

# Implementação e avaliação de sistema OCR para identificação de placas veiculares em robô humanoide NAO: uma prova de conceito

*Implementation and evaluation of an OCR system for vehicle license plate identification in the NAO humanoid robot: a proof of concept*

<sup>1</sup> Vitor Amadeu Souza  

<sup>1</sup> UniFOA, Centro Universitário de Volta Redonda, Volta Redonda, RJ, Brasil

## RESUMO

O presente trabalho descreve o desenvolvimento e avaliação de um sistema integrado de reconhecimento óptico de caracteres (OCR) para o robô humanoide NAO, com foco na identificação de placas veiculares brasileiras em ambiente laboratorial controlado. O sistema utiliza a biblioteca EasyOCR baseada em aprendizado profundo, integrada ao *framework* qi do NAO, por meio de arquitetura *off-board* de processamento remoto via WiFi, com captura de imagens pela câmera frontal em resolução 640x480 a 15 quadros por segundo (do inglês, *Frames Per Second* - FPS). Cinco casos experimentais foram conduzidos em ambiente controlado, com tempos de processamento variando entre 0,070 s e 0,345 s. Nos cinco casos avaliados, o módulo de detecção de regiões textuais apresentou menos problemas do que o módulo de classificação de caracteres, sendo o desempenho deste último influenciado pela qualidade visual da imagem, pelo contraste e pela complexidade composicional da placa. A confiança média das detecções foi de 0,75 (desvio-padrão de 0,25), com variações entre casos simples e complexos. Os resultados fornecem evidências experimentais preliminares sobre a viabilidade da integração proposta e permitem identificar limitações do reconhecimento de caracteres, indicando o pré-processamento de imagens como uma possibilidade de aprimoramento em trabalhos futuros.

## Palavras-chave:

Robótica Humanoide, Reconhecimento Óptico de Caracteres, Robô NAO, Visão Computacional, Identificação de Placas Veiculares.

## ABSTRACT

This study describes the development and evaluation of an integrated optical character recognition (OCR) system for the NAO humanoid robot, focusing on the identification of Brazilian vehicle license plates in a controlled laboratory environment. The system uses the EasyOCR deep learning library, integrated with the NAO qi framework through an off-board remote processing architecture via WiFi, with images captured by the front camera at a resolution of 640 × 480 pixels and 15 frames per second (FPS). Five experimental cases were conducted under controlled conditions, with processing times ranging from 0.070 s to 0.345 s. In the five evaluated cases, the text-region detection module exhibited fewer observed problems than the character classification module, whose recognition performance was influenced by image quality, contrast, and the compositional complexity of the license plate. The mean confidence of the detections was 0.75, with a standard deviation of 0.25, showing variations between simpler and more complex cases. The results provide preliminary experimental evidence regarding the feasibility of the proposed integration and help identify limitations in character recognition, indicating image preprocessing as a potential avenue for improvement in future work.

## Keywords:

Humanoid Robotics, Optical Character Recognition, NAO Robot, Computer Vision, Vehicle License Plate Identification.

## 1 INTRODUÇÃO

A crescente demanda por sistemas automatizados de reconhecimento e identificação em ambientes urbanos tem impulsionado o desenvolvimento de tecnologias robóticas integradas com técnicas avançadas de visão computacional (Luo *et al.*, 2024). Nesse contexto, o reconhecimento óptico de caracteres (OCR) representa uma das aplicações mais relevantes, especialmente quando implementado em plataformas robóticas móveis capazes de operar em tempo real (Chen *et al.*, 2010). A capacidade de extrair e interpretar informações textuais do ambiente confere aos robôs um papel ativo em tarefas de monitoramento, controle de acesso e interação com a infraestrutura urbana, ampliando enormemente seu espectro de aplicações práticas.

O robô humanoide NAO, desenvolvido pela SoftBank Robotics, tem se consolidado como plataforma de referência para pesquisa em visão computacional aplicada à robótica, graças à sua arquitetura modular, capacidades sensoriais integradas e ampla base de suporte na literatura científica (Magallán-Ramírez, 2022). No entanto, suas limitações computacionais intrínsecas, em particular o suporte restrito ao Python 2.7 na versão NAO V6, impõem desafios relevantes à implementação de algoritmos modernos de aprendizado profundo diretamente na plataforma, tornando necessária a adoção de estratégias alternativas de processamento.

A identificação automática de placas veiculares representa um domínio de aplicação particularmente exigente para sistemas OCR embarcados em robôs móveis, uma vez que envolve variações em condições de iluminação, ângulos de captura, distâncias e qualidade de imagem (Guruprasad *et al.*, 2021). Diante desse contexto, o presente trabalho tem como objetivo desenvolver e avaliar, em caráter de prova de conceito, um sistema integrado de OCR para o robô NAO, com foco na identificação de placas veiculares brasileiras em ambiente laboratorial controlado, por meio de uma arquitetura de processamento remoto (*off-board*). A pesquisa busca responder a duas questões complementares: se é viável operar algoritmos modernos de OCR baseados em aprendizado profundo em uma plataforma humanoide com recursos computacionais limitados, como o NAO, e quais componentes do *pipeline* de reconhecimento apresentam as principais limitações em condições controladas de laboratório.

A contribuição do trabalho não reside no algoritmo de OCR em si, já consolidado na literatura, mas na caracterização experimental de sua integração com uma plataforma robótica de recursos computacionais restritos (Python 2.7, NAO V6), por meio de arquitetura *off-board*, ainda pouco documentada nesse contexto específico, incluindo a análise do tempo de processamento, da detecção de regiões textuais e dos erros de reconhecimento observados nos cinco casos experimentais. Por seu caráter exploratório, o estudo não pretende estabelecer uma avaliação estatisticamente representativa de sistemas de reconhecimento de placas, mas demonstrar a viabilidade da arquitetura e identificar limitações relevantes para seu aprimoramento.

A estrutura do presente trabalho está organizada da seguinte forma: a seção 2 apresenta a fundamentação teórica, abordando os princípios do reconhecimento óptico de caracteres, a arquitetura da biblioteca EasyOCR e as características técnicas do robô NAO relevantes para o sistema desenvolvido. A seção 3 apresenta a metodologia adotada, incluindo a arquitetura de *software*, a configuração experimental e as métricas de avaliação empregadas. A seção 4 expõe e discute os resultados obtidos nos experimentos de reconhecimento de placas veiculares, com análise dos tempos de processamento, valores de confiança e erros de reconhecimento observados. A seção 5 apresenta as limitações do trabalho. Por fim, a seção 6 apresenta as conclusões do trabalho e as direções propostas para investigações futuras.

## 2 FUNDAMENTAÇÃO TEÓRICA

O reconhecimento óptico de caracteres em sistemas robóticos fundamenta-se na convergência de técnicas avançadas de visão computacional, aprendizado de máquina e processamento de imagens digitais

(Manakitsa et al., 2024). A evolução dos algoritmos OCR tem sido marcada pela transição de abordagens baseadas em características manuais para métodos fundamentados em aprendizado profundo, que demonstram capacidade de generalização em condições variadas de captura de imagem. Long et al. (2020) destacam que sistemas OCR baseados em redes neurais convolucionais apresentam robustez aprimorada, se comparados a métodos tradicionais, quando aplicados a cenários com variações visuais complexas, como o reconhecimento de placas veiculares em ambientes urbanos.

## 2.1 Arquitetura de sistemas OCR

A arquitetura de sistemas OCR modernos tipicamente incorpora múltiplas etapas de processamento, incluindo detecção de regiões de interesse, segmentação de caracteres, extração de características e classificação final. Em aplicações robóticas, essas etapas devem ser otimizadas para operação em tempo real, considerando as limitações computacionais inerentes às plataformas móveis (Memon, 2020). Wang et al. (2020) propõem que a eficiência computacional em sistemas OCR pode ser melhorada através da implementação de algoritmos de detecção de texto baseados em redes neurais recorrentes combinadas com mecanismos de atenção espacial.

## 2.2 A biblioteca EasyOCR

A biblioteca EasyOCR, utilizada no sistema desenvolvido, implementa uma arquitetura híbrida que combina redes neurais convolucionais para detecção de texto com redes neurais recorrentes, para o reconhecimento de sequências de caracteres. Baek et al. (2019) descrevem que essa abordagem permite reconhecimento robusto de texto em múltiplos idiomas e sistemas de escrita, mantendo eficiência computacional adequada para aplicações em tempo real. A implementação do EasyOCR baseia-se no *framework CRAFT* (*Character Region Awareness for Text Detection*) para detecção de regiões textuais.

O robô NAO oferece uma plataforma de desenvolvimento robótica que integra capacidades avançadas de percepção visual através de duas câmeras de alta resolução, processador Intel Atom com capacidade de processamento distribuído e *framework* de *software* NAOqi que facilita o desenvolvimento de aplicações em Python (Aldebaran Robotics, 2021). Gouaillier et al. (2009) descrevem a arquitetura do NAO como um sistema modular que permite integração eficiente de algoritmos de visão computacional através de interfaces de programação padronizadas. A utilização do *middleware* qi permite comunicação assíncrona entre módulos de *software*, facilitando a implementação de sistemas de processamento em tempo real.

## 2.3 A integração de sistemas OCR

A integração de sistemas OCR em plataformas robóticas apresenta desafios específicos relacionados à sincronização entre captura de imagens, processamento e controle de movimento do robô. Chen et al. (2015) argumentam que sistemas eficientes devem implementar arquiteturas de processamento paralelo que permitam captura contínua de imagens enquanto processamento OCR é executado em *threads* separados. Essa abordagem minimiza a latência do sistema e permite responsividade adequada para aplicações interativas.

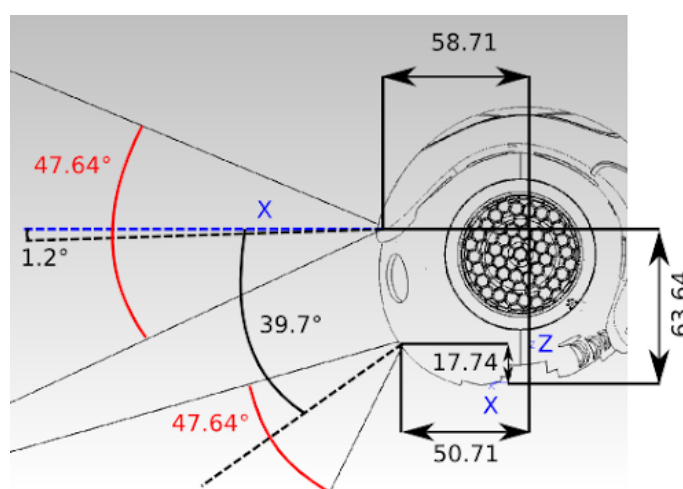
O processamento de placas veiculares representa um domínio específico dentro do reconhecimento OCR que demanda considerações particulares sobre variações de formato, padrões regionais e condições ambientais de captura (Anagnostopoulos et al., 2008). Montazzoli e Jung (2017) identificam que sistemas OCR para placas veiculares devem ser capazes de reconhecer múltiplos padrões alfanuméricos, lidar com variações de fonte e tamanho, e manter precisão sob condições adversas de iluminação. A implementação de técnicas de pré-processamento de imagem, incluindo correção de perspectiva, normalização de contraste e filtragem de ruído, é fundamental para otimizar a *performance* de reconhecimento (Anagnostopoulos et al., 2008).

## 2.4 Câmeras e sistema híbrido

O robô NAO é equipado com duas câmeras CMOS, uma posicionada na parte superior e outra na parte inferior, que lhe conferem capacidade de visão computacional (Gouaillier *et al*, 2008). O campo de visão (*Field of View – FOV*) da câmera apresenta ângulo diagonal de  $47,64^\circ$ , abrangendo uma área efetiva de  $39,7^\circ$  a partir do ponto focal. Esses parâmetros técnicos são fundamentais para o desenvolvimento de algoritmos de visão computacional eficientes (Aldebaran Robotics, 2021).

A precisão das medidas do campo de visão é essencial para o processamento de imagens e para o reconhecimento de objetos em tempo real, possibilitando a calibração adequada de algoritmos de detecção e rastreamento (Ficht *et al*, 2018). A Figura 1 apresenta o campo de visão do Robô NAO.

Figura 1 - Campo de visão (FOV) do Robô NAO.

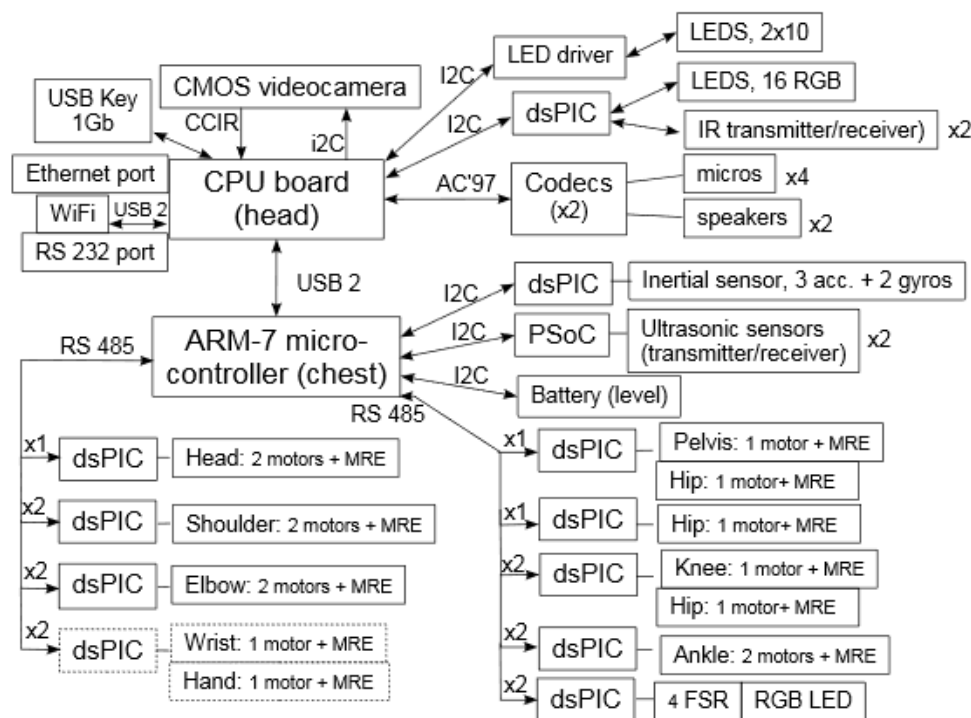


Fonte: Aldebaran, 2021

A arquitetura interna do NAO é composta por diversos módulos integrados que configuram um sistema híbrido. Entre esses elementos, destaca-se a câmera CMOS, responsável por fornecer os dados visuais que serão processados (Gouaillier *et al*, 2008). Essa câmera está conectada à placa principal (CPU board), que utiliza um processador Intel Atom E3845, auxiliado por um microcontrolador ARM-7. A comunicação com os módulos periféricos ocorre por meio de interfaces como o barramento I2C (Gouaillier *et al*, 2008).

Além disso, o robô conta com módulos dsPIC, que controlam os motores e sensores distribuídos pelo corpo, incluindo cabeça, ombros, cotovelos, quadris, joelhos e tornozelos. Essa arquitetura híbrida garante a execução coordenada de movimentos e interações complexas com o ambiente (Heremans, 2016). A Figura 2 apresenta o sistema híbrido do Robô NAO.

Figura 2 - Arquitetura interna do Robô NAO.



Fonte: Aldebaran, 2021.

A arquitetura interna configura um sistema de processamento distribuído, no qual a *CPU board*, localizada na cabeça, centraliza o processamento principal e se comunica com os periféricos por meio dos protocolos I<sup>2</sup>C, USB e RS 485. Um microcontrolador ARM-7 instalado no tórax coordena módulos dsPIC distribuídos pelos membros superiores e inferiores, responsáveis pelo acionamento dos motores e leitura dos *encoders* nas articulações de cabeça, ombros, cotovelos, quadris, joelhos e tornozelos. Sensores inerciais, ultrassônicos e de força nos pés complementam o conjunto de percepção do robô, garantindo a execução coordenada de movimentos e interações complexas com o ambiente em tempo real (Gouaillier *et al.*, 2008; Heremans *et al.*, 2016).

### 3 METODOLOGIA

O desenvolvimento do sistema de reconhecimento óptico de caracteres para o robô NAO foi estruturado em uma abordagem metodológica que integra desenvolvimento de *software*, implementação de algoritmos de visão computacional e validação experimental através de testes controlados. A metodologia empregada segue os princípios de desenvolvimento iterativo com validação contínua, permitindo refinamento progressivo do sistema baseado nos resultados obtidos em cada etapa de implementação.

A arquitetura do sistema foi projetada seguindo o padrão de design modular, com separação clara entre os componentes de interface com *hardware*, processamento de imagens, reconhecimento OCR e interface de usuário. O sistema foi implementado em Python 3.8, utilizando o *framework* *qi*, para comunicação com o robô NAO; OpenCV 4.12, para processamento de imagens; EasyOCR 1.7.2, para reconhecimento de caracteres e; NumPy, para operações matemáticas sobre *arrays* de imagens. A escolha dessas tecnologias

baseou-se em sua comprovada eficiência em aplicações de visão computacional robótica e compatibilidade com a plataforma NAO.

### 3.1 Arquitetura e Configuração do Sistema

A configuração do ambiente de desenvolvimento incluiu a instalação do NAOqi SDK versão 2.7, configuração da conectividade TCP/IP com o robô através da rede local e implementação de rotinas de teste de conectividade, para garantir comunicação estável durante as operações. O robô NAO foi configurado com endereço IP 172.15.1.29 na porta 9559, permitindo acesso consistente aos serviços de câmera e controle de movimento. A câmera frontal foi configurada para captura em resolução VGA (640x480 pixels) com taxa de 15 quadros por segundo, proporcionando equilíbrio entre qualidade de imagem e eficiência de processamento.

O módulo de captura de imagens implementa *streaming* contínuo através do serviço ALVideoDevice do NAO, utilizando programação *multithread* para separar as operações de captura de imagem da interface de usuário e processamento OCR. A implementação de *threads* dedicados para captura permite manutenção de fluxo de vídeo estável enquanto o processamento computacionalmente intensivo é executado em paralelo. O sistema implementa *buffer* circular para *frames* de imagem, garantindo que o processamento OCR sempre acesse a imagem mais recente disponível.

### 3.2 Implementação dos Módulos

A integração da biblioteca EasyOCR foi realizada com configuração para reconhecimento de caracteres em inglês e português, considerando os padrões de placas veiculares brasileiras que incorporam caracteres alfanuméricos em ambos idiomas. O sistema foi configurado para utilizar aceleração GPU, quando disponível, com *fallback* automático para processamento CPU em sistemas sem suporte CUDA. A inicialização do modelo EasyOCR é realizada durante a fase de *setup* do sistema para minimizar a latência durante operações de reconhecimento.

### 3.3 Armazenamento e Avaliação de Resultados

A interface de usuário foi implementada por meio do *OpenCV* para exibição contínua do fluxo de vídeo, com sobreposição de informações do sistema e dos resultados de OCR. O reconhecimento é executado sob demanda pelo operador, enquanto o *streaming* da câmera permanece ativo. O sistema também implementa controles de teclado para captura de imagens, reinicialização do *streaming* e finalização da aplicação. A interface exibe informações de *performance* incluindo tempo de processamento OCR, número de textos detectados e estatísticas de confiança das detecções.

O armazenamento de resultados foi implementado através de sistema de arquivos estruturado que salva imagens originais, imagens anotadas com resultados de OCR e metadados em formato JSON contendo informações detalhadas sobre cada detecção. O sistema gera *timestamps* únicos para cada captura, facilitando organização e análise posterior dos resultados. Os dados salvos incluem coordenadas das caixas delimitadoras, texto reconhecido, níveis de confiança e métricas de *performance*.

Os testes foram realizados em ambiente interno controlado com iluminação artificial de 250 lux. Para cada caso experimental, foram registrados o tempo de processamento OCR, os textos reconhecidos, os valores de confiança fornecidos pelo modelo e o número de regiões textuais detectadas. A análise dos resultados foi realizada de forma descritiva, considerando os tempos de processamento, os valores de confiança e os erros de reconhecimento observados nos cinco casos experimentais conduzidos em condição padronizada. Para os valores de confiança, foram calculados a média e o desvio-padrão do conjunto de elementos textuais detectados.

O robô humanoide NAO foi posicionado em frente a uma tela exibindo imagens de placas veiculares (Wikipedia, 2025), garantindo o alinhamento adequado para a captura visual. Durante o experimento, a câmera frontal do robô adquiriu continuamente as imagens exibidas na tela. O fluxo de vídeo permaneceu ativo durante a operação, enquanto o reconhecimento OCR foi executado sob demanda a partir do quadro disponível no momento da captura. Os resultados obtidos, incluindo os caracteres reconhecidos, valores de confiança e tempo de processamento do OCR, foram armazenados para posterior análise (Figura 3).

Figura 3 - Configuração experimental com o robô NAO posicionado em frente a uma tela exibindo placas veiculares para processamento por OCR



Fonte: (Blanson Henkemans, 2025; LG Electronics, 2025).

Como o robô NAO V6 suporta apenas Python 2.7, foi necessário implementar um processamento remoto usando a técnica de *off-board processing*, no qual o computador se comunica com o robô via WiFi para obter os dados da câmera. O código-fonte desta pesquisa encontra-se disponível para *download* através do link <https://github.com/vitor-souza-ime/ocrnao>.

Considerando o caráter de prova de conceito do estudo, a avaliação experimental foi deliberadamente restrita a cinco casos em ambiente laboratorial controlado. Dessa forma, os resultados apresentados devem ser interpretados como uma caracterização exploratória do comportamento da arquitetura implementada, não sendo utilizados para inferir desempenho generalizável em cenários veiculares reais. Além disso, os tempos medidos correspondem à etapa de processamento OCR executada sob demanda e não representam uma medição da latência fim a fim de um sistema de reconhecimento automático de placas veiculares (*Automatic License Plate Recognition – ALPR*) operando continuamente.

### 3.4 Descrição do algoritmo implementado

O sistema desenvolvido segue uma arquitetura de processamento em *pipeline*, organizada em quatro módulos funcionais principais: inicialização e conexão, captura de imagens por *streaming*, reconhecimento OCR e armazenamento de resultados. O Quadro 1 apresenta o pseudocódigo que descreve o fluxo geral do algoritmo implementado.

Quadro 1 - Pseudocódigo da leitura de OCR no Robô NAO

| ALGORITMO SistemaOCR_NAO                                        |   |        |    |                |     |               |
|-----------------------------------------------------------------|---|--------|----|----------------|-----|---------------|
| //                                                              | — | Módulo | 1: | Inicialização  |     |               |
| <hr/>                                                           |   |        |    |                |     |               |
| INÍCIO                                                          |   |        |    |                |     |               |
| Inicializar leitor EasyOCR com idiomas ['pt', 'en']             |   |        |    |                |     |               |
| SE GPU disponível ENTÃO utilizar aceleração CUDA                |   |        |    |                |     |               |
| SENÃO utilizar processamento CPU                                |   |        |    |                |     |               |
| Conectar ao robô NAO via protocolo TCP/IP (porta 9559)          |   |        |    |                |     |               |
| Subscrever câmera frontal via serviço ALVideoDevice             |   |        |    |                |     |               |
| resolução ← VGA (640×480 pixels)                                |   |        |    |                |     |               |
| espaço de cores ← RGB                                           |   |        |    |                |     |               |
| taxa ← 15 quadros/segundo                                       |   |        |    |                |     |               |
| //                                                              | — | Módulo | 2: | Streaming      | e   | Captura       |
| <hr/>                                                           |   |        |    |                |     |               |
| Iniciar thread dedicada para streaming contínuo                 |   |        |    |                |     |               |
| ENQUANTO sistema_ativo FAÇA                                     |   |        |    |                |     |               |
| frame ← capturar imagem via <a href="#">getImageRemote()</a>    |   |        |    |                |     |               |
| Converter array de bytes para matriz NumPy                      |   |        |    |                |     |               |
| Converter espaço de cores RGB para BGR (compatibilidade OpenCV) |   |        |    |                |     |               |
| Armazenar frame no buffer circular                              |   |        |    |                |     |               |
| //                                                              | — | Módulo | 3: | Reconhecimento | OCR | (sob demanda) |
| <hr/>                                                           |   |        |    |                |     |               |
| SE tecla_espaço pressionada ENTÃO                               |   |        |    |                |     |               |
| imagem_capturada ← frame atual do buffer                        |   |        |    |                |     |               |

```

t_início ← tempo atual
resultados ← EasyOCR.readtext(imagem_capturada)
t_fim ← tempo atual
tempo_processamento ← t_fim - t_início

PARA CADA (bbox, texto, confiança) EM resultados FAÇA
SE confiança > 0,8 ENTÃO cor ← verde
SENÃO SE confiança > 0,5 ENTÃO cor ← amarelo
SENÃO cor ← vermelho
Desenhar caixa delimitadora com cor correspondente
Sobrepôr texto e valor de confiança na imagem

//      —      Módulo      4:      Armazenamento


---


Salvar imagem original em arquivo JPEG
Salvar imagem anotada em arquivo JPEG
Salvar metadados em arquivo JSON contendo:
timestamp, tempo_processamento,
total de textos detectados,
PARA CADA detecção: texto, confiança, coordenadas bbox,
estatísticas: confiança média, máxima, contagem alta confiança

SE tecla_q pressionada ENTÃO encerrar sistema
SE tecla_r pressionada ENTÃO reiniciar streaming
FIM

```

Fonte: Elaborado pelo autor, 2026

O primeiro módulo realiza a inicialização do sistema, incluindo a instanciación do modelo EasyOCR com suporte aos idiomas português e inglês, considerando os padrões alfanuméricos das placas veiculares brasileiras e o estabelecimento da conexão TCP/IP com o robô. A detecção automática de GPU permite que o sistema utilize aceleração por *hardware*, quando disponível, com *fallback* automático para CPU, estratégia que otimiza o desempenho sem comprometer a portabilidade do código (Baek *et al.*, 2019).

O segundo módulo implementa o *streaming* contínuo de imagens, por meio de uma *thread* dedicada, seguindo a abordagem de processamento paralelo recomendada por Chen *et al.* (2015) para sistemas OCR em plataformas robóticas. Os *frames* capturados via serviço ALVideoDevice são convertidos de *arrays* de *bytes* para matrizes NumPy e armazenados em um *buffer* circular, garantindo que o módulo de reconhecimento sempre acesse a imagem mais recente disponível sem bloquear o fluxo de captura.

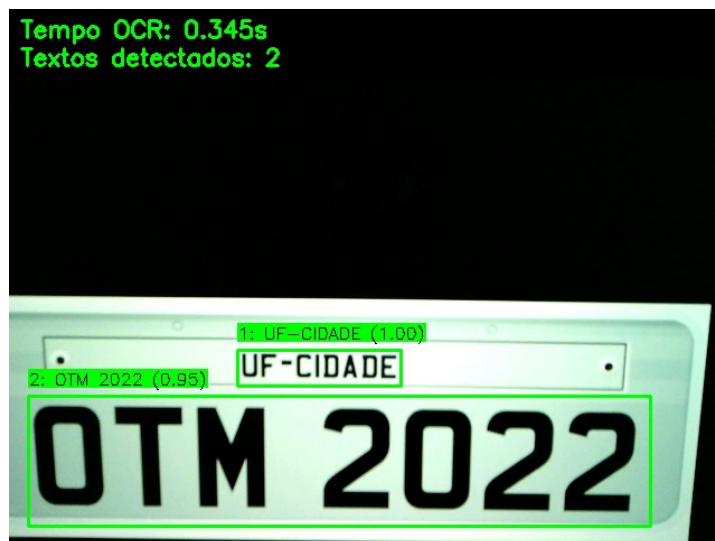
O terceiro módulo executa o reconhecimento OCR sob demanda, acionado pelo operador, por meio de controle de teclado. O tempo de processamento é medido com precisão por meio da função *perf\_counter()*, e os resultados são anotados visualmente na imagem, por meio de caixas delimitadoras com codificação de cores proporcional ao nível de confiança de cada detecção: verde, para confiança superior a 0,8; amarelo, para valores entre 0,5 e 0,8; e vermelho, para valores inferiores a 0,5. Essa codificação fornece *feedback* visual imediato ao operador sobre a qualidade de cada reconhecimento, sem necessidade de interpretação numérica dos resultados (Long; He; Yao, 2020).

O quarto módulo persiste os resultados de cada captura em três arquivos: a imagem original, a imagem anotada com as detecções e um arquivo JSON contendo os metadados completos da operação, incluindo *timestamp*, tempo de processamento, coordenadas das caixas delimitadoras, textos reconhecidos, valores de confiança individuais e estatísticas agregadas. Essa estrutura de armazenamento facilita tanto a auditoria das detecções quanto a análise estatística posterior dos resultados, atendendo a requisitos de rastreabilidade em aplicações de monitoramento (Memon, 2020).

## 4 RESULTADOS E DISCUSSÃO

Nos cinco casos experimentais avaliados, foram observados tempos de processamento OCR entre 0,070 e 0,345 segundos por imagem capturada. Os resultados apresentaram variações no reconhecimento dos caracteres entre as diferentes placas, permitindo analisar qualitativamente limitações relacionadas à qualidade visual, ao contraste e às características dos elementos textuais. No primeiro caso experimental, a placa "OTM 2022" e o elemento "UF-CIDADE" foram corretamente detectados, totalizando dois elementos textuais reconhecidos integralmente nesse caso específico, com tempo de processamento de 0,345 s. (Figura 4).

Figura 4 - Resultado da detecção para a placa "OTM 2022" com dois elementos textuais. As caixas delimitadoras indicam as regiões reconhecidas



Fonte: Wikimedia Commons, domínio público (Wikipedia, 2025).

O segundo experimento apresentou tempo de processamento de 0,117 s para a placa brasileira "LSN4I49", com detecção de três elementos textuais, incluindo "Brasil" e identificação regional (Figura 5). Nesse caso, a imagem apresentava boa diferenciação visual entre os caracteres principais e o fundo. Embora o sistema tenha reconhecido a sequência "LSN4I49", a placa correta contém o caractere "l" no lugar do numeral "1", evidenciando uma limitação do OCR na distinção entre caracteres visualmente similares, problema recorrente em sistemas de reconhecimento baseados em aprendizado profundo, quando aplicados a fontes com baixa diferenciação morfológica (Memon, 2020).

Figura 5 - Reconhecimento da placa "LSN4I49" com três elementos textuais. As caixas delimitadoras indicam as regiões detectadas

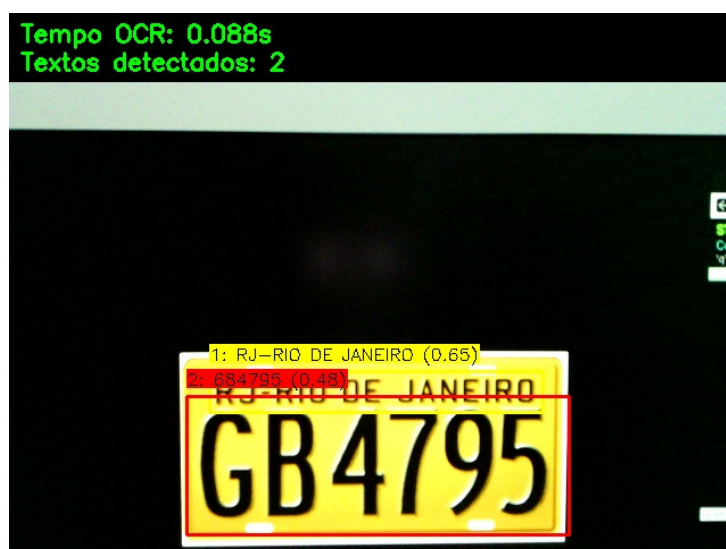


Fonte: Olympiobr, Wikimedia Commons, CC BY-SA 3.0 (Wikipedia, 2025).

O terceiro caso apresentou tempo de processamento de 0,088 s para a placa "GB4795", com detecção de dois elementos textuais (Figura 6). No entanto, a saída do OCR foi incorreta ("684795"), com confiança de reconhecimento de 0,48. Esse caso ilustra que um menor tempo de processamento não implica necessariamente

mente em reconhecimento correto. A literatura indica que características visuais da imagem e dos caracteres podem influenciar o desempenho de sistemas de reconhecimento de texto (Li et al., 2019).

Figura 6 - Reconhecimento da placa "GB4795" com dois elementos textuais detectados e erro de classificação pelo OCR



Fonte: Olympiobr, Wikimedia Commons, CC BY-SA 3.0 (Wikipedia, 2025).

O quarto experimento processou a placa "BCD-3456" em 0,110 s, detectando três elementos textuais. Contudo, o nome da cidade não foi reconhecido de forma completa (Figura 7), sendo registrado baixo valor de confiança para esse elemento (0,16). Essa dificuldade é compreensível considerando que os nomes de cidades em placas veiculares apresentam tipografia de menor tamanho e contraste reduzido em relação aos caracteres principais, condições que aumentam a taxa de erro em sistemas OCR (Anagnostopoulos et al., 2008).

Figura 7 - Detecção da placa "BCD-3456" com três elementos textuais, com nome da cidade reconhecido parcialmente



Fonte: Olympiobr, Wikimedia Commons, CC BY-SA 3.0 (Wikipedia, 2025).

O último caso, relativo à placa "A.13987", alcançou o menor tempo de processamento do conjunto experimental, de 0,070 s, com detecção de apenas um elemento textual (Figura 8). Embora o valor correto da placa seja "A.13987", o modelo classificou incorretamente o caractere inicial como "4", sugerindo sensibilidade do sistema a pontuações e caracteres especiais adjacentes a letras. Nesse caso específico, a imagem apresentou o menor tempo de processamento observado no conjunto experimental, de 0,070 s.

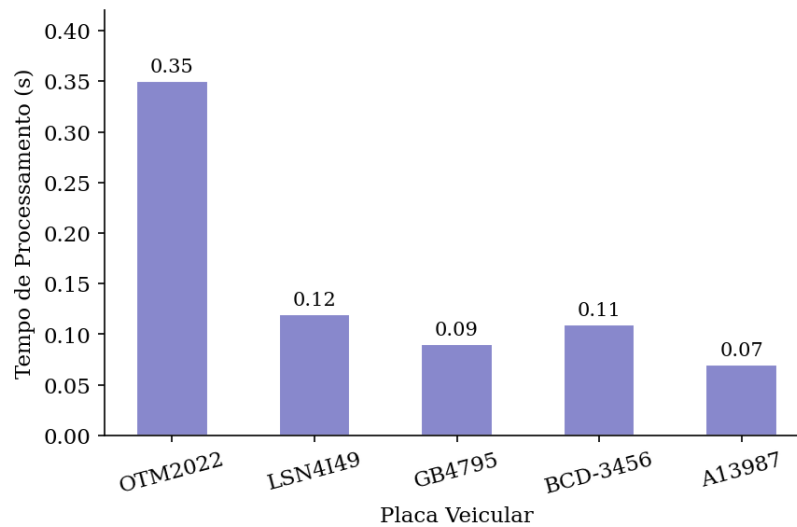
Figura 8 - Reconhecimento da placa "A.13987" com um único elemento textual



Fonte: Olympiobr, Wikimedia Commons, CC BY-SA 3.0 (Wikipedia, 2025).

A Figura 9 apresenta um gráfico de barras com os tempos de processamento registrados para cada placa analisada. Nos casos avaliados, observou-se variação do tempo de processamento em função das características das imagens: a placa "A13987" apresentou o menor tempo, de 0,070 s, enquanto a placa "OTM2022" apresentou o maior tempo, de 0,345 s. Devido ao número reduzido de casos, esses resultados devem ser interpretados de forma descritiva, não permitindo estabelecer uma relação causal entre complexidade visual e tempo de processamento. Esse comportamento é consistente com a literatura sobre sistemas OCR baseados em redes neurais, nos quais o custo computacional tende a crescer proporcionalmente à quantidade e complexidade dos elementos a serem segmentados e classificados (Long; He; Yao, 2020).

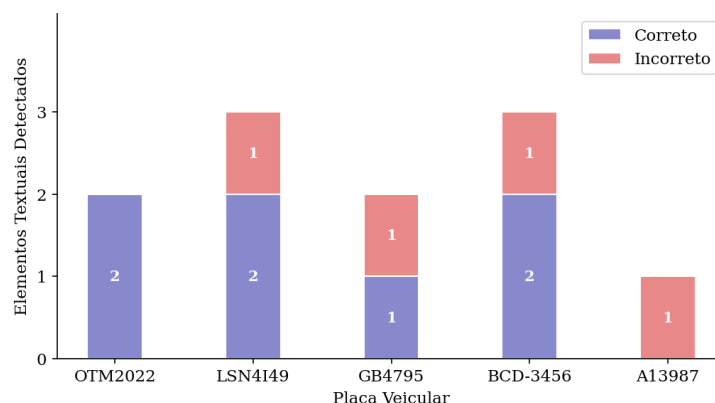
Figura 9 - Tempo de processamento por placa veicular.



Fonte: Elaborado pelo autor, 2026.

A Figura 10 apresenta o número de elementos textuais detectados por placa, distinguindo entre reconhecimentos corretos e incorretos por meio de barras empilhadas. Observa-se que as regiões textuais foram detectadas nos casos avaliados, ainda que a classificação dos caracteres tenha apresentado erros em determinados casos, como nas placas "LSN4I49" e "GB4795". Essa distinção entre detecção e reconhecimento é relevante, pois, nos cinco casos analisados, foram observados menos problemas na localização das regiões textuais do que na classificação dos caracteres. Esse resultado exploratório sugere que o reconhecimento dos caracteres constitui um dos principais pontos a serem investigados em trabalhos futuros.

Figura 10 - Número de elementos textuais detectados por placa veicular (Corretos vs. Incorretos).

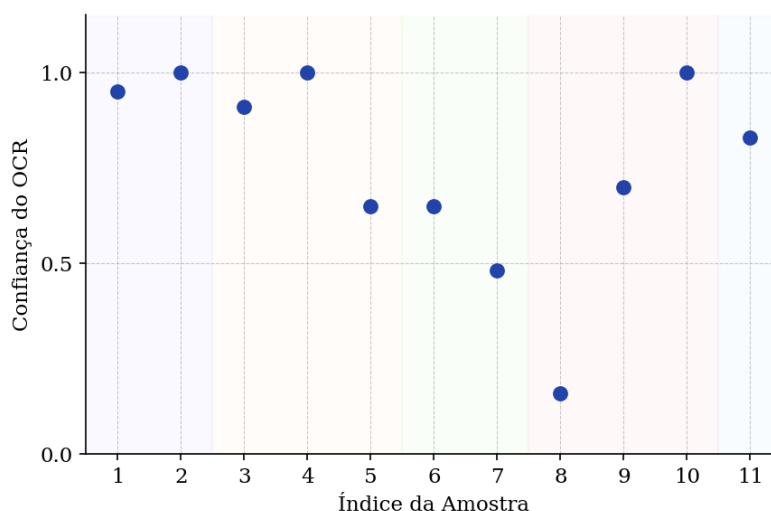


Fonte: Elaborado pelo autor, 2026.

Os valores de confiança do OCR para todos os elementos textuais detectados são apresentados na Figura 11. Considerando os 11 elementos no total, a confiança média foi de, aproximadamente, 0,75, com desvio-padrão de 0,25. Os resultados indicam que, embora a maioria das detecções alcance valores elevados de confiança (próximos a 1,0), casos com condições visuais desfavoráveis reduzem substancialmente a confiabilidade do sistema, como observado no elemento com confiança de 0,16 na placa "BCD-3456". Essa

variabilidade indica o interesse em investigar etapas de pré-processamento de imagem como estratégia para homogeneizar as condições de entrada e potencialmente melhorar o reconhecimento dos caracteres.

Figura 11 - Valores de confiança do OCR para todos os elementos textuais detectados. Cada ponto representa um elemento detectado.



Fonte: Elaborado pelo autor, 2026.

De modo geral, nos casos experimentais analisados, os melhores resultados de reconhecimento foram observados nas imagens com condições visuais mais favoráveis, enquanto erros de classificação ocorreram em situações envolvendo caracteres visualmente semelhantes ou elementos textuais de menor tamanho. Devido ao caráter exploratório da avaliação, essas observações não devem ser generalizadas para outros conjuntos de placas ou condições de aquisição. A distinção entre os módulos de detecção e reconhecimento, evidenciada pela análise, contribui para a identificação precisa dos pontos de melhoria do sistema. O design modular e a execução interativa do reconhecimento OCR permitem a visualização imediata dos resultados, dos valores de confiança e dos tempos de processamento, características úteis para avaliação do sistema em ambiente laboratorial controlado.

Nos cinco casos experimentais avaliados, as regiões textuais foram localizadas pelo sistema e apresentadas por meio das respectivas caixas delimitadoras. A interface com exibição contínua do fluxo de vídeo possibilitou operação interativa e visualização imediata dos resultados. O armazenamento estruturado em formato JSON permite análise posterior dos dados das detecções. As variações de confiança são sinalizadas visualmente por meio de codificação por cores na interface, com verde, para alta confiança; amarelo, para confiança intermediária; e vermelho, para baixa confiança, fornecendo *feedback* imediato ao operador. No contexto dessa prova de conceito, a integração entre o *framework* qi do NAO e as bibliotecas de visão computacional demonstraram a viabilidade técnica da arquitetura *off-board* implementada, permitindo futuras extensões para reconhecimento de outros tipos de texto ou integração com módulos de controle de movimento para rastreamento automático de alvos.

## 5 LIMITAÇÕES

Como limitação, destaca-se que a avaliação foi realizada com cinco casos experimentais, utilizando imagens de placas exibidas em monitor e sob condições laboratoriais controladas. Consequentemente, os resultados não permitem estimar o desempenho do sistema em tráfego real nem avaliar de forma abrangente

fatores como movimento, oclusão, variações de distância, perspectiva e iluminação. Além disso, os tempos registrados correspondem especificamente à execução do módulo OCR sob demanda. Assim, os resultados devem ser interpretados como evidências exploratórias da viabilidade da integração entre o NAO e o processamento *off-board*, e não como uma avaliação de desempenho generalizável de um sistema ALPR.

## 6 CONCLUSÕES

O sistema desenvolvido demonstrou a viabilidade técnica da integração entre a câmera do robô humanoide NAO, uma arquitetura de processamento *off-board* e a biblioteca EasyOCR em ambiente laboratorial controlado. Nos cinco casos experimentais avaliados, os tempos de processamento do módulo OCR variaram entre 0,070 e 0,345 segundos. A arquitetura adotada permitiu manter o fluxo de imagens proveniente do robô e executar o reconhecimento sob demanda em um computador externo.

A análise dos casos revelou comportamentos distintos entre as etapas de detecção e reconhecimento. As regiões textuais foram localizadas nos casos avaliados, enquanto o reconhecimento dos caracteres apresentou erros relacionados a caracteres visualmente semelhantes, como nas confusões entre "l" e "1" e entre "G" e "6". Esses resultados exploratórios indicam que o módulo de reconhecimento de caracteres e as condições de entrada das imagens constituem pontos relevantes para o aprimoramento do sistema.

Considerando que a avaliação foi realizada com cinco casos e em condições laboratoriais controladas, os resultados não permitem generalizações sobre o desempenho do sistema em cenários veiculares reais. Assim, o estudo deve ser interpretado como uma prova de conceito da integração proposta e como uma caracterização experimental preliminar de seu comportamento.

Trabalhos futuros poderão ampliar o conjunto experimental, incorporar técnicas adaptativas de pré-processamento, aplicar regras sintáticas ou expressões regulares para validação dos padrões de placas brasileiras, avaliar a latência completa do *pipeline* em operação contínua e integrar o sistema aos módulos de movimento do NAO para rastreamento automático de alvos.

## DECLARAÇÃO DE USO DE INTELIGÊNCIA ARTIFICIAL

O presente trabalho utilizou ferramentas de inteligência artificial exclusivamente para revisão ortográfica e gramatical do texto. Todo o conteúdo, argumentação, análise e conclusões são de inteira responsabilidade dos autores.

## REFERÊNCIAS

- ALDEBARAN ROBOTICS. **NAO Technical Documentation**. Softbank Robotics, 2021. Disponível em: [http://doc.aldebaran.com/2-8/home\\_nao.html](http://doc.aldebaran.com/2-8/home_nao.html). Acesso em: 2 set. 2025.
- ANAGNOSTOPOULOS, Christos-Nikolaos E. et al. License plate recognition from still images and video sequences: A survey. **IEEE Transactions on Intelligent Transportation Systems**, v. 9, n. 3, p. 377-391, 2008.
- BAEK, Y.; LEE, B.; HAN, D.; YOON, S.; LEE, H. Character region awareness for text detection. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**, 2019, Long Beach. Anais... Long Beach: IEEE, 2019. p. 9365-9374. DOI: [10.1109/CVPR.2019.00959](https://doi.org/10.1109/CVPR.2019.00959).
- BLANSON HENKEMANS, O. **NAO robot by Aldebaran Robotics**. 2025. Disponível em: [\[https://www.aldebaran.com/en/nao\]](https://www.aldebaran.com/en/nao) (<https://www.aldebaran.com/en/nao>). Acesso em: 21 maio 2026.
- CHEN, Li et al. Beyond human recognition: A CNN-based framework for handwritten character recognition. In: 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR). IEEE, 2015. p. 695-699.
- CHEN, Song et al. A real-time face detection and recognition system for a mobile robot in a complex background. **Artificial Life and Robotics**, v. 15, n. 4, p. 439-443, 2010.
- FICHT, Grzegorz et al. NimbRo-OP2X: Adult-sized open-source 3D printed humanoid robot. In: 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids). IEEE, 2018. p. 1-9.
- GOUAILLIER, D.; HUGEL, V.; BLAZEVIC, P.; KILNER, C.; MONCEAUX, J.; LAFOURCADE, P.; MARNIER, B.; SERRE, J.; MAISONNIER, B. Mechatronic design of NAO humanoid. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2009, Kobe. Proceedings... Kobe: IEEE, 2009. p. 769-774. DOI: [10.1109/ROBOT.2009.5152516](https://doi.org/10.1109/ROBOT.2009.5152516). Disponível em: <https://ieeexplore.ieee.org/document/5152516>. Acesso em: 10 set. 2024.
- GOUAILLIER, David et al. The nao humanoid: a combination of performance and affordability. arXiv preprint arXiv:0807.3223, 2008.
- HEREMANS, François et al. Bio-inspired balance controller for a humanoid robot. In: IEEE INTERNATIONAL CONFERENCE ON BIOMEDICAL ROBOTICS AND BIOMECHATRONICS, 6., 2016. Anais... IEEE, 2016. p. 441-448.
- GURUPRASAD, Arvind; PAUL RAJ, Kevin; NAGARATHNA, N. Automatic License Plate Recognition Under Harsh Conditions. In: Soft Computing and Signal Processing: Proceedings of 3rd ICSCSP 2020, Volume 1. **Singapore: Springer Singapore**, 2021. p. 705-716.
- LG ELECTRONICS. **Monitor TV LED LCD 21.5" M2280A**. 2025. Disponível em: <https://www.lg.com>. Acesso em: 21 maio 2026.
- LI, Hui et al. Show, attend and read: A simple and strong baseline for irregular text recognition. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. 2019. p. 8610-8617.
- LONG, S.; HE, X.; YAO, C. Scene text detection and recognition: The deep learning era. **International Journal of Computer Vision**, New York, v. 129, n. 1, p. 161-184, 2020. DOI: [10.1007/s11263-020-01369-0](https://doi.org/10.1007/s11263-020-01369-0). Disponível em: <https://link.springer.com/article/10.1007/s11263-020-01369-0>. Acesso em: 11 set. 2024.

LUO, Jing et al. Robotics perception and control: Key technologies and applications. **Micromachines**, v. 15, n. 4, p. 531, 2024.

MAGALLÁN-RAMÍREZ, D. et al. Implementation of NAO Robot Maze Navigation Based on Computer Vision and Collaborative Learning. **Frontiers in Robotics and AI**, v. 9, p. 834021, 2022. DOI: [10.3389/frobt.2022.834021](https://doi.org/10.3389/frobt.2022.834021).

MANAKITSA, Nikoleta et al. A review of machine learning and deep learning for object detection, semantic segmentation, and human action recognition in machine and robotic vision. **Technologies**, v. 12, n. 2, p. 15, 2024.

MEMON, Jamshed et al. Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR). **IEEE Access**, v. 8, p. 142642-142668, 2020.

MONTAZZOLLI, Sérgio; JUNG, Claudio. Real-time brazilian license plate detection and recognition using deep convolutional neural networks. In: 2017 30th SIBGRAPI Conference on Graphics, patterns and Images (SIBGRAPI). IEEE, 2017. p. 55-62.

WANG, Tianwei et al. Decoupled attention network for text recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2020. p. 12216-12224.

WIKIPEDIA. Placas de identificação de veículos no Brasil. 2025. Disponível em: [https://pt.wikipedia.org/wiki/Placas\\_de\\_identifica%C3%A7%C3%A3o\\_de\\_ve%C3%ADculos\\_no\\_Brasil](https://pt.wikipedia.org/wiki/Placas_de_identifica%C3%A7%C3%A3o_de_ve%C3%ADculos_no_Brasil). Acesso em: 21 maio 2026.