469
Tentativa 3	2.670	Tentativa 3	2.703	2.704	2.711	2.706
Tentativa 4	2.713	Tentativa 4	2.728	2.728	2.712	2.722
Tentativa 5	2.557	Tentativa 5	2.724	2.637	2.636	2.666
Tentativa 6	2.480	Tentativa 6	2.586	2.570	2.569	2.575
Tentativa 7	2.652	Tentativa 7	2.703	2.703	2.703	2.703
Tentativa 8	2.497	Tentativa 8	2.543	2.561	2.544	2.549
Tentativa 9	2.326	Tentativa 9	2.336	2.338	2.336	2.336
Tentativa 10	2.753	Tentativa 10	2.769	2.770	2.770	2.769

Tabela 1 – Tempo de cronometragem por tentativa para aplicação com Sensor E18-D80NK e Visão Computacional com Python

O protótipo apresentou algumas limitações na padronização da corrente de água, resultando em variações de altura de 1 a 2 cm, o que afetou a velocidade da corrente e, conseqüentemente, os tempos medidos. No entanto, essas variações não influenciaram significativamente os resultados, pois o programa em Python registrou tempos semelhantes aos capturados pelos sensores, com um erro médio de 1,87% e uma diferença de menos de 4,30% entre as medições. Isso resultou em uma precisão de mais de 95%. Os sensores se destacaram pela alta taxa de resposta de 2 ms, enquanto o programa em Python dependia da qualidade do vídeo, taxa de quadros, poder de processamento do computador e a precisão na seleção da região de interesse (ROI), o que introduziu pequenas variações nos tempos

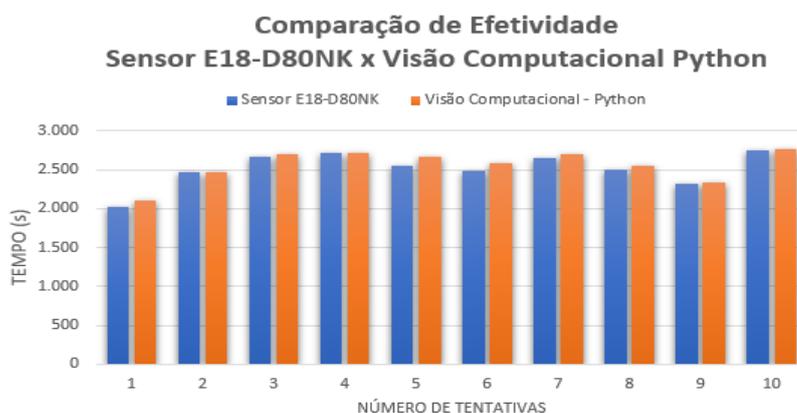


Figura 5 – Gráfico de Barra de Comparação de Efetividade – fonte: Coletânea do Autor

medidos, mas que do mesmo modo se desempenhou muito bem e com potencial de melhoramento. As figuras 5 e 6 demonstram visualmente a equiparidade dos resultados obtidos pelos dois métodos

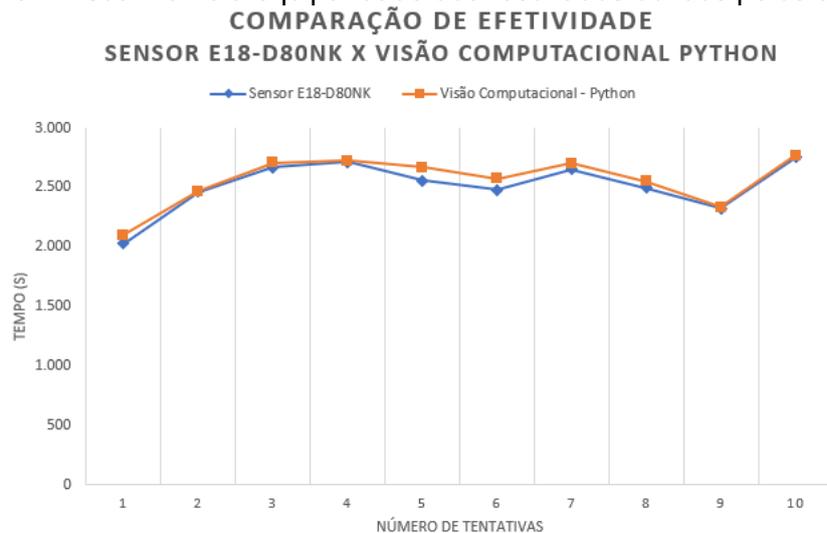


Figura 6 – Gráfico de Linha de Comparação de Efetividade – fonte: Coletânea do Autor

## CONCLUSÕES:

Os sensores E18-D80NK são precisos e cumprem sua função, porém nota-se que é limitado em termos de flexibilidade, tipos de ambiente e objetos, sendo recomendado para lugares abertos, no qual pode desempenhar e haver melhor efetividade em diferentes ramos comparado com a hidráulica. A utilização do Python para análises de vídeos e captura de imagens, mesmo em um protótipo não muito favorável e com uma lógica programacional menos complexa, conseguiu se desempenhar e mostrar potencial de melhoria, a visão computacional traz mais possibilidades de soluções para problemas e pesquisas, além de contribuir mais efetivamente para a inovação, principalmente quando se trata de *SmartCities* e *IoT*, pois permite que seja aprofundado em conceitos de Inteligência Artificial e *Machine Learning*, abrindo desse modo, um leque para aprimoramento de pesquisas iniciadas e melhor gerenciamento e melhor controle de sistemas hídricos no mundo.

---

## BIBLIOGRAFIA

- [1] ANAFISCO. **Como acontece a gestão da água em cidades inteligentes?** Disponível em: <https://anafisco.org.br/como-acontece-a-gestao-da-agua-em-cidades-inteligentes/>.
- [2] SAMIR, Nourhan; KANSO, Rawya; ELBARKI, Walid; FLEIFLE, Amr. (2017). **Pressure control for minimizing leakage in water distribution systems**. Alexandria Engineering Journal, [s.l.], v. 56, n. 4, p. 601-612.
- [3] BRENTAN, Bruno Melo. **Sistemas de abastecimento de água inteligentes: Previsão de demanda e controle ótimo em tempo real**. 2017. Tese (Doutorado em Engenharia Civil) - Universidade Estadual de Campinas.
- [4] Rasekh A., Hassanzadeh A., Mulchandani S., Modi S., Banks M. K., 2016. **Smart Water Networks and Cyber Security**.
- [5] WALSKI, T.M., Chase D.V., Savic, D.A., Grayman, W., Beckwith, S., Koelle, E. (2003). **Advanced Water Distribution Modelling and Management**. Haestad Press, Waterbury, CT, USA.
- [6] Andrich, P., & Silva, F. (2021). **Computational Methods for Hydraulic Analysis Using Python**. Journal of Water Resources Planning and Management, 147(3), 04021007.