CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA
DE MINAS GERAIS
Departamento de Mecatrônica

Hadler Henrique Tempesta

CONTROLE E SUPERVISÃO DE PROCESSOS INDUSTRIAIS POR MEIO DO SIMULADOR 3D FACTORY I/O

Varginha - MG
2022

CONTROLE E SUPERVISÃO DE PROCESSOS INDUSTRIAIS POR MEIO DO SIMULADOR 3D FACTORY I/O

Hadler Henrique Tempesta

Relatório de conclusão de estágio apresentado como requisito para obtenção do título de técnico em mecatrônica pelo Centro Federal de Educação Tecnológica de Minas Gerais.

Orientação: Prof. Dr. Juliano Coêlho Miranda

Varginha - MG
2022

Lista de Figuras

Lista de Tabelas

Sumário

1 Introdução

A simulação é uma ferramenta importante no desenvolvimento de sistemas mais eficientes. No ambiente industrial é amplamente utilizada, desde o projeto para novas aquisições de equipamentos e produtos até a adequação da capacidade produtiva. No cenário da simulação, a realidade virtual, com ambiente 3D, pode ser utilizada como uma forma de visualização e interação do usuário com o meio simulado. No ensino, a simulação pode ser utilizada para uma aprendizagem significativa. Os softwares de simulação permitem interatividade e a possibilidade do discente simular situações experimentais e de visualizar processos que muitas vezes são de difícil obtenção pelas instituições de ensino.

1.1 Factory I/0

O Factory I/O é uma ferramenta de simulação 3D para o aprendizado de tecnologias de automação. O software foi desenvolvido pela empresa Real Games, que atua no ramo de softwares de simulação 3D a mais de uma década, e conta com uma ampla rede de apoiadores, assim como pesquisadores e desenvolvedores em universidades e centros de pesquisa [1].

Dentro do ambiente do Factory I/O é possível criar plantas industriais virtuais que podem ser controladas por tecnologias externas, tais como: Soft PLCs, PLCs e plataformas micro controladas. A comunicação do factory I/O com controladores externos pode ser feita de diversas maneiras, para esse projeto, foi escolhido o protocolo Modbus TCP/IP.

Altamente flexível, essa ferramenta possibilita a construção uma fábrica virtual completamente personalizada usando uma seleção de peças. Dentre os componentes disponíveis estão: atuadores elétricos, como: esteiras e posicionadores cartesianos; atuadores eletropneumáticos, como: pistões acionados por solenoide; e sensores comuns no setor industrial, tais como: cortina de luz, capacitivo e indutivo. Além dos componentes discretos para construção da fábrica, existem também módulos prontos, como: o paletizador, e o tanque de água.

Paralelo à possibilidade de criação de uma cena própria, o Factory I/O ainda disponibiliza várias cenas prontas que exemplificam aplicações típicas dos componentes e que podem ser usadas pelo usuário.

1.2 Outseal PLC

A Outseal é a primeira empresa indonésia a atuar no setor de desenvolvimento de tecnologia para automação. A companhia trabalha com o desenvolvimento de CLPs *open source*, assim como da ferramenta para sua programação: o Outseal Studio [2].

Os PLCs da Outseal são baseados no bootloader das placas embarcadas Arduino, de modo que o código escrito no Outseal Studio pode facilmente ser carregado em placas Arduino [2].

Embora ainda não esteja em sua versão final, o Outseal Studio conta com todas as funcionalidades essenciais da linguagem ladder, além de outros recursos, tais como: geração de PWM, controle PID e suporte ao protocolo Modbus RTU.

O Outseal PLC foi escolhido como principal interface de programação para implementação das lógicas de controle ao longo do trabalho. Ressalta-se que, um dos recursos essenciais para a execução do projeto é o suporte ao protocolo de comunicação Modbus. Na versão utilizada no projeto (Outseal Studio 3.6 Beta 5) somente é possível utilizar as funções 01 a 06 do protocolo.

1.3 Sistemas SCADA

O termo SCADA é derivado da expressão *Supervision, Control and Data Acquisition*, ou `Supervisão, Controle e Aquisição de Dados`. Os sistemas SCADA são amplamente utilizados na industria junto a automatização de processos, realizando um papel essencial na centralização do monitoramento e controle de uma ou mais linhas produtivas.

Com a conexão dos controladores de uma planta a um sistema SCADA, é possível realizar a coleta de todos os dados de saída, entrada e internos de cada um desses dispositivos. Os dados da planta são então centralizados, e podem ser exibidos de forma conveniente em interfaces gráficas.

Além de monitoramento, o sistema SCADA também permite a escrita de valores nos controladores de sua rede, seja essa escrita manual - através da interface gráfica, por exemplo; ou por meio de rotinas pré-construídas, chamadas de *scripts*.

1.4 Protocolo de Comunicação Modbus

O Modbus é um protocolo de comunicação desenvolvido pela empresa Modicon em 1979. Hoje em dia este é o protocolo mais utilizado para redes de comunicação industriais e sistemas SCADA. O protocolo é especializado na centralização da troca de dados de entrada e saída entre dispositivos de controle [3].

O funcionamento do protocolo é baseado em uma rede cliente servidor (*Client-Server*) para monitorar e controlar os dispositivos conectados. Diversos servidores são conectados na rede, cada um com seu ID próprio (valores válidos: 1 a 254). O cliente controla o fluxo de dados no barramento, sendo capaz de requisitar ações de leitura e escrita nos dispositivos servidores, seja em suas portas I/O ou memória interna [3].

Existem duas versões do protocolo Modbus: RTU e TCP/IP

1.4.1 Modbus RTU

A versão RTU é a clássica do protocolo, ela é compatível com canais de comunicação físicos baseados na tecnologia USART (*Universal Serial Assincronous Receiver-Transmiter*).

O padrão de tensão do canal que transmite Modbus RTU é comumente o RS-485, pois este utiliza a tecnologia *differential signaling*, o que possibilita a montagem de uma rede com multiplos dispositivos. No entanto, também é possível utilizar o padrão RS-232 para redes com apenas dois dispositivos (*single-endend signaling*).

Já a velocidade de transmissão varia entre 1200 e 115200 bps; e a topologia de rede utilizada é a barramento.

Uma rede Modbus RTU genérica é mostrada na Figura 1.

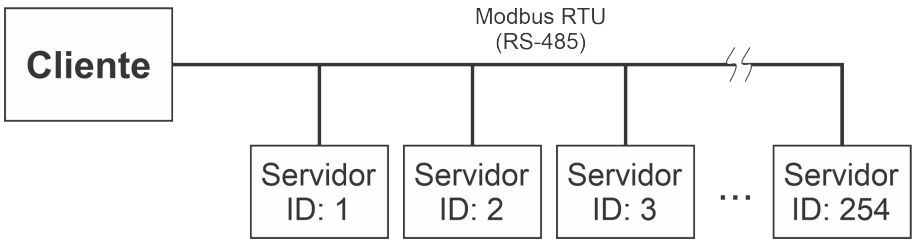

Figura 1: Topologia de rede Modbus RTU.

Quanto à transferência de dados na versão RTU, uma informação enviada pelo cliente diversos Bytes, formando o pacote Modbus RTU. Os Bytes são enviados de forma independente, porém consecutiva; o receptor deve ser então capaz de identificar um pacote completo, isso é, detectar seu início e fim. Para isso, segundo a documentação do Modbus, foi determinado que o tempo máximo entre Bytes transmitidos é de 1.5 vezes a duração de um caractere Modbus, e o tempo mínimo entre o fim de um pacote e início do seguinte é de 3.5 caracteres Modbus. Esse parâmetro é bastante importante e garante a leitura correta dos pacotes durante a implementação do protocolo.

Caractere Modbus: Um caractere Modbus é essencialmente uma unidade de tempo no contexto de transmissão de dados por meio do protocolo na versão RTU. Esse valor se refere ao tempo gasto para transmitir um Byte de dados úteis, esse valor é dependente do baudrate e também das configurações da serial.

Exemplificando: Se o canal serial tem baudrate de 9600bps, dado que o Modbus utiliza 11 bits para transferir um Byte de informação útil, o tempo de transmissão de 1 caractere completo será o tempo de envio de 11 bits a uma velocidade de 9600bps, ou seja, cerca de 1.15ms. Nessa situação, o tempo equivalente a 3.5 caracteres é de cerca de 4.01ms.

A documentação indica, no entanto, que para baudrates acima de 22000 bps, um valor fixo deve ser utilizado para o tempo de 3.5 caracteres, sendo este 1.750ms

1.4.2 Modbus TCP/IP

O TCP/IP é o protocolo mais comum de transporte de dados via internet, provendo uma comunicação confiável entre máquinas [3].

A tecnologia Ethernet tornou-se o padrão comercial para construção de redes de computadores, e portanto sua aparição no ramo industrial foi natural. Com a evolução da tecnlogia, a implementação de uma rede Ethernet garante simplicidade, baixos custos e

compatibilidade generalizada com outros dispositivos. O uso da versão TCP/IP do protocolo Modbus permite uma integração generalizada do processo fabril com a intranet da empresa [3].

As características da camada física do protocolo Modbus TCP/IP seguem aquelas definidas na norma IEEE 802.3, que define as características para implementação de uma rede Ethernet. Ressalta-se que, diferentemente do Modbus RTU, o Modbus TCP/IP trabalha, normalmente, com uma topologia de rede estrela, o que permite a criação de uma rede com múltiplos dispositivos.

A troca de informações na versão TCP/IP é semelhante à RTU, existe um dispositivo Cliente, que controla o fluxo de dados na rede, podendo realizar funções de leitura e escrita nos Servidores, que recebem IDs entre 1 e 254. No entanto, esses 254 servidores podem ser alocados de forma arbitrária a endereços IP ao longo da rede Ethernet.

Na Figura 2 é mostrada a topologia de uma rede Modbus TCP/IP genérica.

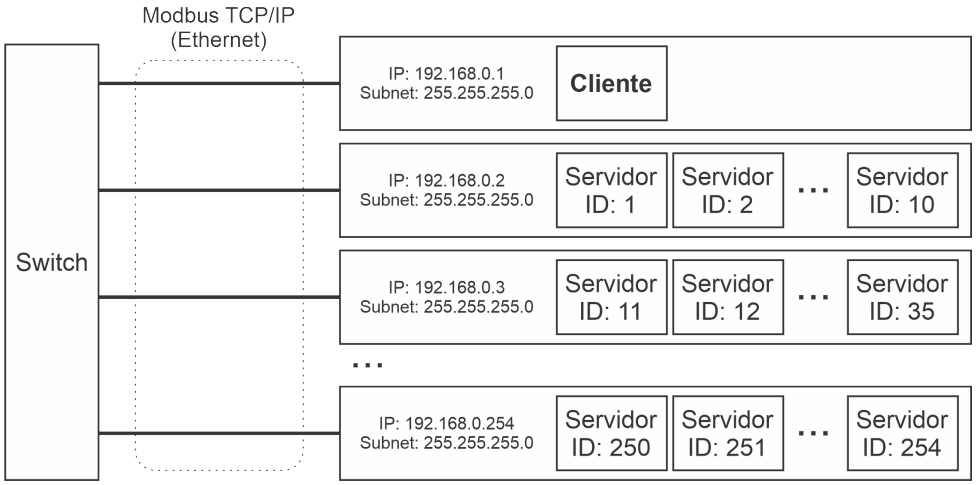

Figura 2: Topologia de rede Modbus TCP/IP.

1.4.3 Tipos de Dados Modbus

O protocolo Modbus utiliza 4 tipos de dados para transferência de informações. Esses são classificados quanto às suas permissões para leitura e escrita, e também pelo tipo de informação que ele carrega. Os nomes atribuídos aos dados Modbus são: **Coil**, **Input Status**, **Holding Register** e **Input Register**.

Os dados Coil e Input Status são do tipo booleano, isso é, são usados para variáveis do tipo True/False ou Ligado/Desligado. Já os dados Holding Register e Input Register são capazes de armazenar valores numéricos discretos.

Quanto às permissões de acesso das variáveis Modbus, elas podem ser classificadas como somente leitura (R), ou leitura e escrita (RW). As variáveis Input Status e Input Register são utilizado para armazenar dados de entrada, e, portanto, são do tipo somente leitura. Já as variáveis Coil e Holding Register são comumente usadas para saídas, e podem receber ações de leitura e escrita.

Na Tabela 1 é resumido os tipos de variável Modbus, com suas classificações.

Tabela 1: Tipos de Variáveis do Protocolo Modbus.

Nome	Permissão	Tipo de Dado
Coil	RW	Booleano
Input Status	R	Booleano
Holding Register	RW	Numérico
Input Register	R	Numérico

1.4.4 Pacote de Dados Modbus

O campo de informação do pacote de Modbus é compartilhado entre as versões Modbus RTU e TCP/IP. Os dados são divididos entre 3 partes principais:

- **ID (1 Byte):** O campo de ID é o primeiro que aparece no pacote Modbus, ele identifica o ID do servidor que está sendo requisitado. No caso do pacote enviado pelo cliente, o ID indica o endereço para qual o pacote deve ser enviado; quando o pacote é de resposta, ou seja, enviado pelo servidor, esse campo indica a origem do pacote.

- **Função (1 Byte):** O funcionamento do Modbus é baseado em funções, elas são codificadas por um byte, e indicam o tipo de ação que é requisitada pelo cliente. Algumas das funções Modbus mais comumente usadas são mostradas na Tabela 2.

Tabela 2: Funções Modbus Comumente Utilizadas.

Função	Definição	Descrição
01	Read Coil	Leitura de Saída(s) Booleana(s)
02	Read Input Status	Leitura de Entrada(s) Booleana(s)
03	Read Holding Register	Leitura de Saída(s) Numérica(s)
04	Read Input Register	Leitura de Entrada(s) Numérica(s)
05	Write Single Coil	Escrita de Saída Booleana
06	Write Single Register	Escrita de Saída Numérica
15	Write Multiple Coils	Escrita de Múltiplas Saídas Booleanas
16	Write Multiple Registers	Escrita de Múltiplas Saídas Numéricas

  Vale ressaltar que os servidores são capazes de informar erros ao cliente por meio deste campo. Para isso, a função requisitada é retornada com o bit mais significativo setado, por exemplo: um erro na execução da função 0x03 seria respondido com este campo no valor 0x83.

- **Parâmetros de Função (4+ Bytes):** O campo de parâmetros tem pelo menos 4 Bytes de tamanho, e o significado de seus dados é dependente da função Modbus.

Exemplificando, para o caso da função 01 (Read Coil), são utilizados dois bytes para indicar o primeiro endereço de um conjunto de coils, e os segundos dois bytes indicam a quantidade de coils a serem lidos a partir desse endereço.

A função 15 (Write Multiple Coils) já utiliza, além dos 4 bytes para indicação do alvo da ação, bytes extras para carregar os valores a serem escritos nos coils.

Particularidades do pacote Modbus RTU: A versão RTU do protocolo Modbus, conta ainda com 2 Bytes extras ao fim do pacote básico:

- **CRC - Cyclic Redundancy Check (2 Bytes):** Mecanismo de detecção e erros pelo método de Checagem de Redundância Cíclica.

Particularidades do pacote Modbus TCP/IP: Na versão TCP/IP existem 6 bytes que precedem os dados do pacote Modbus básico, esses dados são:

- **TI - Transaction Identifier (2 Byte):** Utilizado para identificação dos pacotes enviados, possibilitando o controle de fluxo de dados.
- **PI - Protocol Identifier (2 Bytes):** Utilizado para identificar sub protocolos, possibilita a implementação de funções personalizadas. Valor padrão: 0x0000, (Modbus Convencional).
- **Length (2 Bytes):** Indica o comprimento do restante do pacote em bytes.

Para a transmissão, todo esse conjunto de dados (TI + PI + Length + Modbus) irá formar apenas o campo de dados do pacote TCP/IP, que é precedido pelo cabeçalho do protocolo.

Na Figura 3 é possível comparar os formatos dos pacotes Modbus em suas diferentes versões.

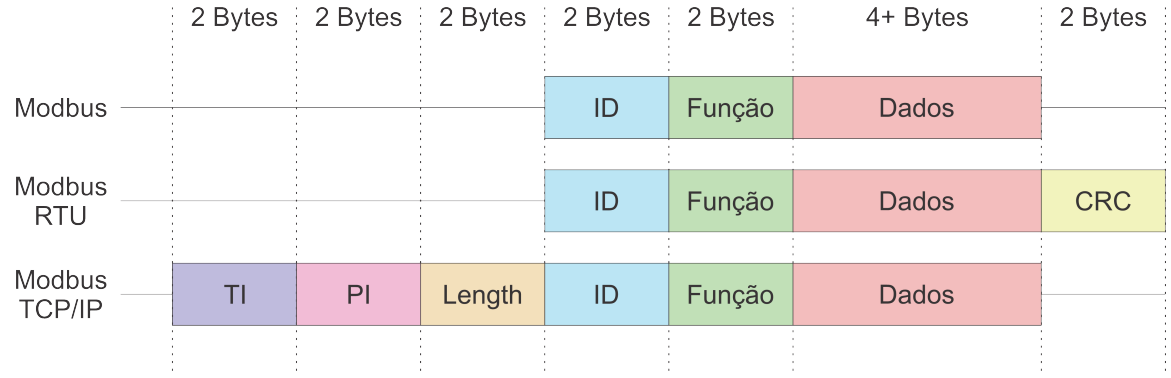

Figura 3: Formato do pacote Modbus RTU e TCP/IP.

1.4.5 Gateway Modbus

Como já citado, o Modbus é o protocolo de comunicação mais utilizado na industria, dessa forma, grande parte das fabricantes de equipamentos de automação (como, por

exemplo, CLPs) adicionam compatibilidade com o protocolo, normalmente para a versão RTU, cujo canal de comunicação é mais robusto para o chão de fabrica.

No entanto, muitas vezes é desejado uma compatibilidade com a versão TCP/IP por seu potencial de integração. Embora ambos os protocolos sejam Modbus, eles apresentam diferenças consideráveis na estrutura de seus pacotes, e, portanto, não podem ser conectados diretamente.

A solução é a criação de um gateway, que é um dispositivo capaz de estabelecer uma comunicação entre dispositivos que utilizam protocolos de comunicação diferentes. Isso é possível pois o gateway é capaz de compreender os dois protocolos usados. Ao receber dados do protocolo A, o gateway é capaz de interpretá-los, e converter a mensagem para protocolo B, e vice-versa. Enquanto que o gateway resolve o problema em questão, sua maior desvantagem é a latência adicional gerada no canal de comunicação devido à conversão, essa grandeza deve ser conhecida para garantir que não afete o funcionamento da rede

Para o Modbus, é possível criar um gateway Modbus, capaz de converter os pacotes do Modbus RTU para o Modbus TCP/IP, e vice-versa. No entanto, deve-se atentar ao cliente Modbus, que é um dispositivo único. Ele pode estar no lado RTU ou TCP/IP da rede, o que gera dois casos diferentes.

Cliente RTU: Caso o cliente seja conectado no lado RTU da rede, a função do gateway será monitorar o canal RTU. Quando um pacote vindo do cliente é identificado, é feita a conversão para o TCP/IP e o dado é encaminhado para o canal Ethernet.

Uma resposta é então aguardada pelo cliente e , caso o servidor requisitado seja um servidor Modbus TCP/IP, a resposta será coletada pelo gateway, que converte a informação de volta para RTU, e a injeta no canal original.

Na Figura 4 é mostrada um exemplo de rede Modbus com gateway e cliente RTU.

Cliente TCP/IP: Já no caso de o cliente trabalhar com Modbus TCP, o gateway também fica encarregado de fazer o controle do fluxo de dados entre os dispositivos.

Diferentemente do cliente RTU, o cliente Modbus TCP/IP não necessariamente aguarda a resposta de uma requisição para enviar a próxima. Esse controle de fluxo de informações é natural do Modbus TCP/IP e previsto no protocolo, no entanto, o comportamento em questão não é compatível como o Modbus RTU, e causaria certamente colisão de dados entre os servidores RTU e o gateway no canal RS-485.

As colisões do canal RTU podem ser evitadas com um manejo correto dos pacotes pelo gateway, o que aumenta o grau de complexidade do dispositivo.

Na Figura 5 é mostrado um exemplo de rede Modbus mista com cliente TCP/IP.

O gateway Modbus é um dispositivo necessário para o desenvolvimento desse projeto, uma vez que o Factory I/O oferece suporte apenas para o Modbus TCP/IP, enquanto que o Outseal Studio permite uso apenas do Modbus RTU.

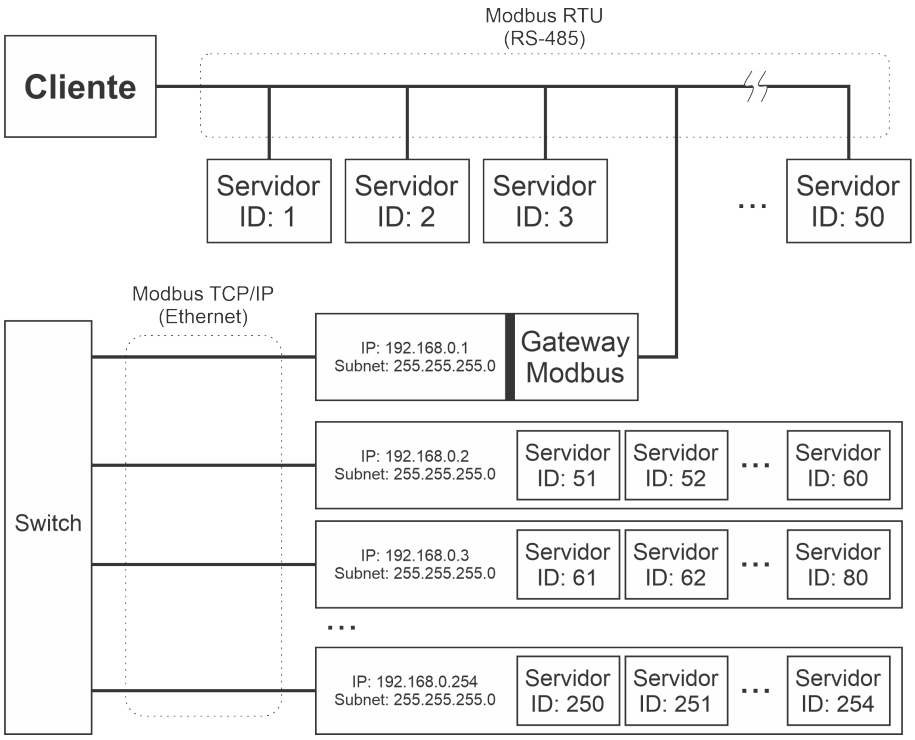

Figura 4: Topologia de rede Modbus mista com cliente RTU.

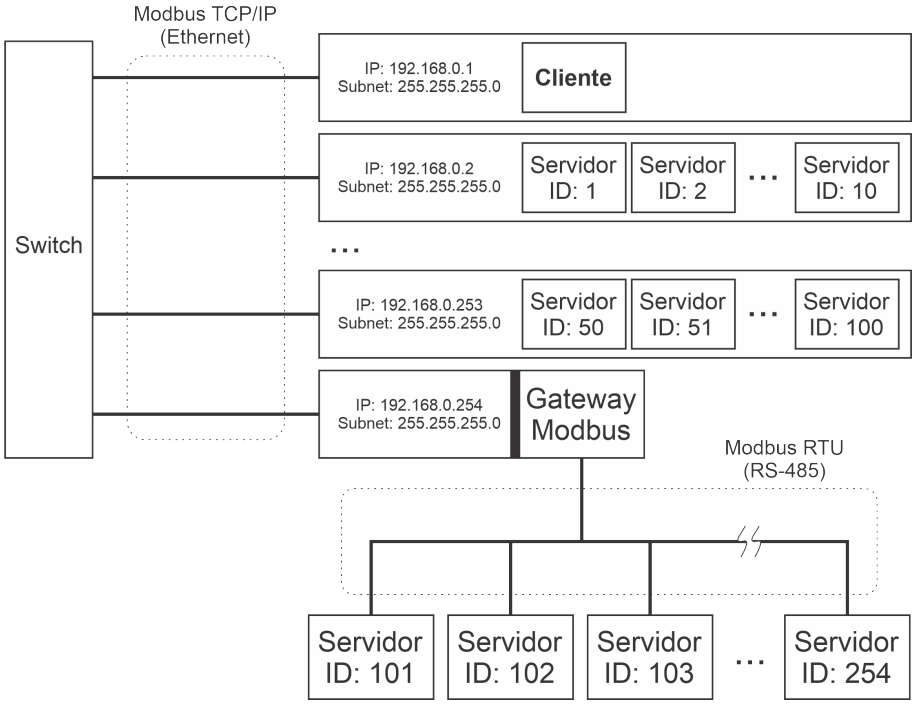

Figura 5: Topologia de rede Modbus mista com cliente TCP/IP.

1.5 Objetivo Geral

O objetivo primário da pesquisa foi estabelecer um conjunto de etapas para o uso inicial da ferramenta Factory I/O no contexto de sistemas SCADA. Os casos de estudo serão associados ao contexto educativo das disciplinas de automação industrial, robótica, instrumentação, controle e desenvolvimento de sistemas supervisórios.

1.5.1 Objetivos Específicos

- Desenvolver cenas de plantas fabris no Factory I/O;
- Criar um algoritmo de controle para as plantas desenvolvidas;
- Codificar os algoritmos de controle em linguagem Ladder no Outseal Studio;
- Elaborar um algoritmo de gateway Modbus;
- Implementar um gateway Modbus na plataforma micro controlada Arduino;
- Estabelecer uma rede de comunicação entre o Outseal PLC e o Factory I/O;
- Documentar as etapas para criação de uma rede de comunicação entre o Factory I/O e o Outseal PLC.