

UNIVERSIDADE DE CAXIAS DO SUL
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E
ENGENHARIAS

JOÃO VITOR BUFFON CAMELLO

SISTEMA SCADA PARA MONITORAMENTO E TESTES DE
PRODUTO

CAXIAS DO SUL

2025

JOÃO VITOR BUFFON CAMELLO

**SISTEMA SCADA PARA MONITORAMENTO E TESTES DE
PRODUTO**

Trabalho de Conclusão de Curso
apresentado como requisito parcial
à obtenção do título de Engenheiro
Eletricista na Área do Conhecimento
de Ciências Exatas e Engenharias da
Universidade de Caxias do Sul.

Orientador: Daniel Corteletti

CAXIAS DO SUL

2025

JOÃO VITOR BUFFON CAMELLO

**SISTEMA SCADA PARA MONITORAMENTO E TESTES DE
PRODUTO**

Trabalho de Conclusão de Curso
apresentado como requisito parcial
à obtenção do título de Engenheiro
Eletricista na Área do Conhecimento
de Ciências Exatas e Engenharias da
Universidade de Caxias do Sul.

BANCA EXAMINADORA

Daniel Corteletti
Universidade de Caxias do Sul - UCS

Angelo Zerbetto Neto
Universidade de Caxias do Sul - UCS

Ricardo Leal Costi
Universidade de Caxias do Sul - UCS

Este trabalho é dedicado aos meus pais que sempre me apoiaram muito durante todo o percurso de conclusão do curso.

AGRADECIMENTOS

Agradeço aos meus pais Luci Maria Buffon e Saulo Camello, que não mediram esforços para me ajudar nessa etapa tão importante da minha vida.

Aos meus amigos e colegas, que me incentivaram todos os dias e ofereceram apoio nos momentos críticos.

Agradeço ao Professor Daniel Corteletti, responsável pela orientação deste trabalho, me apoiando em cada etapa da pesquisa e contribuindo com as revisões do conteúdo.

“Não foi sorte. Foi esforço, dedicação e noites mal dormidas.”

Anônimo

RESUMO

Este Trabalho de Conclusão de Curso apresenta o desenvolvimento de um Sistema SCADA (Supervisory Control and Data Acquisition) destinado ao monitoramento e à realização de testes de produtos em ambientes industriais. O projeto visa oferecer uma solução eficiente para a supervisão em tempo real, coleta e armazenamento de dados, análise de desempenho e execução de testes automatizados, contribuindo para a melhoria da qualidade e confiabilidade dos processos produtivos.

O sistema desenvolvido contempla recursos como monitoramento de variáveis, registro histórico de dados para análises posteriores, geração de relatórios personalizados, emissão de alarmes e notificações em caso de desvios operacionais, além da execução de rotinas de teste automatizadas nos produtos monitorados. Essas funcionalidades têm como objetivo aumentar a eficiência operacional, reduzir falhas e otimizar o controle de qualidade.

Durante o desenvolvimento, são consideradas boas práticas de comunicação industrial, segurança da informação, usabilidade de interface e integridade de dados. A estrutura modular do sistema é planejada para permitir expansões e adaptações futuras, facilitando sua aplicação em diferentes cenários industriais conforme as necessidades específicas de cada processo.

Os resultados obtidos indicam que o sistema SCADA proposto é capaz de atender às exigências de monitoramento e teste de produtos de forma segura, ágil e confiável. Dessa maneira, o projeto contribui para a modernização dos processos industriais, apoiando a tomada de decisão baseada em dados e promovendo a melhoria contínua da qualidade e da produtividade.

Palavras-chave: SCADA, Automação, Indústria 4.0, Teste de Produto.

ABSTRACT

This Final Project presents the development of a SCADA (Supervisory Control and Data Acquisition) system designed for monitoring and testing products in industrial environments. The project aims to offer an efficient solution for real-time supervision, data collection and storage, performance analysis, and automated testing execution, contributing to the improvement of quality and reliability in production processes.

The developed system includes features such as real-time variable monitoring, historical data logging for further analysis, customized report generation, alarm triggering and notifications in case of operational deviations, and the execution of automated testing routines on monitored products. These functionalities aim to increase operational efficiency, reduce failures, and optimize quality control.

During development, best practices in industrial communication, information security, interface usability, and data integrity were considered. The modular structure of the system was designed to allow future expansions and adaptations, facilitating its application in different industrial scenarios according to the specific needs of each process.

The results indicate that the proposed SCADA system is capable of meeting the requirements for product monitoring and testing in a secure, agile, and reliable manner. Thus, the project contributes to the modernization of industrial processes, supporting data-driven decision-making and promoting continuous improvement of quality and productivity.

Keywords: SCADA, Automation, Industry 4.0, Product Testing.

LISTA DE FIGURAS

Figura 1 – Test Points presentes em placas eletrônicas	22
Figura 2 – Diagrama da rede de conexões do sistema SCADA proposto	34
Figura 3 – Tela Inicial do Sistema SCADA Proposto	42
Figura 4 – Tela de Configuração do Hardware do Sistema SCADA Proposto	43
Figura 5 – Tela de Cadastramento de Endereços do Sistema SCADA Proposto	44
Figura 6 – Tela de Testes do Sistema SCADA Proposto	45
Figura 7 – Tela de Criação e Edição de Testes do Sistema SCADA Proposto	46
Figura 8 – Tela de Receitas do Sistema SCADA Proposto	48
Figura 9 – Tela de Edição e Criação de Receitas do Sistema SCADA Proposto	49
Figura 10 – Tela de Processo com receita em execução no Sistema SCADA Proposto . .	50
Figura 11 – Tela de gerenciamento de usuário para adição de usuários no Sistema SCADA Proposto	51
Figura 12 – Tela de gerenciamento de usuário para edição de usuário do Sistema SCADA Proposto	52
Figura 13 – Tela de gerenciamento de usuário para exclusão de usuários no Sistema SCADA Proposto	52
Figura 14 – Diagrama Entidade Relação (DER)	54
Figura 15 – Exemplo de arquivo CSV	55
Figura 16 – Protoboard montada para testes e validação do sistema	60

LISTA DE TABELAS

Tabela 1 – Comparação entre Oracle, Microsoft SQL Server e PostgreSQL	27
Tabela 2 – Comparação entre Ethernet/IP, PROFINET e Modbus (CONTROLDESIGN, 2025), (PROCENDEC, 2023), (AUTOMATION, 2021), (ODVA, 2025), (Siemens, 2025), (Modbus Organization, 2023)	31
Tabela 3 – Comparação entre o sistema proposto, NI TestStand, Keysight PathWave e ScadaBR	40

LISTA DE ALGORITMOS

Algoritmo 1	Gerenciamento da comunicação Modbus/TCP	56
Algoritmo 2	Query para inserir relatórios no banco de dados	58

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
API	<i>Application Programming Interface</i> – Interface de Programação de Aplicações
CAT5/CAT6/CAT7	Categorias de cabos de rede Ethernet
CLP	<i>Controlador Lógico Programável</i>
CLR	<i>Common Language Runtime</i> - Tempo de Execução de Linguagem de Comando
CMD	<i>Command Prompt</i> – Terminal de comandos do Windows
CPU	<i>Central Processing Unit</i> – Unidade Central de Processamento
GUI	<i>Graphical User Interface</i> – Interface Gráfica do Usuário
IDE	<i>Integrated Development Environment</i> – Ambiente de Desenvolvimento Integrado
JSON	<i>JavaScript Object Notation</i> - Formato de documento baseado em texto para armazenar e trocar dados de uma forma que seja legível por humanos e analisável por máquinas
LAN	<i>Local Area Network</i> – Rede Local
MAUI	<i>Multi-platform App UI</i> – Interface de Aplicações Multiplataforma da Microsoft
Modbus	Protocolo de comunicação industrial
.NET	Plataforma de desenvolvimento da Microsoft
OS	<i>Operating System</i> – Sistema Operacional
PC	<i>Personal Computer</i> – Computador Pessoal
PowerShell	Ferramenta de automação e shell de linha de comando desenvolvida pela Microsoft
RAC	<i>Real Application Cluster</i> - Arquitetura de Banco de Dados Oracle para sistemas Oracle compartilharem do mesmo armazenamento

RAM	<i>Random Access Memory</i> – Memória de Acesso Aleatório
SCADA	<i>Supervisory Control and Data Acquisition</i> – Sistema de Supervisão e Aquisição de Dados
SGBD	<i>Sistema de Gestão de Banco de Dados</i>
SGBDR	<i>Sistema de Gestão de Banco de Dados Relacional</i>
SQL	<i>Structured Query Language</i> – Linguagem de Consulta Estruturada
SSMS	<i>SQL Server Management Studio</i> - Plataforma de dados analíticos do Microsoft SQL Server
TCC	<i>Trabalho de Conclusão de Curso</i>
TCP	<i>Transmission Control Protocol</i> - Protocolo de Controle de Transmissão
UCS	<i>Universidade de Caxias do Sul</i>
XML	<i>Extensible Markup Language</i> - Formato para a criação de documentos com dados organizados de forma hierárquica.

LISTA DE SÍMBOLOS

A_e	Área efetiva da antena
λ	Comprimento de onda
Γ	Coefficiente de reflexão
B	Largura de faixa em que o ruído é medido em Hertz
d	Distância em metros
I	Corrente elétrica
k	Constante de Boltzmann's
K	Eficiência de irradiação
M	Variação do patamar de ruído em função da RBW
N	Condutor de neutro
NF	Figura de ruído
P	Potência
R	Resistência
E	Campo elétrico
FA	Fator da antena
Gr	Ganho de recepção
h	Altura efetiva ou comprimento efetivo de uma antena
S_i	Potência do sinal na entrada
S_o	Potência do sinal na saída
t	Tempo
V	Tensão
Z_L	Impedância da antena
Z_o	Impedância de referência (50Ω)

SUMÁRIO

1	INTRODUÇÃO	16
1.1	OBJETIVOS	16
1.2	ESTRUTURA DO TRABALHO	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Sistemas SCADA	18
2.2	Jigas de Teste em Linhas de Produto	20
2.2.1	Como Funcionam os Testes nas Jigas de Placas Eletrônicas	21
2.3	Interfaces Homem-Máquina (IHMs)	23
2.4	Banco de Dados na Automação Industrial	24
2.4.1	Características de Bancos de Dados na Automação Industrial	25
2.4.2	Características de Bancos de Dados Específicos	26
2.4.3	Comparação Geral e Justificativa para a Escolha do PostgreSQL	27
2.5	Linguagem de Programação para o Desenvolvimento do Sistema SCADA	28
2.5.1	Justificativa para a Escolha do C#	28
2.6	Comunicação Entre Sistemas Supervisórios e Hardware	29
2.6.1	Comparação entre os Protocolos Ethernet/IP, PROFINET e Modbus	30
2.6.2	Escolha do Protocolo Modbus	31
2.6.3	Estrutura e Funcionamento do Protocolo Modbus TCP/IP	32
2.6.4	Infraestrutura de Rede	33
2.6.5	Considerações de Desempenho e Alternativas ao Modbus	34
2.7	Execução de chamadas externas via linha de comando no Sistema SCADA	35
2.7.1	Vantagens da Arquitetura com Execução Externa	35
2.8	Interface de Execução	36
2.8.1	Normas Aplicadas ao Desenvolvimento de Sistemas SCADA	36
2.9	Comparação com Soluções já Desenvolvidas	38
2.9.1	Tabela Comparativa	40
3	DESENVOLVIMENTO	41
3.1	Funcionamento Principal do Software	41
3.1.1	Tela Principal	41
3.1.2	Níveis de Acesso	42
3.1.3	Configuração do Hardware	43
3.1.4	Testes	45
3.1.4.1	Edição e Criação de Teste	45
3.1.5	Comandos e Funcionalidades de Teste	46

3.1.6	Receitas	48
3.1.6.1	Edição e Criação de Receita	49
3.1.7	Processo	49
3.1.8	Gerenciamento de Usuários	51
3.2	Infraestrutura do banco de dados	53
3.3	CSV de Sequenciamento de Testes	53
3.4	Arquitetura do código	55
3.4.1	Modbus/TCP	56
3.4.2	Query de Banco de Dados	58
3.5	Execução dos Testes	60
4	RESULTADOS	62
5	CONSIDERAÇÕES FINAIS	64
	REFERÊNCIAS	65

1 INTRODUÇÃO

O avanço das tecnologias de automação industrial tem impulsionado o desenvolvimento de sistemas capazes de integrar supervisão em tempo real, controle distribuído e registro histórico de dados. Entre essas tecnologias, os sistemas SCADA (*Supervisory Control and Data Acquisition*) se destacam pela capacidade de coletar, processar e apresentar informações de maneira confiável, atendendo às exigências de rastreabilidade, precisão e segurança em ambientes industriais.

No contexto da engenharia elétrica e da automação, o teste e monitoramento de produtos desempenham papel fundamental na garantia da qualidade e na validação funcional de dispositivos antes de sua implementação ou comercialização. Em diversos cenários, porém, essas atividades ainda são realizadas de forma manual, resultando em inconsistências operacionais, retrabalho e custos adicionais.

Diante desse cenário, este trabalho propõe o desenvolvimento de um sistema SCADA voltado ao monitoramento e execução de testes de produto, integrando supervisão, controle e persistência de dados em uma única plataforma. O objetivo central é demonstrar a viabilidade técnica de uma solução prática, modular e adaptável, capaz de reduzir a intervenção humana em tarefas repetitivas e aumentar a confiabilidade dos processos de teste e validação.

Além de seu propósito técnico, o projeto também busca explorar seu potencial de aplicação prática no ambiente industrial. A solução desenvolvida possui características que permitem sua evolução futura para um produto comercializável, podendo ser apresentada a empresas do setor de automação, eletrônica e manufatura como alternativa flexível aos sistemas proprietários utilizados atualmente. Nesse sentido, a presente pesquisa constitui não apenas um estudo acadêmico, mas também um passo inicial em direção ao desenvolvimento de uma ferramenta com aplicabilidade real e potencial de adoção em diferentes cenários industriais.

1.1 OBJETIVOS

O objetivo geral deste trabalho é desenvolver e validar um sistema SCADA para monitoramento e execução de testes em uma montagem protoboard, integrando supervisão, controle e registro de dados em uma solução única e funcional.

Os objetivos específicos deste trabalho são:

1. **Desenvolver um sistema SCADA funcional** para monitoramento e controle de jigas de teste, acessível e independente de softwares proprietários.
2. **Padronizar a interface de operação** por meio de uma IHM única, intuitiva e adaptável

a diferentes configurações de hardware.

3. **Implementar uma interface gráfica interativa** para acompanhamento em tempo real de entradas e saídas digitais durante a execução dos testes.
4. **Implementar a persistência de dados em PostgreSQL** para garantir integridade, segurança e rastreabilidade dos registros de teste.
5. **Disponibilizar consulta ao histórico de testes** para análise comparativa de resultados e identificação de padrões.
6. **Validar o sistema por meio de testes práticos em protoboard** utilizando LEDs e botões para simular entradas e saídas digitais.

1.2 ESTRUTURA DO TRABALHO

O presente trabalho está organizado da seguinte forma:

- O Capítulo 2 desenvolve a fundamentação teórica, abordando os principais conceitos relacionados a sistemas SCADA, automação industrial e tecnologias utilizadas no desenvolvimento da solução proposta.
- O Capítulo 3 apresenta o desenvolvimento do sistema, seus principais componentes e funcionalidades, bem como as decisões técnicas tomadas ao longo do processo.
- O Capítulo 4 discute os resultados obtidos, realiza a validação do sistema e apresenta as considerações finais do trabalho, apontando possíveis melhorias e sugestões para trabalhos futuros.

A seção seguinte apresenta a fundamentação teórica necessária para entender as escolhas de arquitetura e comunicação adotadas.

2 FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica tem como objetivo apresentar os conceitos e tecnologias essenciais que embasam o desenvolvimento do sistema proposto neste trabalho. Por meio de uma revisão dos principais tópicos relacionados à automação industrial, busca-se fornecer o embasamento necessário para a compreensão do funcionamento e das motivações por trás da implementação de um sistema SCADA voltado ao monitoramento e à execução de testes em jigas de teste.

Neste contexto, serão abordados os fundamentos dos sistemas SCADA, suas arquiteturas e componentes, bem como sua aplicação na supervisão de processos industriais. Em seguida, discute-se a importância dos CLPs (Controladores Lógicos Programáveis) como elemento central na automação, além das interfaces homem-máquina (IHMs), que viabilizam a comunicação entre o operador e o sistema. Por fim, são exploradas as tecnologias e ferramentas utilizadas no desenvolvimento do software, como a linguagem de programação C#, bancos de dados relacionais e protocolos de comunicação com dispositivos industriais.

A construção deste embasamento é essencial para compreender as decisões técnicas adotadas ao longo do projeto e demonstrar a relevância da proposta frente aos problemas enfrentados na operação e padronização de sistemas de testes industriais.

2.1 SISTEMAS SCADA

Os Sistemas SCADA (Supervisory Control and Data Acquisition) são essenciais para o monitoramento e controle de processos industriais e de infraestrutura crítica. Eles operam coletando dados de sensores e dispositivos distribuídos no campo e transmitindo essas informações para uma central onde podem ser supervisionadas e controladas. Um sistema SCADA é composto por elementos de software e hardware que trabalham em conjunto para viabilizar a automação e supervisão remota de processos industriais complexos (UPADHYAY; SAMPALLI, 2020).

Essa arquitetura permite que operadores monitorem dispositivos e realizem ações de controle em tempo real, o que é particularmente importante em ambientes como redes de energia elétrica, sistemas de água e esgoto, fábricas e plantas de geração de energia. A visibilidade centralizada e a capacidade de controle remoto tornam os SCADAs fundamentais para a eficiência e segurança operacional de sistemas distribuídos.

Esses sistemas oferecem diversas vantagens, como:

- **Monitoramento em Tempo Real:** Os SCADAs permitem que os dados de sensores e dispositivos sejam visualizados instantaneamente, ajudando na tomada de decisões rápidas.

- **Automação de Processos:** Através da automação, processos repetitivos podem ser controlados automaticamente, reduzindo a necessidade de intervenção humana.
- **Escalabilidade:** Sistemas SCADA podem ser facilmente ampliados para incluir mais dispositivos e sensores, conforme a necessidade de expansão da infraestrutura.
- **Eficiência Operacional:** A centralização das informações e o controle remoto tornam a operação mais eficiente, permitindo ajustes rápidos no sistema sem a necessidade de presença física em cada dispositivo (ONLOGIC, 2025).

Contudo, como qualquer tecnologia, os sistemas SCADA também apresentam desvantagens. As principais incluem:

- **Custo de Implementação:** A implantação de sistemas SCADA pode ser cara, especialmente para grandes operações que exigem infraestrutura robusta e software especializado.
- **Vulnerabilidade a Ciberataques:** Como os sistemas SCADA estão frequentemente conectados a redes externas. Contudo, a crescente conectividade desses sistemas os torna vulneráveis a ameaças cibernéticas, exigindo estratégias específicas de segurança para garantir a integridade das operações.
- **Manutenção Complexa:** Embora a automação traga eficiência, a manutenção do sistema SCADA pode ser desafiadora, principalmente em sistemas mais antigos ou complexos.
- **Dependência de Energia e Redes:** A falha nos sistemas de energia ou nas redes de comunicação pode levar à perda de dados ou falhas no controle de processos (UBIDOTS, 2025).

Considerando isso, também é possível mitigar as vulnerabilidades identificadas nos sistemas SCADA (UPADHYAY; SAMPALLI, 2020):

- **Implementação de Firewalls e Sistemas de Detecção de Intrusos (IDS):** Essas ferramentas ajudam a monitorar e controlar o tráfego de rede, prevenindo acessos não autorizados.
- **Atualizações Regulares de Software e Patches de Segurança:** Manter o sistema atualizado é crucial para proteger contra vulnerabilidades conhecidas.
- **Segmentação de Redes:** Dividir a rede em segmentos pode limitar o alcance de um possível ataque, confinando-o a uma parte específica do sistema.
- **Treinamento Contínuo para Operadores:** Capacitar os operadores para reconhecer e responder a ameaças potenciais é essencial para a segurança do sistema.

Diante dessas características, os sistemas SCADA assumem um papel central na automação industrial, viabilizando o monitoramento e o controle eficiente de operações cada vez mais complexas. Assim, compreender o funcionamento das jigas de teste é indispensável para estabelecer os requisitos funcionais do SCADA desenvolvido neste trabalho.

2.2 JIGAS DE TESTE EM LINHAS DE PRODUTO

As jigas de teste são ferramentas essenciais na linha de produção de dispositivos eletrônicos, especialmente para garantir a qualidade e a funcionalidade dos componentes. Elas são projetadas para realizar testes automáticos ou manuais em placas de circuito impresso (PCBs), dispositivos e sistemas, ajudando a identificar falhas de fabricação e assegurando que os produtos atendam aos requisitos de qualidade. De maneira geral, as jigas de teste atuam como interfaces entre o dispositivo sob teste e os sistemas de medição ou controle, permitindo a automação do processo de verificação (Pride Industries, 2025).

Na automação industrial, as jigas de teste têm uma presença significativa, pois são amplamente utilizadas para otimizar o processo de testes em larga escala. Elas garantem que as verificações sejam realizadas de forma rápida e eficiente, com alta precisão e repetibilidade, o que é essencial em linhas de produção de alta demanda (Venturus, 2025). O uso dessas ferramentas é crucial para reduzir custos de retrabalho e falhas no produto final, melhorando a eficiência da produção e a qualidade dos produtos. Segundo (ELANGO VAN, 2025), a utilização de jigas de teste em automação permite maior controle sobre os processos de fabricação e contribui para a minimização de erros durante os testes.

As jigas de teste agregam valor ao processo de fabricação de diversas maneiras. Elas oferecem uma forma eficiente e automatizada de testar vários parâmetros de um produto, como a funcionalidade elétrica, as conexões e o comportamento de sinais. Além disso, permitem realizar testes repetitivos sem a intervenção constante de operadores, o que acelera o processo de produção e diminui o risco de erro humano (Pride Industries, 2025). As jigas também são capazes de verificar múltiplas variáveis ao mesmo tempo, como corrente, tensão, resistência e outros fatores, garantindo que os produtos estejam em conformidade com os padrões de qualidade.

O funcionamento de uma jiga de teste geralmente envolve a fixação do dispositivo em questão, enquanto a jiga aplica os testes necessários, como sinais de entrada e medições de saída. As jigas podem ser altamente personalizadas dependendo das necessidades específicas do produto e do tipo de teste a ser realizado. Elas podem ser controladas manualmente ou, em muitos casos, automatizadas através de sistemas SCADA ou outros sistemas de controle industrial, integrando-se perfeitamente a processos de fabricação automatizados (Venturus, 2025). Segundo (ELANGO VAN, 2025), a automatização dos testes em linha com o uso de jigas tem sido uma prática crescente, contribuindo para a redução de erros e otimização do tempo de produção.

2.2.1 Como Funcionam os Testes nas Jigas de Placas Eletrônicas

As jigas de teste são dispositivos fundamentais no processo de produção de placas eletrônicas, desempenhando um papel crucial na garantia da qualidade e funcionalidade dos produtos finais. Elas permitem a realização de testes automatizados e repetitivos com segurança, eficiência e consistência, sendo utilizadas para verificar parâmetros elétricos, funcionais e de comunicação das placas eletrônicas (PEREIRA, 2023).

Segundo (PEREIRA, 2023), a implementação de jigas automáticas no teste de placas eletrônicas industriais resulta em melhorias significativas na produtividade e na qualidade do processo. O estudo de caso apresentado no artigo destaca que a automação dos testes reduz o tempo de execução, minimiza a fadiga operacional e proporciona maior controle sobre o processo produtivo. Além disso, a utilização de jigas automáticas permite a padronização dos testes, assegurando que cada unidade fabricada esteja em conformidade com os padrões de qualidade exigidos.

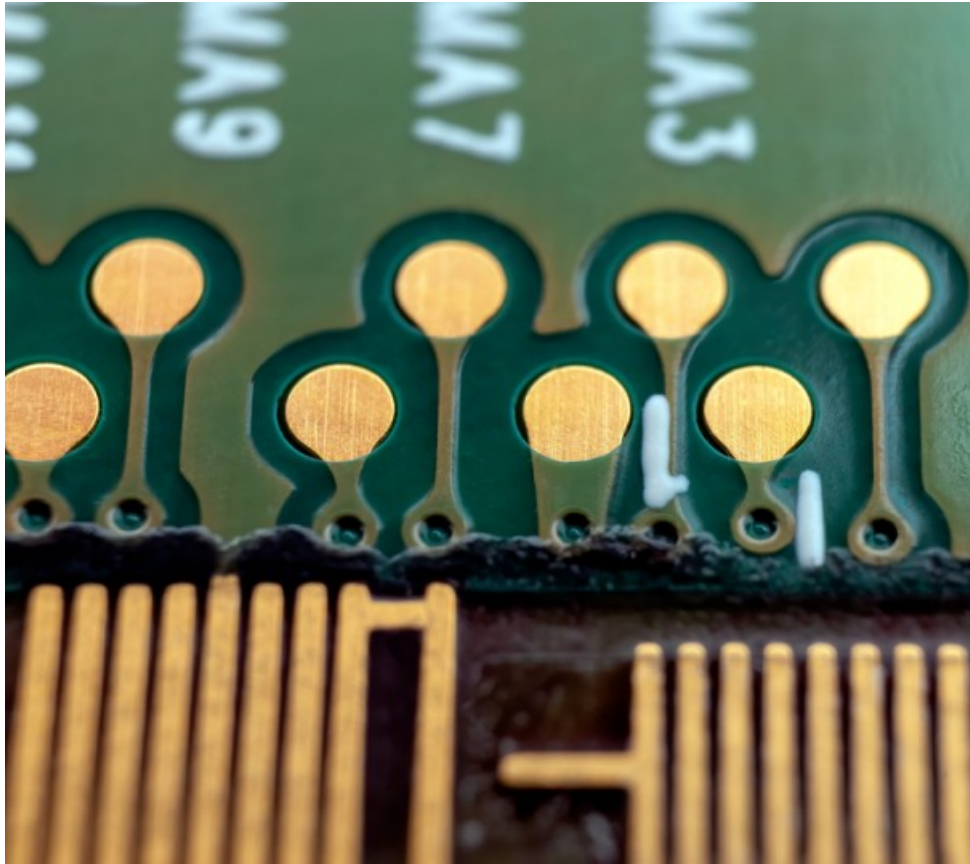
Os testes realizados nas jigas de placas eletrônicas são altamente técnicos e visam garantir a conformidade do produto com as especificações elétricas e funcionais. Quando uma placa eletrônica é fixada na jiga, ela é conectada a um sistema automatizado que realiza uma série de verificações críticas. Entre as verificações mais comuns estão a medição de **tensão** e **corrente** em diversos pontos da placa, utilizando *test points* (pontos de teste) específicos que foram projetados para este fim. Esses pontos são conectados aos sistemas de medição da jiga, que fornecem os valores reais de tensão e corrente para comparação com as especificações de projeto (Venturus, 2025).

Além da verificação de tensão e corrente, outro teste fundamental é a **verificação de curto-circuito**. Durante esse teste, a jiga avalia se há conexões indesejadas entre os diferentes pinos ou trilhas da placa, o que indicaria um curto-circuito. Isso é feito aplicando um sinal de baixa tensão nos circuitos e verificando se há resistência zero ou valores muito baixos entre os pontos, o que poderia indicar uma falha no design da placa (Pride Industries, 2025). A capacidade da jiga de detectar esses problemas de forma precisa e rápida é uma das razões pelas quais elas são tão úteis na automação industrial.

Outro parâmetro crítico frequentemente verificado nas jigas de teste é o **duty cycle**, que mede a relação entre o tempo de operação e o tempo de inatividade de um sinal. Esse teste é particularmente importante em circuitos de comunicação e controle, onde o comportamento dinâmico da placa precisa ser validado. A jiga pode simular sinais de entrada e verificar se o circuito responde de maneira esperada, mantendo o **duty cycle** dentro dos limites especificados para o funcionamento correto do dispositivo (ELANGOVAN, 2025).

Esses testes não só ajudam a garantir que as placas estejam funcionando corretamente, mas também permitem que as falhas sejam identificadas e corrigidas antes que os produtos cheguem ao mercado. A automação dos testes nas jigas aumenta a precisão e a eficiência desses

Figura 1 – Test Points presentes em placas eletrônicas



Fonte: Altium (2025).

processos, eliminando a possibilidade de erro humano e assegurando que cada unidade fabricada esteja em conformidade com os padrões de qualidade exigidos.

Além disso, as jigas de teste podem ser programadas para realizar testes simultâneos em múltiplos pontos da placa, acelerando o processo de verificação e reduzindo significativamente o tempo necessário para testar cada unidade. Com o uso dessas ferramentas, as fábricas conseguem manter um ritmo de produção rápido, enquanto garantem que cada produto que sai da linha de produção seja funcional e confiável (Venturus, 2025).

Para que a jiga de testes possa ser operada de forma eficiente e para que o andamento dos procedimentos seja acompanhado em tempo real, é indispensável a presença de uma interface adequada. Essa interface pode assumir diferentes formas — desde uma IHM gráfica em um computador até indicadores físicos, como LEDs e botões — permitindo que o operador visualize o status das operações, identifique eventuais falhas e tome decisões de maneira rápida e informada.

2.3 INTERFACES HOMEM-MÁQUINA (IHMS)

As Interfaces Homem-Máquina (IHMs) são componentes essenciais em sistemas de automação, permitindo que os operadores interajam com os processos controlados de maneira intuitiva e eficiente. Elas funcionam como a interface de comunicação entre o ser humano e os sistemas automatizados, proporcionando um meio visual e interativo para monitoramento, controle e ajustes dos processos. Em sistemas industriais, a IHM permite que os operadores visualizem o status das máquinas, ajustem parâmetros, realizem diagnósticos e até controlem a operação de forma remota (ALTUS, 2021), (INDUSTRIAL, 2021), (Dakol Automação, 2022), (CHAVES, 2023), (Kalatec Automação, 2023).

A presença das IHMs na automação industrial tem sido crescente, uma vez que oferecem vantagens significativas em termos de usabilidade e eficiência. Elas substituem métodos de controle tradicionais, como painéis de controle com botões físicos e luzes indicativas, oferecendo uma interface gráfica que facilita a visualização do processo. Além disso, elas integram funções avançadas, como alarmes e gráficos dinâmicos, que ajudam na tomada de decisões rápidas e precisas. As IHMs são encontradas em diversas aplicações industriais, desde fábricas de manufatura até sistemas de controle de processos em setores como energia e petroquímica (INDUSTRIAL, 2021), (Kalatec Automação, 2023).

O funcionamento das IHMs baseia-se em um sistema de entrada e saída, onde o operador utiliza dispositivos como teclados, mouses, ou telas sensíveis ao toque para interagir com a interface. A partir dessa interação, o sistema processa os dados inseridos e apresenta as informações relevantes para o operador por meio de displays gráficos, como telas de visualização, gráficos de tendências, e outros elementos visuais. A IHM comunica-se com o sistema automatizado, como PLCs (Controladores Lógicos Programáveis), para enviar e receber dados do processo controlado (ALTUS, 2021), (SCIELO, 2021), (Dakol Automação, 2022).

Em muitos sistemas modernos, as IHMs possuem uma variedade de recursos adicionais, como conectividade com redes industriais (Ethernet, Modbus, etc.), integração com sistemas SCADA, e suporte a diversos protocolos de comunicação. Esses recursos tornam as IHMs ferramentas ainda mais poderosas e flexíveis, permitindo que os operadores possam controlar, monitorar e ajustar processos em tempo real, otimizando a eficiência e a segurança das operações industriais.

As IHMs oferecem diversas vantagens no contexto da automação industrial. Elas tornam o processo de operação mais simples e rápido, melhorando a produtividade ao permitir que os operadores visualizem e ajustem parâmetros de forma eficiente. Além disso, as IHMs proporcionam maior segurança, pois oferecem mecanismos de diagnóstico, detecção de falhas e alarmes, que permitem a tomada de ações corretivas antes que problemas maiores ocorram. Outro benefício importante é a flexibilidade, já que muitas IHMs são projetadas para serem facilmente adaptáveis a diferentes sistemas e processos (INDUSTRIAL, 2021), (Kalatec Automação,

2023).

Embora as IHMs apresentem várias vantagens, existem também algumas limitações. Uma das principais desvantagens é o custo, uma vez que as IHMs de alta qualidade, com telas sensíveis ao toque e interfaces gráficas avançadas, podem ser relativamente caras para implementar em sistemas de automação. Além disso, a complexidade do design e da programação das IHMs também pode representar um desafio, especialmente em sistemas com muitas variáveis e parâmetros a serem monitorados. No entanto, com o avanço da tecnologia, esses custos têm diminuído, tornando as IHMs mais acessíveis para uma variedade de aplicações (ALTUS, 2021) (Kalatec Automação, 2023).

Apesar dessas considerações, a eficácia de uma IHM depende não apenas da qualidade da interface, mas também da robustez e da organização das informações que ela exibe. Para que os operadores possam visualizar dados consistentes, históricos confiáveis e resultados de testes em tempo real, é fundamental que o sistema disponha de uma estrutura de armazenamento adequada. Nesse contexto, o uso de bancos de dados torna-se essencial, pois permite registrar, organizar e recuperar informações de forma eficiente, garantindo integridade, rastreabilidade e suporte às funcionalidades da IHM.

2.4 BANCO DE DADOS NA AUTOMAÇÃO INDUSTRIAL

A automação industrial demanda a integração de sistemas de controle, dispositivos de campo e softwares de supervisão, exigindo um gerenciamento eficiente dos dados gerados. Nesse cenário, os bancos de dados desempenham um papel essencial, permitindo a coleta, armazenamento, processamento e análise de grandes volumes de informações, fundamentais para a otimização dos processos industriais.

Segundo Silva (SILVA, 2018), o uso de bancos de dados em sistemas de automação industrial permite a organização estruturada das informações provenientes de sensores, atuadores e controladores, assegurando a integridade, a disponibilidade e a confiabilidade dos dados. A autora destaca que a centralização desses dados facilita a supervisão e o controle dos processos, possibilitando análises mais precisas e suporte à tomada de decisões estratégicas.

Com o avanço da Indústria 4.0, a gestão de dados industriais tornou-se ainda mais complexa e desafiadora. De acordo com (RAPTIS; PASSARELLA; CONTI, 2019), o volume, a variedade e a velocidade dos dados gerados em ambientes industriais modernos exigem novos paradigmas de armazenamento e processamento. A utilização de arquiteturas distribuídas, computação em nuvem e tecnologias de análise em tempo real tornam-se fundamentais para atender às demandas de desempenho, escalabilidade e segurança.

Além disso, Silva (SILVA, 2018) ressalta que os bancos de dados aplicados à automação precisam ser adaptáveis a operações em tempo real, exigindo alta disponibilidade e baixa latência para que não comprometam a operação dos sistemas de controle. Nesse sentido, o projeto

adequado da estrutura do banco de dados é crucial para garantir a eficiência do sistema como um todo.

Raptis complementa que o gerenciamento de dados na Indústria 4.0 deve considerar não apenas o armazenamento eficiente, mas também a interoperabilidade entre diferentes plataformas e a aplicação de técnicas de inteligência artificial para a extração de conhecimento a partir dos dados capturados.

Portanto, o banco de dados na automação industrial não é apenas um repositório de informações, mas sim um componente estratégico que influencia diretamente a qualidade, a eficiência e a inovação nos processos produtivos.

2.4.1 Características de Bancos de Dados na Automação Industrial

A escolha do sistema de gerenciamento de banco de dados (SGBD) em ambientes industriais precisa considerar requisitos rigorosos de desempenho, escalabilidade, confiabilidade e integração com outros sistemas. Conforme destacado por Silva, os bancos de dados industriais devem ser capazes de operar continuamente, suportando falhas sem perda de dados e oferecendo respostas rápidas às consultas realizadas pelos sistemas de supervisão e controle.

Entre as características essenciais de bancos de dados voltados à automação, destacam-se:

- **Alta Disponibilidade:** A continuidade das operações industriais depende da capacidade do banco de dados em se manter ativo e acessível mesmo em casos de falhas parciais de hardware ou software.
- **Baixa Latência:** Em aplicações de tempo real, é fundamental que o banco de dados responda rapidamente às operações de leitura e escrita.
- **Escalabilidade:** A capacidade de expandir o sistema para acomodar o aumento no volume de dados é essencial, especialmente em ambientes com múltiplos sensores e dispositivos conectados.
- **Segurança:** A proteção dos dados industriais contra acessos não autorizados e ataques cibernéticos é crítica para a integridade e a confiabilidade dos processos.
- **Compatibilidade e Integração:** A facilidade de integrar o banco de dados a diferentes plataformas e protocolos industriais facilita a interoperabilidade e reduz a complexidade dos sistemas.

Segundo Raptis, o surgimento da Indústria 4.0 ampliou ainda mais esses requisitos, introduzindo a necessidade de bancos de dados que suportem arquiteturas distribuídas, big data

e análise em tempo real, com a utilização de soluções híbridas que combinem armazenamento local e em nuvem.

2.4.2 Características de Bancos de Dados Específicos

Diversos sistemas de gerenciamento de banco de dados (SGBD) são utilizados na automação industrial, cada um com suas características próprias e adequação a diferentes necessidades. A seguir, são apresentados três dos bancos mais empregados no setor: *Microsoft SQL Server*, *Oracle Database* e *PostgreSQL*.

- **Microsoft SQL Server:** O Microsoft SQL Server é um SGBDR desenvolvido pela Microsoft, conhecido por sua integração com o ecossistema Windows e ferramentas como o Microsoft Azure. Utiliza a linguagem Transact-SQL (T-SQL) para manipulação de dados e oferece recursos como Always On Availability Groups para alta disponibilidade. É amplamente adotado em ambientes corporativos que utilizam tecnologias Microsoft, oferecendo uma curva de aprendizado moderada e suporte técnico abrangente (ENTERPRISEDB, 2025) (FUTURAMO, 2025).

Também é amplamente utilizado em aplicações industriais devido à sua robustez, segurança e integração com o ecossistema Microsoft. De acordo com o site DB-Engines (DB-Engines, 2025), o SQL Server é consistentemente classificado como um dos bancos de dados mais populares do mundo, evidenciando sua confiabilidade para aplicações industriais críticas.

- **Oracle Database:** O Oracle Database também é um sistema de gerenciamento de banco de dados relacional (SGBDR) proprietário, amplamente utilizado em ambientes corporativos de grande escala. Oferece recursos avançados como Oracle Real Application Clusters (RAC) para alta disponibilidade, suporte a replicação multimaster e particionamento de dados eficiente. Sua linguagem procedural PL/SQL permite a criação de procedimentos armazenados complexos. Apesar de seu alto custo de licenciamento, é preferido por organizações que necessitam de desempenho robusto, escalabilidade e suporte técnico especializado (FUTURAMO, 2025) (BAIRESDEV, 2025).

Também destaca-se por sua alta capacidade de processamento, escalabilidade e segurança. De acordo com DB-Engines (DB-Engines, 2025), o Oracle é considerado uma das soluções líderes no mercado para grandes corporações, especialmente em sistemas críticos que exigem desempenho e confiabilidade elevados.

- **PostgreSQL:** O PostgreSQL assim como os outros, também é um SGBDR, porém é de código aberto, reconhecido por sua conformidade com os padrões SQL e extensibilidade. Suporta uma ampla variedade de tipos de dados, incluindo JSON, XML e geoespaciais, e permite a criação de tipos de dados personalizados. Utiliza o controle de concorrência multiversão (MVCC) para gerenciar transações simultâneas de forma eficiente. Sua

comunidade ativa contribui para melhorias contínuas, tornando-o uma escolha popular para aplicações que requerem flexibilidade e desempenho (ENTERPRISEDB, 2025) (BAIRESDEV, 2025).

Também é amplamente adotado na automação industrial devido à sua flexibilidade, estabilidade e aderência aos padrões SQL. Segundo o DB-Engines (DB-Engines, 2025), o PostgreSQL tem demonstrado crescimento constante em popularidade, sendo reconhecido como uma opção viável para empresas que buscam uma solução robusta e escalável sem custos de licenciamento.

2.4.3 Comparação Geral e Justificativa para a Escolha do PostgreSQL

Para ilustrar a comparação entre os bancos de dados, segue uma tabela com todas as informações indicadas anteriormente e todos os dados disponíveis nas referências bibliográficas utilizadas.

Aspecto	Oracle	SQL Server	PostgreSQL
Licenciamento	Proprietário	Proprietário	Código Aberto
Suporte a Interfaces	Amplio	Excelente c/ Microsoft	Amplio e flexível
Alta Disponibilidade	RAC e Data Guard	Always On	Replic. física/lógica
Suporte a JSON/XML	Sim (mais XML)	Sim	Avançado
Particion. de Dados	Completo	Automático (limitado)	Declarativo

Tabela 1 – Comparação entre Oracle, Microsoft SQL Server e PostgreSQL

A comparação entre os bancos de dados evidencia que tanto o *Microsoft SQL Server* quanto o *Oracle Database* oferecem recursos avançados e elevada confiabilidade, mas seus custos de licenciamento e a complexidade de administração podem ser entraves, especialmente em projetos com restrições orçamentárias. Em contrapartida, o *PostgreSQL* destaca-se como uma solução open-source que combina robustez, escalabilidade e flexibilidade, proporcionando redução de custos e facilidade de personalização.

Portanto, com base nos dados apresentados pelas fontes de referência, a escolha do *PostgreSQL* para este projeto de automação industrial é justificada pela sua viabilidade econômica.

Contudo, a eficiência de um banco de dados como o PostgreSQL depende diretamente da forma como o software é construído para interagir com ele. A implementação de rotinas de armazenamento, consulta e tratamento das informações exige uma linguagem de programação capaz de lidar tanto com a complexidade da comunicação com o banco quanto com os requisitos de desempenho e interface do sistema SCADA. Dessa forma, a definição da linguagem utilizada torna-se um elemento estratégico do projeto, justificando a análise das opções disponíveis e a escolha da plataforma mais adequada para o desenvolvimento do sistema.

2.5 LINGUAGEM DE PROGRAMAÇÃO PARA O DESENVOLVIMENTO DO SISTEMA SCADA

A escolha da linguagem de programação é um fator determinante para o desenvolvimento eficiente de um sistema SCADA (Supervisory Control and Data Acquisition). Aspectos como integração com hardware, suporte a interfaces gráficas, disponibilidade de bibliotecas, manutenção do código e produtividade do desenvolvedor influenciam diretamente o resultado final. Além disso, o contexto de execução — neste caso, um ambiente Windows — exige uma linguagem com ferramentas e suporte nativos que simplifiquem o fluxo de desenvolvimento e reduzam o tempo necessário para validar funcionalidades. Entre as opções amplamente utilizadas em sistemas industriais, destacam-se linguagens como o C++, Python e C#, cada uma com características distintas.

Diversas linguagens oferecem recursos suficientes para o desenvolvimento de sistemas SCADA, porém com abordagens e capacidades distintas. O C++ é reconhecido pelo alto desempenho e controle fino de memória, sendo amplamente utilizado em aplicações industriais onde a otimização é essencial (DANTAS, 2021). Contudo, sua complexidade e a dificuldade de manutenção tornam o desenvolvimento mais demorado.

O Python, por sua vez, destaca-se pela facilidade de escrita e pela grande quantidade de bibliotecas disponíveis, além de oferecer um ecossistema rico para automação e integração com dispositivos (FLUMINENSE, 2021; FREITAS, 2023). Entretanto, por ser interpretado, tende a apresentar desempenho inferior em operações que exigem comunicação contínua e em tempo quase real, o que pode ser um fator limitante em sistemas SCADA.

Já o C#, apoiado pela plataforma .NET, apresenta um equilíbrio entre desempenho, facilidade de desenvolvimento e suporte a interfaces gráficas, além de possuir bibliotecas maduras adequadas a aplicações industriais (RESEARCHGATE, 2020; SOUZA, 2023). Essa combinação o torna uma opção robusta e prática para o desenvolvimento de sistemas que exigem responsividade e integração com múltiplos módulos de comunicação.

2.5.1 Justificativa para a Escolha do C#

Embora a análise comparativa seja relevante, a escolha final do C# neste projeto foi guiada principalmente por fatores práticos e estratégicos relacionados ao desenvolvimento. A familiaridade prévia do autor com a linguagem, com as ferramentas do ecossistema .NET e com a plataforma Windows foi determinante para garantir um desenvolvimento mais ágil, seguro e consistente dentro do tempo limitado disponível para o projeto. O domínio prévio do ambiente e das bibliotecas reduziu significativamente a curva de aprendizado e permitiu concentrar os esforços na arquitetura do sistema e na implementação das funcionalidades centrais do SCADA.

Além disso, o C# oferece um conjunto robusto de recursos que facilitam a construção de

interfaces gráficas estáveis e responsivas, fator essencial para a visualização de dados em tempo real, monitoramento de estados e interação com o operador. A disponibilidade de bibliotecas consolidadas para comunicação industrial, como Modbus via *NModbus*, e a integração nativa com bancos de dados relacionais como o *PostgreSQL*, também contribuíram para assegurar uma implementação eficiente dos módulos de comunicação e armazenamento.

Considerando a natureza aplicada do projeto, a necessidade de produzir um sistema funcional e testável, e as restrições de tempo e recursos, o C# mostrou-se a opção mais eficiente do ponto de vista técnico e prático. A escolha favorece não apenas o desenvolvimento atual, mas também a expansão futura do sistema, garantindo facilidade para incorporar novas IOs, protocolos ou funcionalidades conforme as demandas de aplicações reais.

Uma vez definida a linguagem de desenvolvimento e estabelecida a base tecnológica do sistema, torna-se fundamental compreender como o software se conecta ao mundo físico. Em sistemas SCADA, essa integração ocorre por meio de protocolos de comunicação capazes de transmitir comandos e adquirir dados de equipamentos, sensores e controladores industriais. Assim, a próxima etapa envolve analisar os métodos e padrões que possibilitam essa troca de informações entre o ambiente supervisão implementado em C# e os dispositivos de campo, garantindo confiabilidade, precisão e desempenho durante a operação.

2.6 COMUNICAÇÃO ENTRE SISTEMAS SUPERVISÓRIOS E HARDWARE

A comunicação entre sistemas supervisórios e hardware de campo é um dos pilares fundamentais para a implementação eficiente de sistemas SCADA (Supervisory Control and Data Acquisition). A correta integração entre o software supervisão e dispositivos como CLPs (Controladores Lógicos Programáveis), sensores, atuadores e instrumentos de medição garante a confiabilidade, a precisão e a robustez necessárias para o monitoramento e controle de processos industriais.

Os sistemas supervisórios dependem de protocolos de comunicação padronizados para enviar e receber dados de forma consistente e segura. Esses protocolos podem operar sobre diferentes tipos de redes, como redes seriais tradicionais (RS-232, RS-485) ou redes Ethernet industriais modernas. A escolha do protocolo adequado impacta diretamente a velocidade de resposta, a integridade dos dados, a escalabilidade do sistema e a compatibilidade com equipamentos de diferentes fabricantes.

No contexto industrial, a diversidade de protocolos disponíveis reflete a necessidade de atender a diferentes requisitos de aplicação, como taxa de atualização, tolerância a falhas, interoperabilidade e facilidade de configuração. Protocolos como Modbus, Ethernet/IP, PROFINET e UDP são amplamente utilizados, cada um apresentando características específicas que influenciam sua aplicabilidade em determinados cenários (FS, 2025).

Segundo a fonte (MODBUSTOOLS, 2025), a seleção do protocolo de comunicação é uma

etapa crítica no projeto de sistemas industriais, pois define as capacidades de coleta e envio de dados em tempo real, além de determinar o grau de abertura e integração do sistema com futuros upgrades tecnológicos. Além disso, a escolha afeta diretamente aspectos como a segurança cibernética, um tema cada vez mais relevante na automação moderna.

Dessa forma, torna-se essencial compreender as diferenças entre os protocolos de comunicação mais utilizados para que se possa adotar a solução que melhor atenda aos objetivos de desempenho, custo, manutenção e expansão do sistema SCADA.

2.6.1 Comparação entre os Protocolos Ethernet/IP, PROFINET e Modbus

A escolha do protocolo de comunicação adequado é crucial para a integração eficaz de sistemas supervisórios com dispositivos de campo. A seguir, uma comparação entre os protocolos Ethernet/IP, PROFINET e Modbus, incluindo os dispositivos onde esses protocolos são comumente encontrados.

- **Ethernet/IP:** O Ethernet/IP é um protocolo amplamente utilizado para a comunicação em redes industriais. Ele é baseado na tecnologia Ethernet, permitindo a integração com diversos dispositivos industriais e suportando tanto redes de controle quanto de informações. O protocolo é altamente flexível e ideal para ambientes de automação que exigem alta escalabilidade e confiabilidade. Ele é amplamente utilizado em equipamentos como CLPs, IHM (Interfaces Homem-Máquina) e sensores, e está presente em diversos fabricantes de automação industrial, como Allen-Bradley e Schneider Electric (ENVISIA, 2022).
- **PROFINET:** PROFINET é um protocolo também baseado em Ethernet, mas focado em automação industrial em tempo real. Desenvolvido pela Siemens, o PROFINET oferece alta performance e confiabilidade, sendo ideal para sistemas que exigem baixa latência e alta disponibilidade. O PROFINET é amplamente utilizado em CLPs Siemens, como a linha S7, e em outros equipamentos industriais de alta performance, como robôs e sistemas de controle de movimento, encontrados em automações de processos e fabricação (PROFINET, 2025). Sua principal aplicação está em ambientes industriais que demandam uma comunicação rápida e precisa, como sistemas de controle de máquinas e processos de manufatura.
- **Modbus:** Modbus é um dos protocolos de comunicação mais antigos e simples, frequentemente utilizado em sistemas de automação mais simples ou em dispositivos com requisitos de comunicação menos exigentes. Embora o Modbus TCP seja mais moderno, sua versão serial (Modbus RTU) ainda é amplamente usada em redes industriais. O Modbus está presente na maioria dos CLPs e dispositivos de campo de diversos fabricantes, como

Siemens, Schneider Electric, ABB e Mitsubishi. É comumente encontrado em equipamentos como sensores, atuadores e medidores, sendo um protocolo acessível e fácil de implementar, ideal para automações de pequeno e médio porte (MODBUSTOOLS, 2025).

Para facilitar a análise das diferenças entre os protocolos Ethernet/IP, PROFINET e Modbus, a tabela 2 resume suas principais características em termos de desempenho e os tipos de dispositivos comumente utilizados. Essa comparação permite uma visão clara das vantagens e desvantagens de cada protocolo, ajudando na escolha da solução mais adequada para a integração entre sistemas supervisórios e hardware em diferentes cenários industriais.

Característica	Ethernet/IP	PROFINET	Modbus
Desemp. (round-trip)	31,25 μ s a 4 ms	1 a 100 ms	10 a 200 ms
Uso Comum	Sistemas distribuídos	Autom. em tempo real	Sistemas simples
Hardwares Comuns	CLPs, IHM, Sensores	CLPs Siemens, Robôs	Hardwares div.

Tabela 2 – Comparação entre Ethernet/IP, PROFINET e Modbus (CONTROLDESIGN, 2025), (PROCENDEC, 2023), (AUTOMATION, 2021), (ODVA, 2025), (Siemens, 2025), (Modbus Organization, 2023)

2.6.2 Escolha do Protocolo Modbus

A escolha do protocolo Modbus para a comunicação entre o sistema supervisório e os dispositivos de campo é motivada principalmente pela sua ampla compatibilidade com diversos tipos de hardware industrial. O Modbus é um dos protocolos mais utilizados em sistemas de automação industrial, presente em praticamente todos os Controladores Lógicos Programáveis (CLPs) e dispositivos de medição, sensores e atuadores. Sua versatilidade permite que diferentes equipamentos, muitas vezes de fabricantes distintos, se integrem facilmente.

Além disso, o Modbus é amplamente adotado devido à sua simplicidade, estabilidade e facilidade de implementação. Por ser um protocolo aberto, sem a necessidade de licenciamento, ele facilita a adoção por diversas empresas e fornecedores de equipamentos.

Para a maior parte das aplicações industriais, a comunicação Modbus atende adequadamente às exigências de desempenho, sendo capaz de garantir a transmissão de dados necessários, sem a necessidade de protocolos mais complexos, que poderiam ser excessivos para muitas situações.

Historicamente, o Modbus operava sobre barramentos seriais, como RS-232 e RS-485, atendendo à maioria das aplicações industriais simples. Entretanto, em sistemas SCADA modernos, que exigem integração via redes Ethernet e comunicação eficiente com múltiplos dispositivos simultaneamente, tornou-se vantajoso utilizar o Modbus TCP/IP. Essa versão mantém os benefícios do Modbus tradicional, como simplicidade e compatibilidade, enquanto possibilita a integração em redes industriais modernas de forma confiável e escalável.

Portanto, a escolha do Modbus visa atender a demanda por uma solução com a garantia de que o sistema será capaz de se comunicar com uma grande diversidade de dispositivos industriais, sem as limitações de compatibilidade que outros protocolos poderiam apresentar.

2.6.3 Estrutura e Funcionamento do Protocolo Modbus TCP/IP

O Modbus TCP/IP é uma extensão do protocolo Modbus tradicional que opera sobre redes Ethernet, utilizando a pilha de protocolos TCP/IP para transmissão de dados. Ele adota uma arquitetura baseada no modelo cliente-servidor, onde o cliente (geralmente o sistema supervisor) envia requisições e o servidor (como um CLP) responde às demandas (MODBUSTOOLS, 2025).

Cada transação Modbus TCP/IP é composta por uma estrutura denominada Unidade de Dados de Aplicação (ADU), que inclui dois componentes principais: o *Modbus Application Protocol Header* (MBAP) e a *Protocol Data Unit* (PDU). O cabeçalho MBAP contém informações essenciais para a comunicação, como:

- **Transaction Identifier:** identifica exclusivamente cada requisição.
- **Protocol Identifier:** valor fixo (0x0000) que indica o protocolo Modbus.
- **Length:** especifica o número de bytes da mensagem restante.
- **Unit Identifier:** identifica o dispositivo escravo destino da mensagem.

A PDU contém o código da função requisitada e os dados associados. Os códigos de função indicam qual operação será executada no servidor, como leitura de registradores ou escrita de bobinas. Entre os códigos mais utilizados, destacam-se:

- **0x01** – Leitura de bobinas digitais.
- **0x03** – Leitura de registradores de retenção.
- **0x05** – Escrita de uma única bobina.
- **0x10** – Escrita múltipla de registradores.

A operação é síncrona e baseada em requisição-resposta: o cliente envia uma solicitação e aguarda a resposta do servidor antes de iniciar uma nova transação. A utilização do protocolo TCP, orientado à conexão, garante a integridade, ordem e confiabilidade na entrega dos pacotes de dados, características fundamentais em ambientes industriais (INSTRUMENTS, 2025).

Apesar de suas vantagens, o Modbus TCP/IP carece de mecanismos nativos de segurança, como criptografia ou autenticação. Por isso, em aplicações críticas, é recomendável utilizar técnicas complementares como firewalls industriais, VPNs ou encapsulamento em protocolos seguros, como TLS (MODBUSTOOLS, 2025).

A comunicação em sistemas industriais envolve aspectos elétricos essenciais para garantir o funcionamento adequado das redes de dados, especialmente em protocolos como o Modbus TCP/IP. A escolha dos cabos, conectores e a proteção contra interferências são fundamentais para garantir uma comunicação eficiente e estável.

2.6.4 Infraestrutura de Rede

Para a implementação de sistemas SCADA modernos, como no caso deste trabalho, a escolha da infraestrutura elétrica para suportar a rede Ethernet é crucial. Redes industriais frequentemente utilizam cabeamento tipo *Ethernet* (cabo *CAT5*, *CAT6*, ou *CAT7*) para garantir uma transmissão de dados eficiente e confiável. O cabeamento precisa ser capaz de suportar altas velocidades de transmissão e minimizar a interferência dos ruídos elétricos comuns em ambientes industriais (ENVISIA, 2022).

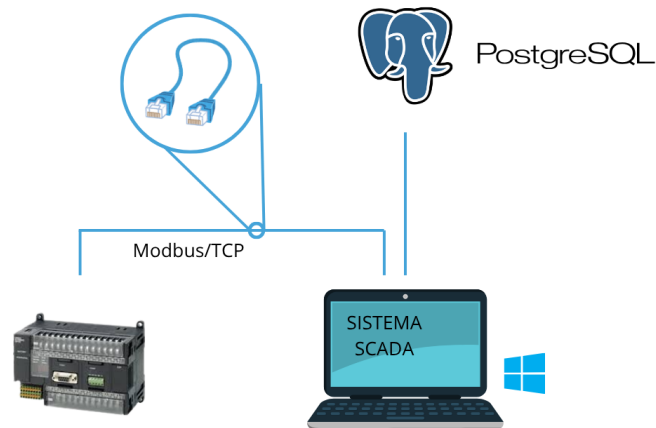
As conexões de rede, geralmente feitas através de *switches* industriais, precisam garantir que os dados sejam transmitidos sem perdas. Esses *switches* são projetados para resistir a condições ambientais adversas, como variações de temperatura e vibrações mecânicas, que são comuns em fábricas e ambientes industriais. A utilização de Ethernet industrial é vantajosa, pois, além de ser uma tecnologia amplamente conhecida, ela é capaz de suportar longas distâncias e garantir a comunicação entre múltiplos dispositivos conectados a uma rede local (LAN) (FS, 2025).

Além da escolha adequada de cabeamento e dispositivos de rede, é essencial considerar parâmetros de desempenho como *latência* e *jitter*, que influenciam diretamente na qualidade da comunicação em redes industriais. A *latência* refere-se ao tempo que um pacote de dados leva para ir de um ponto a outro da rede, sendo um fator crítico em aplicações em tempo real, como sistemas SCADA. Já o *jitter* representa a variação da latência ao longo do tempo, o que pode causar atrasos imprevisíveis na entrega de dados. Altos níveis de jitter e latência podem comprometer a sincronização entre o sistema supervisório e os dispositivos de campo, prejudicando a coleta de dados em tempo real e a execução de comandos. Por isso, recomenda-se o uso de dispositivos com suporte a QoS (Quality of Service) e a segmentação da rede, quando necessário, para garantir um tráfego de dados determinístico e estável, essencial para a operação confiável do sistema (Amazon Web Services, 2025) (Obkio Inc., 2021) (Auvik Networks, 2020).

A Figura 2 ilustra a infraestrutura de rede proposta neste trabalho, representando a comunicação entre o computador que executa o sistema SCADA, o CLP conectado via protocolo

Modbus/TCP e o banco de dados relacional. Todos os dispositivos estão interligados por meio de uma rede Ethernet com cabeamento físico, assegurando a integridade da comunicação e o desempenho adequado do sistema.

Figura 2 – Diagrama da rede de conexões do sistema SCADA proposto



Fonte: O Autor (2025).

Em ambientes industriais, os sinais de comunicação podem ser fortemente afetados por interferências eletromagnéticas (EMI), geradas principalmente por grandes motores, inversores de frequência e outros equipamentos de alta potência. Para minimizar esses efeitos, é essencial que os cabos utilizados sejam blindados (como os cabos *STP*, *Shielded Twisted Pair*) (SCRIBD, 2023). Esses cabos são projetados para proteger os sinais contra interferências externas, garantindo que a comunicação de dados seja preservada durante a transmissão.

Além disso, dispositivos de rede, como *switches* e *routers*, devem ser instalados em gabinetes blindados para reduzir ainda mais os efeitos de EMI. A configuração elétrica deve ser cuidadosamente planejada para garantir que a rede possa operar de forma estável, mesmo em locais com alta carga elétrica ou fontes de ruído.

2.6.5 Considerações de Desempenho e Alternativas ao Modbus

Em determinadas situações de teste, especialmente aquelas que exigem uma elevada taxa de atualização de dados (abaixo de 10 ms), o protocolo Modbus TCP/IP pode não atender plenamente aos requisitos de desempenho devido à sua natureza de comunicação mestre-escravo. Para contornar essa limitação, é possível desenvolver parte do teste internamente no dispositivo CLP integrado com o sistema e apenas ativá-lo através da comunicação Modbus, onde, posteriormente, é possível recolher o resultado do mesmo.

Alternativamente, pode-se isolar a lógica de execução crítica desses testes com alta velocidade em uma chamada externa via linha de comando, onde a mesma pode realizar comunicações alternativas com os dispositivos integrados ao sistema SCADA.

Diante das limitações de desempenho do Modbus TCP/IP em aplicações que exigem elevada taxa de atualização de dados, torna-se necessário adotar mecanismos alternativos para execução de tarefas críticas. Uma abordagem eficiente é a execução de chamadas externas via linha de comando, permitindo que processos de alta velocidade ou lógicas específicas sejam realizados independentemente da comunicação principal do SCADA, garantindo precisão, flexibilidade e modularidade.

2.7 EXECUÇÃO DE CHAMADAS EXTERNAS VIA LINHA DE COMANDO NO SISTEMA SCADA

A integração de chamadas externas via linha de comando no sistema SCADA desenvolvido neste trabalho visa ampliar as possibilidades de automação, flexibilização de rotinas e personalização de testes. Esse recurso permite que o sistema execute qualquer aplicação ou script externo disponível no sistema operacional, desde arquivos batch e executáveis até scripts em linguagens interpretadas como `Python`, `PowerShell` ou mesmo comandos individuais do `CMD`. Essa abordagem oferece grande versatilidade, ao desvincular a lógica principal do SCADA das tarefas específicas de processamento, coleta ou análise de dados, facilitando a modularidade e a manutenção do sistema.

No contexto da Indústria 4.0, essa capacidade de execução externa é especialmente relevante, uma vez que promove a integração com diferentes ferramentas e plataformas, mesmo que estas não compartilhem o mesmo ambiente de desenvolvimento ou linguagem de programação. Ao permitir que o sistema SCADA acione recursos externos via linha de comando, é possível adaptar rapidamente o sistema a novos requisitos produtivos ou operacionais, sem necessidade de reprogramação da interface principal do supervisor.

Com esse mecanismo, um engenheiro ou técnico pode, por exemplo, preparar scripts ou ferramentas específicas para realizar testes, validações, diagnósticos ou até integrações com bancos de dados e sistemas externos. Esses scripts podem ser mantidos e atualizados separadamente do sistema SCADA, promovendo uma arquitetura mais limpa, sustentável e facilmente extensível.

2.7.1 Vantagens da Arquitetura com Execução Externa

A adoção de chamadas externas via `CMD` proporciona diversas vantagens práticas e estruturais:

- **Independência de linguagem:** permite o uso de qualquer linguagem ou ferramenta compatível com o sistema operacional;
- **Modularidade:** os scripts podem ser mantidos de forma independente do núcleo do sistema SCADA;

- **Flexibilidade:** facilita a criação de fluxos personalizados de teste, inspeção e controle;
- **Integração facilitada:** amplia a capacidade de comunicação com sistemas legados, bancos de dados e dispositivos de terceiros;
- **Escalabilidade:** permite a extensão do sistema sem necessidade de alterações profundas em sua arquitetura.

Essa arquitetura torna o sistema SCADA especialmente adequado para ambientes industriais que requerem agilidade na adaptação a novos processos, reduzindo o tempo de resposta às mudanças e promovendo a interoperabilidade entre sistemas heterogêneos.

Entretanto, para que essa capacidade técnica se traduza em eficiência operacional e segurança no ambiente de testes, é necessário que o operador disponha de uma interface capaz de apresentar, contextualizar e controlar essas execuções de maneira clara e padronizada. Nesse sentido, a Interface de Execução assume papel central, funcionando como o elo entre as rotinas internas do software e a interação direta do usuário. A próxima seção detalha como essa interface é estruturada com base em normas consolidadas, assegurando ergonomia, consistência e visibilidade adequada das informações durante a operação do sistema.

2.8 INTERFACE DE EXECUÇÃO

A interface de execução do sistema SCADA é o ponto de interação principal entre os usuários—operadores, supervisores e administradores—e o sistema durante a realização de testes ou ajustes operacionais. Para garantir uma experiência de uso eficiente, segura e padronizada, o design da interface segue diretrizes estabelecidas por normas internacionais, como a ISA-101 e a ISO 11064.

A norma ISA-101, intitulada "Human-Machine Interfaces for Process Automation Systems", fornece um framework abrangente para o design, implementação e manutenção de Interfaces Homem-Máquina (IHMs) em sistemas de automação de processos. Ela enfatiza princípios como usabilidade, consistência e segurança, promovendo a criação de interfaces que minimizam erros operacionais e melhoram a consciência situacional dos usuários (ISA São Paulo, 2016).

Complementarmente, a ISO 11064 trata do design ergonômico de centros de controle, oferecendo recomendações para a disposição física dos elementos da interface, seleção de cores, tipografia e organização das informações, visando otimizar a interação humano-sistema (ARCADIS, 2016).

2.8.1 Normas Aplicadas ao Desenvolvimento de Sistemas SCADA

O desenvolvimento de sistemas SCADA requer aderência a normas técnicas que assegurem qualidade, confiabilidade, segurança e usabilidade, especialmente em ambientes industriais

sensíveis. Tais normas orientam desde a arquitetura e comunicação do sistema, até a concepção de interfaces homem-máquina (IHM) e aspectos ergonômicos de interação com o usuário. Neste trabalho, três normas foram utilizadas como base para fundamentar e direcionar o desenvolvimento do sistema SCADA proposto: a ISA-112, a ISA-101 e a ISO 11064.

A norma ISA-112 aborda as boas práticas na arquitetura e operação de sistemas SCADA, oferecendo diretrizes para gerenciamento de dados, comunicação e segurança. A ISA-101 foca na padronização de interfaces homem-máquina, garantindo clareza, eficiência e consistência visual. Já a ISO 11064 orienta o projeto ergonômico de centros de controle, com ênfase na usabilidade, conforto e eficiência do operador. As subseções a seguir detalham cada uma dessas normas e sua aplicabilidade no projeto desenvolvido.

- **Interface e Consistência Visual segundo a ISA-101:** A norma ISA-101 estabelece diretrizes claras para o projeto e a padronização de Interfaces Homem-Máquina (IHMs) em sistemas de automação de processos. Seu objetivo principal é melhorar a eficácia das interfaces, promovendo uma apresentação consistente das informações, facilitando a tomada de decisões pelos operadores e aumentando a consciência situacional (ISA São Paulo, 2016).

A norma orienta a utilização de gráficos de alta performance, padrões de cores neutras, hierarquia visual clara e navegação intuitiva. Esses princípios foram aplicados no desenvolvimento da interface deste sistema SCADA, justificando escolhas como a adoção de paleta de cores em tons de cinza, uso de ícones representativos e disposição organizada dos elementos na tela. Com isso, busca-se reduzir a sobrecarga cognitiva do usuário e minimizar erros operacionais, alinhando o projeto do sistema com as boas práticas recomendadas pela ISA-101.

- **Arquitetura e Boas Práticas segundo a ISA-112:** A norma ISA-112 fornece um conjunto abrangente de diretrizes para o desenvolvimento, operação e manutenção de sistemas SCADA modernos, com foco em garantir confiabilidade, escalabilidade e segurança. Essa norma estabelece melhores práticas para a arquitetura de sistemas, contemplando aspectos como comunicação entre dispositivos, gerenciamento de dados, redundância, e integração com outros sistemas de controle. Além disso, recomenda a implementação de níveis claros de segurança cibernética e de controle de acesso, alinhados às práticas de gerenciamento do ciclo de vida do sistema (NASBY, 2019).

No contexto deste projeto, a ISA-112 fundamenta as decisões arquiteturais adotadas, como a separação de níveis de usuários, a obrigatoriedade de autenticação no sistema e a integração com banco de dados para armazenamento seguro e rastreável das informações de teste e configuração. Dessa forma, assegura-se que o sistema SCADA desenvolvido atenda aos requisitos de robustez, rastreabilidade e interoperabilidade exigidos para ambientes industriais.

- **Ergonomia e Usabilidade segundo a ISO 11064:** A norma ISO 11064 trata do projeto ergonômico de centros de controle, abrangendo desde a organização física até a apresentação de informações nos sistemas de supervisão. Esta norma destaca a importância de projetar ambientes e interfaces que estejam alinhados às capacidades e limitações humanas, visando garantir segurança, eficiência e conforto (ARCADIS, 2016).

Entre suas recomendações estão a adoção de contrastes adequados, tipografia legível, disposição lógica das informações e otimização dos fluxos de trabalho. Tais diretrizes foram consideradas na concepção da interface deste sistema SCADA, refletindo-se no uso de fontes de tamanho elevado, contraste suficiente entre elementos e agrupamento funcional de informações. Essas medidas contribuem para reduzir a fadiga visual, melhorar a interação homem-máquina e aumentar a confiabilidade das operações de teste de produtos.

A definição de uma interface de execução clara, padronizada e aderente às normas de IHM estabelece a base para uma operação segura e eficiente do sistema SCADA proposto. Com a estrutura funcional consolidada — desde o gerenciamento de testes até a apresentação das informações em tempo real — torna-se possível avaliar como essa solução se posiciona diante das tecnologias e ferramentas já existentes no setor. Assim, a seguir é apresentada uma comparação com soluções já desenvolvidas, permitindo contextualizar o projeto no cenário atual da automação industrial.

2.9 COMPARAÇÃO COM SOLUÇÕES JÁ DESENVOLVIDAS

Para entender melhor o posicionamento do sistema proposto neste trabalho em relação às soluções já consolidadas no mercado, foram selecionadas duas ferramentas amplamente utilizadas na indústria: o **NI TestStand**, da National Instruments, e o **Keysight PathWave Test Automation**, da Keysight Technologies. Ambas são plataformas voltadas à automação de testes, com foco em facilitar o desenvolvimento e execução de sequências de teste em ambientes industriais, reduzindo a necessidade de programação por parte dos usuários finais.

- **NI TestStand:** O TestStand é uma plataforma de automação de testes projetada para permitir que engenheiros de teste criem, executem e implantem sequências de teste com pouca ou nenhuma codificação. Ele fornece uma interface gráfica intuitiva, na qual os usuários podem arrastar e soltar etapas de teste, configurar parâmetros e definir lógicas condicionais de forma visual (National Instruments, 2022).

A plataforma oferece suporte nativo a diversos padrões de comunicação com equipamentos de teste, como GPIB, USB, PXI, LAN e protocolos industriais. Além disso, possui integração com bancos de dados relacionais, permitindo o registro automático de resultados de teste, rastreabilidade e geração de relatórios (National Instruments, 2022).

Seu foco está em ambientes industriais e laboratoriais que demandam testes automatizados de alta confiabilidade, sendo ideal para empresas que desejam padronizar procedimentos e reduzir o tempo de desenvolvimento de sequências de teste (National Instruments, 2022).

- **Keysight PathWave Test Automation:** O PathWave Test Automation é a solução da Keysight para a orquestração de testes complexos em ambientes de validação de hardware. Assim como o TestStand, ele permite que engenheiros desenvolvam sequências de teste sem escrever código, utilizando uma interface baseada em fluxogramas e blocos funcionais (Keysight Technologies, 2022).

A plataforma oferece bibliotecas com suporte nativo a instrumentos da Keysight e de terceiros, permitindo a comunicação direta com os equipamentos. Ela também é compatível com bancos de dados para armazenamento dos resultados de teste, além de fornecer mecanismos para geração de relatórios, controle de versão de sequências e colaboração em equipe (Keysight Technologies, 2022).

Com foco em setores como telecomunicações, automotivo, aeroespacial e defesa, o PathWave Test Automation se destaca pela robustez, escalabilidade e integração com ecossistemas mais amplos de engenharia de testes (Keysight Technologies, 2022).

- **ScadaBR:** O ScadaBR é uma plataforma SCADA open-source amplamente utilizada no Brasil, com suporte a diversos protocolos industriais, incluindo Modbus/TCP, e integração com bancos de dados relacionais. Ele permite a criação de telas de supervisão, execução de scripts externos e implementação de níveis de acesso para diferentes perfis de usuário, oferecendo flexibilidade e modularidade em sistemas de automação industrial (ScadaBR, 2025).

Por ser open-source, o ScadaBR apresenta como vantagem o baixo custo de implementação e a possibilidade de personalização da aplicação de acordo com as necessidades do usuário. Sua documentação e comunidade ativa também facilitam a aprendizagem e o desenvolvimento de novas funcionalidades. Embora seja amplamente utilizado em projetos educativos e pequenas ou médias empresas, ele serve como uma referência consolidada de SCADA no Brasil, especialmente no contexto de integração com protocolos industriais e automação baseada em software livre (ScadaBR, 2025).

- **Sistema Proposto:** O sistema SCADA proposto neste trabalho apresenta características similares às soluções comerciais no que diz respeito à comunicação com hardwares via protocolo Modbus/TCP, integração com banco de dados *PostgreSQL* e execução de procedimentos automatizados. No entanto, diferentemente das ferramentas comerciais, ele é desenvolvido utilizando C#, com lógica personalizada e interface gráfica adaptada para o monitoramento e teste de produtos específicos.

Embora exija conhecimento técnico para configuração inicial, seu design permite a futura implementação de mecanismos de criação de receitas e testes por meio de interfaces gráficas, facilitando o uso por engenheiros de produção sem necessidade de conhecimentos avançados em programação. A arquitetura aberta e flexível o torna uma alternativa viável e de baixo custo para empresas com necessidade de soluções customizadas.

2.9.1 Tabela Comparativa

Tabela 3 – Comparação entre o sistema proposto, NI TestStand, Keysight PathWave e ScadaBR

Critério	Sistema Proposto	NI TestStand	Keysight PathWave	ScadaBR
Necessita programador?	Não	Não	Não	Parcialmente (configuração inicial)
Interface para testes	Sim (Sequenciador)	Sim (Sequenciador)	Sim (Fluxograma)	Sim (Telas de supervisão)
Comunicação com hardware	Modbus/TCP	GPIO, PXI, USB, LAN	SCPI, USB, LAN, GPIO	Mais de 20 protocolos
Integração a Banco de dados	PostgreSQL	SQL Server, Oracle, MySQL	Bancos Relacionais	Bancos Relacionais
Cliente alvo	Eng. de automação/testes/manufatura	Eng. de teste / integradores	Eng. de validação / QA	Eng. Automação / automação educativa

3 DESENVOLVIMENTO

O desenvolvimento do sistema SCADA constitui a etapa em que as definições apresentadas na proposta são efetivamente transformadas em implementação prática. Nesta fase, foram aplicados os requisitos levantados e as tecnologias selecionadas, organizando o projeto em módulos que englobam banco de dados, comunicação, execução de testes e interface gráfica. Cada módulo foi construído com o objetivo de atender aos critérios de desempenho, confiabilidade e usabilidade estabelecidos previamente, garantindo a integração entre hardware, software e processos de monitoramento.

Ao longo deste capítulo, são detalhados os procedimentos de implementação adotados, desde a modelagem do banco de dados até a construção das rotinas de comunicação e execução de testes. Também são descritos os recursos desenvolvidos na interface gráfica, buscando assegurar uma interação clara e eficiente entre o usuário e o sistema. Assim, o capítulo demonstra a evolução do projeto da concepção teórica para sua materialização em um protótipo funcional, consolidando as escolhas técnicas que fundamentam os resultados apresentados posteriormente.

3.1 FUNCIONAMENTO PRINCIPAL DO SOFTWARE

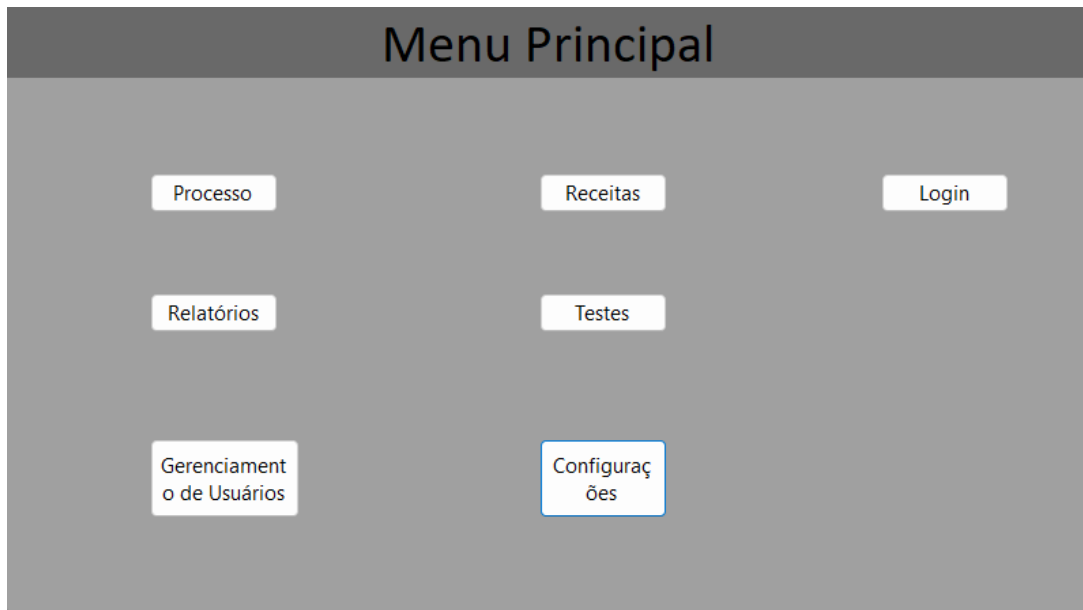
Este projeto é concebido de forma a proporcionar um elevado controle sobre os testes dos produtos, sem que seja necessária a intervenção constante de um programador ou engenheiro, mesmo em casos de alteração ou criação de novos produtos que passarão pela bancada de testes. A seguir, apresentam-se informações específicas sobre cada tela de configuração, montagem de testes e criação de receitas presentes no sistema.

3.1.1 Tela Principal

A figura 3 apresenta a tela principal planejada para o sistema, que é exibida ao iniciar o software na CPU. O design utiliza cores neutras (tons de cinza) para reduzir a fadiga visual e destacar informações críticas quando necessário. Ao interagir com os botões, o usuário é direcionado para a respectiva funcionalidade, mantendo uma navegação lógica e eficiente.

Para acessar o software, é necessário que o usuário realize login com uma conta previamente criada por um administrador. Na ausência de usuários cadastrados, o sistema gera automaticamente um usuário padrão "admin" com senha "admin" na primeira execução, permitindo o acesso inicial e posterior configuração personalizada. Após o login, o nome do usuário é exibido na tela inicial, reforçando a identificação e rastreabilidade das ações realizadas.

Figura 3 – Tela Inicial do Sistema SCADA Proposto



Fonte: O Autor (2025).

3.1.2 Níveis de Acesso

A definição de níveis de acesso é fundamental para assegurar a integridade e segurança operacional do sistema. Cada perfil de usuário possui permissões específicas, alinhadas às suas responsabilidades:

- **Operador:** Responsável pela execução dos testes de produtos, com acesso restrito às telas de Processo e Relatório.
- **Supervisor:** Possui acesso ampliado, incluindo as telas de Processo, Receitas, Relatórios, Testes e Configuração de Hardware, permitindo o controle e ajustes nos testes e configurações do sistema.
- **Administrador:** Detém acesso total ao sistema, incluindo todas as funcionalidades disponíveis para o Supervisor, além da tela de Gerenciamento de Usuários, permitindo a administração completa do sistema.

Os perfis de usuário no sistema SCADA são definidos e configurados na tela de **Gerenciamento de Usuários**, onde é possível atribuir permissões específicas a cada conta. As permissões determinam o acesso às diferentes funcionalidades do software, conforme detalhado nas seções que descrevem cada tela. Essa abordagem garante que somente usuários autorizados possam interagir com funcionalidades críticas, mantendo a segurança e integridade do sistema.

Essa estrutura hierárquica de acessos garante que apenas usuários autorizados possam realizar alterações críticas no sistema, conforme recomendado pelas normas ISA-101, ISA-112 e ISO 11064, promovendo uma operação segura e eficiente.

3.1.3 Configuração do Hardware

Conforme explicado anteriormente, no sistema proposto o hardware será de livre escolha do cliente, sendo necessário apenas que o equipamento suporte comunicação via Modbus TCP. A figura 4, apresenta a imagem da tela de configuração de hardware no software. Esta tela é acessível a partir da tela inicial do software (figura 3) e está disponível apenas para usuários com nível de Supervisor ou Administrador.

Figura 4 – Tela de Configuração do Hardware do Sistema SCADA Proposto

Configurações

Configurações de Comunicação

Endereço de IP: 192.168.0.2

Porta: 502

Tempo de Ciclo: 20

Salvar

Cancelar

Cadastrar Endereços

Configurações de Memórias

Endereços de Escrita

	Inicial	Final
Registrador	0	4
Coil	0	4

Endereços de Leitura

	Inicial	Final
Registrador	5	9
Coil	5	9

Fonte: O Autor (2025).

Para sair desta tela e retornar à tela inicial, o usuário deve clicar em *Salvar* para guardar as alterações ou em *Cancelar* para descartá-las.

Na configuração de comunicação, é solicitado apenas 3 dados para a execução da comunicação Modbus TCP. São eles:

- **Endereço de IP:** Endereço IP do hardware-alvo (escravo) para a comunicação Modbus TCP.
- **Porta:** Porta de comunicação do hardware-alvo (escravo).
- **Tempo de ciclo:** Intervalo, em milissegundos, entre os ciclos de escrita e leitura das memórias Modbus. Após cada ciclo de leitura e escrita, o sistema aguarda o tempo de ciclo antes de repetir a operação.

Nos campos de configurações de memórias, é necessário preencher os endereços das memórias utilizadas. O protocolo Modbus opera por meio de endereços de memória, conforme

descrito no capítulo de comunicações. As configurações solicitadas pelo software são as seguintes:

- **Endereços de Escrita — Registrador:** Endereço inicial e final das memórias de registradores que serão escritas.
- **Endereços de Escrita — Coil:** Endereço inicial e final das memórias booleanas que serão escritas.
- **Endereços de Leitura — Registrador:** Endereço inicial e final das memórias de registradores que serão lidas.
- **Endereços de Leitura — Coil:** Endereço inicial e final das memórias booleanas que serão lidas.

Caso alguma das funções, como a leitura de registradores, não seja necessária durante a execução, o usuário pode simplesmente configurar o valor da memória inicial para um endereço superior ao da memória final. Essa condição é interpretada pelo sistema como uma faixa inválida, fazendo com que o ciclo de comunicação **Modbus/TCP** ignore automaticamente essa operação.

Dentro da tela de configuração de hardware, há uma tela secundária para a configuração dos valores a serem lidos e escritos. A figura 5, apresenta a imagem desta tela.

Figura 5 – Tela de Cadastramento de Endereços do Sistema SCADA Proposto

Endereços de Escrita		Endereços de Leitura	
Registradores		Registradores	
Endereço	Nome	Endereço	Nome
0	writeReg1	5	readReg1
1	writeReg2	6	readReg2
2	writeReg3	7	readReg3
3	writeReg4	8	readReg4
Coils		Coils	
Endereço	Nome	Endereço	Nome
0	writeCoil1	5	readCoil1
1	writeCoil2	6	readCoil2
2	writeCoil3	7	readCoil3
3	writeCoil4	8	readCoil4

Voltar

Fonte: O Autor (2025).

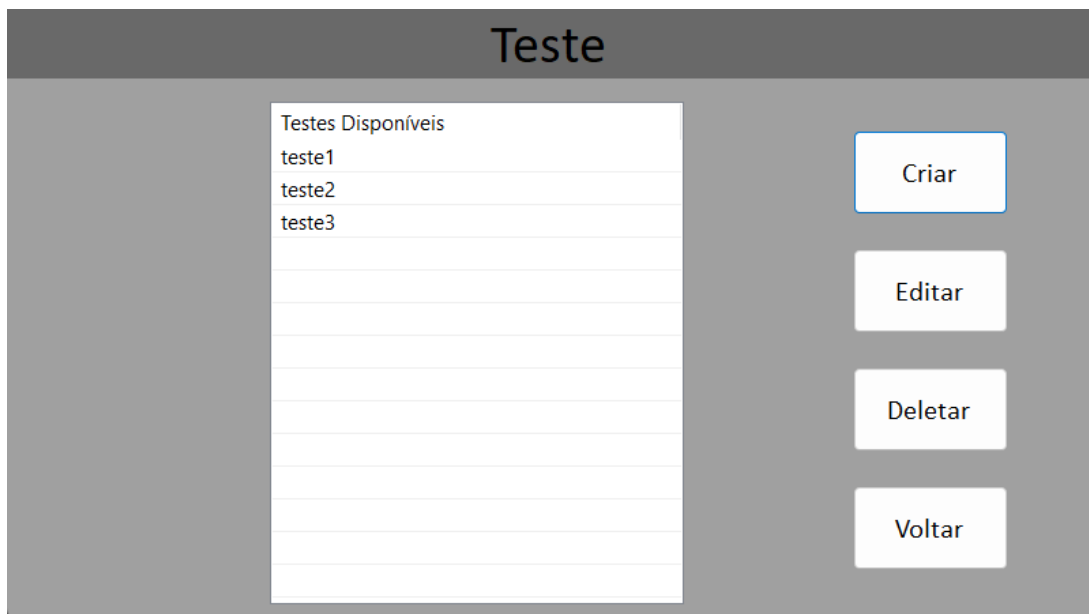
Nesta tela, o usuário define a nomenclatura de cada memória de entrada e saída de leitura e escrita, de forma a facilitar, posteriormente, a criação de testes e receitas. Caso o usuário não defina esses nomes, o próprio software os atribuirá automaticamente.

Para retornar à tela de configuração de hardware, basta clicar em *Voltar*.

3.1.4 Testes

A tela inicial de testes exibe uma lista simples com os testes previamente criados. A figura 6, apresenta a imagem desta tela. Ela pode ser acessada a partir da tela inicial do software (figura 3) e está disponível apenas para usuários com nível de Supervisor ou Administrador.

Figura 6 – Tela de Testes do Sistema SCADA Proposto



Fonte: O Autor (2025).

As ações disponíveis para o usuário nesta tela são:

- **Criar:** Criação de um novo teste.
- **Editar:** Editar um teste existente.
- **Deletar:** Deletar um teste disponível ao clicar no teste da lista e então clicar no botão *Deletar*.
- **Voltar:** Retornar para a tela anterior, no caso é a tela inicial do software.

3.1.4.1 Edição e Criação de Teste

Ao clicar em *Criar* ou *Editar*, a tela referente à figura 7 exibida:

Nesta tela, é possível criar um teste individual a ser executado no produto. Na figura 7, visualiza-se um fluxo de teste. Além dos comandos acessíveis, esta tela contém os seguintes elementos de interface:

retroceder, dependendo da configuração e necessidade do usuário. Caso o comando não possa falhar, a linha de erro dele sempre será 0. Caso a linha de erro não exista, ou não haja sequência de teste para continuar, o teste finaliza em falha.

Ex: Estamos executando a linha 5, ocorreu erro na execução, a linha de erro indicada no comando é 10, então o teste pula automaticamente a linha 6, 7, 8 e 9 e executa diretamente a 10.

Ex2: Executando a linha 5, ocorre erro. A linha de erro é 3. O teste retorna para a linha 3.

Os comandos disponíveis para a realização do teste são:

- **CheckInput:** O comando realiza a verificação de um valor lido de uma memória via comunicação **Modbus/TCP**. Essa verificação compara o valor da memória com limites mínimo e máximo definidos, aprovando o comando somente se o valor estiver dentro do intervalo especificado.

- *Parâmetro 1:* Memória a verificar.
- *Parâmetro 2:* Valor mínimo.
- *Parâmetro 3:* Valor máximo.

Obs.: Se a memória verificada for booleana, os valores mínimo e máximo devem ser iguais a 0 ou 1.

- **ChangeValue:** O comando altera o valor de uma memória que será escrita via comunicação **Modbus/TCP**.

- *Parâmetro 1:* Memória a alterar.
- *Parâmetro 2:* Novo valor.

Obs.: Se a memória alterada for booleana, o novo valor deve ser 0 ou 1.

- **VisualInspection:** O comando solicita ao usuário uma verificação visual, na qual ele informa manualmente o status do comando OK ou NOK. Durante essa etapa, o teste é pausado até que o usuário forneça a resposta.

- *Parâmetro 1:* Mensagem para apresentar.

- **DialogBox:** O comando exibe uma mensagem informativa ao usuário. O teste é pausado até que o operador confirme a leitura clicando no botão *OK*.

- *Parâmetro 1:* Mensagem para apresentar.

- **Delay:** O comando pausa a execução do teste por um tempo fixo, definido em milissegundos.

- *Parâmetro 1*: Tempo para aguardar em milissegundos.
- **ExecCmdLine**: O comando executa uma chamada externa via linha de comando. Durante a execução do comando, o teste é pausado.
 - *Parâmetro 1*: comando a ser executado.
- **End**: O comando finaliza o procedimento de teste, registrando o status como OK ou falha.
 - *Parâmetro 1*: Resultado do teste. 0 - Aprovado; 1 - Reprovado.

3.1.6 Receitas

A tela inicial de receitas é similar à tela de testes, exibindo uma lista de receitas previamente criadas. A figura 8, apresenta a imagem desta tela, acessível a partir da tela inicial do software (figura 3) e disponível apenas para usuários com nível de Supervisor ou Administrador.

Figura 8 – Tela de Receitas do Sistema SCADA Proposto



Fonte: O Autor (2025).

As ações disponíveis para o usuário nesta tela são:

- **Criar**: Criação de uma nova receita.
- **Editar**: Editar uma receita existente.
- **Deletar**: Excluir uma receita existente.
- **Voltar**: Retornar para a tela anterior, no caso é a tela inicial do software.

Para a interação com o usuário, os seguintes elementos estão disponíveis:

- **Receita de Teste:** Campo para informar o nome da receita a ser executada.
- **Tabela de Testes:** Lista com os testes pertencentes à receita e seus respectivos status.
- **Status do Teste:** Indicação do status geral da execução da receita.
- **Iniciar Testes:** Botão para iniciar a execução da receita selecionada.
- **Abortar:** Botão para interromper a execução da receita em andamento.
- **Voltar:** Botão para retornar à tela anterior do sistema.

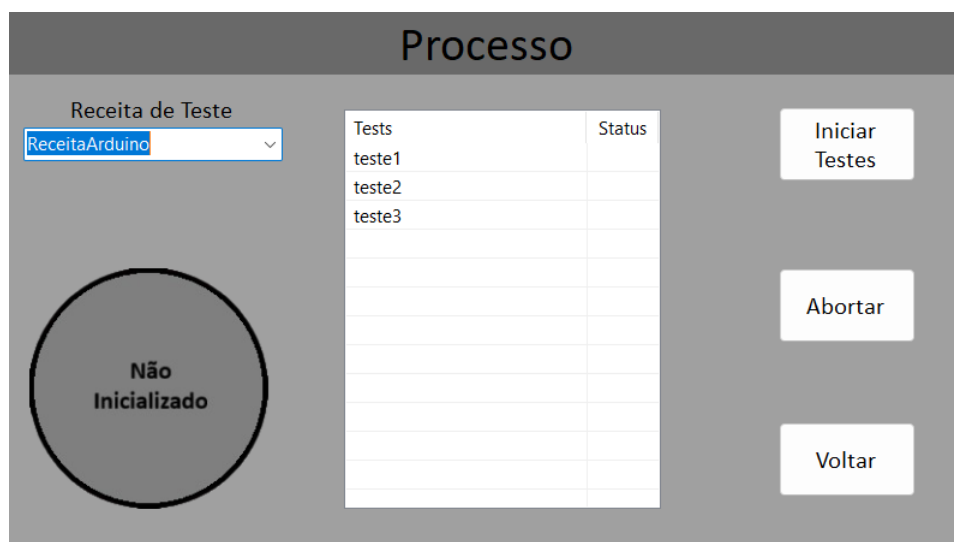
Durante a execução da receita, a interface é atualizada para que o operador visualize claramente os dados e os status relacionados ao teste.

Os status possíveis tanto para a receita quanto para cada teste individualmente são:

- **Aguardando:** Receita ou teste ainda não iniciado. Representado pela cor cinza.
- **Em Processo:** Receita ou teste em execução. Representado pela cor amarela.
- **Aprovado:** Execução concluída com sucesso. Representado pela cor verde.
- **Reprovado:** Execução concluída com falha. Representado pela cor vermelha.

Apresenta-se um exemplo visual da execução de uma receita na figura 10.

Figura 10 – Tela de Processo com receita em execução no Sistema SCADA Proposto



Fonte: O Autor (2025).

3.1.8 Gerenciamento de Usuários

A tela de gerenciamento de usuários permite controlar os acessos ao software, definindo quais usuários estão autorizados e quais permissões possuem. Essa interface contempla as operações: adicionar, deletar e editar usuários. Essa interface só pode ser acessada por usuários com nível de acesso de administrador.

- **Adicionar Usuário:** Essa funcionalidade permite cadastrar novos usuários para utilização do sistema. Os campos obrigatórios são:
 - **Usuário:** Nome de usuário a ser cadastrado.
 - **Senha:** Senha de acesso (digitada de forma oculta).
 - **Permissão:** Nível hierárquico de acesso.

Após preencher os dados, basta clicar em *Adicionar* para concluir o processo de cadastro.

Figura 11 – Tela de gerenciamento de usuário para adição de usuários no Sistema SCADA Proposto

A imagem mostra a interface de usuário para o gerenciamento de usuários. No topo, há um cabeçalho escuro com o título 'Gerenciador de Usuários'. Abaixo dele, o subtítulo 'Criar Usuario' está centralizado. À esquerda, há uma coluna vertical de botões: 'Criar' (destacado com uma borda azul), 'Editar', 'Deletar' e 'Voltar'. À direita, o formulário contém três campos: 'Usuario' (um menu suspenso), 'Senha' (um campo de texto com ícone de olho para alternar visibilidade) e 'Permissão' (um menu suspenso). Um botão 'Criar' largo está posicionado na base do formulário.

Fonte: O Autor (2025).

- **Editar Usuário:** Essa funcionalidade permite modificar os dados de usuários já cadastrados, como senha ou nível de permissão. O nome do usuário não pode ser alterado. Os campos disponíveis para edição são:
 - **Usuário:** Nome do usuário (somente leitura).
 - **Senha:** Nova senha de acesso (digitada de forma oculta).
 - **Permissão:** Novo nível hierárquico de acesso.

Figura 12 – Tela de gerenciamento de usuário para edição de usuário do Sistema SCADA Proposto

Gerenciador de Usuários

Editar Usuario

Criar

Editar

Deletar

Voltar

Usuario

Senha

Permissão

Editar

Fonte: O Autor (2025).

Após atualizar os dados desejados, clicar em *Editar* para concluir o processo.

- **Deletar Usuário:** Essa funcionalidade permite excluir usuários existentes, removendo seu acesso ao sistema. Deve-se preencher o campo correspondente ao nome do usuário. Em seguida, clicar em *Deletar* para concluir a operação.

Figura 13 – Tela de gerenciamento de usuário para exclusão de usuários no Sistema SCADA Proposto

Gerenciador de Usuários

Deletar Usuario

Criar

Editar

Deletar

Voltar

Usuario

Deletar

Fonte: O Autor (2025).

3.2 INFRAESTRUTURA DO BANCO DE DADOS

A infraestrutura do banco de dados foi projetada com base nos requisitos funcionais do sistema SCADA, buscando representar de forma clara as entidades envolvidas, seus atributos e relacionamentos. Além disso, sua estrutura foi definida de modo a permitir que múltiplas aplicações possam operar de forma independente sobre o mesmo servidor, compartilhando o banco de dados sem comprometer a integridade ou o desempenho do sistema. Essa abordagem garante maior flexibilidade e escalabilidade para futuras expansões da plataforma.

Para representar a estrutura lógica do banco, foi elaborado um diagrama Entidade-Relacionamento (ERD), que descreve como os diferentes elementos do sistema se conectam e interagem. Esse modelo serviu de base para a criação das tabelas e para a definição das chaves primárias e estrangeiras utilizadas no PostgreSQL.

O sistema possui cinco grandes blocos funcionais:

- **Configuração da Jiga:** Tabela *generic_scada_config*. Representa a configuração e o identificador de uma jiga.
- **Receitas:** Tabela *generic_scada_recipe*. Armazena as receitas de teste associadas a uma jiga. Representa a receita que será executada, organizando por linha e tipo de teste.
- **Testes:** Tabela *generic_scada_test*. Contém o sequenciamento de execuções dos testes e seus parâmetros.
- **Relatórios:** Tabela *generic_scada_report*. Gera um relatório consolidado da execução da receita. O campo **user** é uma cópia estática do username do usuário que está executando a receita, isso é proposital para auditoria histórica, pois mesmo se o usuário for renomeado, o relatório permanece intacto e fiel ao momento da execução.
- **Usuários:** Tabela *generic_scada_users*. Armazena os usuários do sistema. Realiza o controle de autenticação e autorização dos usuários para acesso às funções do software.

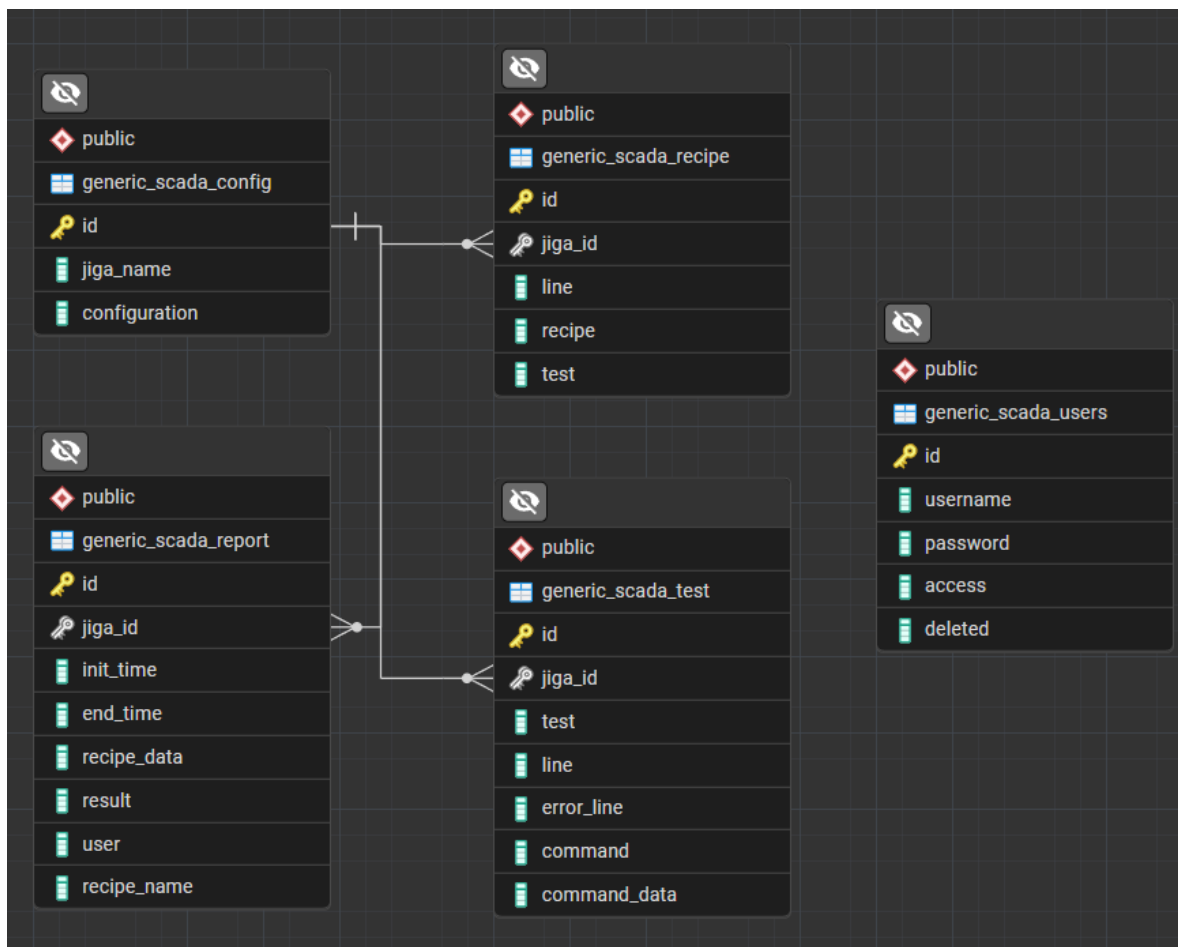
Uma jiga possui apenas uma configuração, mas possui diversas receitas, testes e relatórios. Os usuários são globais para todos as jigas no mesmo servidor.

A Figura 14 apresenta o diagrama Entidade-Relacionamento desenvolvido para o banco de dados do sistema SCADA.

3.3 CSV DE SEQUENCIAMENTO DE TESTES

O sistema SCADA desenvolvido permite a definição do sequenciamento dos testes por meio da importação de um arquivo no formato *CSV (Comma-Separated Values)*. Esse arquivo contém as informações necessárias para orientar a execução automatizada das etapas de teste de

Figura 14 – Diagrama Entidade Relação (DER)



Fonte: O Autor (2025).

cada produto, tornando o processo configurável sem a necessidade de modificar o código-fonte da aplicação.

O arquivo deve ser estruturado de forma padronizada, seguindo a ordem e o tipo de dados definidos pelo sistema. Cada linha representa um comando, enquanto as colunas descrevem os parâmetros necessários para sua execução, como identificadores, instruções, limites de medição e mensagens exibidas na interface.

A tabela deve conter as colunas na seguinte sequência:

- **Linha:** Identificador sequencial de execução, iniciando em 1 e incrementando de 1 em 1.
- **Comando:** Instrução a ser executada pelo sistema.
- **Erro:** Linha de redirecionamento em caso de falha na execução do comando. Caso ocorra erro, o sistema passa automaticamente para a linha indicada nesta coluna.
- **Param1:** Primeiro parâmetro associado ao comando.
- **Param2:** Segundo parâmetro associado ao comando.

- **Param3:** Terceiro parâmetro associado ao comando.
- **Param4:** Quarto parâmetro associado ao comando.

Além de seguir a sequência de colunas correta, o arquivo também deve atender aos seguintes critérios:

- O separador de campos deve ser o ponto e vírgula (;).
- A primeira linha do arquivo é reservada para os cabeçalhos das colunas e, portanto, não é processada pelo sistema.

Caso algum desses critérios não seja atendido, o arquivo poderá não ser processado corretamente pelo sistema.

Segue um exemplo de .csv para a verificação do teste de um botão:

Figura 15 – Exemplo de arquivo CSV

Linha	Comando	Erro	Param1	Param2	Param3	Param4
1	DialogBox	8	Clique e segure o botão branco			
2	Delay	8	2000			
3	CheckInput	8	botao_branco	1	1	
4	DialogBox	8	Solte o botão branco			
5	Delay	8	2000			
6	CheckInput	8	botao_branco	0	0	
7	End	7	0			
8	End	8	1			

Fonte: O Autor (2025).

3.4 ARQUITETURA DO CÓDIGO

A arquitetura do código desenvolvida para o sistema SCADA foi estruturada de forma modular, permitindo que cada parte responsável pela comunicação, processamento e armazenamento de dados opere de maneira independente, porém integrada. Esse modelo facilita tanto a manutenção quanto a escalabilidade do sistema, além de tornar o fluxo de execução mais claro e consistente.

Com o objetivo de apresentar formalmente a lógica empregada na implementação, esta seção descreve os principais algoritmos utilizados ao longo do desenvolvimento. Eles representam rotinas fundamentais, tais como o gerenciamento da comunicação Modbus/TCP, o acesso e manipulação de dados no banco PostgreSQL e o processamento interno de funções específicas do sistema.

A inclusão desses algoritmos tem papel importante no entendimento da solução proposta, pois evidencia a organização interna do software e demonstra a sequência lógica das

operações realizadas. Cada algoritmo é apresentado de forma isolada, acompanhado de sua respectiva descrição, permitindo compreender como os diferentes módulos interagem para garantir o funcionamento confiável e eficiente do SCADA.

3.4.1 Modbus/TCP

Para viabilizar a comunicação entre o sistema SCADA e os equipamentos de campo, foi implementado um módulo responsável pelo gerenciamento do protocolo Modbus/TCP. Esse módulo garante a abertura da conexão, o envio das requisições aos registradores configurados e o tratamento das respostas provenientes dos dispositivos.

Além disso, a implementação inclui mecanismos de verificação de integridade, controle de timeout e reconexão automática, assegurando que a coleta de dados ocorra de forma contínua e confiável.

O Algoritmo 1 apresenta a lógica principal utilizada no gerenciamento da comunicação Modbus/TCP dentro do sistema.

Algoritmo 1 – Gerenciamento da comunicação Modbus/TCP

```
1 using NModbus;
2 using NModbus.Extensions.Enron;
3 using System.Threading.Tasks;
4
5 public void manageConnection()
6 {
7     while (!closeManage)
8     {
9         if (connected)
10        {
11            try //Input Coil Read
12            {
13                if (inputCoils.nCoils > 0)
14                {
15                    inputCoils.coils = ReadCoilInputs(inputCoils.indexIni,
16                    inputCoils.nCoils);
17                }
18            }
19            catch (Exception ex)
20            {
21                error = true;
22                status = $"ERRO_{ex.Message}";
23                connected = false;
24                KeepTryingConnection();
25                break;
26            }
27        }
28    }
29 }
```

```

28
29     if (connected)
30     {
31         try //Output Coil Write
32         {
33             if(outputCoils.nCoils > 0)
34             {
35                 WriteCoilsValue(outputCoils.indexIni ,
36                 outputCoils.coils);
37             }
38         }
39         catch (Exception ex)
40         {
41             error = true;
42             status = $"ERRO_{ex.Message}";
43             connected = false;
44             KeepTryingConnection();
45             break;
46         }
47     }
48
49     if (connected)
50     {
51         try //Input Register Read
52         {
53             if(inputRegisters.nRegisters > 0)
54             {
55                 inputRegisters.register =
56                 ReadRegisterInputs(inputRegisters.indexIni ,
57                 inputRegisters.nRegisters);
58             }
59         }
60         catch (Exception ex)
61         {
62             error = true;
63             status = $"ERRO_{ex.Message}";
64             connected = false;
65             KeepTryingConnection();
66             break;
67         }
68     }
69
70     if (connected)
71     {
72         try //Output Register Write
73         {
74             if(outputRegisters.nRegisters > 0)

```

```

75         {
76             WriteRegistersValue ( outputRegisters . indexIni ,
77                 outputRegisters . register );
78         }
79     }
80     catch ( Exception ex )
81     {
82         error = true ;
83         status = $"ERRO_{ex.Message}";
84         connected = false ;
85         KeepTryingConnection ( ) ;
86         break ;
87     }
88 }
89 if ( ! connected )
90 {
91     KeepTryingConnection ( ) ;
92 }
93 Thread . Sleep ( 200 ) ;
94 }
95 management = false ;
96 closeManage = false ;
97 }

```

Fonte: O Autor (2025)

3.4.2 Query de Banco de Dados

O sistema faz uso de consultas ao banco de dados PostgreSQL para recuperar informações essenciais ao funcionamento da interface e dos módulos de análise. A busca por registros históricos, resultados de testes e dados de identificação dos produtos ocorre em rotinas específicas, otimizadas para reduzir a carga sobre o banco e melhorar a responsividade do SCADA.

O Algoritmo 2 apresenta a estrutura utilizada para executar uma consulta SQL típica, incluindo a preparação da query, a execução da instrução no banco de dados e o tratamento dos resultados retornados.

Algoritmo 2 – Query para inserir relatórios no banco de dados

```

1  using  Newtonsoft . Json ;
2  using  Npgsql ;
3  using  System ;
4  using  System . Collections . Generic ;
5  using  System . Linq ;
6  using  System . Text ;
7  using  System . Threading . Tasks ;
8

```

```

9 public static void InserirReport(Generic_scada_report report)
10 {
11     NpgsqlConnection conn = GeneralParameters.OpenDatabaseConn();
12     try
13     {
14         string sql = @"
15 INSERT INTO generic_scada_report
16 (jiga_id ,
17 recipe_name ,
18 init_time ,
19 end_time ,
20 recipe_data ,
21 result ,
22 "user")
23 VALUES( @jiga_id ,
24 @recipe_id ,
25 @init_time ,
26 @end_time ,
27 @recipe_data ,
28 @result ,
29 @username)
30 ";
31
32     NpgsqlCommand cmd = new NpgsqlCommand(sql, conn);
33     cmd.Parameters.AddWithValue("jiga_id", report.jiga_id);
34     cmd.Parameters.AddWithValue("recipe_id", report.recipe_name);
35     cmd.Parameters.AddWithValue("init_time", report.init_time);
36     cmd.Parameters.AddWithValue("end_time", report.end_time);
37     cmd.Parameters.Add("recipe_data",
38     NpgsqlTypes.NpgsqlDbType.Json).Value =
39     Recipe_data.ListToJson(report.recipe_data);
40     cmd.Parameters.AddWithValue("result", report.result);
41     cmd.Parameters.AddWithValue("username", report.user);
42
43     cmd.ExecuteNonQuery();
44     GeneralParameters.CloseDatabaseConn(conn);
45     }
46     catch (Exception ex)
47     {
48         GeneralParameters.CloseDatabaseConn(conn);
49         throw new Exception("Erro ao inserir report: " + ex.Message);
50     }
51 }

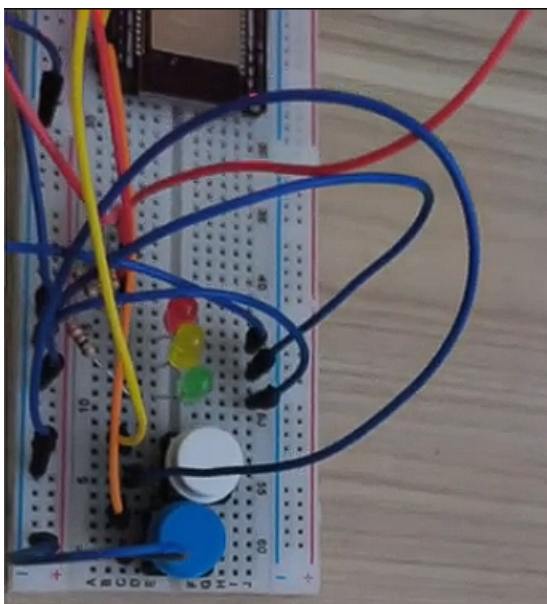
```

Fonte: O Autor (2021)

3.5 EXECUÇÃO DOS TESTES

Após a conclusão da implementação inicial do sistema SCADA, foi iniciada a fase de testes funcionais, com o objetivo de validar a comunicação, a lógica de controle e a integridade das rotinas desenvolvidas. Para simular o ambiente de operação real, foi montado um circuito em protoboard utilizando um microcontrolador ESP32, configurado para atuar como o dispositivo de campo equivalente ao CLP utilizado na indústria.

Figura 16 – Protoboard montada para testes e validação do sistema



Fonte: O Autor (2025).

O ESP32 foi programado para operar como servidor Modbus/TCP via conexão Wi-Fi, recebendo comandos e enviando estados ao sistema SCADA. A interface física consistiu em três LEDs — vermelho, amarelo e verde — conectados a diferentes saídas digitais, além de dois botões (branco e azul) conectados às entradas digitais do microcontrolador. Essa configuração permitiu criar um cenário controlado e suficientemente flexível para testar os principais fluxos de comunicação do software.

Com a comunicação Modbus/TCP devidamente estabelecida e validada, iniciaram-se os testes individuais das funcionalidades. O primeiro conjunto de testes envolveu o acionamento dos LEDs. Foram executados comandos de escrita sequenciais para ligar e desligar cada LED, inicialmente de forma isolada e, posteriormente, compondo uma sequência estruturada, usando o recurso da receita, executada pelo SCADA. Esse procedimento permitiu verificar tanto o envio correto dos comandos quanto a atualização dos estados exibidos na interface do sistema.

Em seguida, foram realizados os testes relacionados aos botões. Para isso, foram criados ciclos de leitura periódica, verificando primeiramente a detecção do botão pressionado e, na sequência, a confirmação de que o botão havia retornado ao estado de repouso. Esses testes

asseguraram que as leituras digitais estavam estáveis, sem falsos positivos, e que o SCADA reagia adequadamente às mudanças de estado enviadas pelo ESP32.

A partir desses experimentos foi possível validar o comportamento geral do sistema, confirmando o correto funcionamento das rotinas de comunicação, escrita, leitura e processamento interno. Os resultados obtidos nesses testes serviram como evidência prática da confiabilidade da arquitetura implementada e orientaram ajustes finos realizados antes da conclusão do projeto.

4 RESULTADOS

A presente seção apresenta os principais resultados obtidos a partir do desenvolvimento e validação do sistema SCADA proposto. Os testes práticos foram conduzidos utilizando uma montagem protoboard composta por LEDs e botões, permitindo simular entradas e saídas digitais representativas de um processo real. Por meio de uma abordagem experimental e iterativa, foram conduzidos testes práticos envolvendo a execução de receitas, a comunicação entre módulos, a aquisição de dados, o registro de eventos e a geração automática de relatórios. Cada funcionalidade implementada foi analisada quanto ao seu desempenho, estabilidade e aderência aos requisitos levantados durante a fase de especificação.

Durante os testes iniciais de validação, foi possível observar que o mecanismo de criação e modificação das receitas apresentou desempenho acima das expectativas. A alteração na ordem de execução de comandos, assim como o ajuste das condições de verificação de sinais, pôde ser realizada em questão de segundos, permitindo reconfigurações rápidas e não intrusivas no fluxo de teste. Essa flexibilidade demonstrou que o sistema oferece elevada eficiência na adaptação dos procedimentos, tornando-se especialmente útil em ambientes onde ajustes frequentes são necessários para validar diferentes combinações de entradas e saídas.

Outro aspecto relevante identificado diz respeito ao controle de acesso por usuário. A separação de permissões garantiu que apenas usuários autorizados pudessem modificar parâmetros essenciais do sistema, reduzindo significativamente o risco de alterações indevidas em dados críticos ou em configurações sensíveis das rotinas de teste. Esse mecanismo reforça a segurança operacional do ambiente e contribui para a rastreabilidade das ações realizadas.

Os relatórios gerados pelo sistema também se mostraram fundamentais para o acompanhamento e análise das execuções. A estruturação clara dos dados — incluindo comandos executados, valores retornados e sequência cronológica das operações — proporcionou uma visão detalhada do comportamento do processo, facilitando a identificação de falhas e o entendimento do fluxo completo de testes. Durante os ensaios, essa visualização contínua dos dados provou-se essencial para ajustes em tempo real e para a validação dos resultados.

Além disso, a arquitetura do código foi planejada de forma a facilitar a integração de novos comandos e funcionalidades. Essa característica permite que o sistema se adapte a diferentes aplicações e requisitos específicos, sem necessidade de refatorações extensas. A inclusão de novas IOs, por exemplo, mostrou-se rápida e direta, o que contribuiu para a agilidade na configuração inicial do ambiente e para a expansão de sistemas já em operação.

Em relação ao desempenho da comunicação, as medições realizadas durante os ensaios indicaram que o tempo médio de resposta das operações Modbus apresentou-se em torno de 8,5 ms por ciclo de requisição. Esse valor foi obtido a partir de instrumentação direta no código-

fonte, registrando o tempo de envio e recebimento das mensagens. Os resultados demonstram que o sistema é capaz de manter uma comunicação estável e de baixa latência com os dispositivos conectados.

Durante os testes da interface gráfica, verificou-se a necessidade de implementar mecanismos de controle de janelas para evitar sobreposições indevidas e comportamentos inesperados durante a navegação entre telas. Para mitigar esse problema, foram criadas rotinas que bloqueiam a interação com janelas inativas, garantindo que a tela anterior não permaneça acessível enquanto uma nova funcionalidade está em execução. Além disso, foi implementado um controle de posicionamento automático, de forma que todas as janelas abertas utilizem a mesma referência espacial no monitor, mantendo a consistência visual e operacional ao longo do uso do sistema.

5 CONSIDERAÇÕES FINAIS

O desenvolvimento do sistema SCADA apresentado ao longo deste trabalho permitiu demonstrar a viabilidade de uma solução flexível, modular e adaptável às necessidades de monitoramento e testes de produtos em ambientes industriais. A implementação em C#, aliada ao uso de banco de dados PostgreSQL e à integração com protocolos industriais, mostrou-se uma combinação eficiente para garantir desempenho, organização estrutural e escalabilidade. Os resultados obtidos comprovam que a arquitetura proposta atende aos objetivos iniciais do projeto, oferecendo uma base sólida para futuras expansões e adaptações.

Apesar da maturidade alcançada nesta etapa, o sistema ainda apresenta um amplo potencial de evolução. Pretende-se, como continuidade deste trabalho, explorar diferentes variações de operação e ampliar os cenários de teste para que o software consiga atender a uma gama ainda maior de aplicações industriais. Além disso, parte das ideias aqui desenvolvidas já está sendo aplicada na empresa onde atuo, utilizando recursos e infraestruturas mais adequados ao ambiente corporativo. Esse contato direto com o setor industrial reforça a relevância prática do projeto e abre perspectivas para, no futuro, apresentar a solução a empresas parceiras, com possibilidade real de comercialização.

Por fim, vale destacar que o projeto estabeleceu uma base capaz de sustentar avanços futuros, como melhorias na interface homem-máquina, integração com sistemas de inteligência artificial para análise preditiva, expansão de protocolos industriais suportados e otimizações de desempenho na comunicação Modbus. Com estas evoluções, o sistema pode se tornar uma ferramenta robusta, moderna e competitiva no mercado de automação e controle, contribuindo para soluções mais eficientes e inteligentes dentro do ambiente industrial.

REFERÊNCIAS

- ALTUS. O que é e por que você deve utilizar uma ihm na sua aplicação. 2021. Disponível em: <https://www.altus.com.br/post/360/o-que-e-e-por-que-voce-deve-utilizar-uma-ihm-na-sua-aplicacao?utm_source=chatgpt.com>.
- Amazon Web Services. **What is Latency?** 2025. Acessado em: 26 maio 2025. Disponível em: <<https://aws.amazon.com/pt/what-is/latency/>>.
- ARCADIS. **Next-Generation SCADA High Performance Human Machine Interfaces.** 2016. <<https://cdn.ymaws.com/oawwa.org/resource/collection/6EE0BF16-E230-4405-8005-29A50CD98EE2/High%20Performance%20Human-Machine%20Interfaces.pdf>>. Acesso em: 10 maio 2025.
- AUTOMATION, C. **Comparison of Modbus Over TCP/IP and PROFINET.** 2021. Acesso em: 15 jun. 2025. Disponível em: <<https://control.com/technical-articles/comparison-of-modbus-over-tcp-ip-and-profinet/>>.
- Auvik Networks. **Jitter vs. Latency: What's the Difference?** 2020. Acessado em: 26 maio 2025. Disponível em: <<https://www.auvik.com/franklyit/blog/jitter-vs-latency/>>.
- BAIRESDEV. **PostgreSQL vs Oracle: Which Database Is Right for Your Project?** 2025. Acesso em: 26 maio 2025. Disponível em: <<https://www.bairesdev.com/blog/postgresql-vs-oracle/>>.
- CHAVES, L. E. **O Uso de CLP e IHM na Automação Industrial.** 2023. Acesso em: 26 maio 2025. Disponível em: <<https://www.engenhariahibrida.com.br/post/uso-de-clp-ihm-automacao-industrial>>.
- CONTROLDESIGN. **Ethernet/IP, EtherCAT, PROFINET and Modbus TCP: Which Protocol Is Right for You?** 2025. Acesso em: 26 abr. 2025. Disponível em: <<https://www.controldesign.com/connections/data-acquisition-monitoring/article/55249378/ethernet-ip-ethercat-profinet-and-modbus-tcp-which-protocol-is-right-for-you>>.
- Dakol Automação. **IHM – o que é e para que serve.** 2022. Acesso em: 26 maio 2025. Disponível em: <<https://www.dakol.com.br/ihm/>>.
- DANTAS, H. A. UtilizaÇÃo da linguagem de programaÇÃo c/c++ no estudo e ensino de alguns tÓpicos da Álgebra linear. **Universidade Federal Rural do semi-árido**, 2021. Acesso em: 26 maio 2025. Disponível em: <<https://repositorio.ufersa.edu.br/server/api/core/bitstreams/7ec5f1c8-18a8-4bd6-b9cc-457a6e29f0be/content>>.
- DB-Engines. **DB-Engines Ranking - Comparison of Microsoft SQL Server, Oracle and PostgreSQL.** 2025. Accessed: 2025-04-26. Disponível em: <<https://db-engines.com/en/system/Microsoft%2BSQL%2BServer%3BOracle%3BPostgreSQL>>.
- ELANGO VAN, Y. Test fixtures in manufacturing. **arXiv**, 2025. Disponível em: <<https://arxiv.org/abs/2503.02072>>.

ENTERPRISEDB. **Microsoft SQL Server (MSSQL) vs PostgreSQL: Comparison Details**. 2025. Acesso em: 26 maio 2025. Disponível em: <<https://www.enterprisedb.com/blog/microsoft-sql-server-mssql-vs-postgresql-comparison-details-what-differences>>.

ENVISIA. **Conceitos Básicos do Protocolo Ethernet/IP**. 2022. Acesso em: 26 abr. 2025. Disponível em: <<https://www.envisia.com.br/2022/08/22/conceitos-basicos-do-protocolo-ethernet-ip/>>.

FLUMINENSE, U. U. F. Aplicação do scadabr para monitoramento de sistemas de controle industrial utilizando python. **UFF - Universidade Federal Fluminense**, 2021. Disponível em: <<https://app.uff.br/riuff/handle/1/27539>>.

FREITAS, C. M. Pesquisa de mercado sobre python no mercado brasileiro. **Universidade Tecnológica Federal do Paraná**, 2023. Acesso em: 26 maio 2025. Disponível em: <<https://repositorio.utfpr.edu.br/jspui/bitstream/1/31600/1/pesquisamercadopython.pdf>>.

FS. **Comparing Industrial Ethernet Protocols**. 2025. Acesso em: 26 abr. 2025. Disponível em: <<https://www.fs.com/blog/comparing-industrial-ethernet-protocols-1735.html>>.

FUTURAMO. **Oracle vs SQL Server: Head-to-Head Comparison**. 2025. Acesso em: 26 maio 2025. Disponível em: <<https://futuramo.com/blog/oracle-vs-sql-server-head-to-head-comparison/>>.

INDUSTRIAL, I. A. O que é a interface homem-máquina (ihm)? 2021. Disponível em: <https://impulsoautomacaoindustrial.com/o-que-e-a-interface-homem-maquina-ihm/?utm_source=chatgpt.com>.

INSTRUMENTS, N. **The Modbus Protocol in Depth**. 2025. Acesso em: 26 abr. 2025. Disponível em: <https://www.ni.com/en/shop/seamlessly-connect-to-third-party-devices-and-supervisory-system/the-modbus-protocol-in-depth.html?srsId=AfmBOorkenRHfVUQsWuu-bk_nDm2_gGq3hDg-29GDSypIXIUthfotkIW>.

ISA São Paulo. **Introdução à Norma ISA-101: Interfaces Homem-Máquina**. 2016. <<https://isasp.org.br/wp-content/uploads/2020/01/ISA-101-III-Simp%C3%B3sio-ISA-S%C3%A3o-Paulo-Sabesp-Nov2016.pdf>>. Acesso em: 10 maio 2025.

Kalatec Automação. **IHM – Interface Homem-Máquina: melhores modelos para comprar**. 2023. Acesso em: 26 maio 2025. Disponível em: <<https://www.kalatec.com.br/ihm/>>.

Keysight Technologies. **PathWave Test Automation Software**. 2022. <<https://www.keysight.com/us/en/products/software/pathwave-test-software/pathwave-test-automation-software.html>>. Acesso em: 1 jun. 2025.

Modbus Organization. **About Modbus**. 2023. <<https://www.modbus.org>>. Acessado em 31 de Agosto de 2025.

MODBUSTOOLS. **Modbus Protocol**. 2025. Acesso em: 26 abr. 2025. Disponível em: <<https://www.modbustools.com/modbus.html>>.

NASBY, G. **Introduction to ISA-112: SCADA Systems**. 2019. <https://www.grahamnasby.com/files_publications/NasbyG_2019_Intro-to-ISA112_Halton-SCADA-Workshop_jun26-2019_slides.pdf>. Acesso em: 10 maio 2025.

National Instruments. **What is TestStand?** 2022. <<https://www.ni.com/en-us/shop/test-stand.html>>. Acesso em: 1 jun. 2025.

Obkio Inc. **Latency vs Jitter: What's the Difference and Why Does It Matter?** 2021. Acessado em: 26 maio 2025. Disponível em: <<https://obkio.com/blog/latency-vs-jitter/>>.

ODVA. **EtherNet/IP**. 2025. <<https://www.odva.org/technology-standards/key-technologies/ethernet-ip/>>. Acessado em 31 de Agosto de 2025.

ONLOGIC. **What is SCADA Automation and How is it Used?** 2025. Acessado em: 26 abr. 2025. Disponível em: <<https://www.onlogic.com/blog/what-is-scada-automation/>>.

PEREIRA, P. I. S. **Processo de melhoria na implementação de jigas automáticas no teste de placas eletrônicas industriais**. 2023. Trabalho de Conclusão de Curso, Centro Universitário Ritter dos Reis. Disponível em: <<https://repositorio-api.animaeducacao.com.br/server/api/core/bitstreams/3fd23d1a-0b72-486f-9fd8-1941c59fb66b/content>>.

Pride Industries. **Fixture Design for Manufacturing**. 2025. Disponível em: <https://www.prideindustries.com/our-stories/fixture-design-for-manufacturing?utm_source=chatgpt.com>.

PROCENTEC. **PROFINET vs EtherNet/IP: The Similarities and Differences**. 2023. Acesso em: 15 jun. 2025. Disponível em: <<https://procentec.com/content/profinet-vs-ethernetip-the-similarities-and-differences/>>.

PROFINET. **PROFINET Explained**. 2025. Acesso em: 26 abr. 2025. Disponível em: <<https://us.profinet.com/profinet-explained/>>.

RAPTIS, T. P.; PASSARELLA, A.; CONTI, M. Data management in industry 4.0: State of the art and open challenges. **arXiv preprint arXiv:1902.06141**, 2019. Disponível em: <<https://arxiv.org/abs/1902.06141>>.

RESEARCHGATE. Protótipo de sistema supervisorio utilizando c# e arduino. **ResearchGate**, 2020. Disponível em: <https://www.researchgate.net/publication/340844870_PROTOTIPO_DE_SISTEMA_SUPERVISORIO>.

ScadaBR. **ScadaBR: Sistema SCADA Livre e Open Source**. 2025. Acesso em: 07 set. 2025. Disponível em: <<https://scadabr.org>>.

SCIELO. Ergonomia e engenharia de usabilidade aplicadas no desenvolvimento da interface homem-máquina para um sistema interativo de soldagem automatizada. 2021. Disponível em: <https://www.scielo.br/j/cr/a/FYrzLGwJJpSpqFgHhQZvgsK/?utm_source=chatgpt.com>.

SCRIBD. **Interferência Eletromagnética na Comunicação em Redes Industriais**. 2023. Acesso em: 26 abr. 2025. Disponível em: <<https://pt.scribd.com/document/821898273/INTERFERENCIA-ELETROMAGNETICA-NA-COMUNICACAO-EM-REDES-INDUSTRIAIS>>.

Siemens. **PROFINET**. 2025. <<https://www.siemens.com/br/pt/produtos/automacao/comunicacao-industrial/profinet.html>>. Acessado em 31 de Agosto de 2025.

SILVA, A. C. S. **Análise de Uso de Banco de Dados por Sistemas de Automação Industrial**. 2018. Monografia (Bacharelado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul. Disponível em: <<https://lume.ufrgs.br/bitstream/handle/10183/190151/001088599.pdf?sequence=1>>.

SOUZA, R. D. C. Uma análise de performance das linguagens de programação no processamento paralelo. **Revista Científica Multidisciplinar Núcleo do Conhecimento**, 2023. Acesso em: 26 maio 2025. Disponível em: <<https://www.nucleodoconhecimento.com.br/engenharia-da-computacao/linguagens-de-programacao>>.

UBIDOTS. **SCADA System Explained: The Heart of Industrial Automation**. 2025. Acessado em: 26 abr. 2025. Disponível em: <<https://ubidots.com/blog/scada-system-explained-the-heart-of-industrial-automation/>>.

UPADHYAY, D.; SAMPALLI, S. Scada (supervisory control and data acquisition) systems: Vulnerability assessment and security recommendations. **Computers & Security**, Elsevier, v. 89, p. 101666, 2020.

Venturus. **Tudo sobre Jigas de Testes de Placas Eletrônicas**. 2025. Disponível em: <https://www.venturus.org.br/insights/blog/tudo-sobre-jigas-de-testes-de-placas-eletronicas?utm_source=chatgpt.com>.