UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
MESTRADO EM ENGENHARIA ELÉTRICA

Compreensão e exemplificação dos fundamentos dos protocolos industriais - Ethernet/IP

MANAUS - AM
SETEMBRO DE 2025

Disciplina: Protocolos de comunicação e aplicações industriais.

Aluno(s):

- Thiago Rodrigo Monteiro Salgado.
- Gledyson Cidade da Costa.

Professor:

- Dr. Rafael da Silva Mendonça

Objetivos do trabalho: utilizando o protocolo Ethernet/IP, este trabalho visa apresentar conceitos, características, diagramas de funcionamento, aplicações na indústria. Além de realizar comparações com outros protocolos bastante utilizados no setor industrial e realizar uma aplicação desenvolvida utilizando o protocolo.

MANAUS - AM
SETEMBRO DE 2025

Sumário

1. Definição

Por volta dos anos de 1980 muitos fabricantes de **CLP** (controlador lógico programável) costumavam desenvolver seu próprio protocolo de comunicação, o que fazia com que houvesse uma enorme dificuldade em integrar CLPs de marcas diferentes, deixando tudo difícil, caro e limitado. Diante disso, a ODVA (*Open DeviceNet Vendors Association*) cria o **CIP** (*Protocol Industrial Comum*), que seria um protocolo padronizado visando a comunicação entre os dispositivos industriais.

O CIP era um protocolo que de aplicação que ficava não ficava preso ao tipo de rede, podendo ser utilizado **CAN** (*Controller Area Network*), Ethernet, entre outros. Sua intenção era realmente facilitar a troca de informações na camada de aplicação entre os dispositivos. Ou seja, era o início de uma ideia onde todos os dispositivos industriais poderiam ser visualizados de uma maneira padronizada. A partir do CIP, surgem então três protocolos: DeviceNet (utilizando CAN) e a ControlNet.

É então que com o sucesso do CIP e a evolução da tecnologia de Ethernet de não determinístico para algo determinístico, escalar e confiável é que nos anos 2000 surge o **Ethernet/IP** (Ethernet Industrial Protocol). A ideia foi usar o CIP por meio de TCP/IP e **UDP/IP**, contudo, utilizando na camada de enlace apenas Ethernet. Mais precisamente Ethernet IEEE 802.3, que é um padrão que descreve como os computadores e dispositivos se comunicam em LAN (Local area network). Com isso o Ethernet/IP permitiu que um modelo de dados padrão fosse utilizado em redes modernas e do mundo inteiro, fornecendo escalabilidade e interoperabilidade, além de ser um protocolo totalmente aberto.

1.1 Características gerais

Neste capítulo será abordado as características de forma genérica e posteriormente elas serão melhor detalhadas.

Sempre que falamos de protocolo de comunicação, falamos também das suas características, sendo elas costumeiramente as camadas de arquitetura de rede que utilizam, sua velocidade, segurança e escalabilidade. Isto porque são as características de um protocolo que definem onde podem ser melhor utilizados. Neste caso, o protocolo Ethernet/IP é um protocolo voltado para o setor de automação industrial e temos as seguintes características:

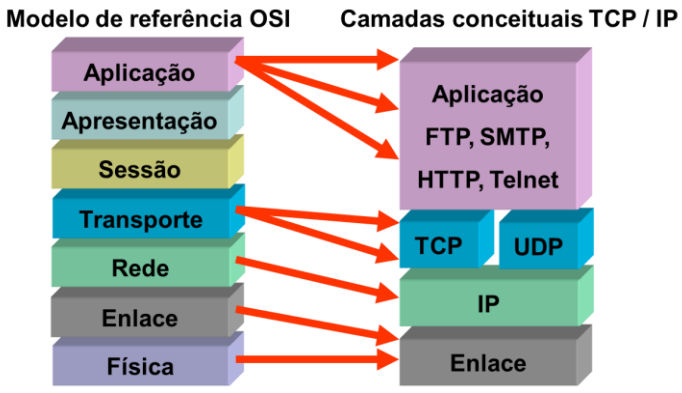
- Arquitetura: utiliza Ethernet (IEEE 802.3) na camada de enlace e nas camadas superiores utiliza TCP/IP e UDP/IP. Já na camada de aplicação utiliza-se do protocolo CIP, que é o que garante interoperabilidade com outras redes CIP.
- Comunicação: utiliza TCP/IP a fim de garantir monitoramento e diagnóstico. Além disso, utiliza comunicação produtora/consumidora baseado em UDP, permitindo um troca cíclica de dados de entradas e saídas de forma determinística, atendendo aos requisitos de tempo real. Ela também representa cada dispositivo como objetos CIP com atributos (dados), serviços (funções) e instâncias (recursos específicos).

- Interoperabilidade: é aberto e padronizado, além de ser totalmente compatível com outros sistemas que também utilizam o protocolo CIP na camada de aplicação como o DeviceNet ou o controleNET.
- Velocidade e determinismo: possui uma comunicação em tempo real, isto porque se utiliza do UDP e os ciclos de atualizações dos dados permitem atender aplicações em milissegundos. Além do que, por estarmos falando de uma rede focada no setor industrial, temos o QoS (*quality of service*), que prioriza pacotes de redes industriais na rede, reduzindo ainda mais a latência e por fim, sua capacidade de ser determinístico.
- Escalabilidade: pode-se utilizar milhares de dispositivos em uma mesma rede e integrar com redes corporativas, facilitando a comunicação vertical também, subindo para o nível de gestão. Muito utilizado para MES (*Manufacturing Execution System*) e ERP (*Enterprise Resource Planning*).
- Segurança e confiabilidade: está integrado a protocolos modernos de segurança como firewalls industriais no nível de rede, e na camada de aplicação utiliza o CIP security que implementa autenticação, integridade e confidencialidade dos dados, além de uma assinatura digital para garantir firmwares não adulterados e gestão de certificados para autenticação de dispositivos. Permitindo que apenas dispositivos autorizados participem da rede.

1.2 Arquitetura

Nesta seção será abordada como a arquitetura do Ethernet/IP está estruturada. Sua arquitetura possui três pilares principais na cama conceitual TCP/IP, visto na Figura 1, que seria a camada de Enlace (seguindo os padrões IEEE 802.3), a cama de transporte (TCP/IP e UDP) e a camada de aplicação, na qual utiliza-se o protocolo CIP.

Figura 1 - camadas conceituais de rede



Fonte 1 – academia de redes

1.2.1 Camada de Enlace

Na camada de enlace, o Ethernet/IP utiliza a tecnologia Ethernet padrão (IEEE 802.3), a mesma utilizada em redes locais de computadores (LANs). É essa camada que fica responsável por transmitir fisicamente os quadros na rede, fazendo com que a comunicação entre os dispositivos ocorra de forma confiável. Dependendo

da infraestrutura disponível, podem ser alcançadas diferentes taxas de transmissão, como 10 Mbps, 100 Mbps, 1 Gbps ou até valores mais altos em redes modernas. Além disso, o uso do padrão Ethernet assegura compatibilidade com equipamentos amplamente difundidos, como switches, roteadores e cabos de rede, o que facilita a integração e reduz custos de implementação.

1.2.2 Camadas de Rede e Transporte

Nas camadas de rede e transporte, o Ethernet/IP faz uso dos protocolos IP, TCP e UDP, seus papéis são:

- **O IP (Internet Protocol):** garante que cada dispositivo da rede possua um endereço único, permitindo a identificação e a comunicação entre CLPs, IHMs, sensores e atuadores de forma padronizada e global;
- **O TCP (Transmission Control Protocol):** é usado em situações que exigem maior confiabilidade, normalmente utilizado na configuração de dispositivos, no monitoramento e em processos de diagnóstico, assegurando a entrega correta dos pacotes, sua ordem e a detecção de perdas durante a transmissão. Por outro lado;
- **O UDP (User Datagram Protocol):** é utilizado para a troca cíclica de dados em tempo real, típica das aplicações industriais, com ela pode-se eliminar a sobrecarga de verificações e retransmissões, resultando em menor latência e permitindo o determinismo necessário para o controle de processos.

1.2.3 Camada de aplicação

Todos os outros subtópicos antes desse exploraram características que são comuns e padronizados por outros protocolos de comunicação, mas o diferencial do Ethernet/IP vem na camada de aplicação, pois é nela que atua o CIP.

Nessa camada abstrai os detalhes das transmissões e adota uma visão orientada a objetos para visualizar os dispositivos industriais. Cada dispositivo é descrito por **objetos CIP**, que se organizam em três elementos:

- **Atributos**: correspondem aos dados do dispositivo, como o valor de uma variável ou o estado de um sensor;
- **Serviços:** são as funções que podem ser executadas, como um comando de reset ou o início de uma leitura e;
- **Instâncias:** que representam recursos específicos dentro de cada objeto. com a essa estrutura padronizada, fabricantes diferentes conseguem desenvolver dispositivos que se comunicam de forma transparente entre si, garantindo interoperabilidade e reduzindo barreiras na integração de sistemas industriais.

1.3 Comunicação Produtor/Consumidor

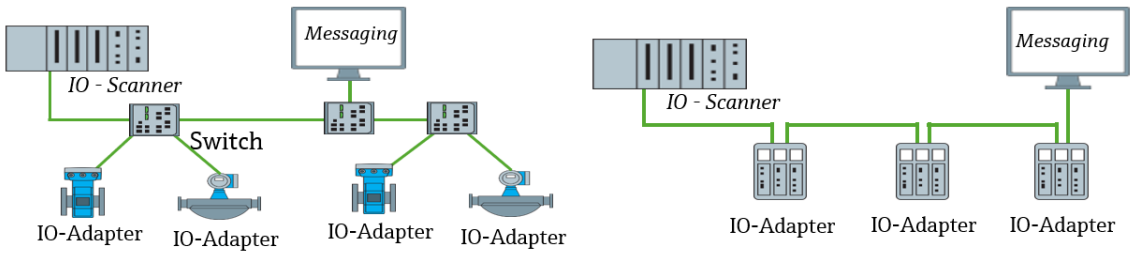
Um dos pontos importantes do Ethernet/IP é o modelo de comunicação **produtor/consumidor**, definido na camada de aplicação pelo **CIP** e suportado pela camada de transporte (**UDP**). Nesse modelo, um único dispositivo (produtor) envia dados que podem ser consumidos por vários dispositivos simultaneamente. Isso é

muito mais eficiente do que o modelo tradicional cliente/servidor, no qual cada troca de informação exigiria uma **solicitação individual**.

Na prática, isso significa que um CLP pode produzir periodicamente os estados de suas entradas e disponibilizá-los para vários outros dispositivos que atuam como consumidores (como IHMs, sistemas supervisórios ou outros CLPs), sem precisar replicar a mesma informação diversas vezes. O uso desse mecanismo, juntamente do UDP, possibilita o **determinismo** e o tempo de resposta em milissegundos, atendendo às necessidades do ambiente industrial. Em alguns locais, é bastante comum chamar o produtor de ***adapter***, que seria o equivalente ao produtor, ou seja, gera os dados. E para o consumidor é comum se chamar também de ***scanner***.

Na Figura 2, podemos ver um CLP como um *scanner* e consumidores como IO-*Adapter*. Podemos ver duas diferentes topologias. Podemos ver que o CLP fica produzindo dados e N consumidores podem consumir deste CLP.

Figura 2 - topologia linear: adapter e scanner exemplo



Fonte 2 – netlion ethernet/IP

1.4 Segurança e confiabilidade

Com relação a segurança, o EtherNet/IP integra mecanismos modernos em diferentes níveis. Na camada de rede, pode ser protegido por firewalls industriais, VLANs e segmentação de tráfego, limitando acessos não autorizados, algo padrão na industria. Já na camada de aplicação, conta com **o CIP Security**, uma extensão do protocolo que aplica conceitos como autenticação de dispositivos, integridade das mensagens e confidencialidade dos dados. O uso de **certificados digitais** e **assinaturas** garante que apenas dispositivos autorizados participem da rede e que firmware ou dados não sejam adulterados. Esses recursos tornam o EtherNet/IP adequado não apenas para redes de automação isoladas, mas também para ambientes integrados ao nível corporativo, onde a segurança da informação é indispensável, já que ele também possui comunicação vertical podendo ser utilizado para MES.

A confiabilidade deste protocolo se da por meio de diferentes recursos em suas camadas de comunicação. No transporte via TCP, ele assegura entrega ordenada e retransmissão de pacotes em aplicações mais críticas como configurações emonitoramentos. Quanto aos dados de tempo real, enviados via UDP, a confiabilidade é obtida pelo próprio modelo cíclico de transmissão, onde os dados são atualizados constantemente em intervalos de milissegundos, reduzindo o impacto de eventuais perdas de pacotes. Além disso, o protocolo adota mecanismos de **QoS** (Quality of Service), muito utilizado também por outros protocolos já consolidados, que priorizam pacotes industriais em relação ao tráfego comum da rede, garantindo baixa latência e determinismo mesmo em redes mais carregadas.

2. Comparação com outros protocolos industriais

Fazer uma tabela comparativa entre os protocolos mais famosos: modbus, profinet, opc-ua, ethercat e o próprio ethernet/ip, comparando velocidade, segurança, forma de comunicação, segurança.

Característica	Modbus	PROFINET	OPC UA	EtherCAT	EtherNet/IP
Ano / Origem	1979 / Modicon	2003 / PI (Siemens)	2008 / OPC Foundation	2003 / Beckhoff	2000 / ODVA
Transporte	RS-485 / TCP/IP	Ethernet / TCP/IP / RT	TCP/IP / Pub-Sub	Ethernet direto (on-the-fly)	Ethernet TCP/UDP/IP
Modelo	Mestre-escravo	Produtor-consumidor	Cliente-servidor / Pub-Sub	Mestre-escravo determinístico	Produtor-consumidor
Determinismo	Baixo	Alto (com RT/IRT)	Médio (boa para TI, não *hard real-time*)	Muito alto (µs)	Alto (UDP + QoS + TSN)
Velocidade típica	até 115 kbps (RTU) / 100 Mbps (TCP)	100 Mbps / 1 Gbps	+100 Mbps!	100 Mbps / 1 Gbps	100 Mbps / 1 Gbps
Escalabilidade	Alta, mas lenta	Alta	Muito alta (íntegra TI/OT)	Alta (milhares)	Alta (milhares)
Interoperabilidade	Alta (muito difundido)	Alta, mas forte presença Siemens	Muito alta (aberto, multi-plataforma)	Boa, mas concentrado em automação de movimento	Alta (ODVA, multi-vendor, CIP security)
Aplicações típicas	CLPs antigos, sensores, energia	Automação de processos, máquinas Siemens	Integração corporativa, IIoT	Robótica, CNC, motion control	Automação discreta, robôs, CLPs, chão de fábrica
Segurança	Quase inexistente	Segurança via TI/TSN	Forte (criptografia, autenticação)	Baixa nativa	CIP Security (TLS/DTLS)
Vantagens	Simples, barato, universal	Determinismo, suporte Siemens	Interoperabilidade, integração TI/OT	Altíssima velocidade e sincronismo	Aberto, baseado em Ethernet padrão
Limitações	Baixa velocidade, sem segurança	Mais dependente do ecossistema Siemens	Não é determinístico rígido	Mais caro, nichado	UDP depende de QoS/TSN para determinismo

3. Aplicações

Neste capítulo veremos algumas aplicações do protocolo no contexto industrial. Trabalhando como uma ferramenta para a integração de diferentes níveis da automação, desde o chão de fábrica até os sistemas corporativos. Como principais exemplos temos:

- **Automação de Processos**: o Ethernet/IP é amplamente utilizado em sistemas de automação de processos industriais, onde é necessário monitorar e controlar equipamentos como bombas, válvulas e sensores em tempo real. A capacidade de comunicação em tempo real e a interoperabilidade com outros dispositivos tornam o Ethernet/IP uma escolha ideal para integrar diferentes componentes em uma planta industrial.
- **Controle de Máquinas e Robótica**: em ambientes que utilizam robôs industriais e máquinas automatizadas, o Ethernet/IP permite a comunicação eficiente entre controladores lógicos programáveis (CLPs) e dispositivos de campo. A comunicação determinística e a alta velocidade garantem que os robôs possam operar de forma sincronizada e coordenada, aumentando a eficiência da produção.
- **Sistemas de Execução de Manufatura (MES)**: o Ethernet/IP é utilizado em sistemas **MES** para conectar equipamentos de chão de fábrica a sistemas corporativos, como ERP. Isso possibilita a coleta e análise de dados em tempo real, melhorando a tomada de decisões e a gestão da produção. A escalabilidade do Ethernet/IP permite integrar milhares de dispositivos, facilitando a comunicação vertical entre níveis operacionais e de gestão.

4. Protótipo

Para demonstração do protocolo, foi desenvolvido, em **python**, um servidor Ethernet/IP e um cliente. Ou seja, um produtor e um consumidor. O python realiza toda a parte do backend. Para demonstração do funcionamento foi desenvolvida uma interface simples utilizando o ***framework node-red***.

Para abrir uma comunicação CIP, no python, temos atualmente três bibliotecas principais:

- **pycomm3**: é muito comum para comunicação com CLPs Allen-Bradley usando EtherNet/IP. Ela oferece suporte a mensagens explícitas e implícitas e fornece uma API de fácil entendimento;

- **eeip.py**: esta biblioteca foca em fornecer um módulo compatível com EtherNet/IP de forma abrangente, com suporte a mensagens explícitas e implícitas, função de IO Scanner e uma biblioteca de objetos com objetos definidos pelo padrão CIP. É uma biblioteca bastante completa e também, assim como as outras abstrai toda a parte baixo nível e fornece uma API de fácil entendimento;

- **cpppo**: assim como a Pycomm, esta biblioteca possui mais outros de protocolos de comunicação, o cpppo inclui funcionalidades de EtherNet/IP e pode ser usado para **simular** e interagir com dispositivos EtherNet/IP.

Para criação de monstração do Ethernet/IP, optou-se por utilizar o CPPPO, já que por meio dele é possível simular e também fazer a comunicação CIP.

4.1 Servidor (produtor)

A criação do servidor Ethernet/IP funciona por meio de um comando que é possível de ser passado pelo próprio terminal de comando após a instalação da biblioteca CPPPO. Onde passamos o nome da TAG e o seu tipo. Na Figura 3, podemos visualizar a criação de uma variável Register1 do tipo inteiro, uma variável temperature do tipo REAL, que seria um FLOAT, e FLAGS que seriam 8 booleanos.

Figura 3 - criando servidor ethernet/IP

Fonte 3 - próprio autor

Para nosso servidor, como visto na Figura 4 foram utilizadas as seguintes Flags:

- Acceleration: como um FLOAT;
- Speed_factor: como um FLOAT;
- Inputs: sendo uma lista com 8 valores booleanos;
- Outputs: outra lista com 8 valores booleanos;

O protocolo funciona por meio dessas TAGs, por isso é importante defini-las.

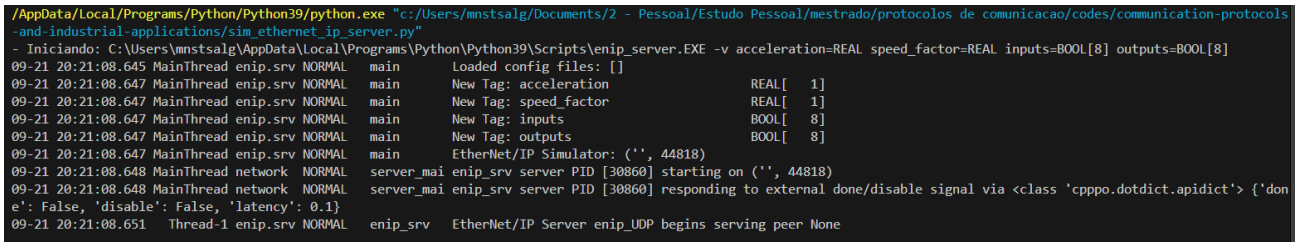
Figura 4 - TAGs utilizadas para criar o servidor

```
tags = {
        "acceleration": "REAL",
        "speed_factor": "REAL",
        "inputs": "BOOL[8]",
        "outputs": "BOOL[8]",
    }
```

Fonte 4 - próprio Autor.

Com a execução do *script*, é então criado o servidor, Figura 5, com as variáveis definidas e na porta padrão do Ethernet/IP, a 44818.

Figura 5 - Terminal durante a criação do servidor Ethernet/IP



```
/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/mnstsalg/Documents/2 - Pessoal/Estudo Pessoal/mestrado/protocolos de comunicacao/codes/communication-protocols
-and-industrial-applications/sim_ethernet_ip_server.py"
- Iniciando: C:\Users\mnstsalg\AppData\Local\Programs\Python\Python39\Scripts\enip_server.EXE -v acceleration=REAL speed_factor=REAL inputs=BOOL[8] outputs=BOOL[8]
09-21 20:21:08.645 MainThread enip.srv NORMAL   main       Loaded config files: []
09-21 20:21:08.647 MainThread enip.srv NORMAL   main       New Tag: acceleration                REAL[   1]
09-21 20:21:08.647 MainThread enip.srv NORMAL   main       New Tag: speed_factor                REAL[   1]
09-21 20:21:08.647 MainThread enip.srv NORMAL   main       New Tag: inputs                      BOOL[   8]
09-21 20:21:08.647 MainThread enip.srv NORMAL   main       New Tag: outputs                     BOOL[   8]
09-21 20:21:08.647 MainThread enip.srv NORMAL   main       EtherNet/IP Simulator: ('', 44818)
09-21 20:21:08.648 MainThread network  NORMAL   server_mai enip_srv server PID [30860] starting on ('', 44818)
09-21 20:21:08.648 MainThread network  NORMAL   server_mai enip_srv server PID [30860] responding to external done/disable signal via <class 'cpppo.dotdict.apidict'> {'don
e': False, 'disable': False, 'latency': 0.1}
09-21 20:21:08.651   Thread-1 enip.srv NORMAL   enip_srv   EtherNet/IP Server enip_UDP begins serving peer None
```

Fonte 5 - próprio autor.

4.2 Client (consumidor)

Já na parte do consumidor, foi utilizada a mesma biblioteca para abrir a comunicação CIP e fazer a comunicação, porém não foi via terminal utilizando parâmetros, mas consumindo as TAGs e escrevendo nelas. Na Figura 6, é possível ver a classe com os métodos de conexão, escrita e leitura de TAGs do tipo FLOAT e booleanas.

Figura 6 - classe de comunicação e conexão com o servidor Ethernet/IP

```python
class EthernetIPProxy:
    class _ClientProtocol(Protocol):
        def read(self, tags: list[str]) -> Iterable[tuple]: ...
        def write(self, tags: list[str]) -> Iterable[bool]: ...

    def __init__(self, host: str = "127.0.0.1", port: int = 44818, timeout: float = 5.0):
        self.host = host
        self.port = port
        self.timeout = timeout
        self.connection: Optional[EthernetIPProxy._ClientProtocol] = None

    def __enter__(self):
        self.connection = proxy_simple(self.host, port=self.port, timeout=self.timeout)
        self.connection.__enter__()
        return self

    def __exit__(self, exc_type, exc_val, exc_tb):
        if self.connection:
            self.connection.__exit__(exc_type, exc_val, exc_tb)

    def write(self, tag: str, value: str) -> bool:
        ok, = self.connection.write([f"{tag}={value}"])
        return ok

    def read(self, tag: str):
        value, = self.connection.read([tag])
        return value[0]

    def write_bool_array(self, tag: str, bools: list[bool]) -> bool:
        data = "(BOOL)" + ",".join("1" if b else "0" for b in bools)
        return self.write(tag, data)

    def read_bool_array(self, tag: str) -> list[bool]:
        values, = self.connection.read([tag])
        return list(values)
```

Fonte 6 – próprio autor.

Neste mesmo programa, foi utilizado Flask, para criar rotas que poderiam ser chamadas pelo Node-red, tanto para leitura quanto para escrita, como visto na Figura 7.

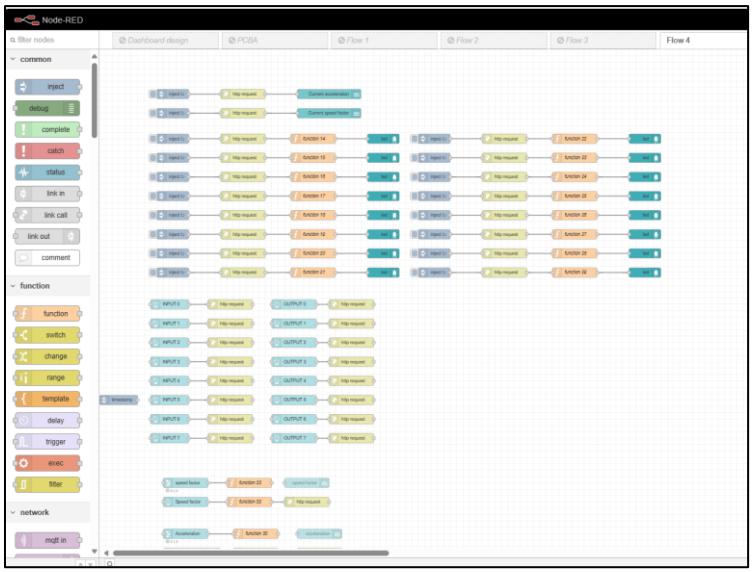
Figura 7 - declarações para chamadas HTTP

```python
@app.route("/inputs/<int:index>", methods=["GET"])
def get_input(index):

    with EthernetIPProxy() as plc:
        outputs = plc.read_bool_array("inputs[0-7]")
        return jsonify(outputs[index])


@app.route("/outputs/<int:index>", methods=["POST"])
def toggle_output(index):
    with EthernetIPProxy() as plc:
        outputs = plc.read_bool_array("outputs[0-7]")

        if index < 0 or index >= len(outputs):
            return jsonify({"error": "Index out of range"}), 400

        outputs[index] = not outputs[index]
```

Fonte 7 - próprio autor.

Para a demonstração desta aplicação, foi utilizado o Node-red, como dito anteriormente, que utilizou-se destas rotas por meio dos nós HTTP. Em conjunto dos nós de *dashboards* como os de LED e de entradas de texto, vistos na Figura 8.

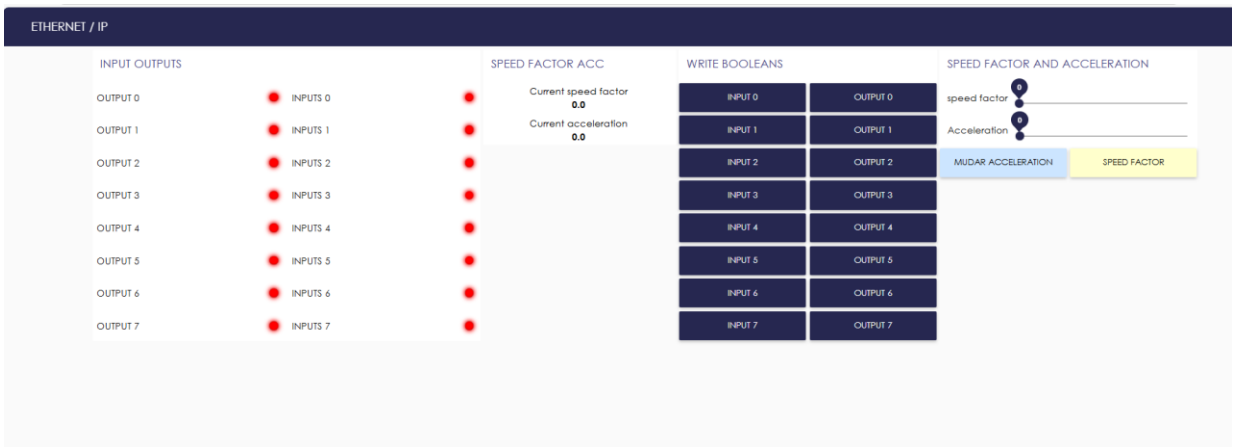
Figura 8 - desenvolvimento dos nós em node-red



Fonte 8 - próprio autor

Com isso, foi possível chegar a interface da Figura 9. Onde temos os visualizadores das *TAGs* de *INPUTs* e *OUTPUTs*, além do valor atual do *speed_factor* e da variável *acceleration*. Do lado mais a direita temos os blocos de escrita, onde podemos alterar o estado das variáveis booleanas e mais a direita é possível deslizar o *slider* e configurar uma nova aceleração ou fator de velocidade.

Figura 9 - interface Node-red para manipulação das variáveis Ethernet/IP



Fonte 9 - próprio autor.

Na Figura 10, foram acionadas algumas entradas e uma saída além de mudar os valores de aceleração e velocidade a fim demonstração.

Figura 10 - manipulando as variáveis

Fonte 10 - próprio autor

5. Conclusão

Por fim, na conclusão deste trabalho, foi possível não somente compreender a o contexto do surgimento, o surgimento em si, sua importância e principais características, mas também de aplicar o protocolo de comunicação Ethernet/IP utilizando a biblioteca CPPPO do python, por meio da simulação de um servidor. Um protocolo robusto, rápido e confiável. Que varia entre TCP/IP e UDP visando a confiabilidade mas também o tempo real.

6. Referências

Para desenvolvimento deste trabalho, foram utilizados conteúdos de sites, livros, publicações e também o auxílio de LLM para pesquisa de conteúdos e desenvolvimento das aplicações. Seguem as referências:

1. Kundert, P.J. *cpppo*. Disponível em: https://github.com/pjkundert/cpppo.
2. Netilion. *Untitled 4*. Disponível em: https://netilion.endress.com/blog/pt/untitled-4/.
3. Academia de Redes. *Protocolos de Roteamento: TCP-IP v4*. Disponível em: https://academiaderedes.com/conteudos/protocolos-de-roteamento/tcp-ip-v4/.
4. Tanenbaum, Andrew S. *Redes de Computadores*. 4. ed.
5. *mm-NHrLtRWI* [vídeo]. Disponível em: https://www.youtube.com/watch?v=mm-NHrLtRWI.
6. *IfeW2MWijG4* [vídeo]. Disponível em: https://www.youtube.com/watch?v=IfeW2MWijG4.
7. ODVA. *Ethernet/IP – Key Technologies*. Disponível em: https://www.odva.org/technology-standards/key-technologies/ethernet-ip/.
8. *vNdaVqsIdHA* [vídeo]. Disponível em: https://www.youtube.com/watch?v=vNdaVqsIdHA.
9. Envisia. *Conceitos básicos do protocolo Ethernet/IP*. Disponível em: https://www.envisia.com.br/2022/08/22/conceitos-basicos-do-protocolo-ethernet-ip/.
10. Rockwell Automation. *Ethernet/IP: White Paper*. Documento técnico ENET-WP001. Disponível em:

https://literature.rockwellautomation.com/idc/groups/literature/documents/wp/enet-wp001_-en-p.pdf

7 Anexos

7.1. Github

Link para o repositório da aplicação:

7.2. Vídeo demonstração no YouTube

Um vídeo no youtube demonstrando a aplicação, segue o link: