

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA - CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA MECATRÔNICA**

MATHEUS REUS VIEIRA

**REALIDADE AUMENTADA INTERATIVA NA INDÚSTRIA 4.0: monitoramento e
ajuste de parâmetros em tempo real**

FLORIANÓPOLIS, 2026.

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA - CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA MECATRÔNICA**

MATHEUS REUS VIEIRA

**REALIDADE AUMENTADA INTERATIVA NA INDÚSTRIA 4.0: monitoramento e
ajuste de parâmetros em tempo real**

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para obtenção do título de Engenheiro em Mecatrônica.

Orientador:
Prof. Gregory Chagas da Costa Gomes,
Msc.

FLORIANÓPOLIS, 2026.

Ficha de identificação da obra elaborada pelo autor.

Vieira, Matheus Reus

REALIDADE AUMENTADA INTERATIVA NA INDÚSTRIA 4.0: monitoramento e ajuste de parâmetros em tempo real / Matheus Reus Vieira; orientação de Gregory Chagas da Costa Gomes. - Florianópolis, SC, 2026.
101 p.

Trabalho de Conclusão de Curso (TCC) - Instituto Federal de Santa Catarina, Câmpus Florianópolis. Bacharelado em Engenharia Mecatrônica. Departamento Acadêmico de Metal Mecânica.
Inclui Referências.

1. Realidade Aumentada. 2. Interface homem-máquina.
3. Visão computacional. 4. Modbus. 5. FastAPI. I. Gomes, Gregory Chagas da Costa. II. Instituto Federal de Santa Catarina. III. REALIDADE AUMENTADA INTERATIVA NA INDÚSTRIA 4.0.

REALIDADE AUMENTADA INTERATIVA NA INDÚSTRIA 4.0: MONITORAMENTO E AJUSTE DE PARÂMETROS EM TEMPO REAL

MATHEUS REUS VIEIRA

FLORIANÓPOLIS, 06 de julho, 2026.

Banca Examinadora:

Gregory Chagas da Costa Gomes, Msc.

Mauricio Edgar Stivanello, Dr. Eng.

Cassiano Bonin, Msc.

Dedico este trabalho a Deus, pela força
em toda a jornada, e ao meu avô Divino
Marcelino Vieira, que sempre me apoiou e
sonhou com meu grau de engenheiro.

AGRADECIMENTOS

A realização deste trabalho não teria sido possível sem o apoio, incentivo e presença de pessoas que foram fundamentais em minha vida acadêmica, pessoal e espiritual.

Agradeço a Deus não apenas pela força nos momentos difíceis, mas sobretudo por ser fonte de propósito e inspiração em minha vida. Foi n'Ele que encontrei sentido para prosseguir, esperança para acreditar no futuro e serenidade para compreender que cada etapa desta jornada fez parte de um plano maior.

À minha esposa, expresso uma gratidão que vai além das palavras. Mais do que companheira de vida, foi parceira também nesta caminhada acadêmica: soube compreender minhas ausências, oferecer incentivo nos momentos de desânimo e celebrar comigo cada pequena conquista ao longo do caminho. Sua presença constante e seu amor me lembraram, em todos os instantes, o verdadeiro motivo de perseverar.

Aos meus pais, agradeço pelos valores transmitidos, pelo exemplo de dedicação e por acreditarem em mim mesmo quando eu duvidava. Aos meus avós, sou grato pelo carinho, pelas palavras de incentivo e pela presença em minha vida, que me fortaleceram ao longo desta jornada.

Aos professores e à instituição, deixo minha gratidão pela formação que recebi, pelo conhecimento compartilhado e pelas oportunidades de aprendizado. Em especial, agradeço ao meu orientador, pela paciência, orientação e pela confiança depositada neste trabalho, contribuindo de forma decisiva para meu crescimento acadêmico e profissional.

Aos colegas de curso, agradeço pela parceria, pelas trocas de conhecimento e pelos momentos de companheirismo que tornaram a trajetória mais leve e significativa.

Aos amigos, que de perto ou de longe se fizeram presentes com palavras de apoio, compreensão e incentivo, meu sincero agradecimento. Cada gesto de amizade e solidariedade fez diferença nos momentos mais desafiadores.

Por fim, a todos que, de alguma forma, deixaram sua marca nesta caminhada, deixo aqui o meu reconhecimento e gratidão. Este trabalho é resultado não apenas do meu esforço, mas também da contribuição de todos vocês.

“Na Quarta Revolução Industrial, quem
dominar os dados dominará a produção.”
— Klaus Schwab

RESUMO

A Realidade Aumentada permite apresentar informações contextualizadas diretamente sobre equipamentos industriais, reduzindo a necessidade de consulta a interfaces externas ao processo observado. Este trabalho desenvolveu uma interface homem-máquina aumentada e interativa para monitoramento, supervisão de falhas e ajuste de parâmetros do torno didático MIRAC PC, integrado ao LinuxCNC e ao CLP Siemens S7-1200. A solução adotou uma arquitetura distribuída composta por uma aplicação web progressiva executada em dispositivo móvel, um *backend* local e uma camada intermediária de comunicação industrial via Modbus TCP. O vídeo permaneceu no dispositivo cliente, enquanto quadros pontuais foram enviados por HTTPS para análise. O *backend* retornou respostas estruturadas em JSON, utilizadas pelo *frontend* para renderizar localmente elementos de sobreposição associados a componentes, estados, comandos e indicações de falha. A identificação visual dos componentes foi realizada sem marcadores artificiais fixados à máquina, utilizando um modelo YOLOv8n treinado com imagens do equipamento e combinado a um mecanismo leve de rastreamento em OpenCV. A interação do operador foi estruturada por menus contextuais, gestos configuráveis e comandos por voz com reconhecimento offline, todos encaminhados a um fluxo comum de validação e confirmação. Na validação offline, o detector apresentou precisão de 0,917, revocação de 0,907, F1-score de 0,912, mAP50 de 0,955 e mAP50-95 de 0,748. Nos ensaios operacionais, a solução registrou latência percebida média de 114,37 ms, continuidade visual de 60,67% e taxa de sucesso de 100,00% nas operações de comunicação industrial avaliadas. Os resultados constituem evidências exploratórias da viabilidade da arquitetura para integrar reconhecimento visual, dados industriais, supervisão contextual e interação do operador em uma interface aumentada acessível por dispositivos móveis.

Palavras-chave: Realidade Aumentada. Interface homem-máquina. Visão computacional. Modbus. FastAPI.

ABSTRACT

Augmented Reality enables contextualized information to be displayed directly over industrial equipment, reducing the need to consult interfaces external to the observed process. This study developed an interactive augmented human-machine interface for monitoring, fault supervision, and parameter adjustment of a didactic MIRAC PC lathe integrated with LinuxCNC and a Siemens S7-1200 PLC. The solution adopted a distributed architecture composed of a progressive web application running on a mobile device, a local backend, and an intermediate industrial communication layer based on Modbus TCP. Video remained on the client device, while individual frames were sent through HTTPS for analysis. The backend returned structured JSON responses, which were used by the frontend to render overlay elements locally associated with components, states, commands, and fault indications. Visual identification of components was performed without artificial markers attached to the machine by using a YOLOv8n model trained with images of the equipment and combined with a lightweight OpenCV tracking mechanism. Operator interaction was structured through contextual menus, configurable gestures, and voice commands with offline recognition, all routed through a common validation and confirmation flow. In offline validation, the detector achieved a precision of 0.917, recall of 0.907, F1-score of 0.912, mAP50 of 0.955, and mAP50-95 of 0.748. During operational tests, the solution achieved an average perceived latency of 114.37 ms, visual continuity of 60.67%, and a success rate of 100.00% for the evaluated industrial communication operations. The results provide exploratory evidence of the feasibility of the architecture for integrating visual recognition, industrial data, contextual supervision, and operator interaction into an augmented interface accessible through mobile devices.

Keywords: Augmented Reality. Human-machine interface. Computer vision. Modbus. FastAPI.

LISTA DE FIGURAS

Figura 1 – Contínuo Real-Virtual	25
Figura 2 – Manutenção assistida por Realidade Aumentada	26
Figura 3 – QR-Code para marcação de objeto	27
Figura 4 – Detecção e classificação por modelo YOLO	30
Figura 5 – Exemplo de arquitetura de uma rede Modbus	32
Figura 6 – Exemplo de IHM tradicional	37
Figura 7 – Torno didático MIRAC-PC utilizado no desenvolvimento	44
Figura 8 – Arquitetura simples da solução	45
Figura 9 – Fluxo operacional geral da arquitetura implementada	51
Figura 10 – Tela de operação da interface em visualização desktop e mobile . .	54
Figura 11 – Ambiente de desenvolvimento	54
Figura 12 – Menus contextuais de comandos associados aos componentes . . .	57
Figura 13 – Fluxo da interação por menus flutuantes	58
Figura 14 – Tela de atribuição de gestos aos comandos	59
Figura 15 – Fluxo da interação por gestos configuráveis	60
Figura 16 – Fluxo da interação por comandos de voz	61
Figura 17 – Reconhecimento de comando por voz em operação	62
Figura 18 – Adequações realizadas no CLP Siemens S7-1200	63
Figura 19 – Adequações realizadas no LinuxCNC em ambiente virtual	64
Figura 20 – Interface de configuração e estado da comunicação industrial . . .	66
Figura 21 – Identificação contextual de falha de pressão baixa associada à cas- tanha	68
Figura 22 – Exemplos de imagens anotadas para treinamento do detector	69
Figura 23 – Fluxo de processamento da imagem e renderização dos overlays .	71
Figura 24 – Detecção visual de componentes e sobreposição aumentada	74
Figura 25 – Tela de login em visualização desktop nos temas claro e escuro . .	85
Figura 26 – Tela de cadastro em visualização desktop nos temas claro e escuro	85
Figura 27 – Tela de operação da câmera e overlays em visualização desktop nos temas claro e escuro	86
Figura 28 – Tela de perfil do usuário em visualização desktop nos temas claro e escuro	86
Figura 29 – Tela de configuração de comunicação em visualização desktop nos temas claro e escuro	86
Figura 30 – Tela de comandos por gestos em visualização desktop nos temas claro e escuro	87
Figura 31 – Tela de comandos por voz em visualização desktop nos temas claro e escuro	87

Figura 32 – Tela de configurações gerais em visualização desktop nos temas claro e escuro	87
Figura 33 – Tela de login em visualização tablet nos temas claro e escuro	88
Figura 34 – Tela de cadastro em visualização tablet nos temas claro e escuro . .	88
Figura 35 – Tela de operação da câmera e overlays em visualização tablet nos temas claro e escuro	89
Figura 36 – Tela de perfil do usuário em visualização tablet nos temas claro e escuro	89
Figura 37 – Tela de configuração de comunicação em visualização tablet nos temas claro e escuro	89
Figura 38 – Tela de comandos por gestos em visualização tablet nos temas claro e escuro	90
Figura 39 – Tela de comandos por voz em visualização tablet nos temas claro e escuro	90
Figura 40 – Tela de configurações gerais em visualização tablet nos temas claro e escuro	90
Figura 41 – Tela de login em visualização mobile nos temas claro e escuro . . .	91
Figura 42 – Tela de cadastro em visualização mobile nos temas claro e escuro .	92
Figura 43 – Tela de operação da câmera e overlays em visualização mobile nos temas claro e escuro	93
Figura 44 – Tela de perfil do usuário em visualização mobile nos temas claro e escuro	94
Figura 45 – Tela de configuração de comunicação em visualização mobile nos temas claro e escuro	95
Figura 46 – Tela de comandos por gestos em visualização mobile nos temas claro e escuro	96
Figura 47 – Tela de comandos por voz em visualização mobile nos temas claro e escuro	97
Figura 48 – Tela de configurações gerais em visualização mobile nos temas claro e escuro	98
Figura 49 – Gestos manuais reconhecidos pelo sistema	100
Figura 50 – Gestos complementares e visualizações da câmera	101
Figura 51 – Acionamento por gesto em orientação paisagem	102

LISTA DE QUADROS

Quadro 1 – Síntese comparativa entre o trabalho relacionado e a proposta metodológica	46
---	----

LISTA DE TABELAS

Tabela 1 – Resultados globais da validação offline do detector YOLO	73
Tabela 2 – Resultados consolidados dos ensaios operacionais	75

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
CNN	Rede Neural Convolucional
CPS	Cyber-Physical Systems
HTTP	<i>Hypertext Transfer Protocol</i>
IoT	<i>Internet of Things</i> (Internet das Coisas)
IoU	<i>Intersection over Union</i>
JSON	<i>JavaScript Object Notation</i>
PWA	<i>Progressive Web App</i>
RA	<i>Realidade Aumentada</i>
RV	Realidade Virtual
YOLO	You Only Look Once

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Justificativa	18
1.2	Definição do Problema	18
1.3	Objetivo Geral	19
1.4	Objetivos Específicos	19
1.5	Estrutura do Trabalho	20
2	REVISÃO BIBLIOGRÁFICA	21
2.1	Fundamentação Teórica	21
2.1.1	Indústria 4.0 e Sistemas Ciberfísicos	21
2.1.1.1	<i>Transformação Digital no Ambiente Industrial</i>	22
2.1.1.2	<i>Sistemas Ciberfísicos (CPS)</i>	22
2.1.1.3	<i>Arquiteturas Distribuídas e Integração de Camadas</i>	23
2.1.1.4	<i>Implicações para Interfaces Homem-Máquina</i>	23
2.1.2	Realidade Aumentada e Problema de Registro Espacial	24
2.1.2.1	<i>O Problema de Registro Espacial</i>	26
2.1.2.2	<i>Abordagens Marker-Based e Markerless</i>	27
2.1.2.3	<i>Desafios da RA em Ambientes Industriais</i>	28
2.1.3	Registro Visual Contextual em Ambientes Industriais	28
2.1.4	Visão Computacional para Detecção de Objetos	29
2.1.4.1	<i>Redes Neurais Convolucionais (CNNs)</i>	29
2.1.4.2	<i>Detectores One-Stage e o Modelo YOLO</i>	30
2.1.4.3	<i>Métricas de Avaliação em Detecção de Objetos</i>	31
2.1.4.4	<i>Detecção de Componentes e Continuidade Temporal</i>	31
2.1.5	Redes Industriais e Protocolo Modbus	32
2.1.5.1	<i>Protocolo Modbus e Estrutura de Comunicação</i>	32
2.1.5.2	<i>Arquiteturas Intermediadas e Separação de Camadas</i>	33
2.1.5.3	<i>Abstração de Variáveis e Modelo de Tags</i>	34
2.1.5.4	<i>Escrita Remota e Segurança Operacional</i>	35
2.1.6	APIs e Serviços Web em Arquiteturas Distribuídas	35
2.1.7	Interfaces Homem-Máquina Contextuais	36
2.1.7.1	<i>IHMs Tradicionais e Supervisórios</i>	37
2.1.7.2	<i>Interfaces Contextuais e Informação In Situ</i>	37
2.1.7.3	<i>Carga Cognitiva e Usabilidade em Ambientes Industriais</i>	38
2.1.7.4	<i>Interação Bidirecional e Controle Assistido</i>	38
2.1.8	Latência e Sistemas Interativos em Tempo Real	39
2.1.8.1	<i>Conceitos de Tempo Real em Sistemas Computacionais</i>	39
2.1.8.2	<i>Latência em Sistemas Distribuídos</i>	40
2.1.8.3	<i>Impacto da Latência na Interação Humano-Sistema</i>	40
2.1.8.4	<i>Considerações sobre Robustez e Recuperação de Falhas</i>	41
2.2	Trabalhos Relacionados	41
2.3	Síntese dos Fundamentos Teóricos	42
3	METODOLOGIA	43
3.1	Delineamento da Pesquisa	43
3.2	Arquitetura do Sistema	43
3.2.1	Camada de Percepção Visual	45

3.2.2	Camada de Análise Automática e Rastreamento	46
3.2.3	Camada de Integração Industrial	47
3.2.4	Camada de Interface Aumentada	47
3.3	Plano de Desenvolvimento do Trabalho	47
3.4	Estratégia de Desenvolvimento Iterativo	48
3.5	Procedimentos de Avaliação	48
3.5.1	Desempenho da Detecção	48
3.5.2	Estabilidade Temporal dos Elementos Visuais	49
3.5.3	Latência Sistêmica	49
3.5.4	Confiabilidade da Comunicação	49
3.5.5	Avaliação Exploratória da Interação	50
3.6	Delimitações do Estudo	50
4	APRESENTAÇÃO DOS RESULTADOS	51
4.1	Desenvolvimento da Solução	51
4.1.1	Consolidação da arquitetura distribuída do sistema	52
4.1.2	Desenvolvimento do <i>backend</i> de processamento e integração	52
4.1.3	Desenvolvimento do <i>webapp</i> de visualização	53
4.1.4	Implementação da captura de vídeo e envio de <i>frames</i>	55
4.1.5	Estruturação do modelo de <i>overlays</i> e renderização local	56
4.1.6	Interação por menus flutuantes associados aos componentes	56
4.1.7	Interação por gestos configuráveis	58
4.1.8	Interação por comandos de voz	60
4.1.9	Implementação da comunicação industrial via Modbus	62
4.1.9.1	<i>Adequações realizadas no CLP Siemens S7-1200</i>	62
4.1.9.2	<i>Adequações realizadas no LinuxCNC</i>	63
4.1.10	Consolidação do fluxo integrado	65
4.2	Validação Funcional da Solução	65
4.2.1	Validação da Comunicação entre Cliente e Backend	65
4.2.2	Validação da Renderização Local dos <i>Overlays</i>	66
4.2.3	Validação da Integração Industrial via Modbus	67
4.2.4	Preparação do Conjunto de Dados e Treinamento do Modelo	68
4.2.5	Implementação da Análise Automática com YOLO	70
4.2.6	Fluxo Híbrido de Inferência e Rastreamento	70
4.2.7	Validação dos Recursos de Visão Computacional	71
4.2.8	Instrumentação Automática dos Ensaios Quantitativos	72
4.2.9	Resultados da Validação Offline do Detector	73
4.2.10	Resultados dos Ensaios Operacionais	74
4.2.11	Validação dos Modos de Interação	75
4.2.12	Validação da Interface em Dispositivo Móvel	76
4.2.13	Síntese dos Resultados Funcionais	76
5	CONSIDERAÇÕES FINAIS	78
5.1	Síntese do Trabalho	78
5.2	Limitações do Estudo	79
5.3	Possibilidades de Continuidade	80
	REFERÊNCIAS	82

	APÊNDICES	84
	APÊNDICE A – CADERNO DE TELAS DA APLICAÇÃO EM DESKTOP	85
A.1	Telas em Visualização Desktop	85
	APÊNDICE B – CADERNO DE TELAS DA APLICAÇÃO EM TABLET	88
B.1	Telas em Visualização Tablet	88
	APÊNDICE C – CADERNO DE TELAS DA APLICAÇÃO EM DIS- POSITIVO MÓVEL	91
C.1	Telas em Visualização Mobile	91
	APÊNDICE D – REGISTROS DOS GESTOS RECONHECIDOS . .	99

1 INTRODUÇÃO

Nas últimas décadas, o setor industrial vem passando por uma transformação marcada pelo avanço de tecnologias digitais, fenômeno conhecido como Indústria 4.0. Essa nova etapa da evolução produtiva caracteriza-se pela integração entre máquinas, sensores, sistemas de informação e inteligência artificial, resultando em ambientes cada vez mais conectados e autônomos. Nesse cenário, informações são geradas em grande volume e velocidade, exigindo ferramentas capazes de torná-las acessíveis de maneira rápida e intuitiva para operadores e gestores.

Entre as tecnologias emergentes, a Realidade Aumentada (RA) tem se destacado por permitir que dados virtuais sejam sobrepostos ao ambiente físico, criando uma camada adicional de informação diretamente vinculada ao processo em execução. Diferente de interfaces tradicionais, que exigem consultas a telas externas ou sistemas paralelos, a RA coloca os indicadores de processo diante do próprio equipamento, aproximando o ser humano da operação e abrindo espaço para novas formas de interação.

Contudo, um dos desafios persistentes na aplicação da Realidade Aumentada no ambiente industrial é a manutenção do alinhamento espacial entre os elementos virtuais e os componentes reais das máquinas. Em condições típicas de operação, com variações de iluminação, vibrações, mudança de ângulo de visão e deslocamento do operador, é comum que os objetos virtuais percam sua referência no espaço, tornando-se imprecisos ou deslocados em relação à máquina física. Essa limitação compromete a clareza e a usabilidade da interface, dificultando a leitura de parâmetros e a interação segura do operador com o equipamento.

Embora já existam iniciativas que utilizam a RA no contexto industrial, muitas delas permanecem restritas à visualização de variáveis em tempo real (WANG *et al.*, 2013; PALMARINI *et al.*, 2018). O desafio atual é avançar para soluções que também possibilitem a interação direta do operador, permitindo o ajuste de parâmetros de funcionamento e a tomada de decisão de forma mais ágil e segura. Essa evolução transforma a RA em uma interface homem-máquina de caráter bidirecional, capaz não apenas de informar, mas também de mediar comandos.

Nesse contexto, este trabalho desenvolveu um sistema de Realidade Aumentada voltado à Indústria 4.0, integrando visão computacional, aprendizado de máquina e comunicação com dispositivos de controle. O objetivo foi demonstrar como a RA pode ser aplicada não apenas como ferramenta de monitoramento, mas também como meio interativo para controle operacional, contribuindo para a eficiência, segurança e inovação no chão de fábrica.

1.1 Justificativa

A Indústria 4.0 representa a integração de tecnologias digitais, automação e inteligência artificial aos processos produtivos, resultando em fábricas mais inteligentes, conectadas e eficientes. Nesse contexto, a Realidade Aumentada (RA) surge como uma ferramenta estratégica capaz de aproximar operadores, máquinas e informações em tempo real, favorecendo a tomada de decisão e aumentando a produtividade.

A escolha do tema se justifica pela necessidade crescente de soluções que tornem os ambientes industriais mais interativos e acessíveis, permitindo que trabalhadores não apenas visualizem dados de sensores, parâmetros de funcionamento e alertas diretamente sobre os equipamentos, mas também interajam com essas informações em tempo real, ajustando configurações de forma prática e segura. Essa integração reduz falhas humanas, otimiza o tempo de resposta e contribui para a manutenção preventiva e corretiva.

Além disso, a aplicação da RA, associada a técnicas de visão computacional e aprendizado de máquina, proporciona meios de reconhecimento automático de máquinas e processos, ampliando a autonomia dos sistemas industriais. Esse tipo de abordagem está alinhado às tendências de digitalização e reforça a importância de investigar soluções aplicáveis em ambientes acadêmicos, didáticos e industriais, especialmente quando se busca integrar tecnologias emergentes a equipamentos já existentes.

Do ponto de vista acadêmico, o tema oferece relevância por unir conhecimento teórico e aplicação prática, possibilitando a exploração de conceitos de inteligência artificial, interfaces homem-máquina e automação, ao mesmo tempo em que abre espaço para experimentação com protótipos e simulações.

Assim, o presente trabalho contribui com uma solução acessível para investigar o problema de ancoragem e estabilidade espacial na Realidade Aumentada industrial, ao mesmo tempo em que amplia o papel da RA como ferramenta interativa e bidirecional de controle. O uso de visão computacional para reconhecer os componentes da máquina e ancorar dinamicamente os elementos virtuais favorece uma sobreposição contextual e coerente nos cenários avaliados, permitindo ao operador monitorar e ajustar parâmetros diretamente sobre o equipamento.

1.2 Definição do Problema

A aplicação da Realidade Aumentada em ambientes industriais ainda é limitada por dois grandes desafios complementares.

O primeiro é de natureza funcional: a falta de soluções práticas e integradas

que permitam o monitoramento e o ajuste em tempo real de variáveis de processo, utilizando a RA como interface homem-máquina. Diversas implementações descritas na literatura concentram-se predominantemente na visualização contextual de informações, especialmente em cenários de manutenção, inspeção e suporte ao operador, enquanto a interação direta com o equipamento exige mecanismos adicionais de validação, confirmação e segurança operacional (HENDERSON; FEINER, 2011; WANG *et al.*, 2013; PALMARINI *et al.*, 2018).

O segundo é de natureza técnica: a instabilidade espacial da interface aumentada. Sistemas baseados em marcadores visuais (como QR Codes ou ImageTargets) são suscetíveis a variações de iluminação, obstruções, vibrações e mudanças de perspectiva, o que resulta em desalinhamento entre os elementos virtuais e os componentes físicos. Esse problema compromete a fidelidade visual e a confiabilidade operacional da RA.

Diante disso, este trabalho se propõe a desenvolver um sistema capaz de unir a interatividade operacional em tempo real ao posicionamento contextual da interface aumentada, por meio de técnicas de visão computacional que reconhecem e localizam componentes de uma máquina industrial didática. A partir desse reconhecimento, os elementos virtuais são ancorados de forma dinâmica conforme a posição real dos componentes, buscando uma representação estável, coerente e contextualizada do ambiente físico.

1.3 Objetivo Geral

Desenvolver um sistema de Realidade Aumentada interativa capaz de realizar o monitoramento e o ajuste em tempo real¹ de parâmetros de uma máquina industrial didática, permitindo o posicionamento contextual dos elementos virtuais sobre os componentes físicos reconhecidos por meio de técnicas de visão computacional.

1.4 Objetivos Específicos

Como forma de direcionamento para o desenvolvimento deste trabalho, foram definidos os seguintes objetivos específicos:

- a) Selecionar e treinar um modelo de visão computacional utilizando *datasets* específicos para identificação de máquinas industriais.
- b) Desenvolver a aplicação de RA integrando visão computacional e sobreposição gráfica em tempo real com utilização de *frameworks* ou bi-

¹ Neste trabalho, a expressão “tempo real” é utilizada no sentido de tempo real flexível, isto é, com atualização suficientemente rápida para manter a fluidez da interação e a coerência perceptiva da interface, sem caracterizar requisitos determinísticos rígidos de controle industrial crítico, conforme a distinção apresentada por Buttazzo (2011).

bibliotecas já existentes.

- c) Implementar um mecanismo de ancoragem espacial dinâmica para posicionar os elementos virtuais de acordo com a localização real dos componentes reconhecidos.
- d) Integrar o sistema com dados de sensores/CLP, de forma a captar e transmitir parâmetros reais de funcionamento das máquinas em tempo real.
- e) Projetar e implementar a interface interativa em RA, permitindo que o usuário visualize e ajuste parâmetros de operação com mecanismos de confirmação e feedback.
- f) Validar o sistema em um equipamento industrial didático, avaliando a estabilidade dos elementos virtuais, a responsividade da interface e aspectos exploratórios de uso em condições controladas de operação.

1.5 Estrutura do Trabalho

A organização deste trabalho foi concebida para conduzir o leitor de forma progressiva, iniciando pelos conceitos fundamentais, avançando por estudos relacionados e, por fim, chegando ao desenvolvimento da solução proposta.

O Capítulo 2 apresenta a fundamentação teórica relacionada à Indústria 4.0, interfaces homem-máquina, Realidade Aumentada, visão computacional e comunicação industrial. O Capítulo 3 descreve a metodologia adotada, a arquitetura da solução e os critérios de validação. O Capítulo 4 detalha o desenvolvimento da aplicação, incluindo a comunicação entre frontend e backend, a integração Modbus, os recursos de visão e a interface aumentada. Por fim, o Capítulo 5 reúne as considerações finais, as limitações identificadas e as possibilidades de continuidade do trabalho.

2 REVISÃO BIBLIOGRÁFICA

2.1 Fundamentação Teórica

Este capítulo apresenta os fundamentos técnicos que sustentam o desenvolvimento da solução proposta. A revisão foi organizada de forma progressiva, partindo do contexto da Indústria 4.0 e dos sistemas ciberfísicos, avançando para Realidade Aumentada, visão computacional, comunicação industrial, APIs e interfaces homem-máquina contextuais. Essa sequência acompanha a própria arquitetura da aplicação, na qual dados industriais, percepção visual e interação com o operador precisam atuar de maneira integrada.

2.1.1 Indústria 4.0 e Sistemas Ciberfísicos

A Indústria 4.0 representa um paradigma de transformação tecnológica caracterizado pela integração entre processos físicos e sistemas digitais. Diferentemente dos modelos industriais anteriores, nos quais automação e informatização eram frequentemente tratadas de forma isolada, o novo paradigma propõe conectividade contínua, coleta de dados em tempo real e integração entre dispositivos, redes e sistemas computacionais (SCHWAB, 2016; KAGERMANN; WAHLSTER; HELBIG, 2013).

Nesse contexto, os Sistemas Ciberfísicos (Cyber-Physical Systems – CPS) emergem como elemento central. Um sistema ciberfísico pode ser compreendido como a integração estruturada entre componentes físicos, responsáveis pela execução do processo industrial, e componentes computacionais, responsáveis pelo monitoramento, análise e tomada de decisão baseada em dados.

A característica distintiva dos CPS reside na bidirecionalidade entre as camadas física e digital. Sensores capturam informações do ambiente e as disponibilizam para processamento computacional, enquanto decisões e comandos podem ser enviados de volta ao sistema físico. Essa interação contínua permite maior adaptabilidade, rastreabilidade e eficiência operacional.

A conectividade proporcionada por redes industriais e protocolos padronizados amplia o alcance dos CPS, viabilizando integração entre múltiplos dispositivos e sistemas distribuídos. A relevância desse tipo de integração também é observada na transformação de produtos e equipamentos conectados, nos quais dados operacionais passam a influenciar serviços, manutenção e formas de interação com o usuário (PORTER; HEPPELMANN, 2014). Entretanto, a simples existência de comunicação digital não caracteriza, por si só, um sistema ciberfísico. O que define esse paradigma é a capacidade de interpretar dados em tempo oportuno e utilizá-los para influenciar o comportamento do processo físico.

A consolidação desse modelo industrial estabelece as bases para aplica-

ções que combinam percepção, processamento computacional e interação humana integrada ao ambiente físico. Nesse cenário, tecnologias de visualização contextual e integração em tempo real passam a desempenhar papel relevante na mediação entre operador e sistema produtivo.

2.1.1.1 Transformação Digital no Ambiente Industrial

A transformação digital na indústria implica a incorporação de sensores, redes industriais e sistemas computacionais que permitem visibilidade ampliada do processo produtivo. Variáveis antes restritas a painéis locais ou sistemas supervisórios passam a ser disponibilizadas em diferentes camadas da organização, possibilitando análise histórica, diagnósticos preditivos e suporte à decisão em tempo real.

Essa nova configuração exige arquiteturas que conciliem confiabilidade operacional com flexibilidade computacional. Enquanto o chão de fábrica demanda determinismo e estabilidade, as camadas digitais agregam processamento intensivo, armazenamento de dados e interfaces mais sofisticadas de interação com o operador.

A integração entre essas camadas requer mecanismos padronizados de comunicação, como protocolos industriais consolidados, além de infraestrutura de rede capaz de suportar transmissão contínua de dados com latência controlada.

2.1.1.2 Sistemas Ciberfísicos (CPS)

Os sistemas ciberfísicos (CPS) são estruturas compostas por elementos físicos integrados a componentes computacionais que monitoram e controlam o comportamento do sistema por meio de ciclos fechados de realimentação.

Em um CPS típico, sensores coletam dados do ambiente físico, algoritmos processam essas informações e atuadores executam ações com base nas decisões computacionais. Esse modelo estabelece uma relação bidirecional entre o domínio físico e o digital, permitindo não apenas observação, mas também intervenção controlada sobre o processo.

No ambiente industrial, os CPS materializam-se na integração entre controladores lógicos programáveis (CLPs), redes de comunicação, sistemas de supervisão e, mais recentemente, tecnologias de visão computacional e interfaces aumentadas. Essa arquitetura possibilita que decisões sejam tomadas com base em informações contextualizadas e atualizadas em tempo real.

A implementação de sistemas ciberfísicos exige atenção especial à sincronização entre camadas, à confiabilidade da comunicação e à segurança operacional, sobretudo quando há possibilidade de atuação remota sobre equipamentos físicos.

2.1.1.3 Arquiteturas Distribuídas e Integração de Camadas

A consolidação da Indústria 4.0 trouxe a adoção de arquiteturas distribuídas, nas quais diferentes camadas desempenham funções específicas dentro do sistema industrial. De forma simplificada, pode-se identificar:

- Camada física: composta por máquinas, sensores e atuadores;
- Camada de controle: responsável pela lógica de operação, geralmente implementada em CLPs;
- Camada de comunicação: que viabiliza a troca de dados entre dispositivos;
- Camada computacional: dedicada ao processamento de informações e análise de dados;
- Camada de interface: responsável pela interação com o operador.

A integração eficiente dessas camadas depende de protocolos de comunicação padronizados e de estratégias que minimizem latência e inconsistências de dados. Além disso, a distribuição funcional exige mecanismos de validação e controle que garantam que comandos emitidos nas camadas superiores não comprometam a segurança ou a estabilidade do processo físico.

Essa organização em camadas é fundamental para viabilizar soluções que combinam percepção visual do ambiente, processamento computacional e atuação controlada sobre o sistema industrial.

2.1.1.4 Implicações para Interfaces Homem-Máquina

A digitalização industrial altera também a forma como operadores interagem com o processo produtivo. As interfaces homem-máquina tradicionais concentram informações em painéis fixos ou telas supervisórias, exigindo que o operador alterne constantemente sua atenção entre o equipamento físico e o sistema digital.

Com o avanço dos sistemas ciberfísicos, surge a possibilidade de interfaces contextuais, nas quais informações digitais são apresentadas diretamente no ambiente de operação, reduzindo a dissociação entre ação física e interpretação de dados. Essa abordagem pode contribuir para menor carga cognitiva, maior rapidez na identificação de falhas e melhor compreensão do estado atual do sistema.

Entretanto, quando a interface passa a permitir também a atuação sobre o processo, torna-se essencial estabelecer mecanismos de validação e controle que assegurem que a interação humana ocorra dentro de limites seguros e previamente definidos.

Assim, a evolução das interfaces industriais está diretamente relacionada à consolidação de arquiteturas ciberfísicas robustas, capazes de integrar percepção, processamento e atuação de maneira coordenada e segura.

A crescente digitalização dos processos industriais amplia o volume de dados disponíveis para monitoramento e controle. Entretanto, a simples existência dessas informações não garante sua utilização eficiente. Torna-se necessário desenvolver mecanismos que permitam associar dados digitais ao ambiente físico de forma contextualizada, favorecendo interpretação imediata e tomada de decisão operacional.

2.1.2 Realidade Aumentada e Problema de Registro Espacial

A Realidade Aumentada (RA) consiste na sobreposição de elementos virtuais ao ambiente físico em tempo real. De acordo com Azuma (1997), um sistema de RA deve combinar objetos reais e virtuais, manter interação contínua e garantir alinhamento tridimensional preciso. Essa compreensão é compatível com a abordagem de Schmalstieg e Höllerer (2016), na qual a RA é tratada como uma interface interativa entre computação espacial e ambiente físico. Esse tipo de interface amplia a percepção do usuário ao apresentar informações contextuais diretamente no campo de visão.

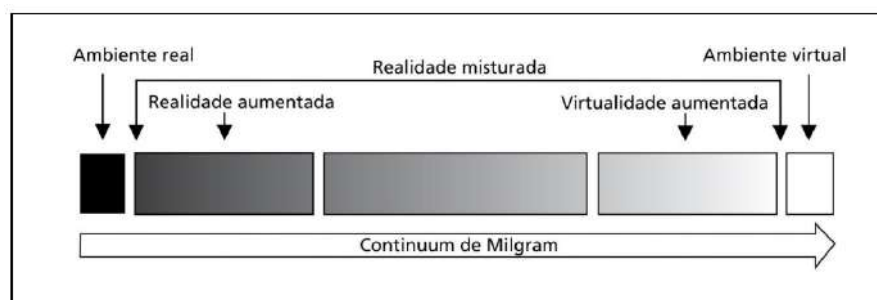
Para que uma aplicação seja caracterizada como realidade aumentada, três requisitos fundamentais devem ser atendidos: a combinação entre elementos reais e virtuais, a interação em tempo real e o alinhamento espacial coerente entre os objetos digitais e o ambiente físico. Este último requisito constitui o principal desafio técnico da RA e é conhecido como problema de registro espacial.

Na definição clássica da RA, esse alinhamento é frequentemente tratado como registro tridimensional preciso, pois envolve estimar a posição e a orientação da câmera em relação ao ambiente (AZUMA, 1997; SCHMALSTIEG; HÖLLERER, 2016). Entretanto, quando a aplicação utiliza detectores de objetos baseados em caixas delimitadoras, a referência espacial disponível passa a ser a região bidimensional ocupada pelo componente no quadro da câmera (REDMON *et al.*, 2016). Nesses casos, a solução não reconstrói a pose tridimensional completa do objeto, mas pode manter uma correspondência visual contextual suficiente para apresentar informações sobre componentes reconhecidos no campo de visão.

Essa discussão sobre registro espacial também ajuda a situar a RA em relação a outras formas de visualização computacional. A distinção entre Realidade Aumentada e Realidade Virtual (RV) é apresentada por Milgram e Kishino (1994), que posicionam ambas dentro do continuum de “realidade misturada”, conforme apresentado na Figura 1. Nesse continuum, a RA mantém o ambiente físico como referência principal e acrescenta informações digitais sobre ele, enquanto a RV cria um ambi-

ente totalmente sintético, desconectado do mundo real. Assim, mesmo quando a ancoragem adotada é bidimensional e contextual, como na solução desenvolvida neste trabalho, a aplicação permanece orientada ao princípio central da RA: complementar a percepção do ambiente físico observado. Essa característica torna a RA particularmente relevante em situações nas quais a referência física permanece indispensável, como operação e manutenção industrial.

Figura 1 – Contínuo Real-Virtual



Fonte: Tori e Hounsell (2018)

Dessa forma, três características são essenciais para a implementação de RA funcional:

1. Registro espacial consistente entre o ambiente físico e o virtual;
2. Atualização em tempo real, permitindo interação dinâmica;
3. Persistência da ancoragem virtual durante deslocamentos da câmera.

Sem esses elementos, a aplicação deixa de ser realidade aumentada e passa a ser apenas uma sobreposição gráfica não contextualizada.

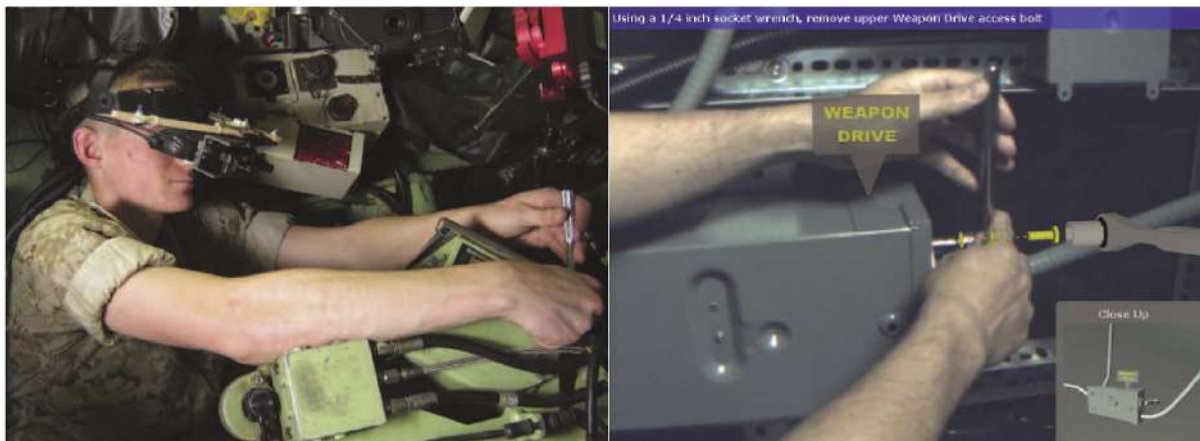
A RA ganhou espaço em diversos segmentos. Na educação, Billinghurst e Duenser (2012) demonstram sua capacidade de enriquecer conteúdos por meio de visualização interativa. Na construção civil, Wang *et al.* (2013) descrevem o uso da RA para inspeções orientadas e análise de estruturas. No setor da saúde, revisões como a de Zhu *et al.* (2014) evidenciam a utilização da RA em treinamento clínico e visualização anatômica.

No contexto industrial, Henderson e Feiner (2011) demonstram que a utilização da Realidade Aumentada em tarefas de manutenção e montagem reduz significativamente o tempo de execução e a taxa de erros ao apresentar instruções diretamente ancoradas sobre o equipamento real. Em seus experimentos, técnicos realizaram procedimentos complexos mais rapidamente e com maior precisão quando comparados às instruções tradicionais em papel ou em telas externas, evidenciando

que a RA atua como um mecanismo de suporte cognitivo ao diminuir a sobrecarga mental e a necessidade de alternância visual entre manual e máquina.

A Figura 2 exemplifica esse tipo de apoio visual aplicado à manutenção assistida por RA.

Figura 2 – Manutenção assistida por Realidade Aumentada



Fonte: Henderson e Feiner (2011)

Além disso, o estudo mostra que a sobreposição de passos, diagramas e alertas em escala real não apenas facilita a compreensão espacial das ações necessárias, mas também melhora a padronização das etapas de manutenção, contribuindo para reduzir variabilidade operacional e aumentar a confiabilidade do processo. Esses resultados consolidam a RA como uma interface industrial capaz de otimizar intervenções técnicas e apoiar decisões diretamente no ambiente produtivo.

2.1.2.1 O Problema de Registro Espacial

O problema de registro espacial consiste em determinar, com precisão, a posição e a orientação da câmera em relação ao ambiente físico, de modo que objetos virtuais possam ser projetados corretamente na imagem capturada (AZUMA, 1997; SCHMALSTIEG; HÖLLERER, 2016).

Em termos práticos, isso implica estimar a transformação geométrica entre o sistema de coordenadas da câmera e o sistema de coordenadas do objeto físico observado (SCHMALSTIEG; HÖLLERER, 2016). Qualquer erro nessa estimativa resulta em desalinhamento visual, instabilidade ou “deslizamento” dos elementos virtuais sobre o ambiente real.

O registro espacial pode ser abordado por diferentes estratégias, como o uso de marcadores artificiais, sensores inerciais ou detecção de características naturais da cena (AZUMA, 1997; SCHMALSTIEG; HÖLLERER, 2016). Em aplicações industriais, onde a confiabilidade visual é crítica, a escolha da estratégia de registro

deve considerar robustez a variações de iluminação, oclusões parciais e superfícies com baixo contraste.

A precisão do registro é particularmente relevante quando a RA é utilizada não apenas para visualização, mas também para interação e atuação sobre o processo físico, pois a posição incorreta de elementos virtuais pode induzir decisões equivocadas do operador.

2.1.2.2 Abordagens Marker-Based e Markerless

As aplicações de RA podem ser classificadas, quanto ao método de rastreamento, em abordagens baseadas em marcadores (*marker-based*) e abordagens sem marcadores artificiais (*markerless*).

Na abordagem *marker-based*, padrões visuais específicos são inseridos no ambiente físico para facilitar o rastreamento e a estimação de pose, como por exemplo os QR-Codes utilizados no trabalho desenvolvido por Rosa (2019), apresentado na Figura 3. Esses marcadores apresentam geometrias conhecidas e alto contraste, simplificando o cálculo da transformação espacial. Embora essa técnica apresente elevada estabilidade, ela exige modificação física do ambiente, o que pode ser indesejável em cenários industriais consolidados.

Figura 3 – QR-Code para marcação de objeto



Fonte: Rosa (2019)

Já a abordagem *markerless* utiliza características naturais do ambiente como bordas, texturas e geometrias estruturais para realizar o rastreamento e a ancoragem virtual (AZUMA, 1997; SCHMALSTIEG; HÖLLERER, 2016). Essa estratégia elimina a necessidade de inserir elementos artificiais na cena, preservando a integridade do ambiente industrial. Contudo, impõe maior complexidade computacional e depende da robustez dos algoritmos de detecção e estimação de pose.

Em aplicações industriais modernas, a abordagem *markerless* tem se destacado por sua flexibilidade e melhor adaptação a ambientes já existentes, embora exija processamento mais sofisticado e mecanismos adicionais de validação espacial.

2.1.2.3 Desafios da RA em Ambientes Industriais

A implementação de RA em ambientes industriais apresenta desafios adicionais quando comparada a aplicações em contextos controlados. Entre os principais fatores que impactam o desempenho do sistema estão:

- Variações intensas de iluminação;
- Presença de superfícies metálicas reflexivas;
- Oclusões parciais por operadores ou componentes móveis;
- Vibrações estruturais;
- Ambientes visualmente homogêneos com baixo contraste.

Essas características podem comprometer a detecção de características visuais e afetar a estabilidade do registro espacial (AZUMA, 1997; SCHMALSTIEG; HÖLLERER, 2016). Além disso, em cenários onde a RA é utilizada como interface operacional, eventuais falhas de ancoragem podem impactar diretamente a confiabilidade da interação.

Portanto, a aplicação de RA em contexto industrial exige não apenas algoritmos robustos de visão computacional, mas também arquitetura de sistema capaz de lidar com latência, perda momentânea de rastreamento e mecanismos de recuperação automática. A superação desses desafios é condição essencial para que a realidade aumentada seja empregada como ferramenta funcional em sistemas ciberfísicos industriais.

2.1.3 Registro Visual Contextual em Ambientes Industriais

A efetividade da Realidade Aumentada em ambientes industriais depende da coerência entre os elementos virtuais apresentados e os componentes físicos observados. Em interfaces de monitoramento e operação, essa coerência pode ser obtida pela associação contextual entre regiões detectadas na imagem e informações digitais correspondentes ao equipamento reconhecido (AZUMA, 1997; HENDERSON; FEINER, 2011).

Uma forma de representação utiliza caixas delimitadoras bidimensionais (*bounding boxes*) produzidas por detectores de objetos (REDMON *et al.*, 2016). Essas caixas indicam a região ocupada por cada componente no quadro capturado e fornecem referências para posicionar rótulos, indicadores e controles sobre a imagem apresentada ao operador.

Em uma câmera móvel, a estabilidade desse registro visual depende não apenas da qualidade de uma detecção isolada, mas também da continuidade temporal

entre quadros (AZUMA, 1997; SCHMALSTIEG; HÖLLERER, 2016). Movimentos rápidos, reflexos, oclusões e variações de enquadramento podem provocar perdas momentâneas ou deslocamentos bruscos. Por esse motivo, estratégias de rastreamento, suavização e permanência visual limitada contribuem para preservar legibilidade sem prolongar informações desatualizadas.

2.1.4 Visão Computacional para Detecção de Objetos

A visão computacional constitui a base perceptiva do sistema proposto, permitindo que informações visuais capturadas por uma câmera sejam processadas para identificação de objetos e extração de características relevantes do ambiente físico.

Em aplicações industriais baseadas em realidade aumentada, a visão computacional não se limita ao reconhecimento genérico de objetos. Ela desempenha papel fundamental na identificação de componentes estruturais específicos utilizados como referências para registro espacial e interação contextualizada.

A detecção robusta e consistente desses elementos é condição essencial para associar informações digitais aos componentes físicos e preservar a estabilidade visual da interface.

2.1.4.1 Redes Neurais Convolucionais (CNNs)

Desde os trabalhos de LeCun *et al.* (1998), redes neurais convolucionais se tornaram fundamentais para esse campo, permitindo automatizar a extração de características e melhorar significativamente o desempenho em tarefas de reconhecimento.

As redes neurais convolucionais (*Convolutional Neural Networks* – CNNs) são arquiteturas de aprendizado profundo amplamente utilizadas em tarefas de classificação e detecção de imagens. Diferentemente de métodos tradicionais baseados em extração manual de características, as CNNs aprendem automaticamente padrões visuais relevantes a partir de dados rotulados.

Uma CNN é composta por camadas convolucionais responsáveis por extrair características espaciais, camadas de ativação não linear e, em alguns casos, camadas de *pooling*¹ que reduzem dimensionalidade. Em tarefas de detecção, essas características são utilizadas para localizar e classificar objetos dentro da imagem.

A capacidade das CNNs de identificar padrões complexos sob diferentes condições de iluminação, escala e perspectiva torna essa abordagem particularmente adequada para ambientes industriais dinâmicos.

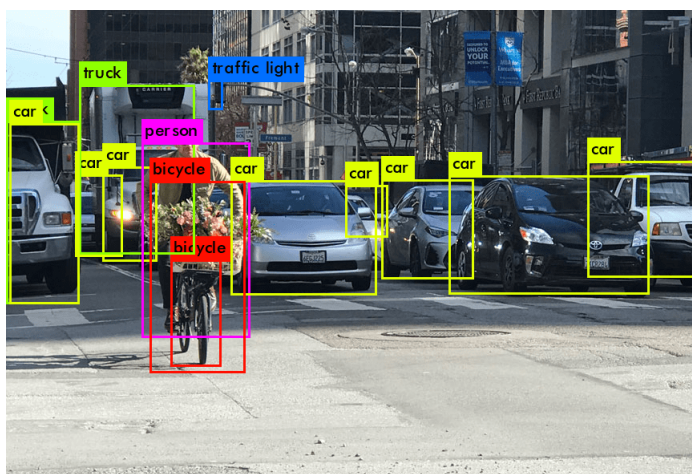
¹ Operação empregada em redes neurais convolucionais para reduzir a dimensão espacial de mapas de características, preservando informações relevantes para a tarefa de reconhecimento.

2.1.4.2 Detectores One-Stage e o Modelo YOLO

Os detectores de objetos baseados em aprendizado profundo podem ser classificados, de forma geral, em duas categorias: *two-stage* e *one-stage*. Os modelos *two-stage* realizam inicialmente a proposta de regiões de interesse e, em seguida, classificam essas regiões. Já os modelos *one-stage* realizam detecção e classificação simultaneamente, oferecendo maior velocidade de processamento.

Entre os detectores *one-stage*, o modelo YOLO (You Only Look Once) destaca-se por sua eficiência computacional e capacidade de operar em tempo próximo ao real (REDMON *et al.*, 2016). O YOLO divide a imagem em regiões e, para cada região, estima diretamente caixas delimitadoras (*bounding boxes*) e probabilidades de classe, como no exemplo apresentado na Figura 4.

Figura 4 – Detecção e classificação por modelo YOLO



Fonte: Platform (2024)

A principal vantagem dessa abordagem em aplicações interativas é a redução de latência, fator crítico quando o sistema precisa fornecer resposta visual imediata ao operador. Em um cenário industrial com integração a redes de controle, atrasos excessivos podem comprometer a experiência de uso e a confiabilidade da interação.

Além da identificação da máquina como um todo, o modelo pode ser treinado para detectar componentes específicos, como painéis, botões, eixos ou elementos estruturais, permitindo granularidade maior na interpretação da cena.

Em aplicações que envolvem registro visual contextual, as regiões detectadas servem como referência bidimensional para posicionamento de informações digitais relacionadas ao componente reconhecido, preservando a associação visual entre o dado apresentado e o objeto físico sem exigir, necessariamente, a estimação completa da pose tridimensional.

2.1.4.3 Métricas de Avaliação em Detecção de Objetos

A avaliação do desempenho de um detector de objetos exige métricas quantitativas que permitam medir sua precisão e robustez.

Uma das métricas fundamentais é o *Intersection over Union* (IoU), que quantifica o grau de sobreposição entre a caixa delimitadora prevista pelo modelo e a caixa real anotada no conjunto de validação. Quanto maior o IoU, maior a precisão da localização estimada.

Outras métricas relevantes incluem:

- Precisão (*Precision*): proporção de detecções corretas entre todas as detecções realizadas;
- Revocação (*Recall*): proporção de objetos corretamente detectados em relação ao total de objetos existentes;
- *F1-score*: média harmônica entre precisão e revocação;
- mAP (*mean Average Precision*): média da precisão ao longo de diferentes limiares de confiança.

Essas métricas são essenciais para validar a confiabilidade do sistema, especialmente quando a detecção de componentes é utilizada como base para posicionar informações operacionais na interface.

2.1.4.4 Detecção de Componentes e Continuidade Temporal

Em aplicações de realidade aumentada industrial, a detecção desempenha papel adicional além do reconhecimento. As caixas delimitadoras fornecem referências bidimensionais para associar informações operacionais aos componentes físicos observados (REDMON *et al.*, 2016; HENDERSON; FEINER, 2011).

Como o detector pode apresentar pequenas variações entre quadros consecutivos, mecanismos de rastreamento e suavização contribuem para reduzir oscilações perceptíveis (AZUMA, 1997; SCHMALSTIEG; HÖLLERER, 2016). A análise da continuidade temporal pode empregar métricas como IoU entre caixas sucessivas, deslocamento normalizado entre centros e taxa de permanência das detecções durante a movimentação da câmera.

Essa integração entre detecção baseada em aprendizado profundo e estabilização temporal viabiliza interfaces contextualizadas sem exigir marcadores artificiais fixados nos equipamentos (REDMON *et al.*, 2016; AZUMA, 1997).

2.1.5 Redes Industriais e Protocolo Modbus

Uma vez estabelecida a camada perceptiva responsável pela identificação e localização de componentes físicos na cena, torna-se necessário integrar essa interpretação ao sistema de controle industrial. Essa integração depende de protocolos de comunicação estruturados, capazes de permitir a leitura e, quando aplicável, a escrita de variáveis operacionais de forma confiável e segura.

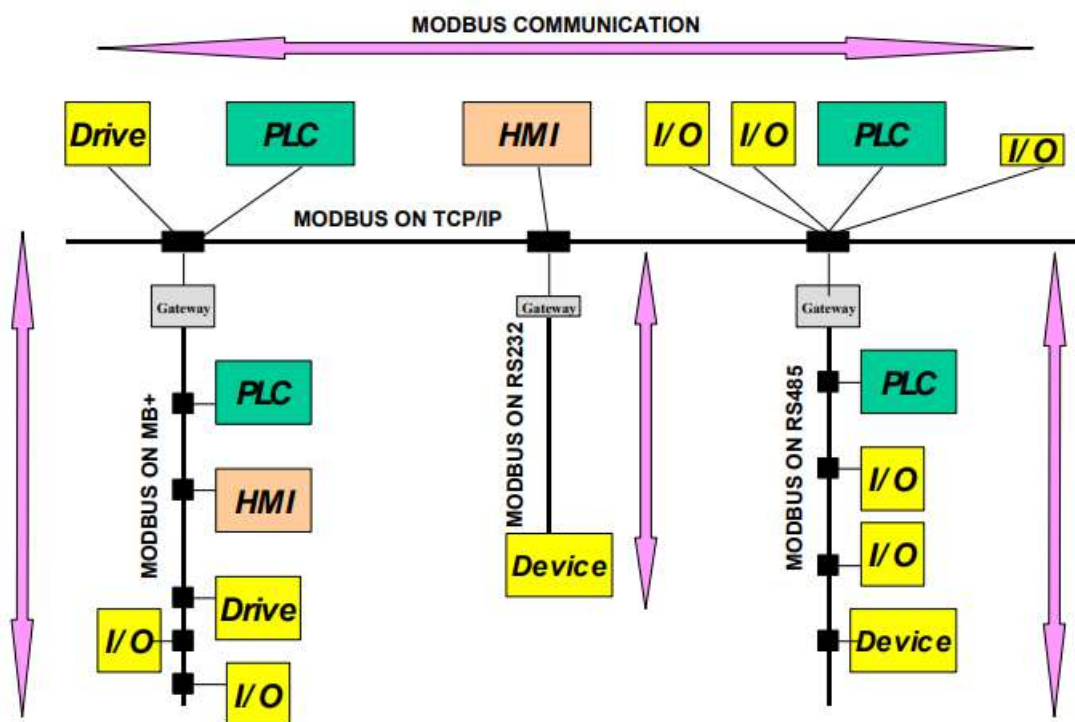
A integração entre camadas digitais e dispositivos físicos, especialmente quando envolve leitura e escrita de variáveis operacionais, requer compreensão detalhada do funcionamento do protocolo e de suas implicações em termos de segurança e confiabilidade.

2.1.5.1 Protocolo Modbus e Estrutura de Comunicação

O Modbus é um protocolo de comunicação industrial especificado originalmente pela Modicon (MODICON, 1999), baseado em modelo de requisição e resposta e tradicionalmente estruturado sob lógica mestre-escravo. No modelo clássico, um dispositivo mestre inicia a comunicação enviando requisições a dispositivos escravos, que respondem com os dados solicitados.

A especificação do protocolo apresenta exemplos de arquitetura para comunicação via Modbus, conforme ilustrado na Figura 5.

Figura 5 – Exemplo de arquitetura de uma rede Modbus



Fonte: Modicon (1999)

Em implementações sobre redes Ethernet, como no Modbus TCP, mantém-se a estrutura lógica de requisição-resposta, mas utilizando infraestrutura de rede padrão. O guia de implementação Modbus TCP/IP define a encapsulação das mensagens Modbus em uma unidade de aplicação própria, transportada sobre TCP, usualmente associada à porta 502 (ORGANIZATION, 2006). Essa adaptação permite maior flexibilidade e integração com sistemas computacionais externos.

Os dados no Modbus são organizados em diferentes categorias de registradores, incluindo:

- *Coils*: variáveis booleanas com leitura e escrita;
- *Discrete Inputs*: variáveis booleanas apenas leitura;
- *Holding Registers*: registradores de 16 bits com leitura e escrita;
- *Input Registers*: registradores de 16 bits apenas leitura.

Essa estrutura permite mapear variáveis físicas do processo como velocidades, posições, estados lógicos ou parâmetros operacionais em endereços específicos acessíveis via comunicação remota.

Embora o protocolo seja eficiente na troca de dados, ele não impõe restrições semânticas sobre os valores transmitidos nem incorpora, em sua concepção clássica, mecanismos nativos de autenticação, autorização ou criptografia. Diretrizes de segurança para sistemas de controle industrial recomendam que comunicações desse tipo sejam tratadas dentro de arquiteturas segmentadas, com controle de acesso, limitação de exposição da rede de controle e validações adicionais nas camadas intermediárias (STOUFFER *et al.*, 2015). Assim, cabe à aplicação que realiza a comunicação implementar mecanismos adicionais de validação quando a operação envolve escrita remota.

2.1.5.2 *Arquiteturas Intermediadas e Separação de Camadas*

Em sistemas industriais que permitem interação remota com controladores, é recomendável adotar arquitetura intermediada entre a interface de usuário e o CLP. Essa separação de camadas introduz componente responsável por gerenciar comunicação com o controlador e aplicar regras de validação antes da execução de comandos. A recomendação é coerente com a defesa em profundidade aplicada a sistemas de controle industrial, na qual a rede de automação não deve ser exposta diretamente a clientes de operação sem mecanismos de filtragem, autenticação e controle de permissões (STOUFFER *et al.*, 2015).

A utilização de camada intermediária apresenta vantagens relevantes:

- Redução da exposição direta do CLP à interface externa;
- Centralização de regras de negócio e validações;
- Facilidade de manutenção e atualização;
- Maior controle sobre operações de escrita.

Esse modelo também favorece desacoplamento entre lógica de controle industrial e lógica de interface, permitindo que modificações na camada de apresentação não impactem diretamente a configuração do controlador físico.

A adoção de arquiteturas distribuídas com separação clara entre camadas física, de controle, de comunicação e de interface está alinhada aos princípios de sistemas ciberfísicos e às recomendações de segmentação e controle de acesso discutidas para ambientes de controle industrial (STOUFFER *et al.*, 2015).

2.1.5.3 Abstração de Variáveis e Modelo de Tags

Em vez de expor diretamente endereços numéricos do protocolo de comunicação à camada de interface, é prática comum empregar modelo de abstração baseado em *tags* ou identificadores lógicos. Essa estratégia reduz a dependência da interface em relação ao mapa físico do controlador e favorece a implementação de permissões e validações em uma camada intermediária, em conformidade com a separação de responsabilidades discutida para sistemas distribuídos e sistemas de controle industrial (STOUFFER *et al.*, 2015).

Nesse modelo, cada variável relevante do processo é associada a um identificador simbólico. O sistema intermediário realiza o mapeamento entre esse identificador e o endereço correspondente no protocolo industrial.

A abstração de variáveis oferece benefícios como:

- Maior legibilidade do sistema;
- Redução de erros associados a endereçamento manual;
- Possibilidade de associar metadados às variáveis, como limites operacionais;
- Facilidade de reconfiguração sem necessidade de alterar a interface.

Além disso, a utilização de modelo baseado em *tags* favorece implementação de políticas diferenciadas de acesso, permitindo definir quais variáveis podem ser apenas monitoradas e quais podem ser alteradas sob determinadas condições.

2.1.5.4 Escrita Remota e Segurança Operacional

Quando sistemas externos possuem capacidade de escrita sobre variáveis de controle industrial, torna-se essencial adotar mecanismos de segurança que minimizem riscos operacionais.

Entre as estratégias recomendadas, em consonância com práticas de proteção de sistemas de controle industrial e com a necessidade de validação semântica não fornecida diretamente pelo protocolo Modbus, estão (MODICON, 1999; STOUFFER *et al.*, 2015):

- Validação de faixa: verificação se o valor solicitado encontra-se dentro de limites previamente definidos;
- Confirmação explícita de operação antes da execução;
- Restrição de escrita a determinados estados operacionais;
- Registro de alterações para fins de rastreabilidade;
- Implementação de intertravamentos lógicos no controlador.

A validação de faixa constitui mecanismo fundamental, pois impede que valores potencialmente perigosos sejam transmitidos ao processo físico. Essa verificação pode ser realizada em camada intermediária antes da emissão do comando de escrita.

Adicionalmente, recomenda-se que o próprio controlador industrial mantenha lógica de proteção redundante, garantindo que alterações indevidas não sejam aplicadas mesmo que cheguem via comunicação externa.

A combinação entre arquitetura intermediada, abstração de variáveis e validação de parâmetros contribui para construção de sistemas robustos, nos quais a interação digital com o processo físico ocorre de maneira controlada e segura.

2.1.6 APIs e Serviços Web em Arquiteturas Distribuídas

Em arquiteturas distribuídas, a comunicação entre camadas de software é frequentemente organizada por meio de APIs (*Application Programming Interfaces*). Uma API define um conjunto de operações, formatos de dados e regras de acesso que permitem que diferentes componentes computacionais troquem informações de forma padronizada. Segundo Fielding (2000), a Web consolidou um estilo arquitetural baseado na separação entre cliente e servidor, no uso de recursos identificáveis e na troca de representações por meio de protocolos de aplicação.

No contexto de aplicações web, o protocolo HTTP atua como base para requisições e respostas entre cliente e servidor. Cada funcionalidade disponibilizada pelo servidor pode ser exposta por meio de um *endpoint*, isto é, um endereço lógico associado a determinada operação, como leitura de estado, envio de comando ou processamento de um quadro de imagem. Essa organização favorece separação de responsabilidades, pois o cliente não precisa conhecer a implementação interna do servidor, apenas o contrato de comunicação disponibilizado pela API.

O formato JSON, definido por Bray (2017), tornou-se amplamente utilizado nesse tipo de integração por representar dados estruturados de forma textual, leve e independente de linguagem de programação. Em sistemas interativos, essa característica facilita a comunicação entre backend e frontend, permitindo que informações de estado, medições, comandos e descrições visuais sejam transmitidas de forma legível e processável.

Em uma interface de realidade aumentada industrial, a API pode funcionar como contrato entre a camada de processamento e a camada de apresentação. O backend concentra tarefas de análise, integração com controladores e organização do estado operacional, enquanto o frontend interpreta as respostas estruturadas e realiza a renderização local dos elementos visuais. Essa abordagem evita o envio de imagens já anotadas pelo servidor e preserva no cliente a composição final dos *overlays*, reduzindo tráfego de dados e ampliando a flexibilidade da interface.

A comunicação estruturada com controladores industriais, por si só, não garante eficiência operacional. A forma como as informações são apresentadas e como o operador interage com elas constitui elemento determinante para a usabilidade e a segurança do sistema. Assim, a análise das Interfaces Homem-Máquina complementa a discussão sobre integração distribuída ao abordar a camada em que os dados industriais se tornam informação operacional para o usuário.

2.1.7 Interfaces Homem-Máquina Contextuais

As Interfaces Homem-Máquina (IHMs) constituem o principal meio de interação entre operadores e sistemas de controle industrial. Historicamente, essa interação ocorreu por meio de painéis físicos com botões, indicadores luminosos e instrumentos analógicos. Com o avanço da automação, tais interfaces evoluíram para painéis digitais e sistemas supervisórios baseados em software.

No contexto da digitalização industrial, a interface deixa de ser apenas elemento de visualização e passa a desempenhar papel central na mediação entre o operador e sistemas complexos distribuídos. A forma como as informações são apresentadas influencia diretamente a capacidade de diagnóstico, a rapidez de tomada de decisão e a segurança operacional.

2.1.7.1 IHMs Tradicionais e Supervisórios

As IHMs tradicionais baseiam-se, em geral, em telas fixas localizadas em painéis próximos ao equipamento ou em estações supervisórias. Nessas interfaces, os dados do processo são apresentados em formatos gráficos, tabelas ou diagramas sinóticos, permitindo ao operador monitorar variáveis e, quando autorizado, alterar parâmetros, conforme exemplificado na Figura 6.

Figura 6 – Exemplo de IHM tradicional



Fonte: Marcio Mello (2023)

Embora essas soluções sejam consolidadas e confiáveis, apresentam limitações relacionadas à dissociação espacial entre o equipamento físico e as informações exibidas. O operador frequentemente precisa alternar sua atenção entre o processo real e a tela digital, o que pode aumentar a carga cognitiva e o tempo de resposta em situações críticas.

Além disso, a navegação em múltiplas telas ou níveis hierárquicos pode dificultar a identificação rápida de variáveis específicas associadas a determinado componente físico.

2.1.7.2 Interfaces Contextuais e Informação In Situ

Com a evolução das tecnologias de visualização e processamento gráfico, surgem as chamadas interfaces contextuais, nas quais a informação digital é apresentada diretamente no ambiente de operação, próxima ao elemento físico ao qual se refere.

O conceito de informação in situ baseia-se na apresentação de dados e elementos de controle diretamente sobre o contexto físico do equipamento, por meio de registro espacial preciso entre mundo real e conteúdo virtual. Segundo Azuma (1997),

a integração espacialmente registrada é uma das características centrais da realidade aumentada, permitindo que informações digitais sejam percebidas como pertencentes ao ambiente físico.

Essa abordagem reduz a necessidade de transformação mental entre diferentes representações, minimizando a carga cognitiva do operador, conforme discutido por Sweller (1988) na teoria da carga cognitiva. Ao posicionar indicadores no próprio ponto de ocorrência do fenômeno, favorece-se a associação direta entre causa e efeito, aspecto particularmente relevante em sistemas complexos compostos por múltiplos componentes interdependentes. Estudos experimentais demonstram que instruções espacialmente integradas reduzem erros operacionais e tempo de execução quando comparadas a interfaces convencionais (HENDERSON; FEINER, 2011).

2.1.7.3 Carga Cognitiva e Usabilidade em Ambientes Industriais

A carga cognitiva refere-se ao esforço mental necessário para interpretar informações e executar tarefas. Segundo Sweller (1988), a capacidade de processamento da memória de trabalho é limitada, de modo que o excesso de informações simultâneas pode comprometer o desempenho e a aprendizagem. Em ambientes industriais, onde decisões podem impactar diretamente a segurança e a continuidade operacional, a clareza e organização das informações são fatores críticos.

Interfaces que exigem constante alternância de foco entre diferentes fontes de informação podem aumentar o risco de erro humano, uma vez que elevam a carga cognitiva extrínseca associada à tarefa. A apresentação contextualizada de dados tende a reduzir etapas intermediárias de interpretação, favorecendo compreensão mais imediata do estado do sistema.

Entretanto, a inserção excessiva de elementos visuais no campo de visão também pode gerar sobrecarga perceptiva. Portanto, o projeto de interfaces aumentadas deve considerar princípios de ergonomia, hierarquia visual e limitação de informações simultaneamente exibidas, de modo a equilibrar suporte informacional e capacidade cognitiva do operador.

2.1.7.4 Interação Bidirecional e Controle Assistido

A evolução das IHMs não se limita à forma de visualização. Sistemas modernos permitem interação bidirecional, na qual o operador pode não apenas monitorar variáveis, mas também atuar sobre o processo por meio da interface.

Quando a interação envolve alteração de parâmetros operacionais, torna-se fundamental implementar mecanismos que garantam que a ação humana seja assistida por validações automáticas e restrições de segurança. Esse modelo pode ser

caracterizado como controle assistido, no qual o operador permanece responsável pela decisão, mas o sistema impõe limites técnicos previamente definidos.

Em interfaces baseadas em realidade aumentada, a interação bidirecional deve considerar não apenas aspectos funcionais, mas também confiabilidade da ancoragem espacial e clareza na identificação do componente afetado pela ação.

A consolidação de interfaces contextuais bidirecionais representa etapa relevante na integração entre sistemas ciberfísicos e operadores humanos, ampliando as possibilidades de monitoramento e controle em ambientes industriais digitalizados.

A incorporação de interfaces contextuais e mecanismos de interação bidirecional introduz novos desafios relacionados ao tempo de resposta do sistema. A sincronização entre captura, processamento, comunicação e atualização visual passa a influenciar diretamente a estabilidade e a percepção de fluidez da interação. Dessa forma, torna-se pertinente analisar os conceitos de latência e comportamento temporal em sistemas interativos distribuídos.

2.1.8 Latência e Sistemas Interativos em Tempo Real

Sistemas industriais que integram percepção visual, processamento computacional e comunicação em rede operam sob restrições temporais que influenciam diretamente sua eficácia e confiabilidade. O tempo decorrido entre a captura de um evento físico e a apresentação de sua interpretação ao operador é denominado **latência** (BUTTAZZO, 2011).

Em aplicações interativas, especialmente aquelas que envolvem sobreposição de informações em tempo real e possibilidade de atuação sobre o processo, a latência torna-se fator determinante para qualidade da experiência e segurança operacional (AZUMA, 1997; BUTTAZZO, 2011).

2.1.8.1 Conceitos de Tempo Real em Sistemas Computacionais

Um sistema computacional pode ser classificado como de tempo real quando seu comportamento correto depende não apenas da lógica funcional, mas também do cumprimento de restrições temporais específicas (BUTTAZZO, 2011).

Sistemas de tempo real são geralmente categorizados como (BUTTAZZO, 2011):

- Tempo real rígido (*hard real-time*): falhas no cumprimento do prazo temporal podem resultar em consequências críticas;
- Tempo real flexível (*soft real-time*): atrasos degradam desempenho, mas não necessariamente comprometem segurança imediata.

Aplicações baseadas em visão computacional e interfaces aumentadas enquadram-se tipicamente como sistemas de tempo real flexível, nos quais atrasos excessivos impactam usabilidade e precisão perceptiva, mas não implicam, por si só, falha catastrófica do sistema.

Ainda assim, a manutenção de latência reduzida é essencial para garantir coerência entre ação física e resposta digital.

2.1.8.2 *Latência em Sistemas Distribuídos*

Em arquiteturas distribuídas, a latência resulta da soma de diferentes componentes, incluindo (FIELDING, 2000; BUTTAZZO, 2011):

- Tempo de captura e processamento da imagem;
- Tempo de inferência do modelo de visão computacional;
- Tempo de transmissão de dados pela rede;
- Tempo de processamento no servidor intermediário;
- Tempo de resposta do controlador industrial.

Cada etapa adiciona atraso acumulado ao ciclo completo de interação. Quando múltiplos processos ocorrem sequencialmente, o tempo total pode comprometer a percepção de fluidez pelo usuário.

A análise da latência deve considerar tanto o tempo médio de resposta quanto variações temporais (*jitter*), que podem gerar instabilidade perceptiva na sobreposição de elementos virtuais (AZUMA, 1997; BUTTAZZO, 2011).

2.1.8.3 *Impacto da Latência na Interação Humano-Sistema*

Em interfaces baseadas em realidade aumentada, a latência afeta diretamente a estabilidade visual da ancoragem virtual (AZUMA, 1997; SCHMALSTIEG; HÖLLERER, 2016). Atrasos significativos podem provocar desalinhamento temporário entre o movimento da câmera e a atualização dos elementos digitais, resultando em sensação de instabilidade.

Do ponto de vista cognitivo, atrasos excessivos entre ação do operador e resposta do sistema podem reduzir confiança na interface e aumentar probabilidade de repetição indevida de comandos.

Portanto, o projeto de sistemas interativos industriais deve buscar equilíbrio entre complexidade computacional e capacidade de resposta, garantindo que o ciclo de atualização permaneça dentro de limites aceitáveis para interação fluida.

2.1.8.4 Considerações sobre Robustez e Recuperação de Falhas

Além da latência média, a robustez do sistema frente a falhas temporárias é aspecto relevante. Em aplicações baseadas em visão computacional, podem ocorrer perdas momentâneas de rastreamento devido a oclusões ou variações de iluminação (AZUMA, 1997; SCHMALSTIEG; HÖLLERER, 2016). Da mesma forma, a comunicação em rede pode apresentar atrasos transitórios.

Sistemas interativos robustos devem prever mecanismos de recuperação, como (STOUFFER *et al.*, 2015; BUTTAZZO, 2011):

- Reestabilização automática da ancoragem após perda temporária de detecção;
- *Timeout* para requisições não respondidas;
- Confirmação de execução de comandos de escrita;
- Sincronização periódica de variáveis críticas.

Essas medidas contribuem para manter consistência operacional mesmo diante de perturbações momentâneas, característica essencial em ambientes industriais.

A consideração explícita de latência e robustez integra a fundamentação necessária para sistemas que combinam percepção visual, processamento computacional e interação com dispositivos de controle.

2.2 Trabalhos Relacionados

A revisão de trabalhos relacionados permite posicionar a solução desenvolvida em relação a iniciativas anteriores que combinaram Realidade Aumentada, Internet das Coisas Industrial e comunicação com sistemas de automação. Entre os estudos diretamente associados ao tema, destaca-se o trabalho de Rosa (2019), desenvolvido no mesmo contexto institucional e orientado à aplicação de RA como ferramenta de suporte à manutenção industrial.

O trabalho de Rosa (2019) propôs uma aplicação de Realidade Aumentada integrada a dados industriais, utilizando protocolos como Modbus TCP, PROFINET e MQTT para disponibilizar informações de processo ao operador. A solução empregou marcadores visuais para identificação dos equipamentos e exibiu indicadores relacionados à operação e manutenção, como dados de produção, alarmes e parâmetros associados ao desempenho da planta.

Essa abordagem apresenta contribuição relevante ao demonstrar a viabilidade de integrar RA e dados industriais em ambiente acadêmico aplicado, aproximando conceitos de Indústria 4.0, automação e visualização contextual. Além disso,

o trabalho evidencia que a RA pode atuar como uma camada informacional sobre o processo físico, reduzindo a necessidade de consulta a painéis externos e ampliando a disponibilidade de dados para o operador.

Assim, o trabalho relacionado de Rosa (2019) constitui referência importante para a motivação da proposta, especialmente por demonstrar a relevância da RA integrada à comunicação industrial em ambiente acadêmico aplicado. A partir dessa base, a definição metodológica do presente trabalho delimita uma alternativa arquitetural voltada à operação distribuída, à separação entre processamento e renderização e à redução da dependência de marcadores físicos.

2.3 Síntese dos Fundamentos Teóricos

A análise desenvolvida ao longo deste capítulo evidencia que a construção de interfaces industriais aumentadas exige a articulação entre múltiplos domínios tecnológicos. A digitalização dos processos produtivos estabelece o contexto no qual dados operacionais podem ser coletados, transmitidos e interpretados de forma estruturada.

A Realidade Aumentada insere-se nesse cenário como tecnologia de visualização capaz de associar informações digitais ao ambiente físico observado. Entretanto, sua efetividade depende da identificação consistente dos componentes e da estabilidade do registro visual, garantindo coerência entre elementos virtuais e componentes reais.

A robustez desse registro está vinculada à identificação consistente de estruturas visuais na cena, etapa que pode ser viabilizada por técnicas de detecção automatizada baseadas em aprendizado profundo. Paralelamente, a integração com sistemas de controle industrial requer protocolos de comunicação estruturados e mecanismos que assegurem confiabilidade na leitura e eventual escrita de variáveis de processo.

Do ponto de vista da interação, a interface deve considerar princípios de clareza informacional, segurança operacional e comportamento temporal adequado, uma vez que a estabilidade perceptiva e a confiabilidade do sistema estão associadas à capacidade de resposta ao longo de múltiplas camadas de processamento.

Observa-se, portanto, que a implementação de soluções industriais aumentadas envolve a convergência entre percepção visual, estabilização temporal, comunicação industrial e interação humano-sistema. A compreensão integrada desses fundamentos constitui base essencial para o desenvolvimento de arquiteturas coerentes e tecnicamente consistentes.

3 METODOLOGIA

Este capítulo apresenta o planejamento metodológico adotado no desenvolvimento do presente trabalho. A proposta consistiu na criação de um sistema de Realidade Aumentada (RA) interativa para monitoramento e ajuste de parâmetros de uma máquina industrial didática. Para isso, a metodologia foi estruturada de modo a descrever o delineamento da pesquisa, a arquitetura adotada, o plano de desenvolvimento do protótipo, a estratégia iterativa de implementação e os procedimentos de avaliação técnica da solução.

3.1 Delineamento da Pesquisa

A presente pesquisa caracteriza-se como aplicada, de natureza experimental e com abordagem predominantemente qualitativa, complementada por métricas quantitativas de desempenho. Seu objetivo foi desenvolver e avaliar uma solução tecnológica para integração entre ambiente industrial físico e interface aumentada digital, permitindo monitoramento e ajuste de parâmetros operacionais em tempo real.

A classificação como pesquisa aplicada decorre de sua orientação para a solução de um problema concreto, enquanto o caráter experimental está associado à construção de um protótipo funcional e à realização de ensaios em condições controladas. A predominância qualitativa decorre do foco na concepção, implementação e análise funcional da interface aumentada, considerando a adequação do fluxo de operação, a clareza das informações apresentadas e a coerência da integração com o equipamento. As métricas quantitativas foram utilizadas como apoio à avaliação técnica, especialmente nos aspectos relacionados ao desempenho da detecção, estabilidade temporal dos elementos visuais, latência de resposta e confiabilidade da comunicação industrial.

Complementarmente, a análise exploratória da interação com a interface observará aspectos como clareza informacional, compreensão do estado operacional e percepção geral de uso da solução proposta. Essa etapa terá caráter descritivo e não busca substituir uma avaliação ergonômica formal ou um estudo estatístico com múltiplos operadores.

3.2 Arquitetura do Sistema

A arquitetura adotada foi definida a partir da separação entre cliente de visualização, processamento computacional e integração industrial. Essa organização estabeleceu a distribuição de responsabilidades entre captura e apresentação da imagem, análise visual e comunicação com o equipamento.

Do ponto de vista de implantação, o sistema foi composto por três blocos principais: um dispositivo móvel executando a interface de RA, um servidor local de processamento executado em um *notebook* conectado à mesma rede e uma camada de integração industrial responsável pela comunicação com o equipamento. Do ponto de vista funcional, a arquitetura foi organizada em quatro camadas lógicas: percepção visual, análise automática e rastreamento, integração industrial e interface aumentada.

A validação foi delimitada ao torno didático MIRAC-PC, equipamento apresentado na Figura 7, controlado por um computador com LinuxCNC e integrado a um CLP Siemens S7-1200 responsável por funções auxiliares, como troca de ferramentas, permissões de acionamento e comando da porta. A arquitetura previu uma camada intermediária para consulta dessas unidades industriais via Modbus TCP, mantendo a interface desacoplada dos endereços físicos de cada controlador.

Figura 7 – Torno didático MIRAC-PC utilizado no desenvolvimento



Fonte: Autoria própria, 2026

Além da composição física já descrita, convém explicitar o papel funcional da planta no estudo. O MIRAC-PC foi utilizado como cenário de ensaio por reunir componentes visíveis, estados industriais monitoráveis e comandos discretos associados à operação de uma máquina-ferramenta. Assim, antes de detalhar a arquitetura computacional, a planta deve ser compreendida como o ambiente no qual a aplicação busca associar aquilo que o operador vê na câmera aos dados e comandos disponíveis no sistema de controle.

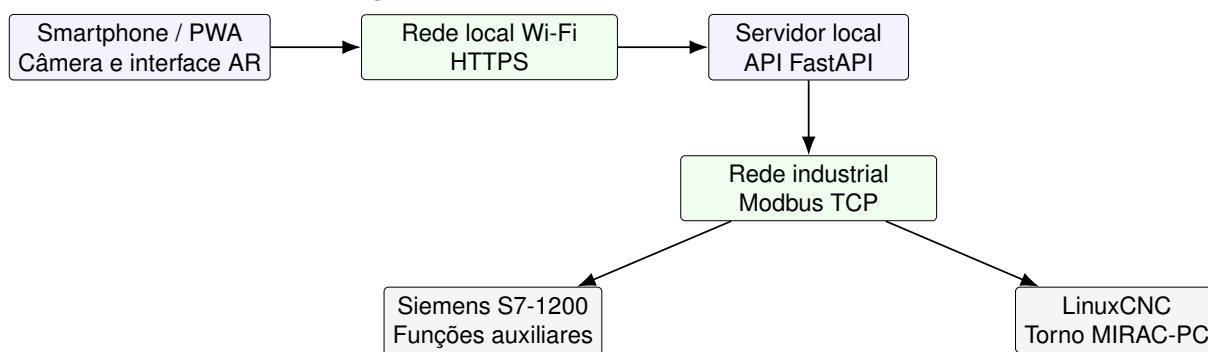
Em termos práticos, o sistema proposto não busca substituir a lógica de controle da máquina, mas oferecer uma camada de visualização e interação contextual sobre componentes observáveis. A aplicação deve permitir que o operador

acompanhe, diretamente sobre a imagem da câmera, estados e variáveis como: porta aberta ou fechada; condição da castanha; estado da iluminação; ferramenta atual e ferramenta desejada; etapa da troca de ferramenta; rotação do spindle; estado do LinuxCNC; sinais de limite; emergência; e condições de falha. Entre as falhas de interesse estão, por exemplo: pressão baixa associada à castanha; falhas de abertura ou fechamento da porta; falha da torre de ferramenta; falhas de drivers ou inversores; e falhas reportadas pelo LinuxCNC.

Também foram previstos comandos operacionais controlados pela interface, sempre mediados pelo servidor e sujeitos a confirmação antes da escrita nos controladores. Entre esses comandos estão: abrir ou fechar a porta; abrir ou fechar a castanha; ligar ou desligar a iluminação; selecionar ferramenta; habilitar a troca automática; ligar ou desligar a máquina pelo LinuxCNC; iniciar, pausar, retomar ou abortar ciclo CNC; ligar ou desligar o spindle; referenciar eixos; e resetar emergência. Dessa forma, a planta funciona como um cenário de validação no qual informações industriais, reconhecimento visual e ações do operador podem ser associados aos componentes físicos observados.

A Figura 8 sintetiza a organização prevista entre dispositivo móvel, servidor local e unidades industriais associadas ao MIRAC-PC.

Figura 8 – Arquitetura simples da solução



Fonte: Autoria própria, 2026

3.2.1 Camada de Percepção Visual

Para obter as imagens necessárias à identificação dos componentes, previu-se a utilização da câmera de um dispositivo móvel acessada por meio de uma aplicação web progressiva (Progressive Web App, abreviado como PWA). Quadros pontuais da cena seriam encaminhados por HTTPS a um servidor local responsável pelo processamento visual.

Também foi prevista a organização de um conjunto de dados composto por imagens reais do MIRAC-PC, contemplando variações de distância, ângulo de

observação e iluminação. A identificação seguiria uma abordagem *markerless*, sem a fixação de marcadores artificiais na máquina.

3.2.2 Camada de Análise Automática e Rastreamento

Para identificar os componentes visíveis do MIRAC-PC, previu-se o treinamento de um modelo de reconhecimento de objetos a partir das imagens coletadas. As regiões identificadas seriam utilizadas como referências bidimensionais para o posicionamento contextual dos elementos aumentados. Neste trabalho, a expressão *markerless* é empregada no sentido de não depender de marcadores artificiais fixados ao equipamento, não implicando estimação tridimensional completa de pose ou registro espacial em seis graus de liberdade.

Para acompanhar os componentes durante o vídeo sem exigir uma nova inferência completa a cada quadro, também foi prevista a avaliação de mecanismos leves de rastreamento entre análises sucessivas. Os métodos e parâmetros efetivamente adotados são apresentados no Capítulo 4.

A delimitação metodológica da proposta considerou o trabalho de Rosa (2019) como referência conceitual por sua integração entre RA e dados industriais. A partir dessa base, o presente trabalho buscou explorar uma organização distinta, centrada em arquitetura web distribuída, renderização local dos *overlays*, abstração das variáveis industriais e identificação visual de componentes sem marcadores fixos. O Quadro 1 sintetiza essa diferenciação, não como comparação de desempenho, mas como posicionamento do escopo desenvolvido.

Quadro 1 – Síntese comparativa entre o trabalho relacionado e a proposta metodológica

Aspecto	Rosa (2019)	Proposta deste trabalho
Referência visual	Marcadores visuais associados aos equipamentos	Detecção de componentes reais sem marcadores fixos
Renderização	Aplicação de RA vinculada ao fluxo de visualização	Renderização local dos <i>overlays</i> no cliente web
Integração industrial	Uso de protocolos industriais para disponibilização de dados	Camada intermediária com <i>tags</i> , comandos e unidades Modbus
Interação	Ênfase na visualização contextual de informações	Visualização e acionamento por menus, gestos e voz com confirmação
Contribuição metodológica	Demonstração de RA integrada a dados industriais	Exploração de uma arquitetura distribuída para IHM aumentada interativa

Fonte: Autoria própria, 2026

3.2.3 Camada de Integração Industrial

Para integrar a interface aos controladores associados ao MIRAC-PC, previu-se a utilização do protocolo Modbus TCP por meio de uma camada intermediária executada no servidor local. Essa camada organizaria o acesso às variáveis industriais sem expor diretamente à interface os endereços físicos dos controladores.

O planejamento contemplou a leitura de estados operacionais e a escrita controlada de comandos discretos e *setpoints*, observando os limites de segurança definidos pela lógica de controle do equipamento.

3.2.4 Camada de Interface Aumentada

Para apresentar as informações digitais sobre os componentes reconhecidos, previu-se o desenvolvimento de uma PWA executada no navegador de um dispositivo móvel, como *smartphone* ou tablet. A câmera embarcada seria utilizada tanto para captura quanto para visualização da cena.

Os elementos de sobreposição seriam renderizados localmente pela interface a partir de informações estruturadas fornecidas pelo servidor. Dessa forma, o vídeo permaneceria no dispositivo cliente, enquanto apenas quadros pontuais seriam encaminhados para análise.

Também foi prevista a avaliação de formas complementares de interação do operador com a interface aumentada, incluindo seleção de menus contextuais, acionamentos por gestos e comandos por voz. Essas alternativas seriam analisadas quanto à clareza de uso, confirmação operacional e integração com a mesma camada de escrita industrial.

3.3 Plano de Desenvolvimento do Trabalho

O desenvolvimento da solução foi planejado em etapas progressivas, considerando a natureza multidisciplinar do problema investigado. O plano contemplou a articulação entre percepção visual, processamento computacional distribuído, comunicação industrial e interação homem-máquina em contexto de Realidade Aumentada.

A primeira etapa prevista correspondeu à implantação da infraestrutura distribuída necessária ao funcionamento do sistema, incluindo comunicação entre dispositivo móvel e backend local, acesso seguro em rede local, captura de vídeo por navegador e troca inicial de dados entre as camadas.

Na etapa seguinte, foi prevista a estruturação do modelo de interface aumentada, com definição do formato utilizado para representar e transmitir os elementos visuais apresentados ao usuário.

O plano também contemplou o desenvolvimento da camada de integração industrial para leitura e escrita de variáveis operacionais do MIRAC-PC por meio de uma representação lógica intermediária.

Por fim, foi prevista a incorporação da percepção visual automática para localização dos componentes e posicionamento contextual dos elementos aumentados. A implementação efetivamente realizada em cada etapa e os ajustes decorrentes do desenvolvimento são apresentados no Capítulo 4.

3.4 Estratégia de Desenvolvimento Iterativo

A estratégia de desenvolvimento foi organizada em ciclos iterativos e incrementais de implementação, verificação funcional e refinamento técnico. Cada ciclo foi delimitado por objetivos associados a um ou mais blocos da arquitetura.

Ao término de cada iteração, previu-se a verificação da funcionalidade implementada, de sua compatibilidade com os módulos existentes e de seu comportamento operacional. A etapa subsequente somente incorporou novos componentes após a obtenção de uma base funcional para continuidade do desenvolvimento.

Essa abordagem permitiu acompanhar progressivamente a integração entre os blocos planejados e orientar os refinamentos necessários ao protótipo.

3.5 Procedimentos de Avaliação

A avaliação da solução foi estruturada de forma a verificar se os métodos empregados respondem adequadamente ao problema de pesquisa e aos objetivos estabelecidos. Em consonância com os fundamentos discutidos na revisão bibliográfica, os procedimentos de avaliação foram organizados em quatro eixos principais: desempenho da detecção, estabilidade temporal dos elementos visuais, latência sistêmica e confiabilidade da comunicação. Complementarmente, a avaliação exploratória da interação observou aspectos qualitativos de uso.

As métricas operacionais foram obtidas por instrumentação incorporada à própria aplicação. Durante os ensaios, o frontend e o backend registraram indicadores relacionados à latência, às detecções e à comunicação industrial. Os dados foram agrupados em sessões experimentais identificadas pelo cenário e pelo dispositivo utilizado, permitindo comparar condições distintas de teste.

3.5.1 Desempenho da Detecção

O desempenho do detector foi avaliado em conjunto de imagens separado das amostras utilizadas no treinamento. Foram consideradas as métricas de precisão,

revocação, *F1-score* e *mean Average Precision* (mAP), além da análise do *Intersection over Union* (IoU) entre caixas delimitadoras previstas e anotações de referência.

Durante a operação da aplicação, também foram registrados a confiança das detecções apresentadas ao usuário, a quantidade de componentes reconhecidos e a origem das atualizações visuais, distinguindo detecção, rastreamento e permanência temporária.

3.5.2 Estabilidade Temporal dos Elementos Visuais

A estabilidade temporal foi avaliada a partir da variação das caixas delimitadoras ao longo do tempo durante a operação da aplicação. Foram observados o deslocamento normalizado entre centros de caixas consecutivas, a IoU entre quadros e a continuidade visual dos componentes reconhecidos.

Essas métricas permitiram examinar oscilações, perdas temporárias de detecção e transições durante o movimento da câmera. A análise também distinguiu detecções produzidas diretamente pelo modelo, posições atualizadas por rastreamento e sobreposições mantidas por permanência visual temporária.

3.5.3 Latência Sistêmica

A latência sistêmica foi avaliada por meio de registros temporais internos associados às etapas de captura no dispositivo móvel, transmissão dos quadros pela rede local, processamento visual no servidor e atualização da interface.

Entre os intervalos temporais de interesse estiveram o tempo entre captura do quadro e renderização da sobreposição, o tempo total de atendimento da rota de análise, o tempo consumido pelo processamento visual e a duração das operações de comunicação industrial quando aplicável.

Os resultados foram analisados por meio de estatísticas descritivas, permitindo observar o desempenho médio da solução e eventuais variações críticas que comprometeram a fluidez da interface.

3.5.4 Confiabilidade da Comunicação

A confiabilidade da comunicação foi avaliada a partir da observação do comportamento do sistema durante a troca de dados com os controladores associados ao MIRAC-PC. A análise considerou a taxa de sucesso de leituras e escritas, a ocorrência de falhas de comunicação, *timeouts*, confirmações tardias e divergências entre valores solicitados e valores efetivamente aplicados.

Essa análise verificou se a integração industrial proposta apresentou robustez suficiente para sustentar a interface aumentada como instrumento de monitoramento e ajuste de parâmetros.

3.5.5 Avaliação Exploratória da Interação

Além das métricas quantitativas, realizou-se uma avaliação exploratória qualitativa da interação com a interface, voltada à observação de aspectos como clareza das informações apresentadas, legibilidade dos elementos visuais, percepção das confirmações operacionais e facilidade geral de compreensão da solução no contexto de uso proposto. Essa avaliação tem a finalidade de apoiar a análise funcional do protótipo, sem caracterizar ensaio formal de usabilidade com amostra representativa de operadores.

3.6 Delimitações do Estudo

O escopo deste trabalho foi delimitado de modo a manter o foco na viabilidade técnica da solução proposta, respeitando o tempo disponível para execução do TCC e os recursos acessíveis no ambiente acadêmico.

A arquitetura experimental centrou-se no uso de dispositivo móvel como cliente de visualização e de um computador na mesma rede local como servidor de processamento e integração. Assim, o estudo não contemplou análise aprofundada de operação remota pela internet pública, comparação entre diferentes modelos de dispositivos móveis ou avaliação extensiva de escalabilidade para múltiplos clientes simultâneos.

A validação industrial foi delimitada ao torno didático MIRAC-PC, ao computador LinuxCNC e ao CLP Siemens S7-1200 associado ao equipamento. A avaliação não abrangeu a coordenação de uma célula de manufatura completa nem a integração com controladores externos ao conjunto analisado.

Também não fizeram parte do escopo deste estudo testes com múltiplos operadores em quantidade suficiente para análise estatística, avaliação ergonômica formal, comparação quantitativa com IHMs industriais tradicionais ou medição de impacto produtivo sobre o equipamento.

4 APRESENTAÇÃO DOS RESULTADOS

4.1 Desenvolvimento da Solução

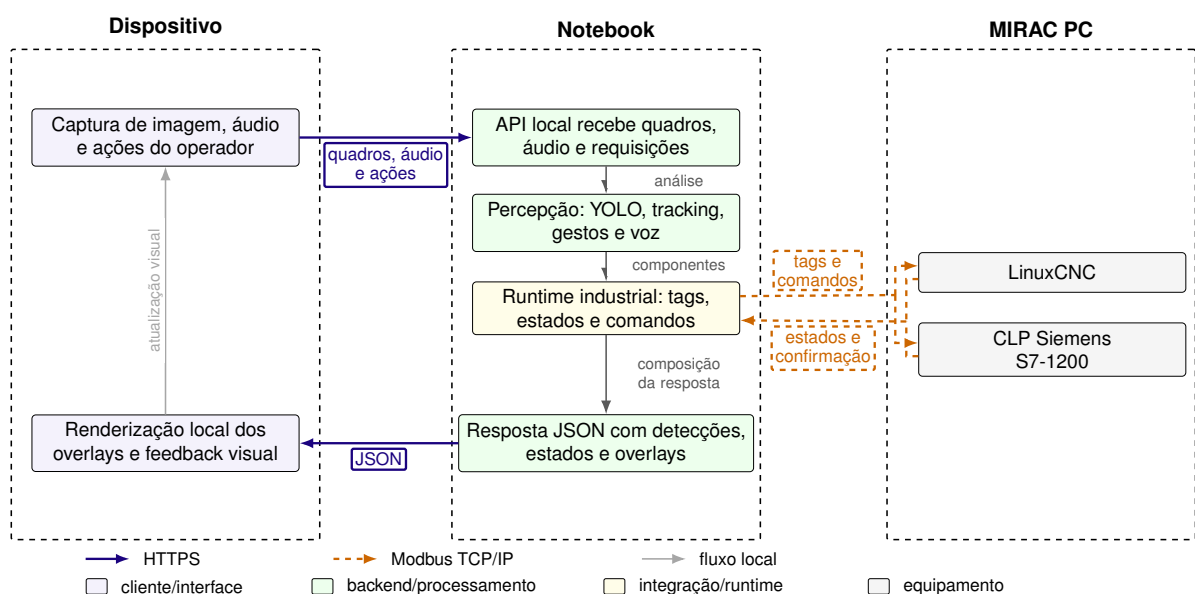
O desenvolvimento da solução foi conduzido segundo a lógica incremental definida na metodologia, de modo que a arquitetura adotada fosse materializada e progressivamente validada em suas camadas constituintes. As evidências apresentadas nesta seção demonstram a consolidação técnica dos módulos que compõem o sistema.

Este item reúne as etapas de implementação e as validações funcionais de baixo e médio nível realizadas durante a construção do protótipo, explicitando como a solução evoluiu de uma arquitetura planejada para um sistema operacionalmente executável.

A organização adotada separa dois planos complementares do trabalho. O primeiro é construtivo e descreve a implementação e a verificação operacional dos módulos. O segundo é avaliativo e examina o comportamento do sistema enquanto solução técnica submetida a ensaios e observações mais amplas.

Durante o desenvolvimento, a decomposição do sistema em blocos funcionais progressivos reduziu a complexidade inicial da implementação, favoreceu o isolamento de falhas e possibilitou validações parciais antes da incorporação de novos componentes. Essa abordagem foi especialmente importante para introduzir gradualmente os recursos de percepção visual sem comprometer a verificação das demais camadas da aplicação.

Figura 9 – Fluxo operacional geral da arquitetura implementada



Fonte: Autoria própria, 2026

A Figura 9 mostrada acima apresenta, de forma conceitual, o fluxo operacional adotado para introduzir o leitor às camadas que serão detalhadas ao longo deste capítulo.

4.1.1 Consolidação da arquitetura distribuída do sistema

A primeira etapa efetiva do desenvolvimento concentrou-se na consolidação da arquitetura distribuída do sistema, definida metodologicamente a partir da separação entre dispositivo cliente, *backend* de processamento e camada de integração industrial. Essa separação não constituiu apenas uma decisão de organização de *software*, mas um princípio estruturante da própria solução, uma vez que permitiu distribuir adequadamente responsabilidades entre captura e exibição local, processamento intermediário e acesso às variáveis do sistema físico.

No plano de implantação, estabeleceu-se um arranjo em que o dispositivo móvel passou a executar a interface de visualização aumentada, enquanto o *backend* local assumiu a recepção dos quadros enviados para análise, a organização da lógica de resposta e a articulação com a camada de integração industrial. Essa definição mostrou-se compatível com a necessidade de preservar o vídeo no dispositivo do usuário, reduzir o envio de dados visuais redundantes e manter um ponto central de processamento e controle do estado global da aplicação.

Durante o desenvolvimento e os ensaios operacionais, o servidor local foi executado em um *notebook* Acer Nitro V15, modelo ANV15-52-737P, com processador Intel Core i7-13620H, 16 GB de memória RAM DDR5, placa de vídeo NVIDIA GeForce RTX 4050 com 6 GB de VRAM dedicada e sistema operacional Windows 11. Esse equipamento concentrou a execução do *backend*, da API local, dos serviços de visão computacional e da comunicação com as unidades industriais.

A validação inicial dessa arquitetura concentrou-se na verificação do fluxo de comunicação entre cliente e *backend* em ambiente local com conexão segura. Essa etapa estabeleceu a base operacional sobre a qual foram construídas as demais capacidades do sistema, incluindo análise visual, integração industrial e renderização aumentada.

4.1.2 Desenvolvimento do *backend* de processamento e integração

Uma vez estabelecida a arquitetura base, procedeu-se à implementação do *backend* de processamento, responsável por receber os quadros capturados pelo dispositivo móvel, executar a lógica de análise associada a cada ciclo e retornar ao *frontend* informações estruturadas para composição da interface aumentada. A centralização dessa lógica em uma camada intermediária permitiu desacoplar a interface de visualização tanto das rotinas de processamento quanto do acesso direto aos da-

dos industriais, favorecendo manutenção, extensibilidade e segurança operacional.

O backend foi implementado como uma API local, organizada em rotas HTTP acessadas pela aplicação cliente por meio de conexão HTTPS na rede local. Essa API concentrou as operações de autenticação, análise de quadros, consulta do estado operacional, configuração de comunicação e escrita controlada de comandos. Dessa forma, o navegador do dispositivo móvel comunicou-se com o servidor por meio de requisições bem definidas, recebendo respostas em formato JSON para atualização da interface.

Para a implementação do backend e dos serviços intermediários, optou-se pela linguagem Python. Essa escolha esteve associada à ampla disponibilidade de bibliotecas voltadas tanto à visão computacional quanto à comunicação industrial, além da elevada produtividade de desenvolvimento proporcionada pela linguagem em contexto de prototipagem experimental. Sobre essa base, empregou-se o *framework* FastAPI para implementar a API da aplicação, estruturando as rotas HTTP responsáveis por receber os quadros do cliente, organizar a lógica de processamento, consultar o estado industrial e fornecer ao *frontend* respostas estruturadas em formato JSON¹.

A adoção do FastAPI mostrou-se apropriada por permitir definição clara de rotas, organização modular dos serviços e integração simples com estruturas de dados tipadas, características particularmente úteis em uma arquitetura que depende da troca recorrente de informações entre dispositivo móvel, processamento local e camada industrial. Dessa forma, o backend passou a desempenhar papel central de articulação entre a percepção visual, o estado operacional do MIRAC PC e a composição dos overlays exibidos na interface aumentada.

4.1.3 Desenvolvimento do *webapp* de visualização

A camada cliente da solução foi implementada como aplicação web progressiva executada em navegador móvel, em coerência com a diretriz metodológica de utilização de tecnologias amplamente acessíveis e independentes de plataforma. Tal escolha também está alinhada ao desenho arquitetural previamente definido, no qual a interface aumentada deve operar diretamente no dispositivo do usuário, explorando a câmera embarcada como meio de captura e visualização da cena industrial.

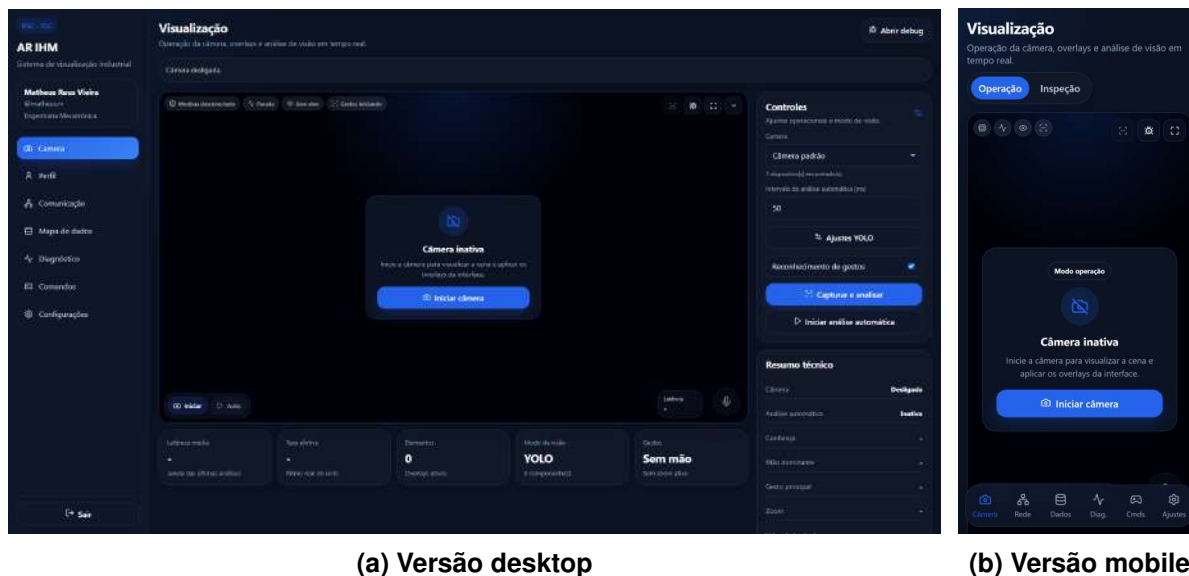
Do ponto de vista de implementação, optou-se pela utilização de React associado à ferramenta de empacotamento Vite. Essa combinação mostrou-se adequada ao contexto do trabalho por favorecer modularização da interface, atualização reativa dos elementos exibidos e rápida iteração durante o desenvolvimento do protótipo. A abordagem em componentes contribuiu para separar de forma mais clara as

¹ Formato textual leve para representação e troca de dados estruturados, amplamente utilizado na comunicação entre sistemas e aplicações web.

responsabilidades entre captura de vídeo, camada de *overlay*, controles de operação e painel de depuração.

A Figura 10 apresenta a tela de operação em visualizações desktop e mobile, evidenciando a adaptação da interface ao principal fluxo de uso.

Figura 10 – Tela de operação da interface em visualização desktop e mobile



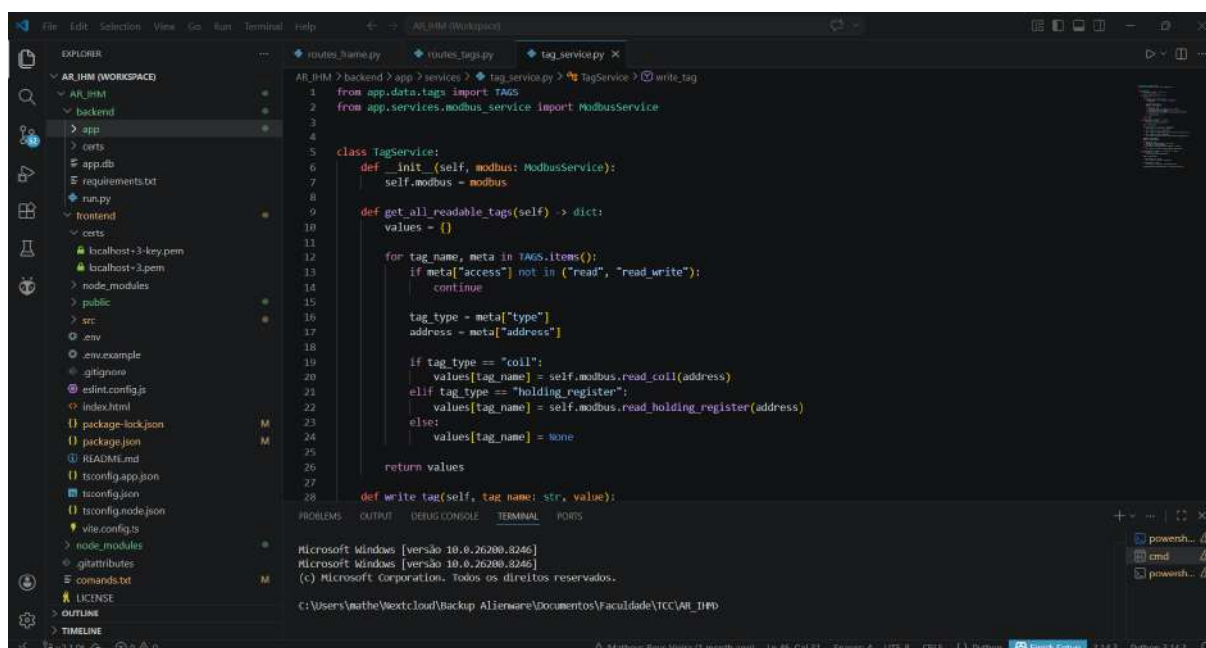
(a) Versão desktop

(b) Versão mobile

Fonte: Autoria própria, 2026

O conjunto completo de telas da aplicação foi organizado em cadernos específicos para visualização desktop, tablet e mobile, apresentados nos Apêndices A, B e C. Essa organização permite registrar a evolução visual da interface sem interromper a análise principal do desenvolvimento.

Figura 11 – Ambiente de desenvolvimento



Fonte: Autoria própria, 2026

O ambiente de desenvolvimento utilizado foi a plataforma Visual Studio Code (VSCode), mostrado na Figura 11. Esta plataforma possui uma gama diversa de extensões e integrações com outros serviços que facilitam a escrita de código, como sugestões de auto-completar tags, navegar entre diferentes arquivos vinculando definição de funções, terminal integrado em Powershell, entre outros.

Além disso, todo o desenvolvimento do projeto foi realizado utilizando o versionamento por meio da plataforma GitHub. A IDE escolhida também possui integração direta com este serviço e auxilia em questões práticas como *commit*² e identificação de arquivos modificados.

A escolha pela abordagem web, em lugar de uma aplicação nativa dedicada, decorreu principalmente de três fatores: independência em relação ao sistema operacional do dispositivo, facilidade de implantação em ambiente local e maior agilidade para experimentação arquitetural durante as etapas iniciais do trabalho. Assim, tornou-se possível concentrar esforços na validação do comportamento da solução e na integração com a planta industrial, sem introduzir complexidades adicionais de distribuição e manutenção de aplicações móveis nativas.

Com a interface cliente estabelecida, a etapa seguinte consistiu em integrar o fluxo de câmera ao ciclo de análise da aplicação, mantendo a captura e a visualização no dispositivo móvel e delegando ao backend apenas o processamento necessário à composição dos dados aumentados.

4.1.4 Implementação da captura de vídeo e envio de *frames*

A integração com a câmera do dispositivo móvel foi desenvolvida no próprio ambiente do navegador, permitindo que o fluxo visual permanecesse local ao cliente. Essa decisão foi particularmente relevante para a arquitetura do sistema, pois evitou a necessidade de transmissão contínua de vídeo processado e reduziu o volume de dados trafegados entre as camadas.

Em termos funcionais, a câmera passou a fornecer quadros periódicos ao *backend* para análise, mantendo no *frontend* a responsabilidade pela exibição da cena e pela renderização dos elementos gráficos de sobreposição. O quadro capturado foi convertido no cliente e enviado por HTTPS a uma rota HTTP da API de análise. O servidor processou o conteúdo recebido, agregou as informações de visão computacional e de estado operacional do MIRAC PC e retornou uma resposta JSON contendo elementos de *overlay*, dados de *runtime*, informações de conexão e metadados da análise.

² *Commit* é um registro de alteração feito em um projeto. Cada *commit* salva um “ponto no tempo” do código, com uma descrição do que foi alterado, permitindo acompanhar a evolução do projeto e voltar a versões anteriores se necessário.

Tal arranjo reforça a separação entre captura, processamento e apresentação. O vídeo permaneceu no dispositivo móvel, enquanto apenas quadros pontuais e dados estruturados trafegaram pela rede. Como resultado, a interface manteve a renderização local dos elementos aumentados e o backend permaneceu responsável pela interpretação da imagem, pela composição lógica dos dados e pela integração com a camada industrial.

4.1.5 Estruturação do modelo de *overlays* e renderização local

Na sequência, definiu-se o modelo de *overlays* utilizado pela interface. Em lugar de delegar ao backend a composição visual final, optou-se pelo envio de estruturas descritivas em formato JSON, contendo posicionamento, conteúdo textual, estado lógico e demais atributos necessários à renderização local pelo *frontend*. Essa decisão reforçou a separação entre lógica e apresentação, aumentou a flexibilidade visual da aplicação e preservou a imagem original no próprio dispositivo cliente.

O modelo adotado também favoreceu a evolução incremental da interface. Alterações de layout, estilo ou comportamento dos elementos puderam ser realizadas no frontend sem modificar a lógica de processamento do backend. Durante as etapas iniciais, a ancoragem dos elementos visuais foi mantida em regime controlado, por meio de referência manual e estruturas de fallback, a fim de validar a coerência do pipeline completo antes da introdução da localização visual automática.

4.1.6 Interação por menus flutuantes associados aos componentes

A primeira forma de interação aumentada foi estruturada por meio de menus flutuantes associados aos componentes identificados na cena. Cada região detectada recebeu uma caixa delimitadora bidimensional com um identificador lógico de componente. Quando o operador seleciona essa região sobre o vídeo, a interface destaca o elemento correspondente e apresenta um painel contextual posicionado nas proximidades da caixa, mantendo a relação visual entre o menu e o objeto físico observado.

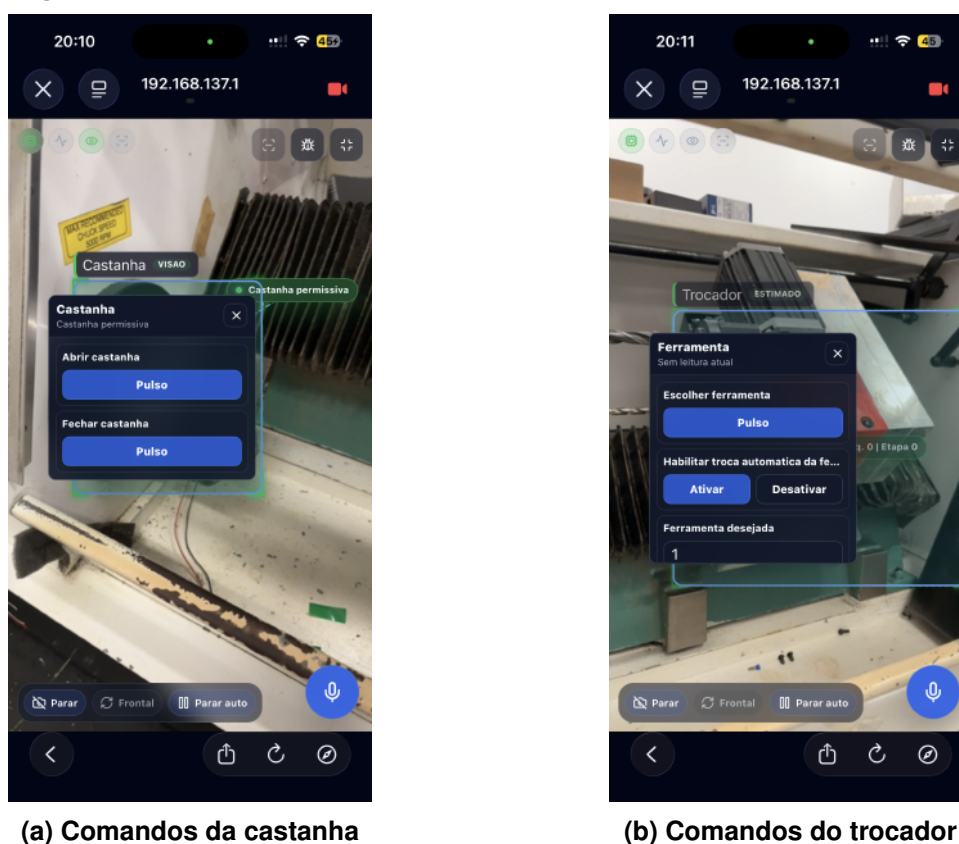
O painel reúne o estado operacional atual do componente e os comandos disponíveis para sua atuação. Essa associação não é definida diretamente no frontend: o catálogo fornecido pelo backend relaciona cada comando gravável à respectiva *tag*, unidade industrial e componente lógico. Dessa forma, a interface apresenta somente as ações pertinentes ao elemento selecionado, como comandos discretos, pulsos temporários ou ajuste de valores numéricos dentro dos limites cadastrados.

Antes de realizar qualquer escrita, a aplicação apresenta uma etapa obrigatória de confirmação contendo o componente de destino, a ação selecionada e o valor solicitado. Após a confirmação, o frontend encaminha a solicitação à rota de escrita da API. Caso o alvo industrial esteja inalcançável ou a operação não possa ser concluída,

o painel informa ao operador que o comando não foi executado. Essa sequência preserva a contextualização visual da RA sem eliminar a confirmação explícita necessária para ações capazes de alterar o estado físico do equipamento.

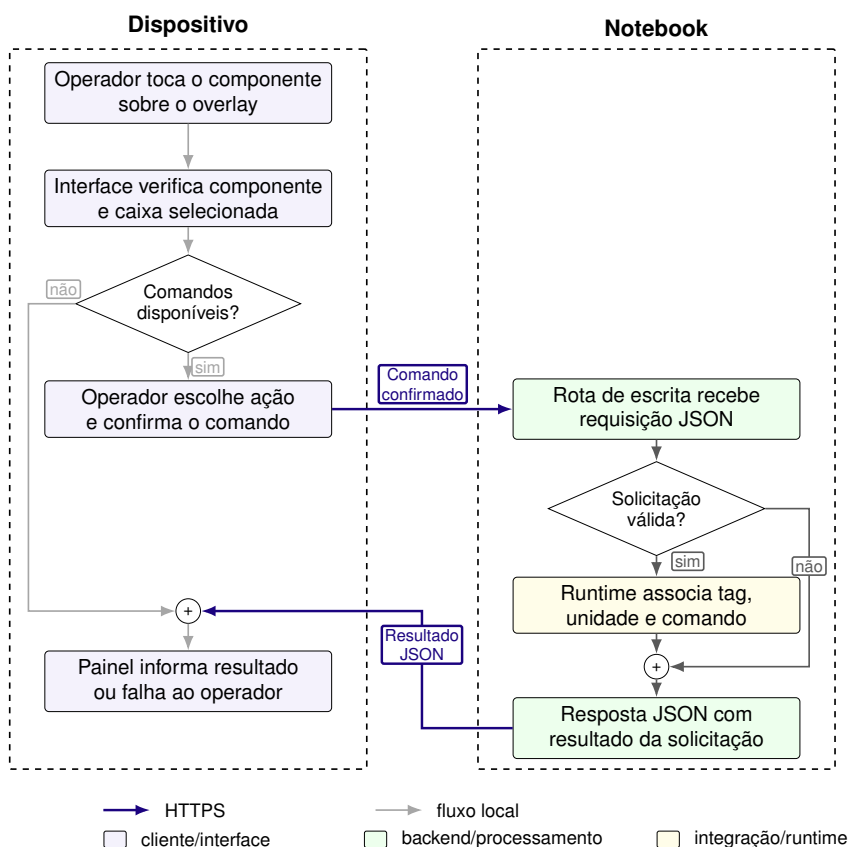
A Figura 12 apresenta exemplos desses menus durante a operação. No caso da castanha, são disponibilizados comandos diretos de abertura e fechamento. No caso do trocador de ferramentas, o painel reúne ações de seleção, habilitação da troca automática e ajuste da ferramenta desejada, evidenciando que o conteúdo do menu é definido pelo componente selecionado e pelas *tags* graváveis associadas a ele.

Figura 12 – Menus contextuais de comandos associados aos componentes



Fonte: Autoria própria, 2026

A Figura 13 sintetiza o fluxo funcional adotado para a interação por menus flutuantes, desde a seleção do componente no cliente até a validação da solicitação de comando.

Figura 13 – Fluxo da interação por menus flutuantes

Fonte: Autoria própria, 2026

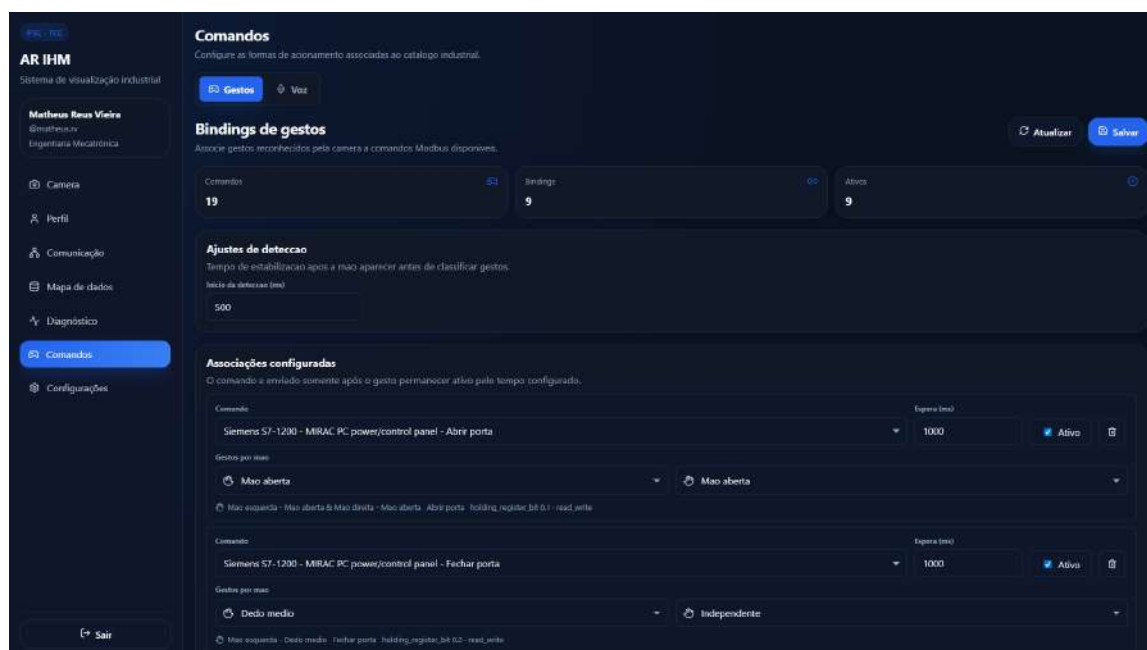
4.1.7 Interação por gestos configuráveis

A segunda forma de interação foi implementada por meio de gestos manuais configuráveis. O reconhecimento ocorre localmente no dispositivo cliente: a aplicação utiliza o MediaPipe para localizar os pontos de referência das mãos no vídeo e aplica regras de classificação sobre esses pontos para identificar poses previamente suportadas. Como esse processamento é realizado no frontend, a análise gestual não exige o envio contínuo das imagens das mãos ao backend.

Exemplos visuais das poses reconhecidas pelo sistema são apresentados isoladamente no Apêndice D, permitindo observar as formas de mão utilizadas na configuração dos comandos.

A página de configuração, mostrada na Figura 14 permite associar comandos disponíveis no catálogo industrial a gestos executados com a mão esquerda, com a mão direita ou com ambas em conjunto. Para cada associação, o usuário define o comando, o valor enviado e o intervalo durante o qual o gesto deve permanecer ativo antes do acionamento. Também pode ser configurado um tempo inicial de estabilização após a entrada da mão na cena, reduzindo ativações causadas pela transição entre poses.

Figura 14 – Tela de atribuição de gestos aos comandos



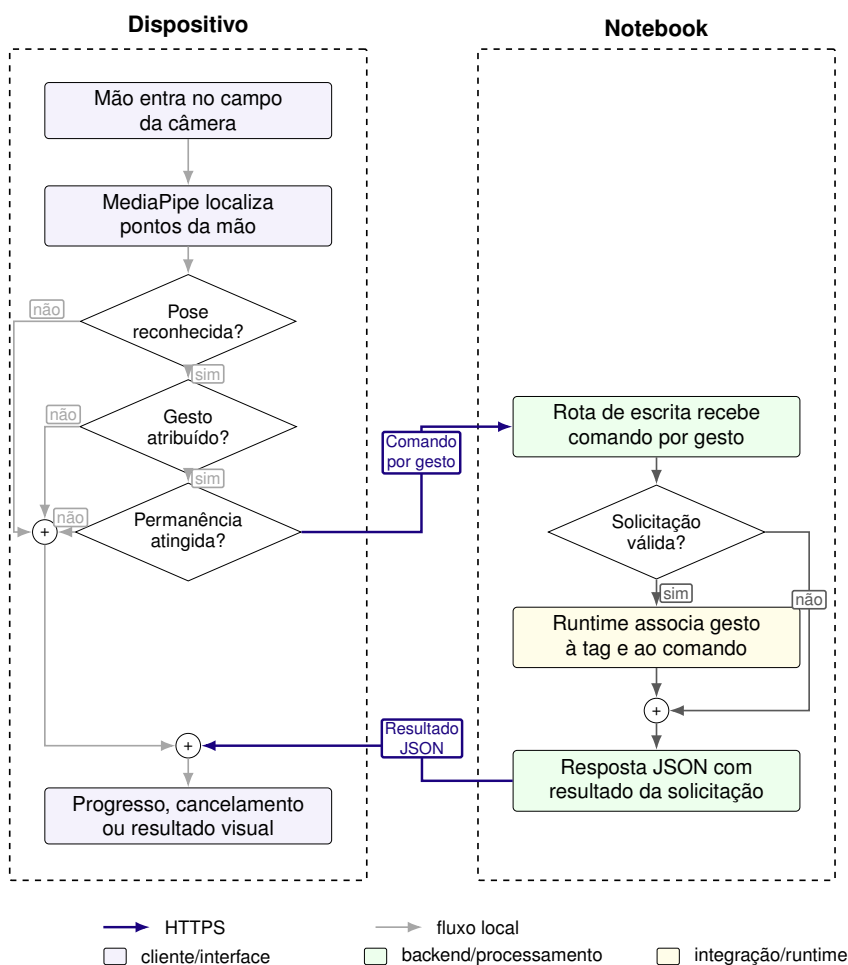
Fonte: Autoria própria, 2026

Durante a operação, a interface apresenta visualmente o progresso de permanência do gesto. Quando o intervalo configurado é atingido, o comando é enviado pela mesma rota de escrita utilizada pelos menus contextuais. Nos comandos booleanos ativados por gesto, a aplicação também envia a desativação quando a pose deixa de ser reconhecida. As associações e os parâmetros de detecção são persistidos por usuário autenticado, permitindo manter a mesma configuração em diferentes dispositivos de acesso.

Como ambos os mecanismos compartilham a mesma camada de escrita industrial, os comandos por gesto também se beneficiam das verificações de alcançabilidade, do tratamento de falhas Modbus e do registro de métricas implementados no backend. A diferença entre os fluxos está na forma de confirmação: os menus exigem confirmação explícita na interface, enquanto os gestos exigem permanência intencional da pose pelo intervalo configurado e exibem o andamento do acionamento ao operador.

A Figura 15 apresenta o fluxo de acionamento por gestos configuráveis, destacando que a identificação da pose ocorre no cliente e que a solicitação de comando permanece concentrada na mesma rota validada do backend.

Figura 15 – Fluxo da interação por gestos configuráveis



Fonte: Autoria própria, 2026

4.1.8 Interação por comandos de voz

Como terceira forma de interação, foi incorporado o acionamento por comandos de voz. O objetivo desse recurso foi oferecer uma alternativa complementar aos menus flutuantes e aos gestos, especialmente em situações nas quais o operador precisa manter a atenção visual sobre a máquina ou evitar múltiplos toques na tela durante a observação do equipamento.

A estratégia adotada utilizou reconhecimento de fala offline no backend local, por meio da biblioteca Vosk e de um modelo em português armazenado no próprio servidor. Durante a operação, o usuário mantém pressionado o botão de voz na tela da câmera; nesse período, o dispositivo captura o áudio do microfone e exibe um indicador visual de escuta. Ao soltar o botão, o frontend converte a gravação para WAV mono PCM, com taxa de amostragem de 16 kHz, e envia o arquivo à rota de transcrição da API.

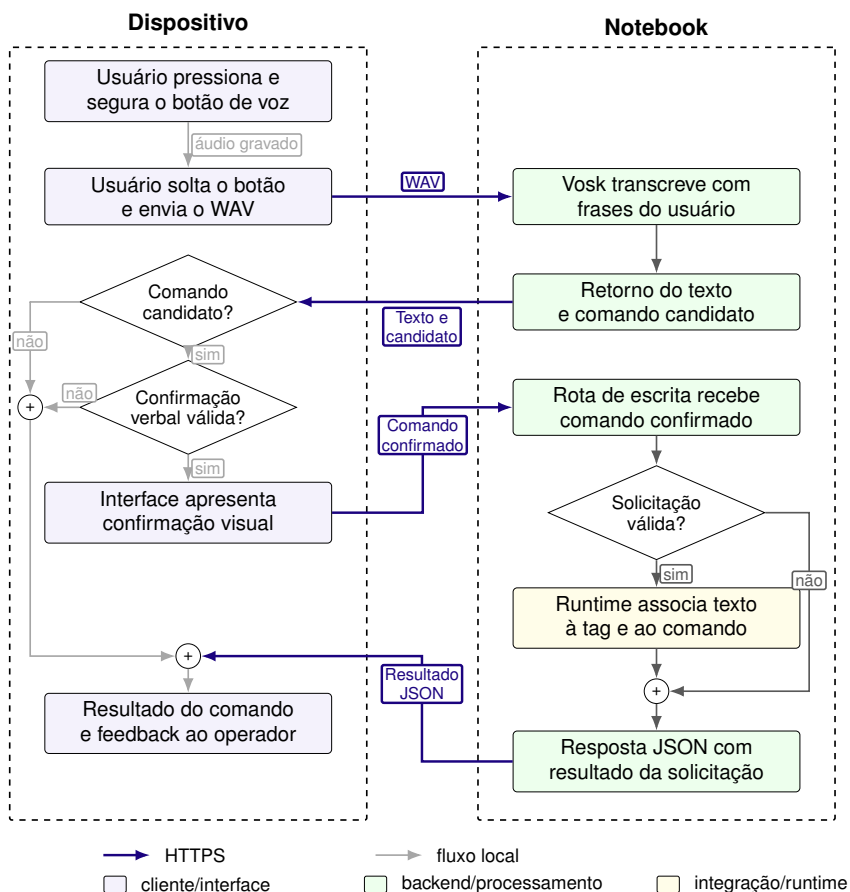
O backend retorna o texto reconhecido, sua versão normalizada e, quando possível, um comando candidato obtido a partir das frases configuradas para o usuário

autenticado. A interpretação do comando não é livre: as frases de acionamento permanecem associadas a um catálogo fechado de comandos disponíveis, obtido a partir das mesmas *tags* graváveis utilizadas pelos menus e pelos gestos. Também foi prevista uma palavra de confirmação configurável por usuário, como “confirma”, que pode ser exigida ao final da frase para reduzir acionamentos acidentais. Assim, uma frase como “abrir porta confirma” é normalizada e associada ao comando correspondente somente quando existe correspondência com o catálogo configurado.

Após o reconhecimento, a execução não ocorre de forma direta e silenciosa. A aplicação reutiliza a confirmação operacional já existente, apresentando ao operador o comando interpretado antes de enviar a escrita final. Com isso, os comandos por voz permanecem integrados à mesma política de segurança aplicada aos demais modos de interação, incluindo verificação de alcançabilidade do dispositivo, tratamento de falhas Modbus e registro de métricas.

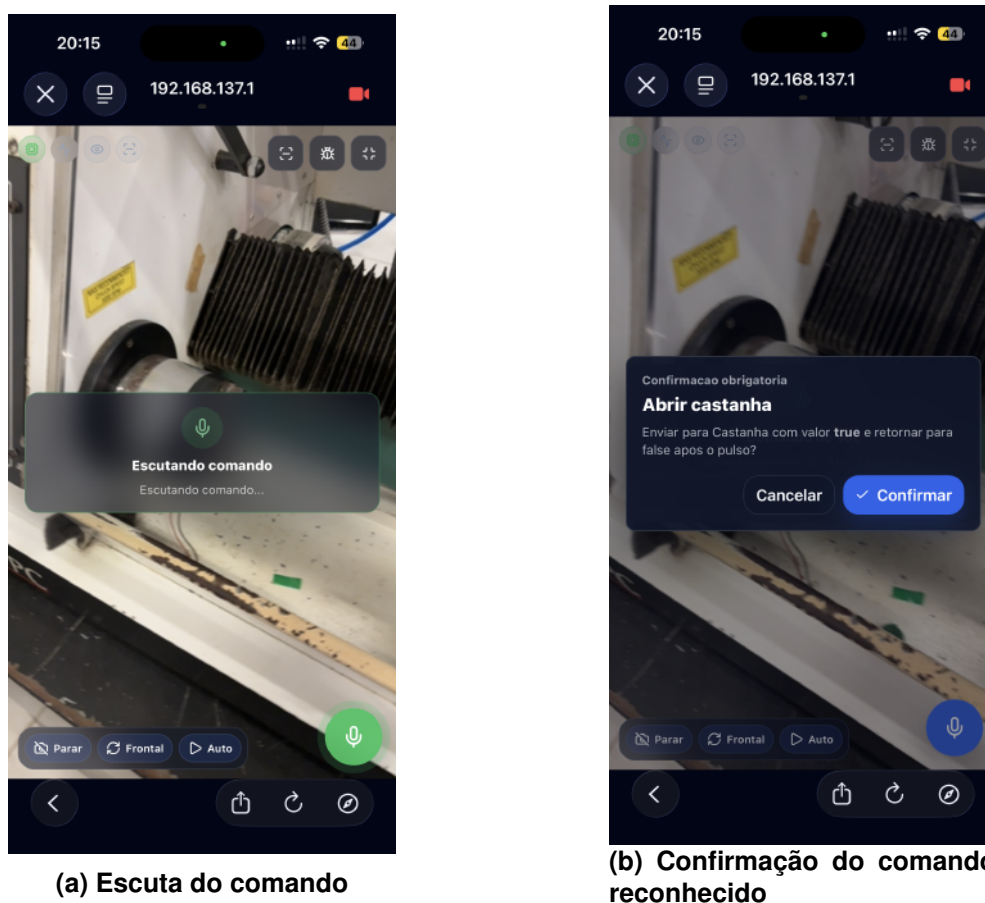
A Figura 16 apresenta o fluxo de interação por voz, no qual a captura do áudio ocorre no cliente, a transcrição é realizada no backend local com apoio das frases configuradas pelo usuário e a solicitação resultante reutiliza a mesma política de confirmação e escrita controlada.

Figura 16 – Fluxo da interação por comandos de voz



A Figura 17 apresenta evidências visuais do recurso de comando por voz durante os testes de uso.

Figura 17 – Reconhecimento de comando por voz em operação



Fonte: Autoria própria, 2026

4.1.9 Implementação da comunicação industrial via Modbus

A integração com os dados operacionais do MIRAC PC foi realizada por meio de uma camada intermediária responsável pela comunicação via protocolo Modbus. Essa camada foi desenvolvida de forma desacoplada da interface de visualização, em conformidade com os princípios de arquitetura intermediada e abstração de variáveis discutidos na fundamentação teórica, segundo os quais a interface não acessa diretamente os endereços do processo físico, mas opera sobre representações lógicas organizadas e controladas.

4.1.9.1 Adequações realizadas no CLP Siemens S7-1200

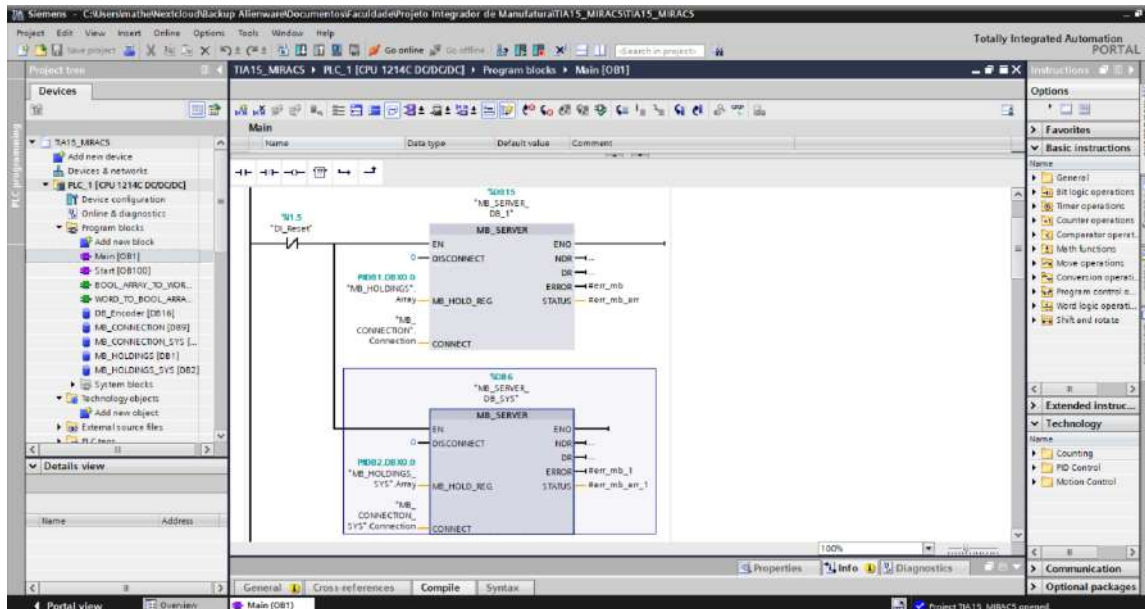
A integração não se limitou ao consumo direto de dados previamente disponíveis. Embora o MIRAC PC já utilizasse comunicação industrial em sua operação original, foram necessárias adaptações no CLP Siemens S7-1200 e no computador

LinuxCNC para disponibilizar à aplicação as variáveis relevantes sem comprometer os fluxos existentes.

No S7-1200, já existia uma instância de servidor Modbus utilizada na troca de dados com o LinuxCNC. Durante os ensaios iniciais, a utilização dessa mesma instância pela aplicação de RA introduziu concorrência entre dois clientes mestres, prejudicando a regularidade das respostas. Para separar os fluxos, foi criada uma segunda instância de servidor Modbus dedicada à aplicação desenvolvida. Essa instância atua como espelho da comunicação original: os estados relevantes são disponibilizados em ambos os servidores, permitindo que LinuxCNC e interface aumentada consultem os dados sem disputar a mesma conexão lógica.

Os comandos recebidos pelas duas instâncias foram agrupados por meio de uma operação lógica OU aplicada às palavras correspondentes. Dessa forma, uma solicitação originada pelo LinuxCNC ou pela interface aumentada pôde alimentar a mesma lógica funcional do equipamento sem eliminar o caminho de comando já existente. Os *setpoints* também foram tratados de acordo com o modo de operação vigente no LinuxCNC, preservando a coerência entre a origem da solicitação e o estado operacional do torno.

Figura 18 – Adequações realizadas no CLP Siemens S7-1200



Fonte: Autoria própria, 2026

4.1.9.2 Adequações realizadas no LinuxCNC

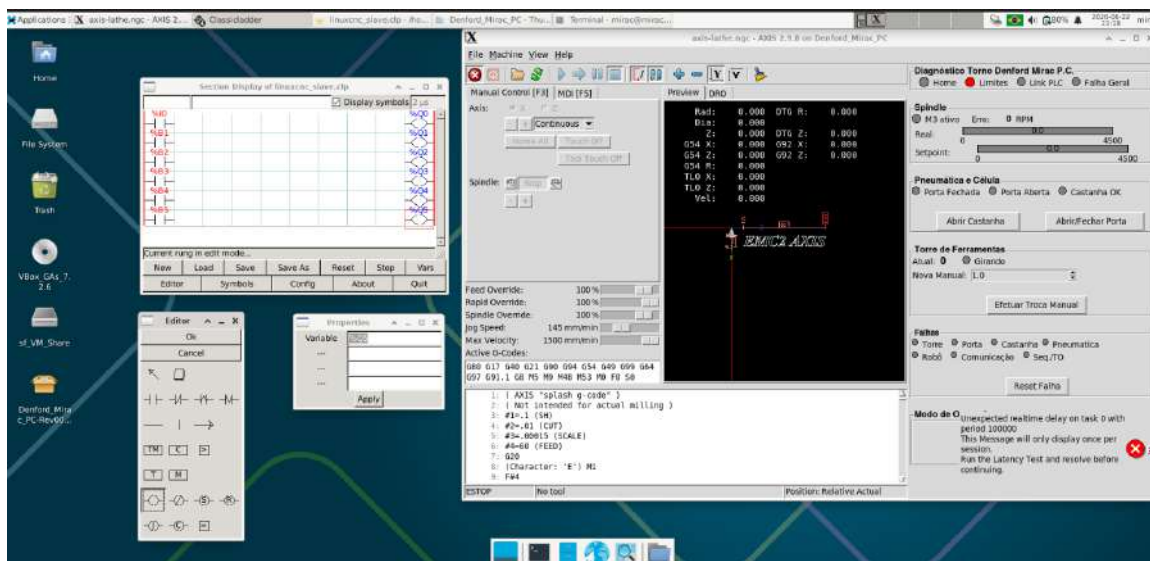
No computador LinuxCNC, o projeto original já possuía uma estrutura de comunicação destinada ao sistema supervisor do MIRAC PC. Essa base foi reaproveitada e ampliada com informações adicionais de estado, permitindo uma monitoração mais completa pela aplicação sem substituir a comunicação supervisória

existente. A estratégia adotada reduziu o impacto sobre a automação original e tornou explícita a separação entre os sinais pertencentes ao controle da máquina e as representações consumidas pela interface aumentada.

O desenvolvimento dessas adequações ocorreu em paralelo ao desenvolvimento da interface LinuxCNC na disciplina de Projeto Integrador de Automação da Manufatura. Nesse contexto, o LinuxCNC foi configurado para operar como servidor, ou escravo, Modbus por meio do Classic Ladder. As memórias internas e as *coils* utilizadas na comunicação foram associadas aos sinais do LinuxCNC por meio do arquivo HAL, responsável por estabelecer as conexões entre pinos, sinais e componentes da configuração da máquina.

Inicialmente, as modificações foram implementadas e testadas em uma máquina virtual, o que permitiu validar a configuração do Classic Ladder e a organização dos endereços Modbus antes da aplicação no equipamento físico. A Figura 19 apresenta esse ambiente de testes, no qual a interface do LinuxCNC é executada em conjunto com o Classic Ladder.

Figura 19 – Adequações realizadas no LinuxCNC em ambiente virtual



Fonte: Autoria própria, 2026

Após a validação no ambiente virtual, a configuração foi replicada no computador real do MIRAC PC. Os comandos enviados pela interface web foram então testados diretamente na máquina e apresentaram o mesmo comportamento observado na etapa de validação, confirmando que o mapeamento Modbus, as associações no HAL e a lógica implementada no Classic Ladder permaneceram compatíveis com a operação do equipamento real.

4.1.10 Consolidação do fluxo integrado

Além da leitura de variáveis operacionais, a estrutura incorporou operações controladas de escrita e ajuste de parâmetros, respeitando os princípios de segurança operacional já discutidos no referencial teórico. Também foram incorporados mecanismos de reconexão automática e tratamento de falhas temporárias, buscando preservar a continuidade funcional da aplicação diante de instabilidades momentâneas de comunicação.

Com a conclusão dessas etapas, a interface aumentada passou a operar como composição local da imagem da câmera com os dados estruturados retornados pelo backend. Em paralelo, foram implementadas páginas de apoio para diagnóstico e monitoramento dos dados, permitindo acompanhar o ciclo de análise, o estado da comunicação, as respostas recebidas e a relação entre componentes, *tags* e comandos disponíveis. Esses recursos se mostraram relevantes tanto para a fase de implementação quanto para a avaliação técnica da solução.

4.2 Validação Funcional da Solução

A validação funcional da solução foi organizada de acordo com os eixos definidos na metodologia, priorizando a verificação incremental das camadas que compõem o sistema. Essa abordagem permitiu analisar separadamente a comunicação entre cliente e backend, a estrutura de resposta da API, a renderização local dos *overlays*, a integração com variáveis industriais e os recursos de visão computacional associados à ancoragem visual.

Os resultados apresentados nesta etapa caracterizam a consolidação funcional do protótipo e estabelecem a base para a interpretação dos ensaios quantitativos. Dessa forma, a análise não se restringe à existência isolada de funcionalidades, mas observa a coerência do fluxo completo entre captura, processamento, comunicação industrial e apresentação aumentada ao operador.

4.2.1 Validação da Comunicação entre Cliente e Backend

A comunicação entre o dispositivo móvel e o backend local foi validada em rede local por meio de conexão HTTPS. O acesso seguro mostrou-se necessário tanto para atender às exigências dos navegadores modernos quanto para permitir o uso da câmera pelo frontend em dispositivos móveis.

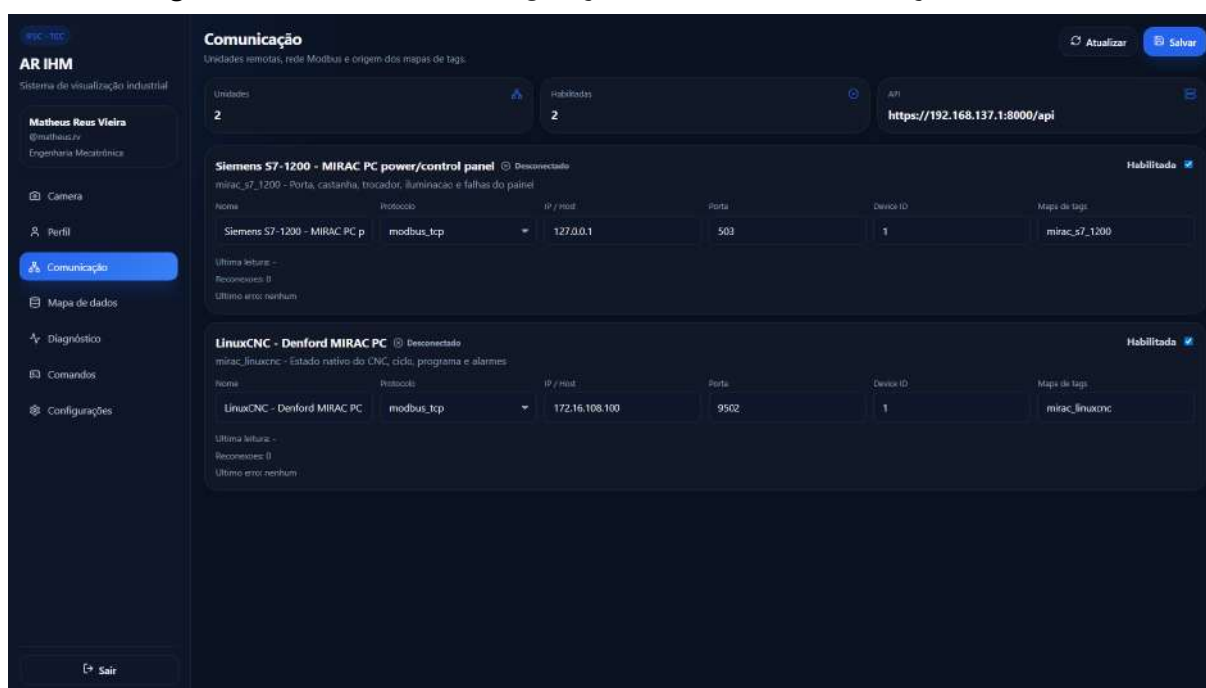
Durante os ensaios funcionais, o navegador do dispositivo móvel realizou requisições HTTP à API do backend e recebeu respostas estruturadas em JSON. Esse fluxo confirmou a separação de responsabilidades definida na arquitetura: o cliente permaneceu responsável pela captura da imagem e pela renderização visual,

enquanto o backend concentrou a análise do quadro, a organização do estado operacional e a integração com a camada industrial.

Também foi verificado que a indisponibilidade de comunicação industrial não impediu a inicialização do backend nem o acesso à aplicação. Essa característica confirmou a independência entre disponibilidade da API e disponibilidade momentânea dos dispositivos Modbus, evitando que falhas de comunicação com controladores bloqueassem o uso geral da interface.

A Figura 20 apresenta a interface utilizada para acompanhar e configurar o estado da comunicação industrial.

Figura 20 – Interface de configuração e estado da comunicação industrial



Fonte: Autoria própria, 2026

4.2.2 Validação da Renderização Local dos *Overlays*

A renderização dos *overlays* foi validada a partir do modelo no qual o backend retorna descrições estruturadas dos elementos visuais, sem enviar imagens anotadas ao frontend. As respostas da API forneceram dados como identificadores de componente, posições, textos, estados lógicos e metadados de análise, permitindo que a interface montasse localmente a camada aumentada sobre o vídeo.

Esse comportamento confirmou a viabilidade do contrato JSON adotado entre backend e frontend. A composição local preservou o vídeo no dispositivo móvel, reduziu a necessidade de tráfego contínuo de imagens processadas e permitiu ajustes visuais no frontend sem alterar a lógica de processamento do backend.

Além disso, a interface apresentou elementos semânticos associados a componentes do MIRAC PC, como estados de porta, castanha, iluminação e trocador de ferramentas. A organização dos *overlays* por componente contribuiu para associar dados industriais ao equipamento físico observado, mantendo coerência com o conceito de informação contextual discutido na fundamentação teórica.

4.2.3 Validação da Integração Industrial via Modbus

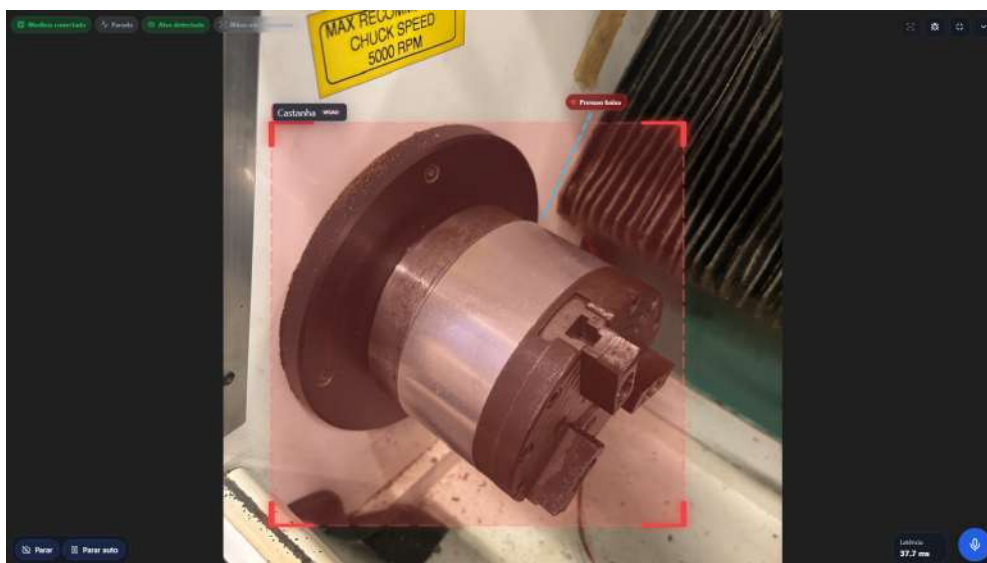
A integração industrial foi validada com o mapa de comunicação do CLP Siemens S7-1200 associado ao torno MIRAC PC. Esse mapa permitiu representar variáveis operacionais e comandos relevantes do equipamento, como estados de porta, castanha, iluminação, ferramenta selecionada e parâmetros associados ao trocador de ferramentas.

O backend organizou essas variáveis por meio de uma camada de *runtime*, abstraindo endereços Modbus em *tags* lógicas utilizadas pela aplicação. Essa estrutura confirmou a adequação do modelo intermediado, pois a interface passou a operar sobre identificadores semânticos em vez de depender diretamente de endereços físicos do controlador.

Além das variáveis de estado e comando, o mapa de comunicação também contemplou *tags* destinadas à supervisão de falhas do equipamento. No CLP Siemens S7-1200, foram representadas indicações como tempo excessivo de abertura ou fechamento da porta, baixa pressão, falhas de *drivers* ou inversores, comando inválido, falha da torre de ferramentas e condição de *trip*. No LinuxCNC, foram monitoradas condições como emergência ativa, falha reportada pelo controle e limite de eixo ativo. Essas variáveis foram tratadas como *tags* de leitura e associadas aos respectivos componentes lógicos da interface, permitindo que condições anormais fossem apresentadas tanto nos elementos aumentados quanto nas telas de diagnóstico.

Desse modo, a monitoração em tempo real implementada não se restringiu à apresentação de estados operacionais favoráveis, mas também incluiu sinais de alarme e falha disponibilizados pelos controladores. Essa supervisão, entretanto, permaneceu limitada às indicações expostas no mapa de comunicação, não constituindo um diagnóstico autônomo de causa raiz nem um sistema certificado de segurança funcional.

A Figura 21 exemplifica esse comportamento ao apresentar a falha de pressão baixa vinculada à castanha detectada na imagem, evidenciando a associação entre componente físico, *tag* industrial e informação contextual exibida ao operador.

Figura 21 – Identificação contextual de falha de pressão baixa associada à castanha

Fonte: Autoria própria, 2026

Os ensaios também confirmaram o comportamento de reconexão e degradação parcial do sistema. Quando uma unidade de comunicação não estava disponível, a aplicação preservou a operação da interface e indicou o estado de conexão ao usuário, sem interromper o restante do fluxo de análise e visualização. Esse resultado reforça a separação entre estado do backend, estado da comunicação industrial e estado visual da interface.

4.2.4 Preparação do Conjunto de Dados e Treinamento do Modelo

O reconhecimento dos componentes reais do MIRAC PC foi implementado com um modelo YOLOv8 treinado a partir de imagens do torno e de seus elementos visíveis. As amostras foram organizadas e anotadas com auxílio da plataforma Roboflow, preservando divisões distintas para treinamento, validação e teste. Essa separação permitiu utilizar parte das imagens no ajuste dos pesos do modelo e manter conjuntos independentes para acompanhamento do desempenho.

O conjunto de dados consolidado contemplou onze classes: MIRAC-PC, castanha, CLP, *drivers*, ferramentas, IHM local, iluminação, inversores, porta, presostato e transformador. A escolha das classes considerou a utilidade operacional de cada componente e a possibilidade de associar sua detecção aos dados industriais apresentados na interface.

O procedimento de treinamento foi automatizado por um *script* Python responsável por preparar o conjunto de dados no formato esperado pelo detector, converter anotações poligonais em caixas delimitadoras quando necessário e iniciar o treinamento por meio da biblioteca Ultralytics. Utilizou-se como ponto de partida o modelo pré-treinado YOLOv8n, escolhido por apresentar dimensão reduzida e custo

computacional compatível com a necessidade de inferência interativa. O treinamento foi executado com imagens redimensionadas para 640 pixels e 80 épocas como configuração de referência. Ao final, o melhor peso obtido foi copiado para o diretório de modelos carregado pelo backend.

Nos ensaios realizados, o processamento do modelo YOLO foi executado no servidor local descrito anteriormente, utilizando a GPU NVIDIA GeForce RTX 4050 por meio de CUDA. Essa informação é relevante para a interpretação dos tempos de processamento obtidos, pois a inferência do detector não foi avaliada em condição exclusivamente baseada em CPU.

A Figura 22 apresenta exemplos de imagens anotadas utilizadas nesse processo.

Figura 22 – Exemplos de imagens anotadas para treinamento do detector



Fonte: Autoria própria, 2026

4.2.5 Implementação da Análise Automática com YOLO

A análise automática foi integrada ao mesmo contrato JSON já utilizado entre frontend e backend. Com a câmera ativa, o dispositivo cliente captura periodicamente um quadro reduzido da cena, codifica a imagem e realiza uma requisição HTTPS para a rota de análise. O backend processa o quadro, consulta o estado industrial necessário à composição dos elementos aumentados e retorna uma resposta estruturada com detecções, informações de execução e descrições dos *overlays*. A renderização permanece local ao dispositivo móvel.

Ao receber um quadro, o serviço de visão carrega o modelo sob demanda e executa inferência com dimensão reduzida para o modo automático. A implementação seleciona a GPU via CUDA quando disponível, utilizando a CPU como alternativa de compatibilidade caso esse recurso não esteja acessível. No ambiente dos ensaios, a inferência YOLO foi executada na GPU do servidor. Cada caixa produzida pelo detector contém classe, nível de confiança e coordenadas normalizadas. A classe é convertida em identificador lógico de componente, permitindo que o serviço de *overlay* associe a região visível às respectivas informações industriais. Quando mais de uma caixa representa o mesmo componente, conserva-se a detecção de maior confiança.

O limiar mínimo de confiança utilizado para apresentação dos *overlays* é configurável por usuário. Assim, candidatos abaixo do valor selecionado podem permanecer disponíveis para depuração, mas não são apresentados na interface operacional. Essa distinção permite ajustar o compromisso entre sensibilidade e ocorrência de falsos positivos sem modificar o modelo treinado.

4.2.6 Fluxo Híbrido de Inferência e Rastreamento

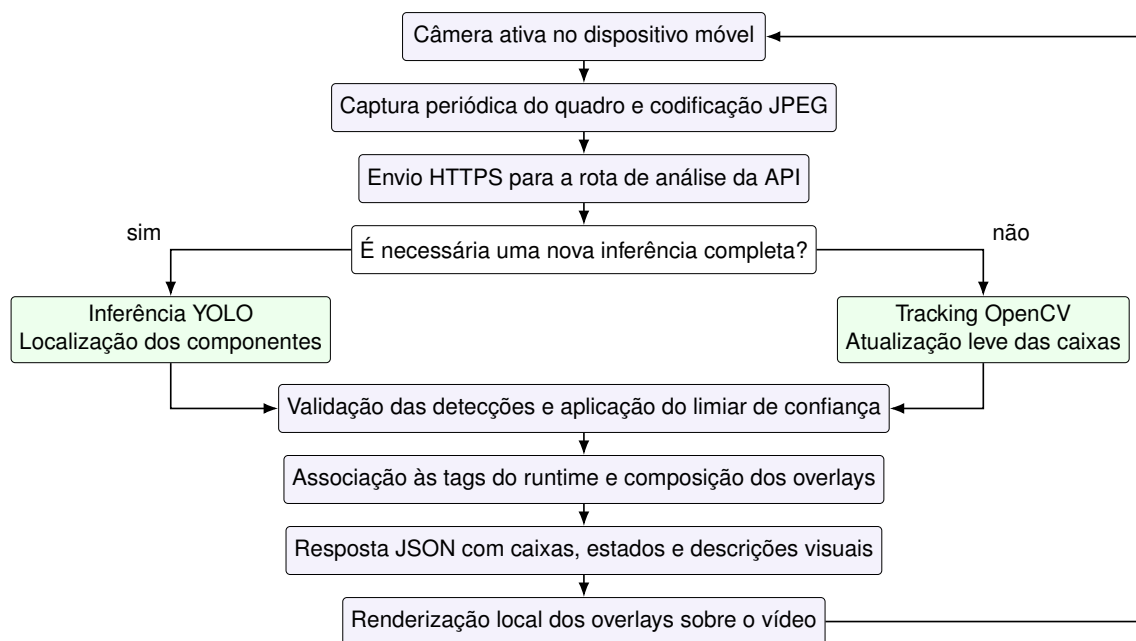
A execução de uma inferência completa em todos os quadros elevaria o custo computacional e reduziria a taxa efetiva de atualização. Para preservar responsividade, foi implementada uma estratégia híbrida que intercala inferências YOLO com rastreamento leve por comparação de *templates* em OpenCV. Entre duas inferências completas, a última região válida de cada componente é utilizada como referência em uma área de busca próxima à posição anterior.

O resultado do rastreamento somente é aceito quando atende simultaneamente aos critérios definidos pela aplicação: confiança mínima da comparação visual, textura suficiente no *template*, IoU mínima entre a caixa anterior e a caixa atual e limite de deslocamento relativo entre seus centros. A confiança retornada também sofre redução progressiva a cada quadro consecutivo atendido exclusivamente pelo rastreamento. Após quatro atualizações consecutivas, o sistema exige nova inferência YOLO, evitando prolongar associações visuais sem reconfirmação do detector.

No frontend, a última posição válida pode permanecer visível por até 450

milissegundos quando um componente desaparece temporariamente. Durante esse intervalo, o elemento é marcado como estimado e sua confiança é reduzida progressivamente. Esse mecanismo reduz cintilações causadas por perdas ocasionais sem manter informações antigas por tempo excessivo quando o enquadramento da câmera muda rapidamente.

Figura 23 – Fluxo de processamento da imagem e renderização dos overlays



Fonte: Autoria própria, 2026

A Figura 23 sintetiza o ciclo executado durante a análise automática. O backend recebe quadros pontuais para processamento, mas o vídeo permanece no dispositivo cliente. Após a identificação ou atualização das regiões visuais, a resposta JSON fornece ao frontend os dados necessários para desenhar os elementos aumentados sobre a imagem exibida ao operador.

4.2.7 Validação dos Recursos de Visão Computacional

Os recursos de visão computacional foram integrados de forma progressiva ao pipeline de análise de quadros. Inicialmente, técnicas clássicas permitiram validar captura, envio de imagens, processamento no backend e retorno de elementos visuais ao frontend. Após a consolidação desse ciclo, o detector YOLO passou a identificar componentes reais do MIRAC PC e a fornecer as regiões utilizadas para contextualização dos *overlays*.

A solução registrou automaticamente informações úteis à avaliação do comportamento visual: quantidade de quadros processados por inferência completa, rastreamento ou reutilização temporária; confiança das detecções; deslocamento normalizado entre centros de caixas; IoU entre quadros consecutivos; continuidade visual;

tempo de processamento e latência percebida entre captura e apresentação. Esses dados foram organizados em sessões históricas de diagnóstico, permitindo comparar cenários de teste sem depender apenas da observação subjetiva do operador.

4.2.8 Instrumentação Automática dos Ensaio Quantitativos

A coleta das métricas quantitativas foi incorporada ao fluxo operacional da aplicação para reduzir a dependência de medições manuais e preservar a correspondência entre os indicadores e o comportamento efetivamente observado pelo usuário. A instrumentação foi distribuída entre frontend e backend, seguindo as responsabilidades já definidas pela arquitetura do sistema.

No frontend, o ciclo de análise registrou o intervalo entre o início da captura do quadro no navegador e o primeiro ciclo de renderização executado após a atualização dos *overlays*. Essa medição representa a latência percebida pelo operador, pois abrange captura, codificação da imagem, comunicação HTTPS, processamento no servidor, recebimento da resposta JSON e aplicação visual do resultado. As amostras foram encaminhadas ao backend juntamente com o quadro subsequente, evitando a criação de requisições adicionais exclusivas para telemetria.

No backend, a rota de análise contabilizou o tempo total de atendimento da requisição, o tempo consumido pelo serviço de visão computacional e o tempo de composição lógica dos *overlays*. Para cada quadro processado, também foram registrados o perfil utilizado pelo fluxo híbrido, a confiança das detecções e a quantidade de componentes localizados. A estabilidade temporal foi mensurada pela comparação entre caixas consecutivas do mesmo componente: calculou-se a IoU entre regiões sucessivas e a distância euclidiana entre seus centros em coordenadas normalizadas pelas dimensões do quadro.

A integração industrial foi instrumentada separadamente. Cada ciclo de *polling* registrou sucesso ou falha, duração e quantidade de erros de leitura. As tentativas de comando registraram resultado e categoria de falha, permitindo consolidar a taxa de sucesso das escritas realizadas durante os ensaios. Antes de uma escrita Modbus, a aplicação verificou a alcançabilidade do dispositivo por meio de ping, permitindo distinguir indisponibilidade de rede de falhas específicas de conexão ou escrita no protocolo industrial.

As amostras numéricas da sessão corrente foram mantidas em memória em uma janela limitada às últimas 2000 observações. A página de diagnóstico permitiu iniciar uma sessão experimental identificada por cenário, dispositivo e observações complementares. Ao término do ensaio, o backend persistiu no banco SQLite um resumo agregado associado ao usuário autenticado, incluindo contagens, médias, percentis 95 e valores máximos aplicáveis. Os registros também puderam ser exportados

em JSON ou CSV para consolidação dos resultados apresentados neste capítulo.

4.2.9 Resultados da Validação Offline do Detector

A validação offline do melhor peso obtido no treinamento foi executada sobre a partição de validação do conjunto de dados, composta por 57 imagens e 93 instâncias anotadas. A Tabela 1 apresenta os indicadores globais obtidos pelo modelo. O F1-score foi calculado a partir da precisão e da revocação globais.

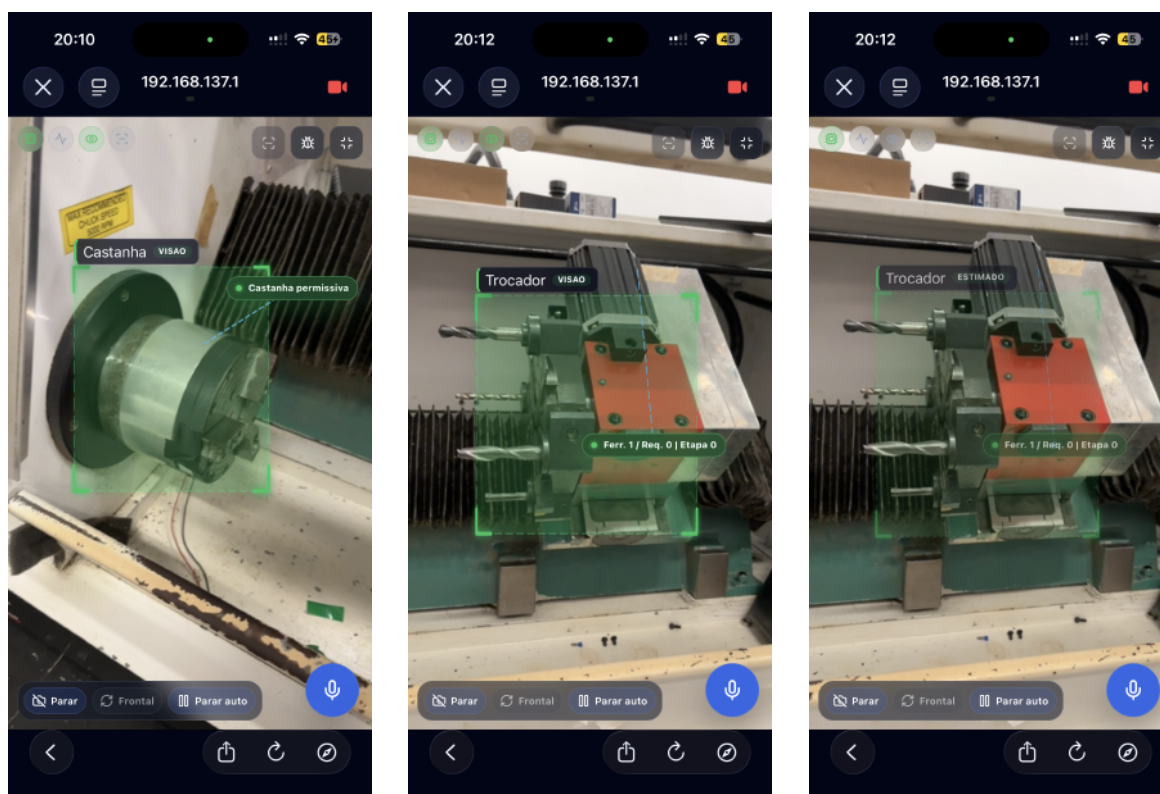
Tabela 1 – Resultados globais da validação offline do detector YOLO

Indicador	Resultado
Imagens de validação	57
Instâncias anotadas	93
Precisão	0,917
Revocação	0,907
F1-score	0,912
mAP50	0,955
mAP50-95	0,748

Fonte: Autoria própria, 2026

Os resultados indicam desempenho global consistente no conjunto utilizado, com mAP50 de 0,955. A diferença entre mAP50 e mAP50-95 evidencia que a localização das caixas se torna mais sensível quando são exigidos critérios progressivamente mais rigorosos de sobreposição. Essa observação é relevante para a interface aumentada, pois a qualidade visual percebida depende não apenas da identificação correta do componente, mas também da estabilidade espacial da região utilizada para posicionar o *overlay*.

Essa relação entre detecção, localização e apresentação contextual é exemplificada na Figura 24. Além de indicar o componente reconhecido, a interface também diferencia a origem da identificação: quando a região é confirmada diretamente pela inferência visual, o *overlay* é apresentado como detecção por visão; quando a região é mantida temporariamente a partir do rastreamento e da última posição válida, o elemento é indicado como estimado. Essa distinção torna visível ao operador o motivo da identificação e evita que uma associação temporária seja interpretada como nova detecção plena do modelo.

Figura 24 – Detecção visual de componentes e sobreposição aumentada**(a) Castanha: visão****(b) Ferramenta: visão****(c) Ferramenta: estimada**

Fonte: Autoria própria, 2026

A apresentação explícita da origem da caixa delimitadora reforça a lógica híbrida adotada no sistema. A identificação por visão representa uma confirmação recente do detector YOLO, enquanto a identificação estimada representa uma permanência controlada baseada no rastreamento temporal. Com isso, a interface reduz cintilações causadas por perdas momentâneas sem ocultar do operador que aquela posição depende de uma estimativa transitória.

4.2.10 Resultados dos Ensaios Operacionais

Os ensaios operacionais complementaram a validação offline do detector ao observar o comportamento da aplicação durante o uso contínuo da interface aumentada. A análise considerou o fluxo completo entre captura do quadro, processamento no backend, atualização dos *overlays* e consulta dos dados industriais. Também foram avaliadas a continuidade visual dos componentes rastreados e a confiabilidade das operações Modbus. A Tabela 2 consolida os indicadores obtidos na sessão experimental completa registrada como “Ensaio 2”, executada em iPhone 15, com limiar de confiança de 0,6, rastreamento habilitado e intervalo de detecção configurado em 200 ms. Esses valores devem ser interpretados como evidências descritivas da viabilidade técnica no cenário ensaiado, e não como generalização estatística para diferentes

operadores, dispositivos ou ambientes industriais.

Tabela 2 – Resultados consolidados dos ensaios operacionais

Indicador	Média ou taxa	P95 ou máximo
Quadros analisados	656	–
Latência percebida entre captura e apresentação do overlay	114,37 ms	259,00 ms
Tempo de processamento visual no backend	41,41 ms	32,25 ms
Continuidade visual dos componentes	60,67%	–
IoU entre caixas consecutivas	0,92	–
Deslocamento normalizado entre centros	2,00%	–
Ciclos de polling Modbus analisados	120	–
Taxa de sucesso do polling Modbus	100,00%	–
Comandos enviados durante os ensaios	9	–
Taxa de sucesso dos comandos	100,00%	–

Fonte: Autoria própria, 2026

Os registros indicaram latência compatível com a interação em tempo real proposta para a aplicação. A latência percebida entre captura do quadro e apresentação do *overlay* apresentou média de 114,37 ms e percentil 95 de 259,00 ms. O processamento visual executado no backend registrou média de 41,41 ms e percentil 95 de 32,25 ms, mantendo resposta adequada durante a análise automática. A diferença entre média e percentil 95 nesse indicador decorre da presença de um evento isolado de maior duração registrado durante a sessão, que elevou a média sem representar o comportamento predominante do processamento visual. Por esse motivo, os indicadores de tempo foram utilizados como caracterização operacional do protótipo no ensaio registrado, não como limite absoluto de desempenho do sistema.

A estabilidade visual mostrou-se satisfatória durante os ensaios, com continuidade dos componentes rastreados de 60,67%, IoU médio de 0,92 entre caixas consecutivas e deslocamento normalizado médio de 2,00% entre seus centros. Esses valores indicam que o rastreamento bidimensional leve e a suavização temporal reduziram oscilações perceptíveis sem comprometer a atualização dos elementos aumentados.

Na integração industrial, foram analisados 120 ciclos de *polling* Modbus e 9 comandos enviados pela interface. As taxas de sucesso obtidas foram de 100,00% para as leituras periódicas e 100,00% para os comandos, indicando confiabilidade adequada para o monitoramento e para a interação controlada com o equipamento durante o ensaio registrado.

4.2.11 Validação dos Modos de Interação

Além da detecção visual e da comunicação industrial, foi verificado se os diferentes modos de interação conduziam o operador ao mesmo fluxo controlado de

solicitação de comandos. Essa validação teve como foco confirmar que menus flutuantes, gestos configuráveis e comandos de voz não criavam caminhos paralelos de escrita industrial, mas apenas formas distintas de selecionar uma ação antes da etapa de confirmação e validação no backend.

Nos menus flutuantes, a seleção do componente manteve a relação espacial entre objeto físico, informação aumentada e comando disponível. Nos gestos, a permanência configurada da pose funcionou como critério adicional de intenção antes do acionamento. Na interação por voz, a frase reconhecida foi comparada ao catálogo fechado de comandos e encaminhada à confirmação operacional já existente.

Assim, a validação dos modos de interação teve caráter funcional e exploratório. Seu objetivo foi verificar a coerência operacional dos mecanismos implementados e sua integração com a mesma política de escrita controlada, sem pretender medir desempenho ergonômico, carga cognitiva ou aceitação por uma amostra representativa de operadores.

4.2.12 Validação da Interface em Dispositivo Móvel

A interface também foi validada no dispositivo que representa o principal cenário de uso da solução. O acesso à PWA em rede local permitiu verificar autorização da câmera pelo navegador, exibição do vídeo em tempo real, renderização dos *overlays* sobre a imagem e navegação entre as telas essenciais da aplicação.

Os ajustes de layout realizados ao longo do desenvolvimento buscaram adequar a interface ao espaço reduzido da tela do celular, mantendo visíveis os controles necessários à operação, à inspeção do estado do sistema e ao acesso aos parâmetros da aplicação. Essa etapa complementou os ensaios funcionais ao verificar se a arquitetura implementada permanecia utilizável no suporte físico previsto para o operador.

Essa verificação permaneceu limitada ao comportamento funcional e visual do protótipo em dispositivo móvel. Avaliações formais de usabilidade, comparação entre diferentes modelos de aparelhos e medições ergonômicas com múltiplos operadores permaneceram fora do escopo delimitado para o trabalho.

4.2.13 Síntese dos Resultados Funcionais

A validação funcional demonstrou que a solução consolidou um fluxo operacional coerente entre dispositivo móvel, backend local, comunicação industrial e renderização aumentada. O sistema preservou o vídeo no cliente, utilizou a API como contrato entre camadas, representou os dados industriais por meio de *tags* e apresentou informações contextuais sobre componentes do MIRAC PC.

Os resultados obtidos nesta etapa confirmam a viabilidade da arquitetura distribuída adotada para uma IHM aumentada em ambiente industrial didático. A estrutura implementada permitiu que dados de automação, análise visual e interação do operador por menus, gestos e voz fossem integrados sem acoplamento direto entre interface gráfica e endereços físicos dos controladores.

Os ensaios operacionais reforçaram essa conclusão ao registrar latência percebida média de 114,37 ms, continuidade visual de 60,67%, taxa de sucesso do *polling* Modbus de 100,00% e taxa de sucesso dos comandos de 100,00%. Em conjunto, esses indicadores demonstraram comportamento adequado para o monitoramento contextual e para a interação controlada com o equipamento.

5 CONSIDERAÇÕES FINAIS

5.1 Síntese do Trabalho

Este trabalho apresentou o desenvolvimento de uma interface homem-máquina baseada em Realidade Aumentada para monitoramento e ajuste de parâmetros em uma máquina industrial didática. A solução integrou captura de vídeo em dispositivo móvel, processamento em backend local, comunicação industrial via Modbus TCP e renderização de *overlays* diretamente no frontend.

A arquitetura adotada manteve separadas as responsabilidades entre cliente, processamento e integração industrial. O dispositivo móvel permaneceu responsável pela câmera e pela apresentação visual, enquanto o backend concentrou a análise dos quadros, a organização dos dados operacionais e a comunicação com os controladores associados ao MIRAC-PC. Essa separação permitiu preservar o vídeo no cliente, reduzir o tráfego de dados e manter um contrato estruturado entre as camadas por meio de respostas JSON.

No contexto industrial, a aplicação foi estruturada para representar componentes reais do torno MIRAC-PC, associando estados de processo a elementos visuais contextualizados. A integração considerou o computador LinuxCNC e o CLP Siemens S7-1200 associado ao equipamento, consolidando o recorte operacional adotado para validação.

Os resultados funcionais demonstraram a viabilidade da proposta como uma IHM aumentada distribuída, capaz de combinar dados industriais, análise visual e interação do operador em uma única interface. A interação foi organizada por menus flutuantes associados aos componentes, comandos por gestos configuráveis e comandos por voz com reconhecimento offline. A solução também estabeleceu uma estratégia de identificação de componentes baseada em YOLO e rastreamento bidimensional leve, mantendo coerência com a abordagem *markerless* adotada no trabalho no sentido de dispensar marcadores artificiais fixados ao equipamento.

A validação offline do detector registrou precisão de 0,917, revocação de 0,907, F1-score de 0,912, mAP50 de 0,955 e mAP50-95 de 0,748 no conjunto de validação. Esses resultados sustentam a utilização do modelo como mecanismo de identificação visual dos componentes analisados e também evidenciam a importância do rastreamento leve e da suavização temporal para preservar a estabilidade percebida dos elementos aumentados.

Nos ensaios operacionais, a aplicação registrou latência percebida média de 114,37 ms entre captura e apresentação do *overlay*, continuidade visual de 60,67%, IoU médio de 0,92 entre caixas consecutivas e deslocamento normalizado médio de 2,00%. A integração industrial apresentou 100,00% de sucesso em 120 ciclos de *pol-*

ling Modbus e 100,00% de sucesso em 9 comandos enviados pela interface durante a sessão completa analisada.

Em comparação ao trabalho de Rosa (2019), que demonstrou a integração entre Realidade Aumentada, protocolos industriais e visualização contextual por meio de marcadores visuais, a solução desenvolvida ampliou a separação entre captura, processamento, integração industrial e renderização local da interface. A adoção de PWA, API local, respostas JSON, camada intermediária de *runtime* e organização por *tags* lógicas consolidou uma arquitetura mais desacoplada, adequada à expansão para múltiplas unidades industriais e ao avanço gradual em direção ao reconhecimento de componentes sem marcadores fixos. Dessa forma, a principal contribuição do trabalho é arquitetural e integradora: demonstrar um fluxo funcional entre percepção visual, dados industriais e comandos controlados em uma interface aumentada acessível por navegador.

5.2 Limitações do Estudo

As limitações observadas estão relacionadas principalmente à escala dos ensaios, às condições físicas do equipamento industrial e à disponibilidade dos mapas de comunicação. A validação funcional demonstrou a operação integrada das camadas principais do sistema, enquanto a avaliação quantitativa permanece sensível ao número de amostras, à diversidade de cenas e à repetição dos ensaios em condições controladas. Assim, os resultados obtidos devem ser interpretados como evidências de viabilidade técnica no recorte experimental adotado, e não como comprovação de desempenho generalizável para diferentes plantas industriais.

Outra limitação refere-se à identificação visual de componentes. A robustez da detecção baseada em aprendizado profundo depende da qualidade, diversidade e volume do conjunto de dados utilizado para treinamento. Em aplicações industriais, variações de iluminação, reflexos, oclusões e mudanças de perspectiva continuam sendo fatores críticos para a estabilidade da detecção e para a continuidade temporal dos elementos apresentados. Além disso, o posicionamento dos *overlays* foi realizado a partir de referências bidimensionais na imagem, não correspondendo a uma estimativa tridimensional completa da pose dos componentes.

Também se destaca que a integração industrial foi delimitada ao MIRAC-PC, considerando o mapa do Siemens S7-1200 associado ao torno e a estrutura de comunicação ampliada no computador LinuxCNC. A disponibilização dos dados exigiu adaptações nas instâncias Modbus dos sistemas existentes, preservando os fluxos originais de comunicação do equipamento. Embora a arquitetura permita múltiplas unidades Modbus, a expansão para demais controladores depende da disponibilidade de mapas de comunicação confiáveis e da validação das permissões de leitura e escrita

em cada equipamento.

Por fim, a avaliação da interação teve caráter exploratório. Embora os fluxos por menus, gestos e voz tenham sido verificados funcionalmente, não foi realizado estudo formal com múltiplos operadores para mensurar usabilidade, carga cognitiva, tempo de execução de tarefas ou aceitação da interface em condições produtivas reais.

5.3 Possibilidades de Continuidade

Como continuidade deste trabalho, destaca-se inicialmente a ampliação da validação experimental da solução proposta. Novos ensaios quantitativos poderiam ser realizados para avaliar latência sistêmica, estabilidade temporal dos *overlays*, continuidade visual da interface, confiabilidade da comunicação Modbus e desempenho do modelo de detecção. A repetição desses testes em diferentes condições de iluminação, distância, enquadramento e movimentação da câmera permitiria consolidar a análise da solução em cenários mais próximos da operação real.

Outro eixo de continuidade envolve o aprimoramento do reconhecimento dos componentes reais do MIRAC-PC. Para isso, o conjunto de imagens poderia ser ampliado com capturas realizadas em diferentes condições de iluminação, distância, zoom, enquadramento e dispositivos de aquisição. A revisão das classes do *dataset*, o treinamento de modelos específicos de detecção ou segmentação e o refinamento dos critérios de rastreamento e suavização também poderiam aumentar a robustez do reconhecimento e reduzir perdas momentâneas de referência durante a movimentação da câmera.

Também se destaca a possibilidade de expandir a arquitetura para conjuntos industriais mais amplos, por meio da criação de catálogos de tags específicos para diferentes tipos de máquinas. Esses catálogos poderiam associar componentes físicos, variáveis de processo, estados operacionais, alarmes, permissões e comandos a estruturas padronizadas de informação. Com isso, a interface de realidade aumentada poderia carregar dinamicamente os dados correspondentes ao equipamento identificado, mantendo a separação entre aplicação, tags lógicas e camada de comunicação.

Nesse contexto, poderiam ser implementados mecanismos de identificação automática da máquina ou unidade em operação, utilizando estratégias como triangulação, reconhecimento de características físicas, marcadores visuais, leitura de códigos associados ao equipamento ou combinação de múltiplas fontes de informação. Essa identificação permitiria distinguir equipamentos semelhantes em uma mesma planta e selecionar automaticamente o conjunto correto de tags, telas, procedimentos e informações operacionais.

A interface também poderia evoluir para contemplar recursos de interação

operacional assistida, como confirmação de comandos, indicação de permissões, exibição contextual de procedimentos, apoio à manutenção passo a passo e integração com registros de eventos. Essa evolução exigiria atenção aos critérios de segurança, rastreabilidade e prevenção de acionamentos indevidos em aplicações industriais reais.

Por fim, a avaliação com operadores permitiria analisar clareza visual, carga cognitiva, tempo de resposta percebido, facilidade de interpretação e segurança da interação. Essa dimensão complementaria as métricas técnicas e ajudaria a verificar a contribuição da interface para a compreensão do processo e o suporte à operação e manutenção.

Assim, este trabalho cumpre seu papel exploratório ao demonstrar que a integração entre realidade aumentada, visão computacional e comunicação industrial pode ser estruturada em uma arquitetura flexível, adaptável e extensível. A solução desenvolvida estabelece uma base técnica capaz de ser expandida para diferentes máquinas, protocolos, conjuntos de tags e contextos operacionais, abrindo possibilidades para interfaces industriais mais inteligentes, contextualizadas e alinhadas aos princípios da Indústria 4.0.

REFERÊNCIAS

- AZUMA, R. T. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, v. 6, n. 4, p. 355–385, 1997. 24, 26, 27, 28, 29, 31, 39, 40, 41
- BILLINGHURST, M.; DUENSER, A. Augmented reality in the classroom. *Computer*, v. 45, n. 7, p. 56–63, 2012. Disponível em: <https://ieeexplore.ieee.org/document/6171143>. 25
- BRAY, T. *The JavaScript Object Notation (JSON) Data Interchange Format*. 2017. RFC 8259. Disponível em: <https://www.rfc-editor.org/rfc/rfc8259>. 36
- BUTTAZZO, G. C. *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*. 3. ed. New York: Springer, 2011. 19, 39, 40, 41
- FIELDING, R. T. *Architectural Styles and the Design of Network-based Software Architectures*. Tese (Doutorado) — University of California, Irvine, 2000. Disponível em: <https://roy.gbiv.com/pubs/dissertation/top.htm>. 35, 40
- HENDERSON, S.; FEINER, S. Exploring the benefits of augmented reality documentation for maintenance and repair. *IEEE Transactions on Visualization and Computer Graphics*, v. 17, n. 10, p. 1355–1368, 2011. Disponível em: <https://ieeexplore.ieee.org/document/5620905>. 19, 25, 26, 28, 31, 38
- KAGERMANN, H.; WAHLSTER, W.; HELBIG, J. *Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0: Final Report of the Industrie 4.0 Working Group*. Munich, 2013. 21
- LECUN, Y. *et al.* Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, 1998. Disponível em: http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf. 29
- Marcio Mello. *IHM: o que é e qual sua importância na automação industrial*. 2023. Disponível em: <https://victorvision.com.br/blog/ihm/>. Acesso em: 2026-03-01. 37
- MILGRAM, P.; KISHINO, F. A taxonomy of mixed reality visual displays. *IEICE Transactions on Information Systems*, E77-D, n. 12, p. 1321–1329, 1994. 24
- MODICON. *Modbus Application Protocol Specification*. 1999. Especificação oficial. Disponível em: <https://www.modbus.org/file/secure/modbusprotocollspecification.pdf>. 32, 35
- ORGANIZATION, M. *Modbus Messaging on TCP/IP Implementation Guide V1.0b*. 2006. Guia de implementação Modbus TCP/IP. 33
- PALMARINI, R. *et al.* A systematic review of augmented reality applications in maintenance. *Robotics and Computer-Integrated Manufacturing*, v. 49, p. 215–228, 2018. Disponível em: <https://doi.org/10.1016/j.rcim.2017.06.002>. 17, 19
- PLATFORM, V. *YOLOv8: Detecção de Objetos de Última Geração em Visão Computacional*. 2024. Disponível em: <https://visionplatform.ai/pt/yolov8-deteccao-de-objetos-de-ultima-geracao-em-visao-computacional-computer-vision/>. Acesso em: 2025-12-04. 30

- PORTER, M. E.; HEPPELMANN, J. E. How smart, connected products are transforming competition. *Harvard Business Review*, v. 92, n. 11, p. 64–88, 2014. 21
- REDMON, J. *et al.* You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [s.n.], 2016. p. 779–788. Disponível em: <https://arxiv.org/pdf/1506.02640>. 24, 28, 30, 31
- ROSA, E. N. A. d. Trabalho de Conclusão de Curso, *Aplicação de Realidade Aumentada Integrada à Internet das Coisas Industrial como Ferramenta de Suporte à Manutenção*. Chapecó: [s.n.], 2019. 27, 41, 42, 46, 79
- SCHMALSTIEG, D.; HÖLLERER, T. *Augmented Reality: Principles and Practice*. Boston: Addison-Wesley, 2016. 24, 26, 27, 28, 29, 31, 40, 41
- SCHWAB, K. *The Fourth Industrial Revolution*. Geneva: World Economic Forum, 2016. 21
- STOUFFER, K. *et al.* *Guide to Industrial Control Systems (ICS) Security*. [S.l.], 2015. Disponível em: <https://doi.org/10.6028/NIST.SP.800-82r2>. 33, 34, 35, 41
- SWELLER, J. Cognitive load during problem solving: Effects on learning. *Cognitive Science*, v. 12, n. 2, p. 257–285, 1988. 38
- TORI, R.; HOUNSELL, M. d. S. (Ed.). *Introdução a Realidade Virtual e Aumentada*. Porto Alegre: Editora SBC, 2018. Organizadores. 25
- WANG, X. *et al.* Augmented reality in built environment: Classification and implications for future research. *Automation in Construction*, v. 32, p. 1–13, 2013. 17, 19, 25
- ZHU, E. *et al.* Augmented reality in healthcare education: An integrative review. *PeerJ*, v. 2, p. e469, 2014. Disponível em: <https://peerj.com/articles/469/>. 25

APÊNDICES

APÊNDICE A – CADERNO DE TELAS DA APLICAÇÃO EM DESKTOP

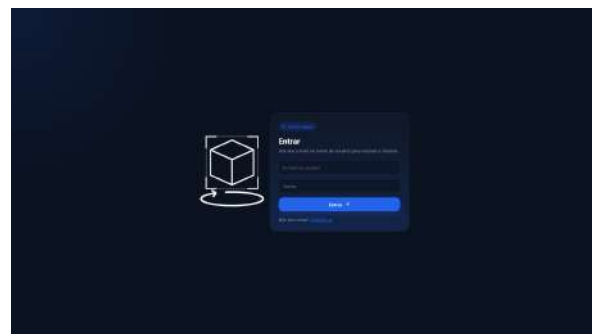
Este apêndice reúne as principais telas da aplicação em visualização desktop. As capturas foram obtidas de forma automatizada com Playwright, utilizando o perfil desktop configurado para o caderno de telas do sistema. O registro evidencia a organização visual da interface, os recursos de operação, os módulos de configuração e as telas de apoio ao uso da IHM aumentada.

A.1 Telas em Visualização Desktop

Figura 25 – Tela de login em visualização desktop nos temas claro e escuro



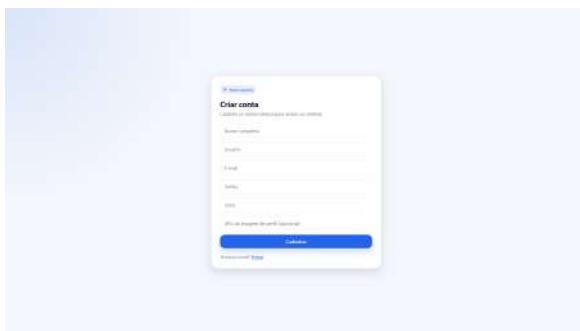
(a) Tema claro



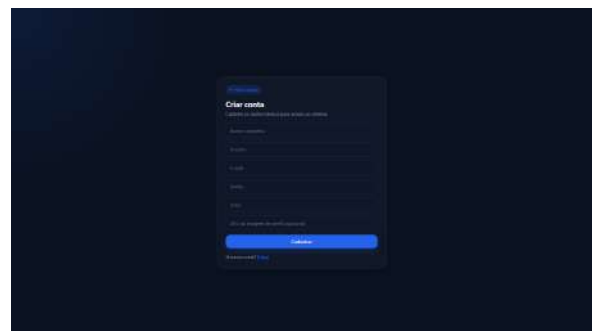
(b) Tema escuro

Fonte: Autoria própria, 2026

Figura 26 – Tela de cadastro em visualização desktop nos temas claro e escuro



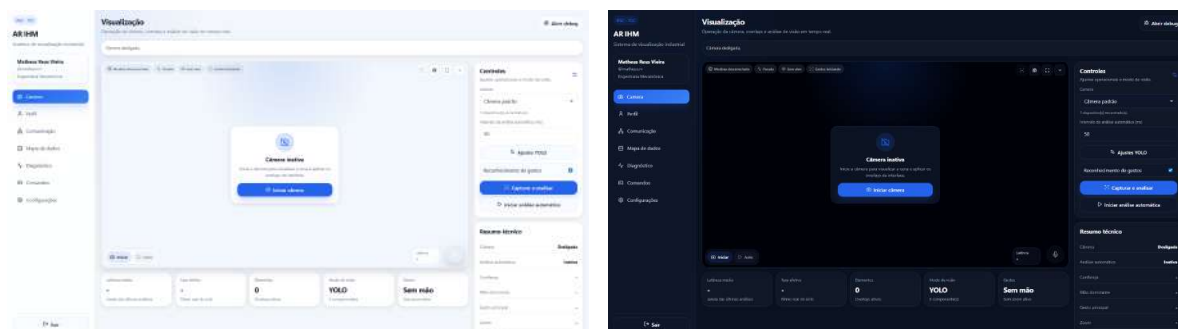
(a) Tema claro



(b) Tema escuro

Fonte: Autoria própria, 2026

Figura 27 – Tela de operação da câmera e overlays em visualização desktop nos temas claro e escuro

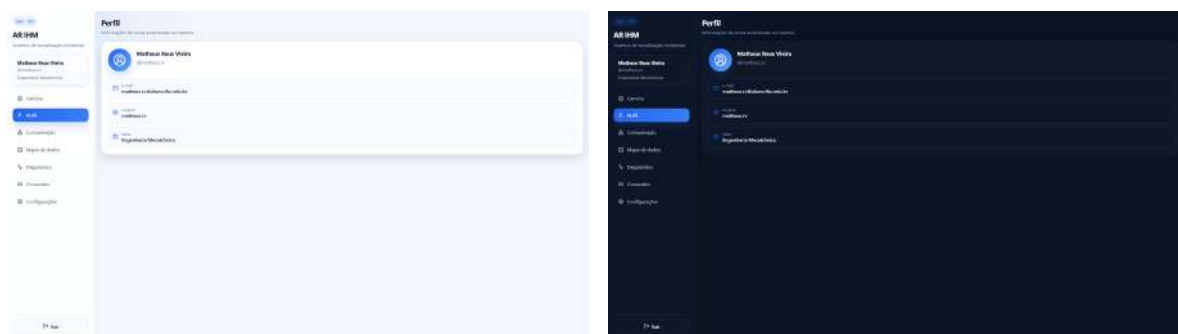


(a) Tema claro

(b) Tema escuro

Fonte: Autoria própria, 2026

Figura 28 – Tela de perfil do usuário em visualização desktop nos temas claro e escuro

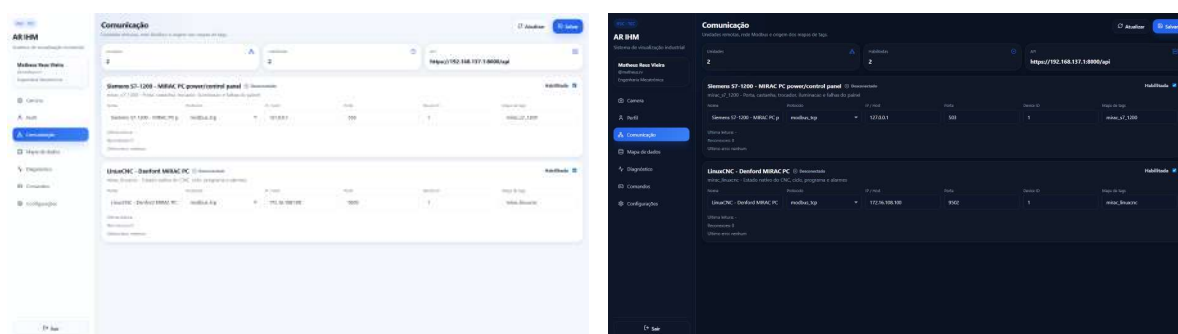


(a) Tema claro

(b) Tema escuro

Fonte: Autoria própria, 2026

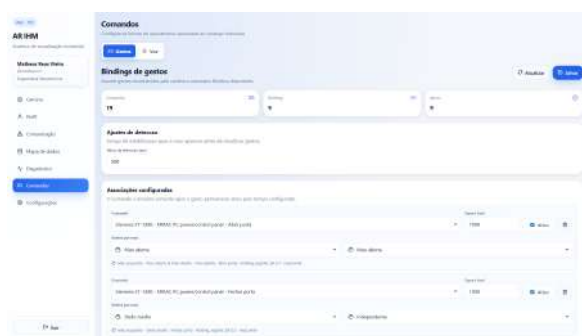
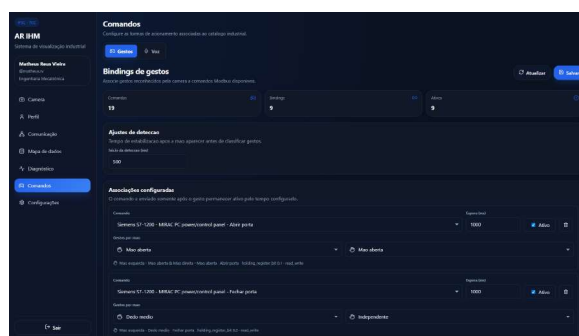
Figura 29 – Tela de configuração de comunicação em visualização desktop nos temas claro e escuro



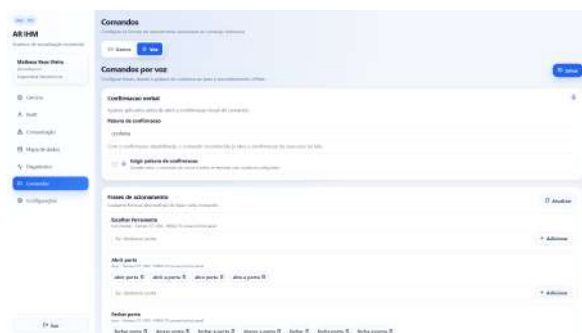
(a) Tema claro

(b) Tema escuro

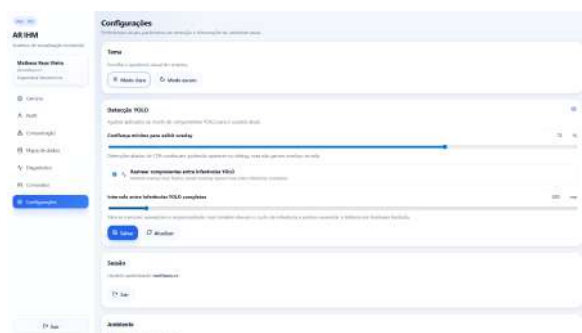
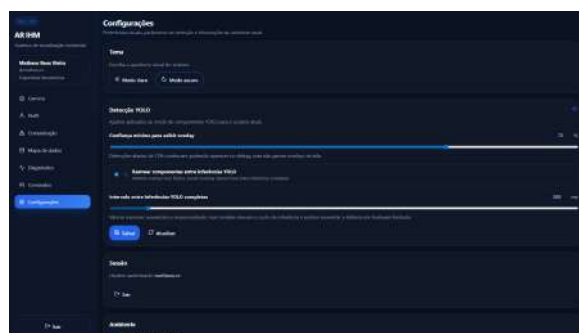
Fonte: Autoria própria, 2026

Figura 30 – Tela de comandos por gestos em visualização desktop nos temas claro e escuro**(a) Tema claro****(b) Tema escuro**

Fonte: Autoria própria, 2026

Figura 31 – Tela de comandos por voz em visualização desktop nos temas claro e escuro**(a) Tema claro****(b) Tema escuro**

Fonte: Autoria própria, 2026

Figura 32 – Tela de configurações gerais em visualização desktop nos temas claro e escuro**(a) Tema claro****(b) Tema escuro**

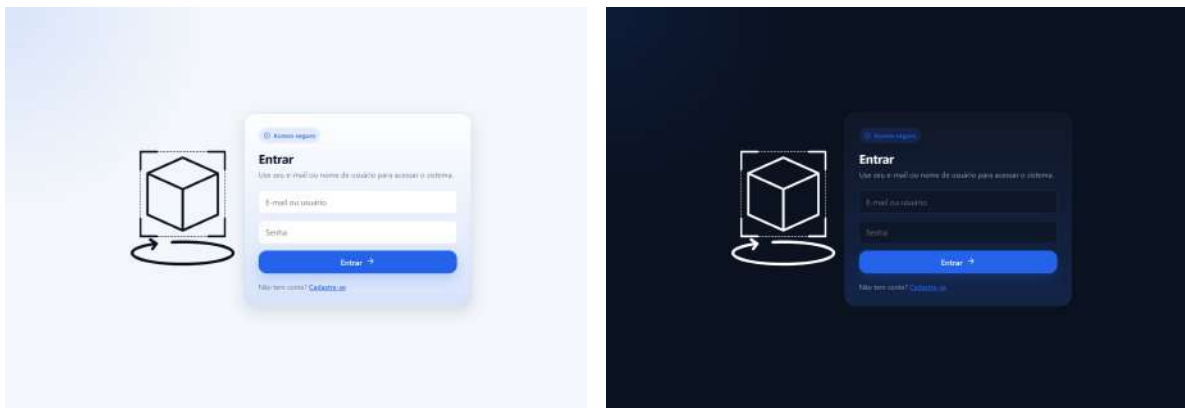
Fonte: Autoria própria, 2026

APÊNDICE B – CADERNO DE TELAS DA APLICAÇÃO EM TABLET

Este apêndice apresenta as principais telas da aplicação em visualização tablet. As capturas utilizam perfil equivalente a um iPad Air 11 polegadas em orientação paisagem, permitindo observar a adaptação da interface para telas intermediárias entre o desktop e o *smartphone*.

B.1 Telas em Visualização Tablet

Figura 33 – Tela de login em visualização tablet nos temas claro e escuro

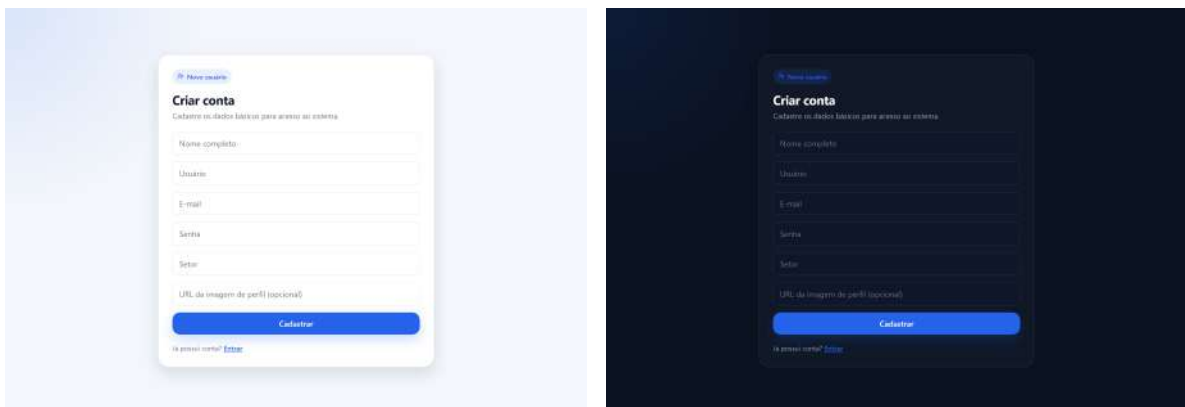


(a) Tema claro

(b) Tema escuro

Fonte: Autoria própria, 2026

Figura 34 – Tela de cadastro em visualização tablet nos temas claro e escuro

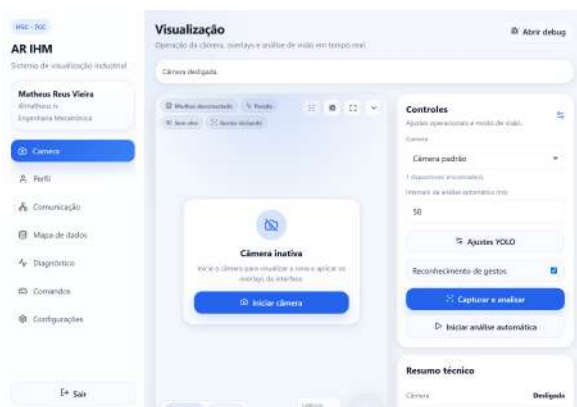


(a) Tema claro

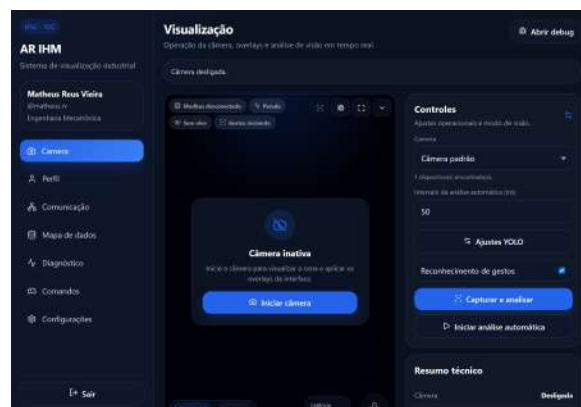
(b) Tema escuro

Fonte: Autoria própria, 2026

Figura 35 – Tela de operação da câmera e overlays em visualização tablet nos temas claro e escuro



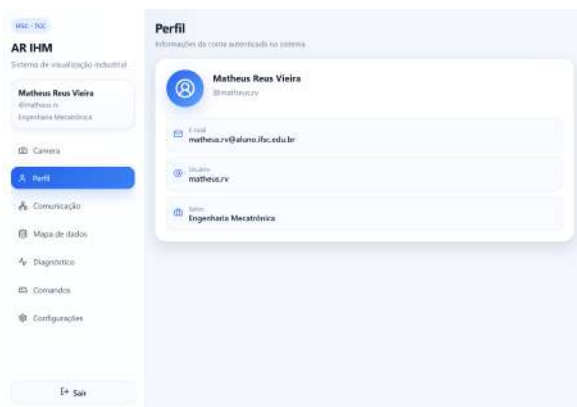
(a) Tema claro



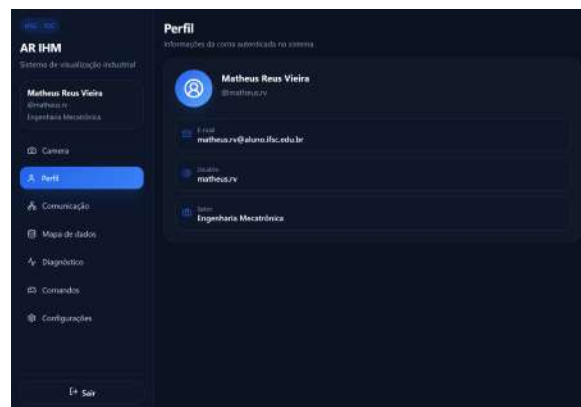
(b) Tema escuro

Fonte: Autoria própria, 2026

Figura 36 – Tela de perfil do usuário em visualização tablet nos temas claro e escuro



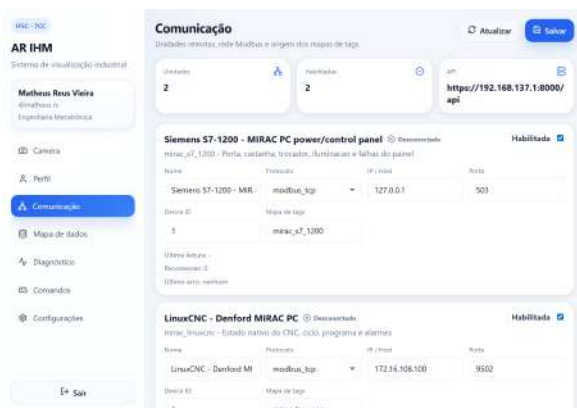
(a) Tema claro



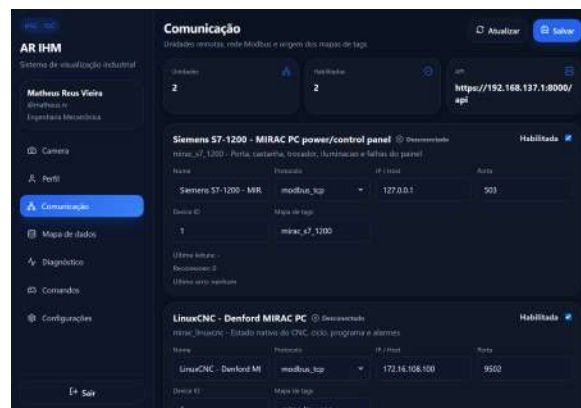
(b) Tema escuro

Fonte: Autoria própria, 2026

Figura 37 – Tela de configuração de comunicação em visualização tablet nos temas claro e escuro

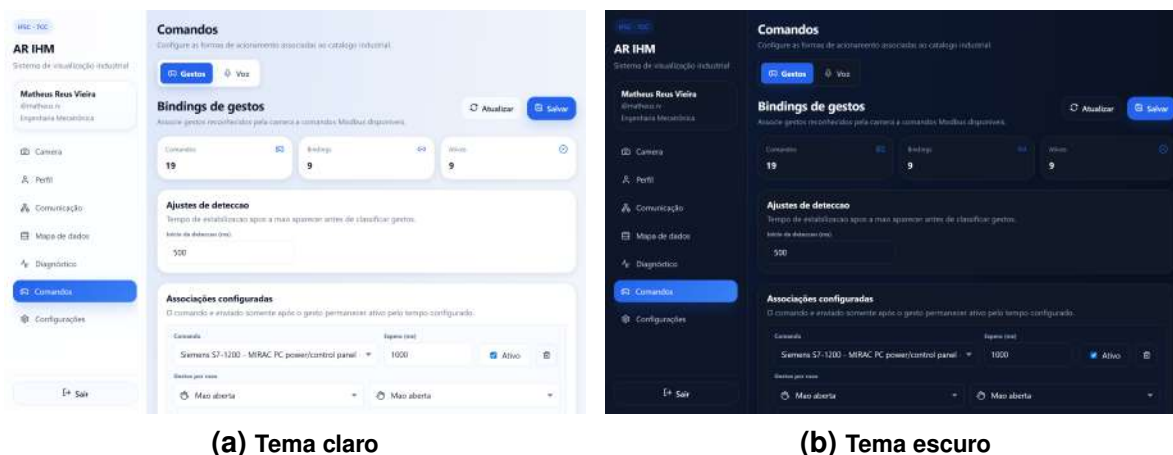


(a) Tema claro

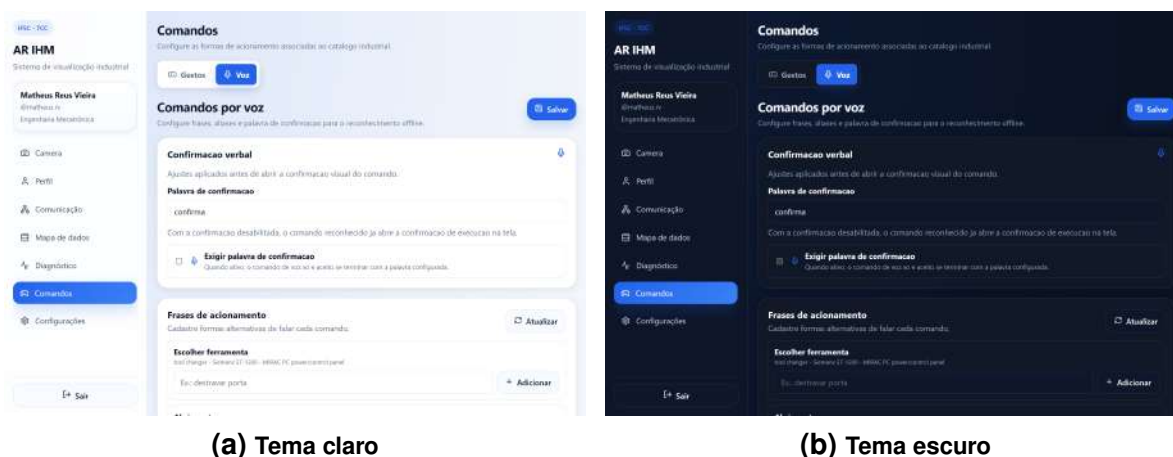


(b) Tema escuro

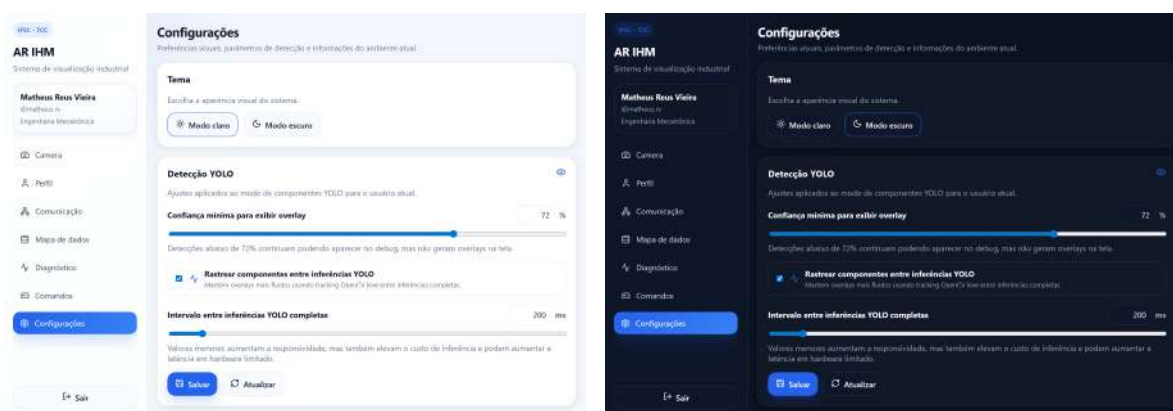
Fonte: Autoria própria, 2026

Figura 38 – Tela de comandos por gestos em visualização tablet nos temas claro e escuro**(a) Tema claro****(b) Tema escuro**

Fonte: Autoria própria, 2026

Figura 39 – Tela de comandos por voz em visualização tablet nos temas claro e escuro**(a) Tema claro****(b) Tema escuro**

Fonte: Autoria própria, 2026

Figura 40 – Tela de configurações gerais em visualização tablet nos temas claro e escuro**(a) Tema claro****(b) Tema escuro**

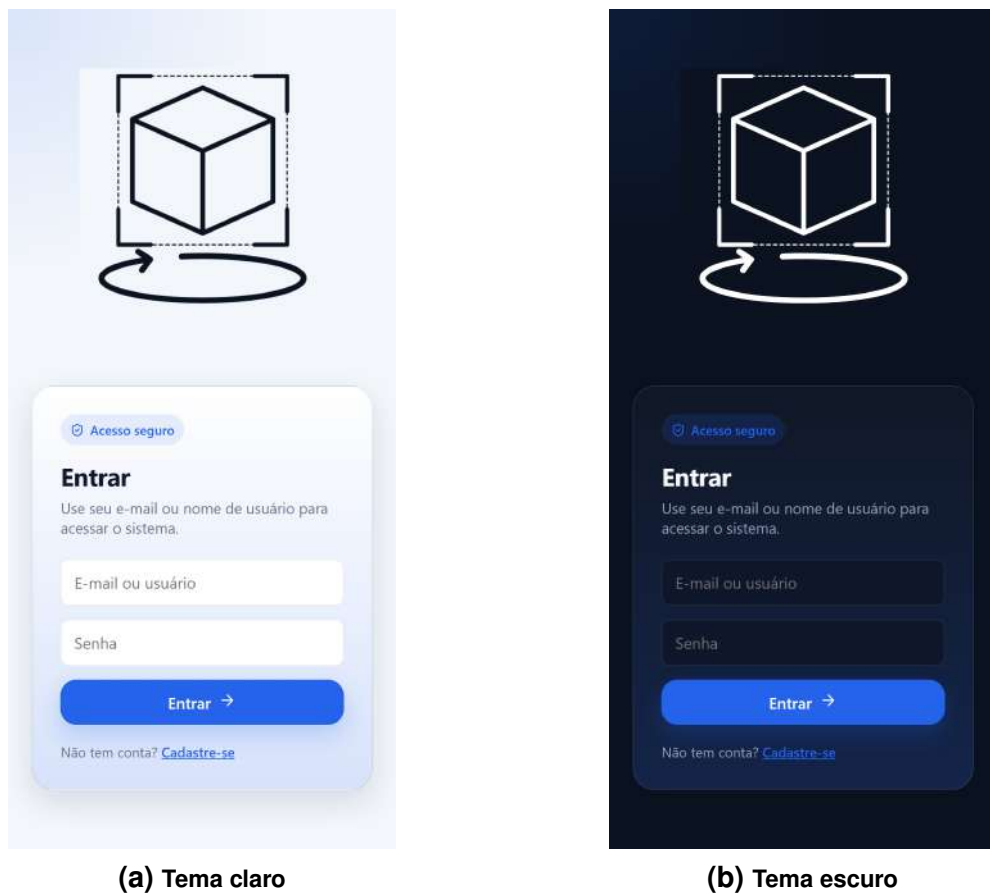
Fonte: Autoria própria, 2026

APÊNDICE C – CADERNO DE TELAS DA APLICAÇÃO EM DISPOSITIVO MÓVEL

Este apêndice apresenta as principais telas da aplicação em visualização mobile. As capturas utilizam perfil equivalente a um iPhone 15, permitindo observar a adaptação da interface para uso em *smartphone*, principal dispositivo de operação da IHM aumentada.

C.1 Telas em Visualização Mobile

Figura 41 – Tela de login em visualização mobile nos temas claro e escuro



Fonte: Autoria própria, 2026

Figura 42 – Tela de cadastro em visualização mobile nos temas claro e escuro

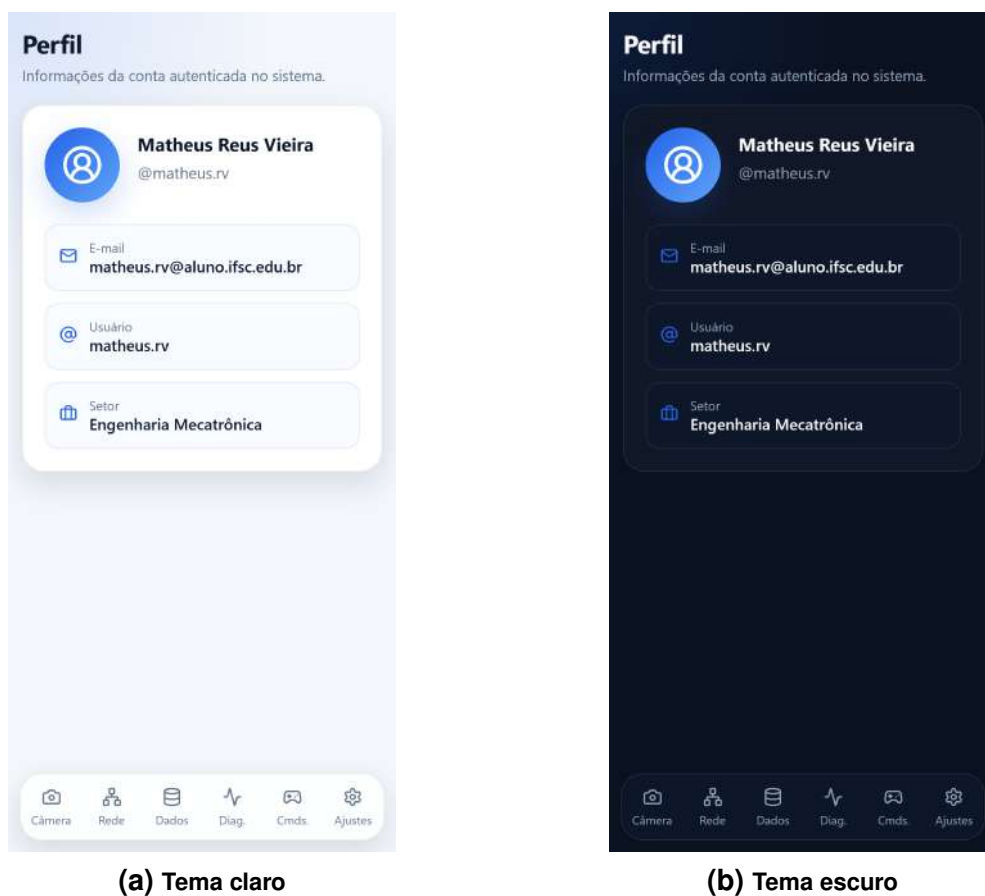
(a) Tema claro

(b) Tema escuro

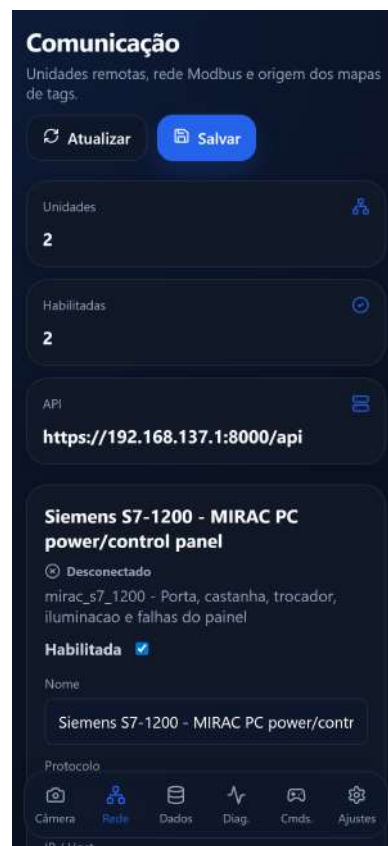
Fonte: Autoria própria, 2026

Figura 43 – Tela de operação da câmera e overlays em visualização mobile nos temas claro e escuro**(a) Tema claro****(b) Tema escuro**

Fonte: Autoria própria, 2026

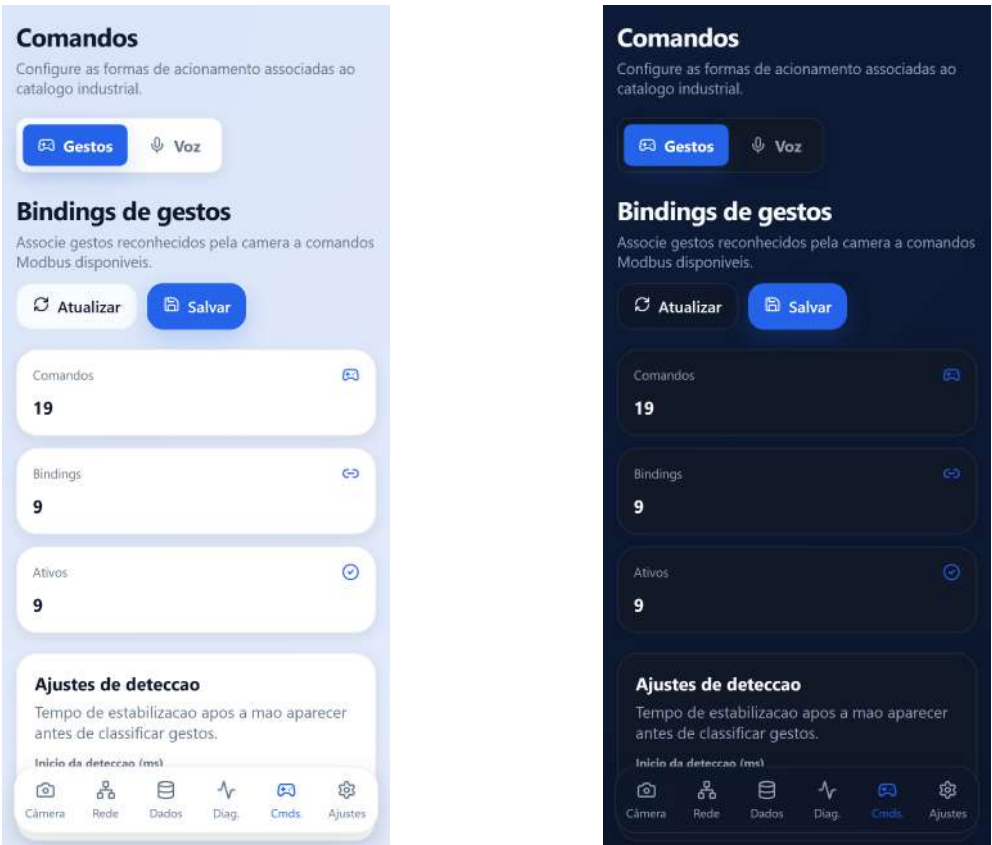
Figura 44 – Tela de perfil do usuário em visualização mobile nos temas claro e escuro

Fonte: Autoria própria, 2026

Figura 45 – Tela de configuração de comunicação em visualização mobile nos temas claro e escuro**(a) Tema claro****(b) Tema escuro**

Fonte: Autoria própria, 2026

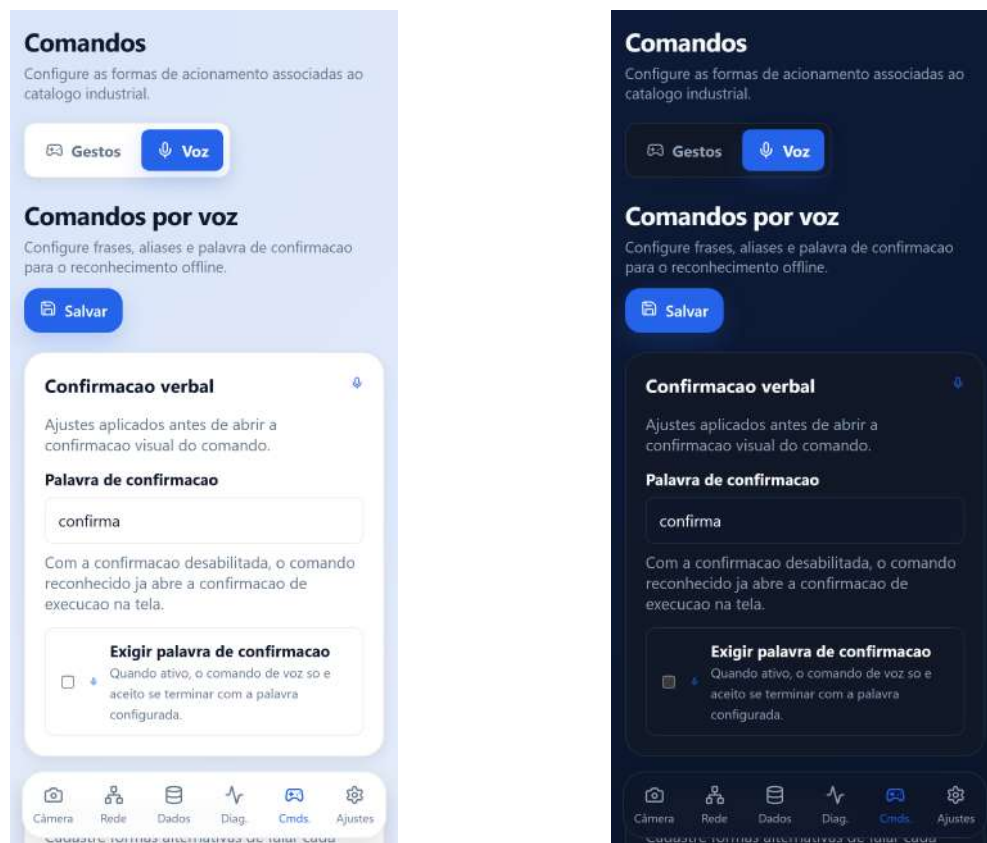
Figura 46 – Tela de comandos por gestos em visualização mobile nos temas claro e escuro



(a) Tema claro

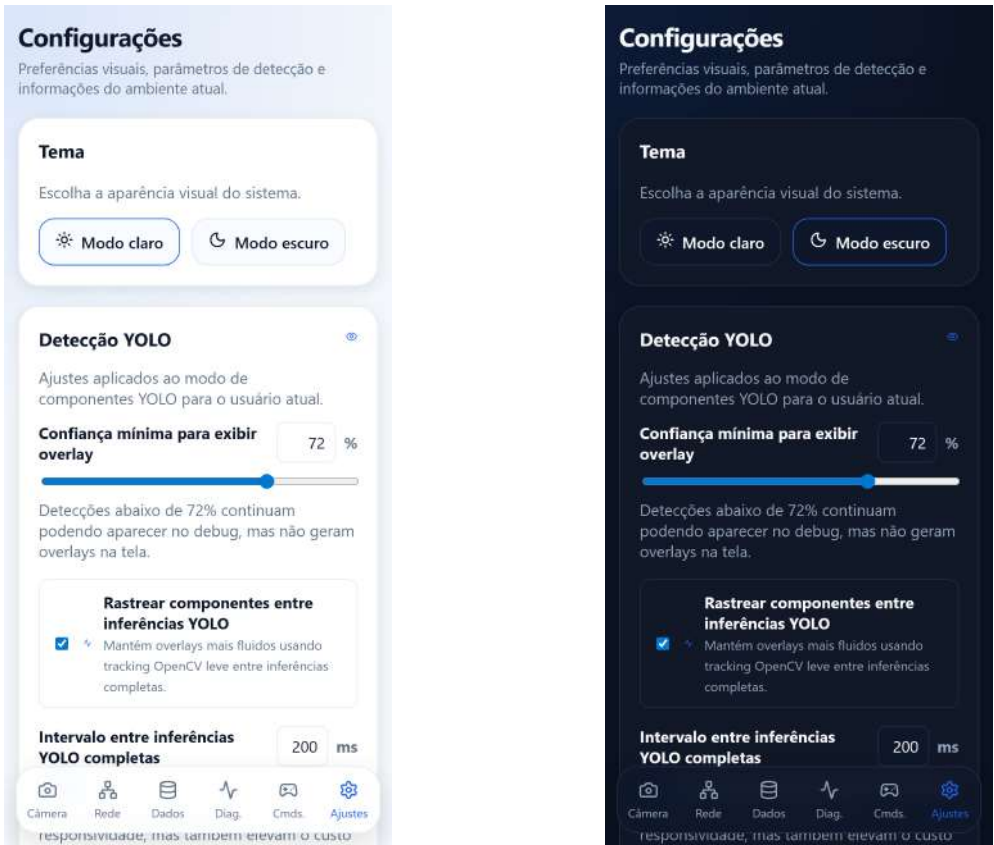
(b) Tema escuro

Fonte: Autoria própria, 2026

Figura 47 – Tela de comandos por voz em visualização mobile nos temas claro e escuro**(a) Tema claro****(b) Tema escuro**

Fonte: Autoria própria, 2026

Figura 48 – Tela de configurações gerais em visualização mobile nos temas claro e escuro



(a) Tema claro

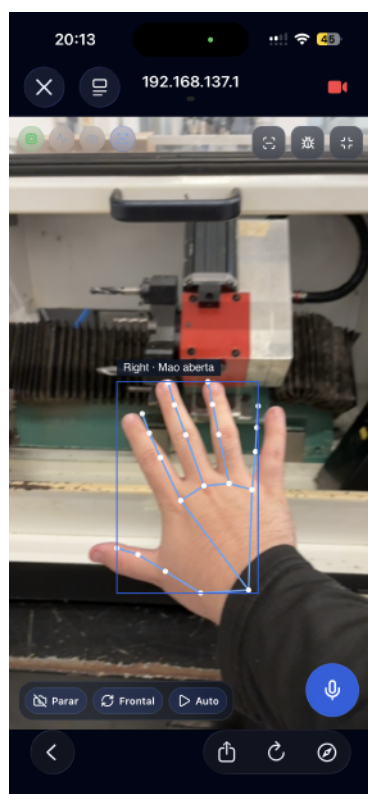
(b) Tema escuro

Fonte: Autoria própria, 2026

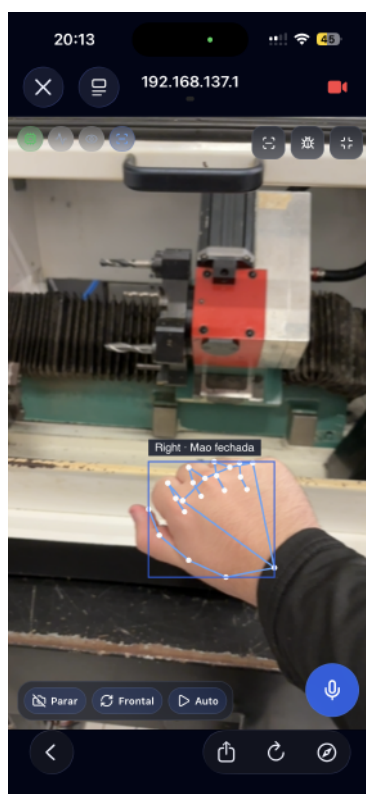
APÊNDICE D – REGISTROS DOS GESTOS RECONHECIDOS

Este apêndice reúne capturas isoladas dos gestos utilizados na validação da interação gestual. As imagens mostram as poses reconhecidas pelo sistema e exemplos do acionamento por permanência antes do envio do comando.

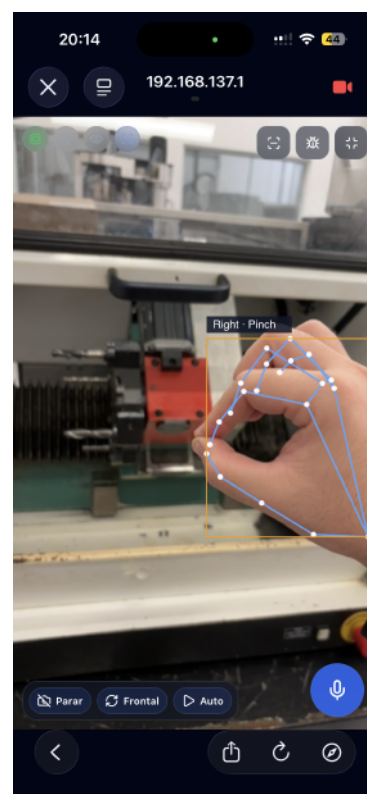
Figura 49 – Gestos manuais reconhecidos pelo sistema



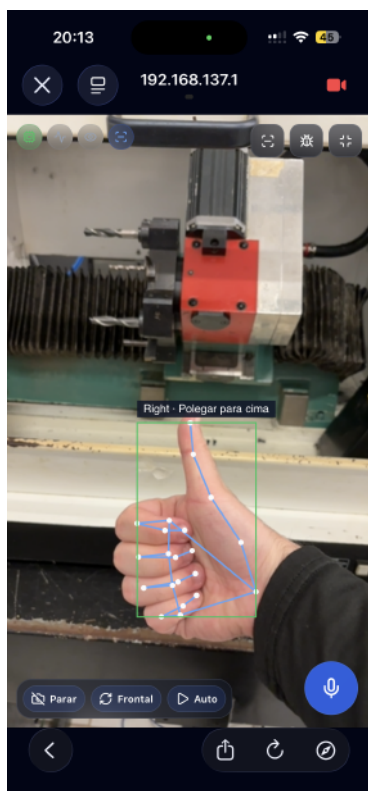
(a) Mão aberta



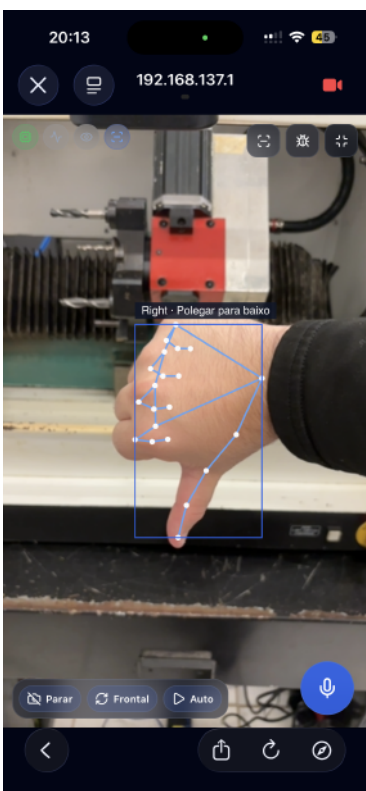
(b) Mão fechada



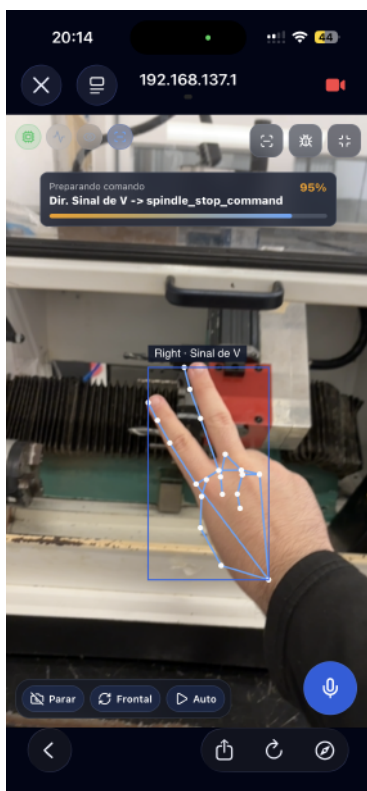
(c) Pinch



(d) Polegar para cima



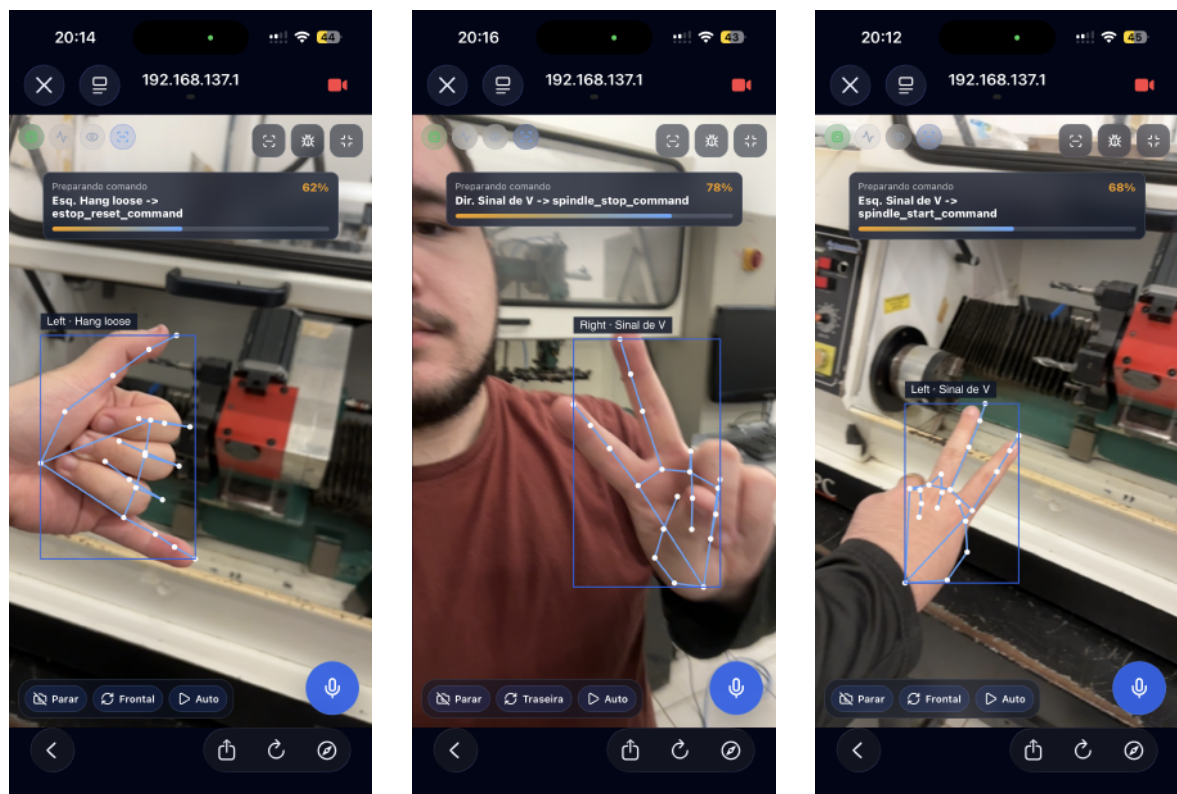
(e) Polegar para baixo



(f) Sinal de V

Fonte: Autoria própria, 2026

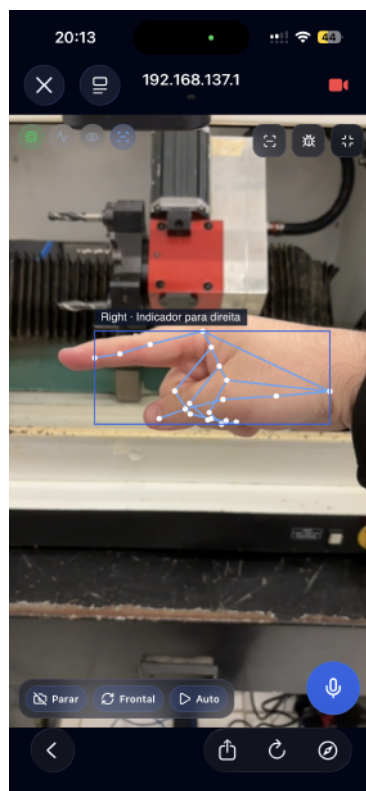
Figura 50 – Gestos complementares e visualizações da câmera



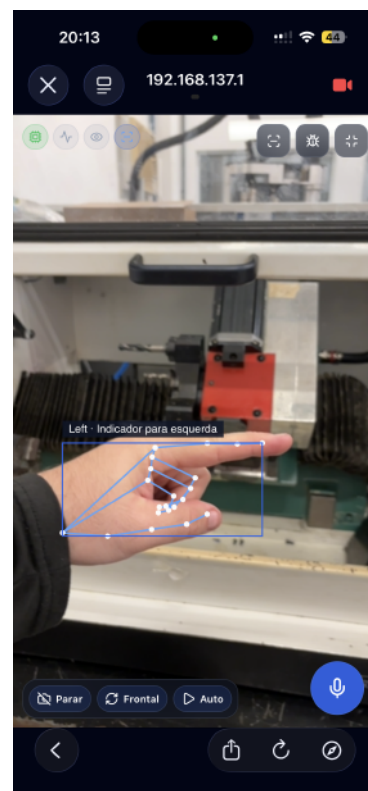
(a) Hang loose

(b) Câmera frontal

(c) Gesto em operação

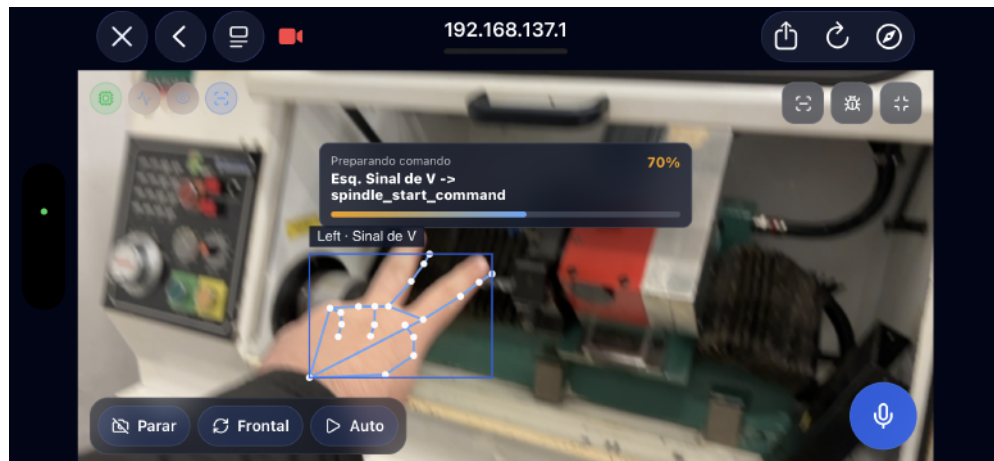
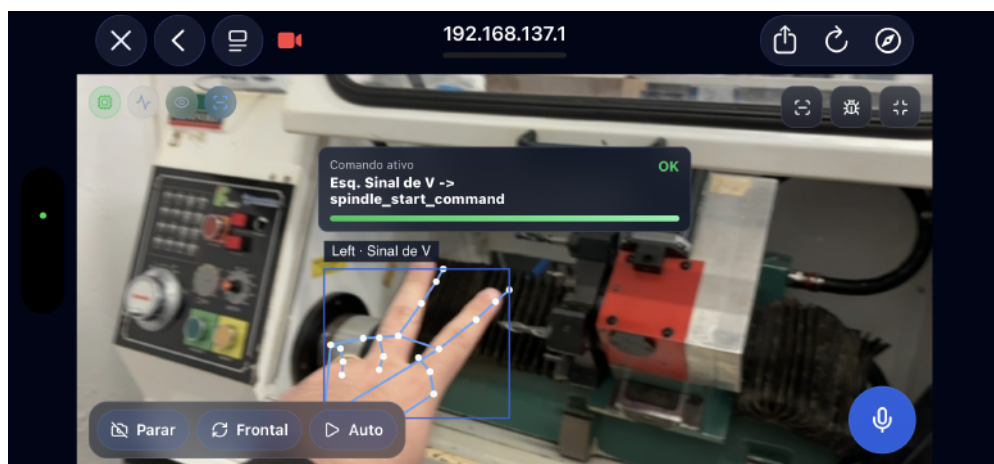


(d) Registro adicional



(e) Registro adicional

Fonte: Autoria própria, 2026

Figura 51 – Acionamento por gesto em orientação paisagem**(a) Permanência em andamento****(b) Comando enviado**

Fonte: Autoria própria, 2026