

# Aplicação de um Veículo Guiado Automatizado (AGV) no Contexto de IoT Aplicada à Inteligência Perimetral

Pedro E. Saraiva

*Instituto Federal de Santa Catarina – Engenharia Mecatrônica,  
Florianópolis, Brasil (e-mail: pedro.es@aluno.ifsc.edu.br)*

---

**Abstract:** This project aims to implement an Autonomous Guided Vehicle (AGV), specifically the Robotino, to enhance perimeter intelligence in an integrated surveillance system, Defense IA, using a Closed-Circuit Television (CCTV). The objectives include automating the control of Robotino in response to external events, autonomously directing it to points of interest, generating digital notifications (events) through Defense IA applied to CCTV, and integrating all systems via a middleware with data management and processing capabilities, allowing efficient exchange of information between all devices involved in the system. Using MQTT and HTTP communication protocols (API commands), along with strategically positioned ESP32 devices as anchors around the CCTV scenario, the desired outcome is a prototype of an autonomous robot capable of moving towards points of interest in response to perimeter intelligence events, significantly contributing to automation and efficiency in security validation.

**Resumo:** Este projeto tem como objetivo principal a implementação de um Veículo Autônomo Guiado (AGV), especificamente o Robotino, para aprimorar a inteligência perimetral em um sistema integrado de vigilância, Defense IA, utilizando um Circuito Fechado de Televisão (CFTV). Com objetivos de automatizar o controle do Robotino em resposta a eventos externos, direcioná-lo autonomamente a pontos de interesse, gerar notificações digitais (eventos) por meio do Defense IA aplicado ao CFTV e integrar todos os sistemas via rede, a integração será realizada por meio de um middleware, com capacidades de gerenciamento e processamento de dados, permitindo a troca eficiente de informações entre todos os dispositivos envolvidos no sistema. Utilizando protocolos de comunicação MQTT e HTTP (comandos API), em conjunto com dispositivos ESP32 posicionados estrategicamente como âncoras ao redor do cenário do CFTV, o resultado almejado é um protótipo de robô autônomo capaz de deslocar-se em direção a pontos de interesse em resposta a eventos de inteligência perimetral, contribuindo significativamente para a automação e eficiência na validação de segurança.

**Keywords:** System integration. Software development. IP CCTV. Mobile robotics. Electronic security.

**Palavras-chaves:** Integração de sistemas. Desenvolvimento de Software. CFTV IP. Robótica móvel. Segurança eletrônica.

---

## 1. INTRODUÇÃO

Sistemas inteligentes e autônomos têm como objetivo reduzir a carga de trabalho humana em atividades designadas; apresentam maior confiabilidade, repetibilidade e, geralmente, maior precisão. A integração dessas tecnologias na vida cotidiana torna-se uma necessidade cada vez mais urgente, influenciando diretamente a segurança pública e o bem-estar da sociedade. À medida em que robôs autônomos tornam-se protagonistas em diversos setores, desde a indústria, à assistência médica, suas aplicações começam a remodelar a maneira como enfrentamos desafios cotidianos. A ascensão desses auxiliares robóticos promete impactar positivamente a vida das pessoas, proporcionando eficiência, inovação e segurança. Setores diversos já colheram benefícios significativos com a implementação desses sistemas. A logística, por exemplo, testemunhou melhorias substanciais na gestão de estoques e distribuição. Na saúde, cirurgias assistidas por robôs estão elevando os padrões de precisão e recuperação. Em paralelo, a indústria automotiva vislumbra

um futuro de veículos autônomos que prometem não apenas revolucionar o transporte, mas também otimizar a segurança viária. À medida que essas inovações tecnológicas se integram à rotina, torna-se viável enxergar o investimento em segurança pública como uma oportunidade estratégica e necessária. Ambientes de alto tráfego, como shoppings centers, estádios, aeroportos e grandes eventos, estão adotando sistemas autônomos para aprimorar a vigilância e resposta a situações críticas. A constante busca por acertabilidade na segurança e a necessidade de garantir redundância em ambientes dinâmicos, onde a quantidade de informação e metadados é massiva, tornam-se desafios iminentes. Nesse contexto, será explorado como esses sistemas autônomos não apenas transformam a paisagem tecnológica, mas também demandam estratégias inovadoras para garantir a segurança pública de maneira eficaz e abrangente.

## 2. MOTIVAÇÃO

A premissa do presente trabalho de conclusão de curso é fundamentada em uma interseção estratégica entre a

engenharia mecatrônica e o cenário prático proporcionado pelo estágio na renomada empresa brasileira Intelbras, líder no mercado brasileiro de segurança eletrônica. Originada a partir de uma oportunidade identificada durante essa experiência profissional, a decisão de integrar essas duas áreas culminou na definição do tema central do projeto: desenvolver uma integração para construir um sistema de segurança perimetral autônomo.

A escolha do Robotino, fornecido pela empresa alemã Festo e disponibilizado pela universidade para a execução do Trabalho de Conclusão de Curso, revelou-se como uma decisão estratégica. Adicionalmente, a utilização do software Defense IA, desenvolvido pela Intelbras e familiar durante o estágio, agrega uma camada crucial de inteligência perimetral ao projeto.

A figura abaixo ilustra um cenário exemplar do tema em questão, demonstrando a sinergia entre o Robotino, atuando como um robô autônomo, e o Defense IA, desempenhando o papel de agente de inteligência perimetral. A imagem evidencia de que forma a integração entre esses componentes em conjunto à aplicação de diversas inteligências ao cenário podem ser eficazes, fluindo de maneira coordenada.

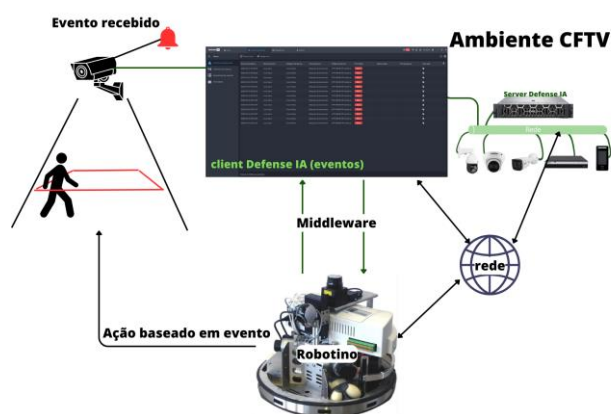


Fig. 1 Cenário exemplo

Um sistema intermediador (Middleware) pode assumir a responsabilidade pela comunicação entre os sistemas, processando informações provenientes de diversas tecnologias e utilizando diferentes protocolos. Dessa forma, ele desempenha o papel crucial de integrar sistemas diversos. Com base nessas premissas, é possível estabelecer objetivos, geral e específicos, para a execução do trabalho, delineando metas a serem alcançadas ao longo do desenvolvimento do protótipo. Estes resumem-se em aplicar um AGV (Robotino) no contexto de inteligência perimetral em um sistema de vigilância integrado a um CFTV para efetuação de rondas realizadas a partir de eventos específicos.

### 3. DESENVOLVIMENTO E RESULTADOS

O Robotino é um robô com um computador integrado e sistema operacional Linux, apresentando softwares próprios e APIs para programação. Os softwares próprios oferecem interface gráfica intuitiva, enquanto as APIs são compatíveis com C, C++ e Java. Para alcançar o objetivo final, o diagrama abaixo apresenta o papel da interface, que deve ser capaz de receber comandos externos (contendo pacotes de dados).

#### 3.1 Open Robotino API 2

A API 2 é uma arquitetura mais moderna comparada à API 1. Ela não contém exemplos ou documentação integrada aos códigos, mas sua estrutura é mais organizada e eficiente. A API 2 é baseada no protocolo SOAP, que é um padrão de comunicação para serviços web. O SOAP é um protocolo baseado em XML que utiliza o formato WSDL para descrever a estrutura de suas solicitações. Este protocolo funciona da seguinte forma: 1. Um cliente envia uma solicitação HTTP para o servidor; 2. O servidor recebe a solicitação e processa-a; 3. O servidor envia uma resposta HTTP para o cliente.

A solicitação HTTP é composta por um cabeçalho e um corpo. O cabeçalho contém informações sobre a solicitação, como o tipo de mensagem, o método e o URI do serviço web. O corpo contém os dados da solicitação. A resposta HTTP também é composta por um cabeçalho e um corpo. O cabeçalho contém informações sobre a resposta, como o código de status HTTP e o tipo de mensagem. O corpo contém os dados da resposta. A estrutura da API 2 é disponibilizada num servidor web instalado no Robotino, alocado na porta padrão web 80; essa porta possui uma interface onde é possível acessá-la, enviando e recebendo solicitações HTTP.

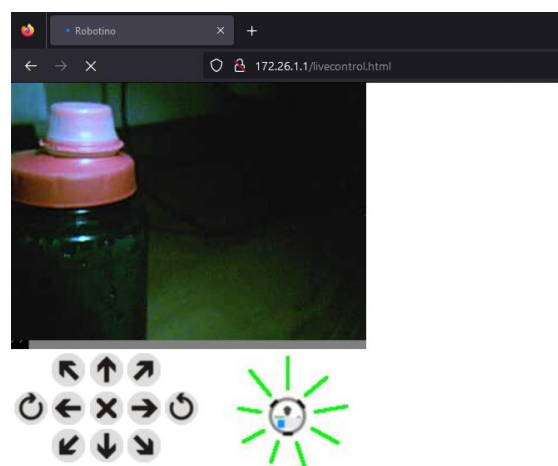


Fig. 2 Interface Web – Robotino

A interface de usuário é dividida entre 3 quadros, um espaço para visualização da web-cam conectada ao robô, apresentando imagens em formato .jpg que são atualizadas a uma taxa de 15 FPS. Um segundo espaço apresenta o estado dos sensores de proximidade e colisão do Robotino, enquanto o terceiro quadro contém botões para controle da navegação do Robô.

Estes comandos são enviados pela própria interface web, a partir de funções descritas em Javascript. São limitados e não possuem camadas de segurança, tampouco estrutura para troca de dados.

Para utilizar a interface de programação da API 2 para acessar os módulos do Robotino a partir da porta 80, deve-se realizar

uma autenticação e manter a conexão utilizando serviços ativos. Durante o próximo capítulo será explicado como a conexão com a API 2 é realizada e como seus serviços são consumidos

### 3.2 Configuração de dispositivos ESP32

Um sistema de posicionamento local será desenvolvido utilizando parâmetros de rede, como a intensidade do sinal entre o robô e dispositivos âncoras, para que o Robotino consiga navegar para uma direção conhecida.

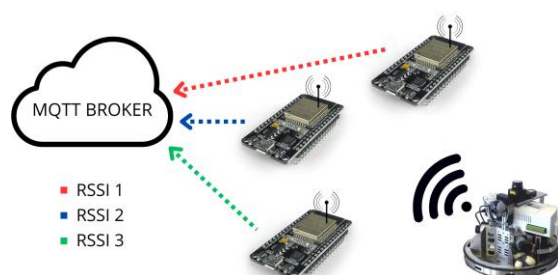


Fig. 3 Captura de intensidade do sinal do Robotino

Para a implementação do sistema de posicionamento, foram escolhidas as ESP32 como dispositivos âncoras devido à sua capacidade de integração com outros dispositivos através de uma rede wireless. As ESP32s são microcontroladores versáteis que oferecem suporte a Wi-Fi e Bluetooth, tornando-as ideais para a comunicação e a medição de intensidade do sinal (RSSI) entre elas e o Robotino. A imagem acima apresenta o fluxo de informação do sistema de posicionamento, onde dispositivos ESP32 conectam-se na (Wireless Local Area Network) do Robotino, cada dispositivo é responsável por capturar a intensidade do sinal recebido (RSSI) e o enviar a um broker MQTT.

### 3.3 Integração de sistemas

o âmbito deste estudo, a integração com o software de segurança eletrônica, Defense IA, é um componente fundamental. Conforme estabelecido pela estrutura lógica do software, é imperativo a implementação de um middleware para permitir o fluxo de dados entre a plataforma do Defense IA e os demais sistemas envolvidos, que incluem o Robotino e o sistema de localização interno. A Figura a seguir ilustra o esquema de intercâmbio de dados entre esses sistemas e o Defense IA por meio do Middleware.

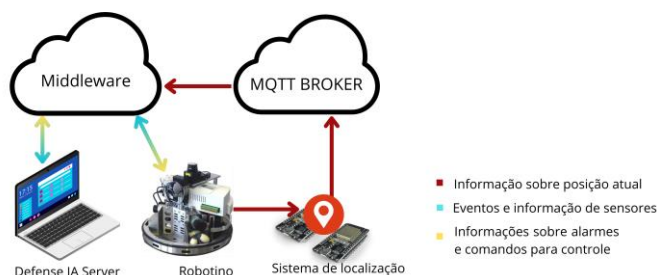


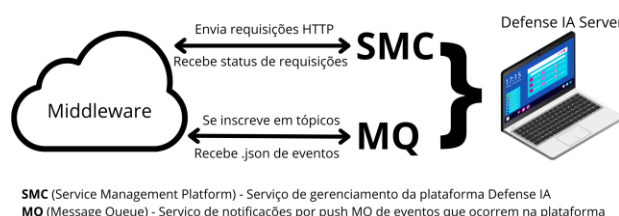
Fig. 4 Diagrama de comunicação de sistemas

#### 3.3.1 Integração com o sistema de localização

As informações sobre a intensidade do sinal entre o Robotino e âncoras estrategicamente posicionadas serão transmitidas através de um broker MQTT. Esses dados serão processados pelo Middleware para referenciar a posição do Robotino no espaço. Com o objetivo de simplificar o desenvolvimento, o Middleware não deve incorporar algoritmos de navegação complexos, mas sim adotar uma lógica de referenciamento simples, utilizando apenas um eixo espacial como trajetória de navegação. Essa abordagem permitirá validar a capacidade de navegação autônoma do Robotino a partir do acionamento remoto por eventos do Defense IA.

#### 3.3.2 Integração com o Defense IA

Na discussão sobre a integração com o Defense IA, é crucial entender a estrutura modular do software, destacando-se por uma ampla variedade de funções e comandos. A arquitetura cliente-servidor do Defense IA reforça a necessidade de uma API para facilitar a comunicação entre cliente e servidor. A API, baseada no modelo RESTFUL, converte cada solicitação do cliente em um comando correspondente, utilizando o protocolo HTTP para efetuar a comunicação de dados com os serviços da plataforma. A comunicação via comandos API desempenha um papel crucial na interoperabilidade e na capacidade de integrar funcionalidades ao Defense IA, aspectos fundamentais abordados na metodologia. A integração entre outros sistemas e o software Defense IA ocorre em duas etapas distintas, implementadas durante o desenvolvimento do middleware. Uma etapa envolve o uso de um cliente HTTP para enviar e receber informações do serviço de gerenciamento do sistema do Defense IA. A estruturação de informações, conforme representada na imagem a seguir, permite gerar outputs desejados (eventos) no sistema.



Já a segunda etapa, um cliente MQTT deve ser desenvolvido com objetivo de conectar-se ao broker próprio do Defense IA, desta forma, é possível acompanhar o fluxo de todos os eventos CFTV ou digitais que ocorrem durante a plataforma. Ao final das duas etapas, será então possível receber e publicar eventos ao software de segurança eletrônica de forma automática a partir de comandos do Middleware.

#### 3.4 Consrtrução do Middleware

Desenvolvido utilizando a linguagem de programação Go, criada pela Google®, o papel do Middleware é receber todos os dados utilizados durante o sistema, tratá-los e distribuir comandos a partir de tais inputs. A figura abaixo apresenta a ordem de execução das rotinas, representando o comportamento do sistema.

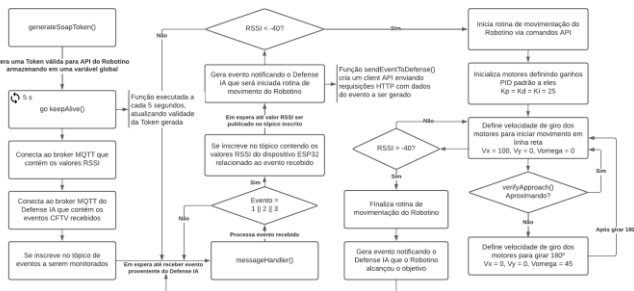


Fig. 5 Fluxograma do Middleware

O Middleware inicia suas operações executando rotinas cruciais. Isso inclui a geração de uma token de conexão para possibilitar o acesso às funcionalidades do Robotino por meio de comandos API. Além disso, estabelece conexões com os brokers MQTT, do sistema de localização e do Defense IA. Essa abordagem permite a obtenção de informações vitais, como os valores RSSI provenientes das ESP32, bem como o acesso aos eventos registrados pela plataforma Defense IA por meio de câmeras de monitoramento. Ao centralizar essas informações no Middleware, torna-se viável a criação de rotinas de navegação para o Robotino, baseadas no processamento lógico desses dados. Essa integração estratégica representa um ponto crucial no desenvolvimento, permitindo a otimização da tomada de decisões e aprimorando a eficiência operacional do sistema como um todo.

### 3.5 Inteligência perimetral

A fim de gerar notificações de eventos ao software de segurança eletrônica, Defense IA, um cenário simples de CFTV foi construído, representando uma aplicação real de inteligência perimetral. O cenário construído é composto pelo Servidor do Defense IA, um cliente do Defense IA, câmeras IP de monitoramento e um roteador de rede.

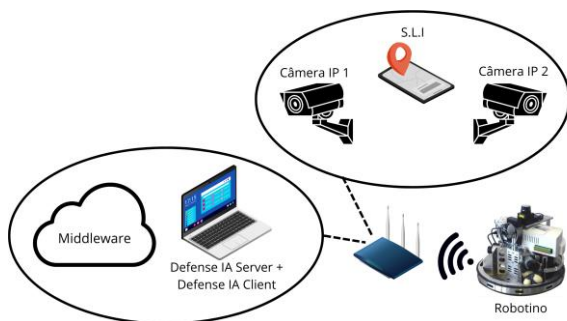


Fig. 5 Diagrama de cenário CFTV

Todos os dispositivos devem se conectar ao roteador, que também deve estar endereçado na mesma rede que o Robotino, o sistema de localização interno e o Middleware, desta forma é possível configurar as câmeras de monitoramento para enviar eventos de CFTV para notificação no Defense IA e direcioná-los ao Middleware. Para o protótipo, o sistema de localização interno (S.L.I) foi vinculado às câmeras, no qual um dispositivo ESP32 representa a posição de uma câmera IP. O Middleware é executado na mesma máquina onde executa-se o servidor do Defense IA e o cliente do Defense IA. Todos os dispositivos conectam-se à WLAN fornecida pelo Robotino.

### 3.6 Acionamento do Robotino em operação

Com todas as funções estruturadas, é possível realizar a integração de todos os sistemas, uma vez que todas as tecnologias se comunicam utilizando o Middleware. O fluxo, executado em etapas é apresentado a seguir.

- Etapa 1: Todos os sistemas, conectados à mesma rede, são inicializados. O Middleware inicializado gera uma token de conexão ao Robotino e a mantém, esperando um evento proveniente do Defense IA.
- Etapa 2: Evento é gerado na plataforma. Eventos cadastrados no Middleware: 1. Reconhecimento Facial; 2. Acionamento por software; 3. Desconexão de dispositivo. Cada evento é vinculado a um dispositivo ESP32, estabelecendo 3 (três) pontos de interesse no espaço.
- Etapa 3: Evento é recebido pelo Middleware e Robotino inicia rotina de movimentação.
- Etapa 4: Robotino alcança objetivo e envia evento ao Defense IA. O Middleware mantém a token de conexão com Robotino, esperando um novo evento proveniente do Defense IA.

## 4. CONCLUSÕES

Este projeto visou o desenvolvimento de um protótipo de integração entre sistemas utilizando diversas tecnologias e abordagens. O objetivo final não vincula-se a apenas o desenvolvimento aplicado. Além do aprofundamento dos aplicativos, diferentes métodos e materiais poderiam ser utilizados para alcançar os objetivos. Alguns foram estudados e serão aqui explorados.

Ao avaliar os objetivos delineados, observa-se que os resultados obtidos foram satisfatórios, alcançando todas as metas estabelecidas. Embora persistam itens passíveis de melhorias, o sistema demonstrou eficácia ao cumprir os requisitos predefinidos. Destaca-se o protagonismo do Middleware no contexto do protótipo, desempenhando a função crucial de integrar e conectar todas as tecnologias envolvidas. Essa peça central revelou-se como o elemento unificador, consolidando a harmonia entre as diferentes partes do sistema.

A comunicação do sistema ocorre por meio de uma rede wireless, e durante o desenvolvimento, identificou-se um gargalo significativo. Durante a rotina de navegação do Robotino, observou-se um volume elevado de dados sendo transmitidos pela rede sem fio, resultando em latência considerável no sistema. Como consequência, a execução da rotina de navegação não se mostrou fluida, frequentemente apresentando interrupções.

Os resultados do protótipo demonstraram eficazmente o acionamento do Robotino em resposta a eventos gerados pelo Defense IA, indicando uma integração bem-sucedida entre os sistemas. Contudo, para validar conceitualmente a ideia, restringiu-se a navegação do Robotino a apenas um eixo. Essa limitação, embora útil para os propósitos iniciais, não explora completamente o potencial do Robotino em termos de movimentação tridimensional.



Com o intuito de aprimorar a capacidade de navegação do Robotino em três eixos, seria vantajoso explorar a implementação de algoritmos de navegação. Utilizar algoritmos adequados permitiria ao Robotino calcular trajetórias eficientes em espaços tridimensionais, otimizando rotas em tempo real. Essa abordagem não apenas expandiria significativamente as capacidades do protótipo, possibilitando uma navegação mais versátil e adaptável, mas também elevaria a precisão e eficiência na execução de tarefas específicas.

Ao incorporar algoritmos de navegação, o protótipo se beneficiaria de uma abordagem flexível, permitindo sua adaptação a diferentes contextos e ambientes. Essa escolha estratégica não apenas enriqueceria a experiência de navegação do Robotino, mas também fortaleceria a utilidade prática do sistema, tornando-o mais eficaz em ambientes dinâmicos e complexos.

INTELBRAS. Manual Defense IA. 2023. Disponível em: [manuais.intelbras.com.br/ manual-software-de-seguranca-eletronica-defense-ia/](https://manuais.intelbras.com.br/manual-software-de-seguranca-eletronica-defense-ia/).

KRACHT, S.; NIELSEN, C. Robots in Everyday Human Environments. 130 p. Dissertação (Mestrado em Sistemas autônomos inteligentes) — Aalborg University - AAU, 2007. Disponível em: <https://core.ac.uk/download/pdf/60678919.pdf>.

RUIZ, A. M. Estudi d'estratègies de seguiment de línia per al robot mòbil robotino. UPC - Escola d'Enginyeria de Terrassa, 2016. Disponível em: <http://hdl.handle.net/2117/107423>.

SHARP. Datasheet - GP2D12. 2023. Disponível em: [https://engineering.purdue.edu/ME588/SpecSheets/sharp\\_gp2d12.pdf](https://engineering.purdue.edu/ME588/SpecSheets/sharp_gp2d12.pdf).

## REFERÊNCIAS

AMAZON. api. 2023. Disponível em: <https://aws.amazon.com/pt/what-is/api/>.

AMAZON. middleware. 2023. Disponível em: <https://aws.amazon.com/pt/what-is/middleware/>.

AMAZON. mqtt. 2023. Disponível em: <https://aws.amazon.com/pt/what-is/mqtt/>.

BEELEN, T.; BERNARD, T.; PAPENMEIER, A. A robot control system for child-robot interaction. HMI - University of Twente, 2016. Disponível em: [https://andrea.papenmeier.io/res/papers/2016\\_HMIp.pdf](https://andrea.papenmeier.io/res/papers/2016_HMIp.pdf).

BELLECIERI, Y.; JABOUR, F. C.; JABOUR, E. G. Localização indoor baseada na leitura bidirecional do rssi. IF Sudeste MG, 2016. Disponível em: [https://www.academia.edu/66369578/Localiza%C3%A7%C3%A3o\\_Indoor\\_Baseada\\_na\\_Leitura\\_Bidirecional\\_do\\_RSSI](https://www.academia.edu/66369578/Localiza%C3%A7%C3%A3o_Indoor_Baseada_na_Leitura_Bidirecional_do_RSSI).

ESPRESSIF. Datasheet - ESP32. 2023. Disponível em: [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e\\_esp32-wroom-32ue\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf).

FESTO. Robotino. 2023. Disponível em: [https://www.festo.com/br/pt/p/sistema-de-robo-movel-id\\_PROD\\_DID\\_8101344/?page=0](https://www.festo.com/br/pt/p/sistema-de-robo-movel-id_PROD_DID_8101344/?page=0).

INTELBRAS. Intelbras líder. 2023. Disponível em: <https://www.intelbras.com/pt-br/noticia/lider-brasileira-intelbras-expande-fronteiras-na-america-latina-marcando-presenca-em-evento>.

INTELBRAS. Inteligência perimetral. 2023. Disponível em: <https://www.intelbras.com/pt-br/linha-future>.