

**UNIVERSIDADE DE CAXIAS DO SUL – UCS
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E
ENGENHARIA**

BRUNO SANTOS DE DAVID

**DESENVOLVIMENTO DE COMUNICAÇÃO DE DADOS
COM CONTROLADORES CNC's**

CAXIAS DO SUL

2022

BRUNO SANTOS DE DAVID

**DESENVOLVIMENTO DE COMUNICAÇÃO DE DADOS
COM CONTROLADORES CNC's**

Trabalho de Conclusão de Curso em Engenharia de controle e Automação apresentado como requisito parcial para à obtenção do título de Engenheiro de Controle e Automação na Área do Conhecimento de Ciências Exatas e Engenharia da Universidade de Caxias do Sul.

Orientador:
Prof. Me. Ricardo Leal Costi

CAXIAS DO SUL

2022

David, Bruno S. De Desenvolvimento de Comunicação de Dados com Controladores CNC's/Bruno Santos De David – Caxias do Sul, 2023- 54 p.: il. ; 30 cm.

Orientador: Prof. Me. Ricardo Leal Costi

Trabalho de Conclusão de Curso – Universidade de Caxias do Sul, 2023.

I. Me. Ricardo Leal Costi. II. Universidade de Caxias do Sul. III. Área do Conhecimento de Ciências Exatas e Engenharias. IV. Graduação em Engenharia de controle e Automação

BRUNO SANTOS DE DAVID

**DESENVOLVIMENTO DE COMUNICAÇÃO DE DADOS EM ETHERNET COM
CONTROLADORES CNC's**

Trabalho de Conclusão de Curso em Engenharia de controle e Automação apresentado como requisito parcial para à obtenção do título de Engenheiro de Controle e Automação na Área do Conhecimento de Ciências Exatas e Engenharia da Universidade de Caxias do Sul.

Aprovado em...../...../.....

Banca Examinadora:

Prof. Me. Ricardo Leal Costi
Universidade de Caxias do Sul - UCS

Prof. Me. Ângelo Zerbetto Neto
Universidade de Caxias do Sul - UCS

Eng. Paulo Rostirolla Junior
Universidade de Caxias do Sul - UCS

Resumo

O presente trabalho explora os conceitos da indústria 4.0 para aquisição de dados de processo de máquinas CNC. As máquinas CNC são utilizadas para fabricação de peças metálicas, de acordo com as características de cada peça a ser produzida. O CNC é programado para executar uma série de etapas de remoção de material, programação essa que é executada por um programador ou operador. Com isso existe a possibilidade de um programa estar exercendo mais ou menos força que o recomendado e com isso existe a possibilidade de desgaste prematuro das ferramentas de corte, ou produção de peça com defeito que em alguns casos é detectada somente pelo cliente após o uso causando um problema de garantia.

A utilização de dados de fabricação coletados, pode ser utilizado no processo de decisão de procedimentos, processos e até máquinas específicas para realização de cada etapa do processo de fabricação. Tais dados podem ser dados atuais ou históricos e a decisão a ser tomada pode e deve levar em consideração ambos os casos pois com o passar do tempo temos modificações ou desgastes nos equipamentos que podem influenciar no processo de fabricação e na qualidade do produto devidamente industrializado.

O estudo de algumas das possibilidades existentes na atualidade para buscar identificar viabilidade, da implementação dessa aquisição de dados de máquinas baseadas em comunicação Ethernet, serial 232 presente nas máquinas e em alguns casos em que a máquina possui uma porta USB, a mesma deverá ser estudada para validar a utilização dela, esse estudo será executado se valendo de informações comerciais e técnicas dos fabricantes para tal análise.

Palavras-Chave: Indústria 4.0, Aquisição de dados, CNC, Decisão, Fabricação. Comunicação.

Abstract

The present work explores the concepts of industry 4.0 for data acquisition from CNC machines. CNC machines are used to manufacture metal parts, according to the characteristics of each part to be produced. The CNC is programmed to perform a series of material removal steps, which programming is performed by a programmer or operator. With that, there is the possibility that a program is exerting more or less force than recommended and with that there is the possibility of premature wear of the cutting tools, or production of a defective piece that in some cases is only detected by the customer after use causing a warranty issue.

The use of collected manufacturing data can be used in the decision-making process of procedures, processes and even specific machines for carrying out each stage of the manufacturing process. Such data can be current or historical data and the decision to be taken can and should take both cases into account because over time we have modifications or wear and tear on equipment that can influence the manufacturing process and the quality of the properly industrialized product.

The study of some of the current possibilities to seek to identify the feasibility of implementing this data acquisition from machines based on Ethernet communication, serial 232 present in the machines and in some cases where the machine has a USB port, it should be studied to validate its use, this study will be carried out using commercial and technical information from the manufacturers for such analysis.

Key words: Industry 4.0, Data acquisition, CNC, Decision, Manufacturing, Communication.

Lista de Figuras

Figura 1 Exemplo de Máquina CNC	15
Figura 2 Imagem de trabalho de um CNC.....	16
Figura 3 Comunicação de dados máquina a máquina.	17
Figura 4 Estrutura do protocolo Modbus RTU.....	20
Figura 5 Diagrama de tempo do protocolo Modbus.....	21
Figura 6 Esquemático básico da comunicação OPC	22
Figura 7 Mapa do protocolo OPC UA.....	23
Figura 8 Estrutura Geral do OPC UA.....	24
Figura 9 Cenário de Aplicação MTConnect.....	29
Figura 10 Exemplo de rede com MTConnect	30
Figura 11 Cenário de Aplicação SINUMERIK.....	31
Figura 12 Software SiOME página inicial	33
Figura 13 Comunicação e armazenamento em banco de dados.	36
Figura 14 Esquemático do desenvolvimento.....	37
Figura 15 Adição servidor OPC Local Matrikon.	39
Figura 16 Configuração do Servidor OPC Local	40
Figura 17 Script Leitura de Variáveis do Servidor OPC	41
Figura 18 Diagrama Barras Comunicação.....	42
Figura 19 Visualização configuração Google Sheets API.....	43
Figura 20 Compartilhamento Planilha Google Sheet	43
Figura 21 Servidor OPC Conectado	44
Figura 22 Software em execução	45
Figura 23 Tabela de Dados Armazenados no Google Sheets.....	46

Lista de Tabelas

Tabela 1 Tabela de Variáveis SINUMERIK	33
Tabela 2 Tabela de Comparação OPC UA e Modbus	35
Tabela 3 Relação de Varáveis Controlador Mazak	38

Lista de Siglas

CLP	Controlador Lógico Programável
CNC	Comando Numérico Computadorizado
COM	Modelo de Objeto Componente
DCOM	Modelo de Objeto Componente Distribuído
SCOM	Modelo de Objeto Específico do Cliente (<i>Customer Specific Object Model</i>)
I/O's	Dispositivos de Entradas e saídas (Sensores e acionamentos)
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
IHM	Interface Homem-Máquina
JSON	Arquivo de Certificação Acesso Remoto
OEE	<i>Overall Equipment Effectiveness</i>
OLE	<i>Object Linking and Embedding</i>
OPC	<i>OLE for Process Control</i>
OPC UA	OPC de Arquitetura Unificada
OSI	<i>Open Systems Interconnect</i>
TCP	Protocolo de Controle de Transmissão
TSN	Rede Sensível ao tempo (<i>Time-Sensitive Networking</i>)
XML	<i>Extensible Markup Language</i>

Sumário

1. INTRODUÇÃO	11
1.1. Justificativa	12
1.2. Objetivo	13
1.2.1. Objetivos Específicos	13
1.3. Limitações Trabalho	13
2. REVISÃO BIBLIOGRÁFICA	14
2.1. Máquinas CNC	14
2.2. Comunicação de Dados	16
2.2.1. Ethernet	17
2.2.2. Modbus	19
2.2.3. OPC UA	21
2.2.4. Arquivo XML.....	24
2.2.5. Segurança de Dados	26
2.3. Fabricantes Equipamentos CNC	28
2.3.1. Controladores Mazak.....	28
2.3.2. Controladores Siemens	31
3. METODOLOGIA	35
3.1. Esquemático desenvolvimento	37
3.2. Variáveis Disponíveis	38
3.3. Configuração Servidor OPC	39
3.3.1. Script para Comunicação OPC.....	40
3.4. Armazenamento de dados	42
4. FUNCIONAMENTO E TESTE	44
5. CONCLUSÃO	47
REFERÊNCIAS	49
APENDICE A- SCRIPT SOFTWARE	51

1. INTRODUÇÃO

Com o advento da indústria 4.0 e a crescente evolução industrial além da globalização, os processos fabris de usinagem e desbaste estão cada vez mais padronizados devido grande parte ao crescente uso da tecnologia CNC, que nos permite executar tais processos com erros cada vez menores e precisões cada vez maiores. Assim sendo, a concorrência entre empresas se torna cada vez maior e presente em qualquer segmento industrial, o que traz a importância de ter-se decisões assertivas e o menor custo possível de fabricação ou processo, quando o produto é um serviço, para que se consiga atingir um valor com cujo qual venha-se a ganhar a concorrência e conseguir fazer tal fornecimento.

Para chegar neste esse ajuste de valores, de forma eficiente é de suma importância a aquisição de dados dos equipamentos do parque fabril, com o intuito de definir o consumo energético, uso de ferramentas, matéria prima, entre outros. Essa aquisição de dados pode se dar de formas diferentes, via leitura direta de informações por sensoriamento em processos específicos de aquisição de dados ou pela extração de tais dados diretamente dos equipamentos presentes no chão de fábrica quando permitido por eles. O fato é que se necessita que, de alguma forma se tenha geração de tais dados, para que se consiga transmiti-los de alguma forma e assim seja executada uma análise desses dados, para que então se consiga uma tomada de decisão de forma assertiva. Essa aquisição direta normalmente se dá através do uso de redes de comunicação, onde diferentes fabricantes possuem diferentes protocolos de comunicação que podem ser utilizados.

Além dos protocolos de comunicação é interessante também que esses equipamentos possuam os dados disponíveis para essa leitura, e que se consiga ter acesso aos mesmos, sabendo onde buscar, endereços de rede, tamanho e formato, de tais dados para que tenham-se os valores corretos de cada um deles, a fim de torná-los utilizáveis, além de fazer a aquisição dos dados importantes e não gerando trabalho na leitura de dados ineficientes ou até mesmo dados inúteis para a tomada de decisão do processo de fabricação em específico.

A indústria 4.0 destina-se ao uso, avaliação e análise de dados de produção em sistemas de TI de nível corporativos. Os controladores dos maquinários registram uma grande variedade de informações de processo e produção e os tornam disponíveis no nível corporativo. Para que tal aquisição de informações seja possível é necessário que se tenham

protocolos de comunicação padrões para que se consiga, no destino final, o entendimento dos dados gerados no chão de fábrica.

Tais protocolos de comunicação são diversos e padronizados por organizações internacionais, cada um deles possui sua formalização e sequência de dados específicas, que forma o padrão do protocolo, nesse padrão estão informações como local de início e fim da mensagem, endereços, número de informações máximas a serem transmitidas, as informações propriamente ditas e informação de verificação de mensagem, para que se certifique que a mensagem foi entregue de forma correta.

1.1. Justificativa

Esse trabalho busca tratar da aquisição de dados de equipamentos CNC's para que dados de processo, tempo, consumo, desgaste de equipamento e ferramentas, falhas, qualidade do produto/processo. Disponibilizando essas informações em banco de dados, para que com tais dados consiga-se definir o melhor processo ou a melhor estratégia de fabricação a fim de se otimizar custos e ter melhor desempenho no processo. Por se tratar de uma máquina programável, por motivo de erro de setup ou até pela troca da matéria prima ou das pastilhas de desgaste, essa programação pode estar com as velocidades erradas para avanço de ferramenta e velocidade do *spindle* de corte, gerando problemas de qualidade na peça produzida. As informações coletadas podem ser utilizadas para ajustes de programa, definição de valores hora-máquina e até mesmo qual equipamento e quais as configurações de setup melhores para cada etapa do processo de fabricação.

Essa tomada de decisão pode ser baseada em dados históricos ou dados em tempo real, dependendo da interação entre tomadores de decisão e o processo de fabricação, para agilizar a fabricação ou até mesmo modificar tempos de processo com o intuito de reduzir a despesa de fabricação do produto a fim de balancear custo com agilidade de fabricação e entrega de material, visando sempre o ponto mais importante em cada determinado momento do processo de fabricação até a finalização do produto.

1.2. Objetivo

Como objetivos para o trabalho, tem-se o estudo de desenvolvimento de aquisição de dados de equipamentos CNC para a disponibilização deles através de um banco de dados, para posterior tratamento e análise a fim de que eles sejam utilizados de forma correta.

1.2.1. Objetivos Específicos

Os objetivos específicos se destacam em:

- Realizar um estudo de fabricantes de CNC;
- Realizar um estudo de protocolos de comunicação presentes nos CNC;
- Realizar pesquisa dados disponíveis para aquisição;
- Desenvolver um projeto de coleta de Dados;
- Realizar testes e ensaios;
- Disponibilizar dados para armazenamento;

1.3. Limitações Trabalho

Com os objetivos traçados para o desenvolvimento do trabalho, nessa seção do trabalho se descreve alguns tópicos que não fazem parte do escopo de desenvolvimento do mesmo.

- A aquisição de dados será de servidor OPC, configurado com as variáveis disponíveis em aplicação de comunicação real com equipamento CNC;
- Não haverá tratamento de informações, somente a disponibilização delas;

2. REVISÃO BIBLIOGRÁFICA

Nesse capítulo serão abordados os conceitos de equipamentos, fabricantes dos controladores CNC, protocolos de comunicação utilizados e apresentados dados técnicos a respeito dos conceitos e equipamentos.

2.1. Máquinas CNC

As máquinas denominadas CNC são equipamento que possuem um comando numérico computadorizado, os equipamentos CNC são máquinas controladas por computador para execução do trabalho, na Figura 1 é possível visualizar um exemplo de máquina CNC. Nesse tipo de máquina, o programa de instruções pode ser alimentado diretamente no computador através de uma pequena placa semelhante ao teclado tradicional, ou ainda utilizando-se de um canal de comunicação, serial ou ethernet, presente no painel de comando. O programa de instruções pode ser armazenado na própria memória do computador, onde podem ser armazenados uma série de programas, facilitando o trabalho do programador que pode escrever e editar os programas salvos de acordo com a necessidade. Para a inserção das programações de instrução via utilização do canal de comunicação, normalmente, essas máquinas possuem conversores e programas próprios de cada fabricante do equipamento para fazer a conversão do programa da linguagem de computador para a linguagem de máquina utilizada pelo controlador do CNC. (MECÂNICA INDUSTRIAL,2017).

A máquina CNC é composta pelo computador no qual o programa é alimentado para o trabalho do corte do metal de acordo com os requisitos. Todos os processos de corte que devem ser realizados e todas as dimensões finais são alimentados no computador através do programa. O computador assim sabe o que exatamente deve ser feito e executa todos os processos de corte. A máquina CNC funciona como um robô, que deve ser alimentado com o programa e segue todas as suas instruções. Algumas máquinas ferramentas comuns que podem ser executadas com CNC são: Tornos, Fresas, Perfuratrizes, Centros de usinagens, entre outros. Todas essas máquinas possuem o propósito de remover parte do material de determinada peça para que se atinja a forma e/ou geometria cuja qual foi programada para forma final do processo. (MECÂNICA INDUSTRIAL,2017).

Figura 1 Exemplo de Máquina CNC



Fonte: Adaptado Mecânica Industrial (2017).

Máquinas CNC são equipamentos que trabalham com precisão, cabendo ao operador/programador apenas dar as instruções de no computador, carregar as ferramentas necessárias na máquina e o resto do trabalho é executado automaticamente pelo computador. Como mostrado na figura 2, o computador direciona a máquina-ferramenta para executar várias operações de usinagem de acordo com o programa de instruções alimentadas pelo operador. (MECÂNICA INDUSTRIAL,2017).

Figura 2 Imagem de trabalho de um CNC



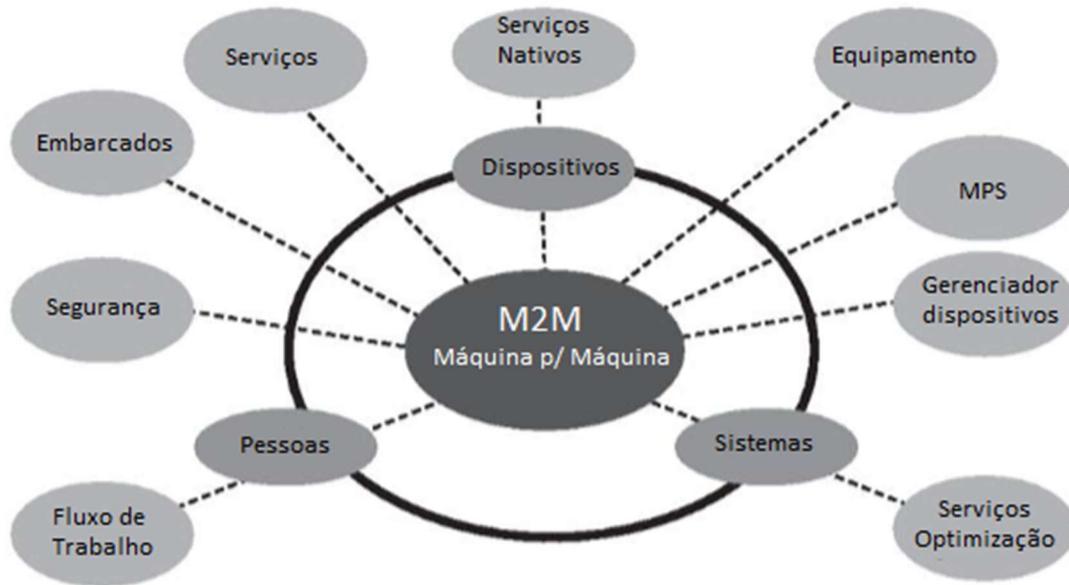
Fonte: Adaptado Mecânica Industrial (2017).

Ao final do trabalho basta realimentar o equipamento e dar novo início de ciclo para que a máquina comece a fazer novo trabalho e como a precisão é fornecida pelo computador e não pelo operador, gerando novas peças padronizadas com medidas e formas exatas, desde que operadas com o mesmo programa de instruções. A maioria das empresas de fabricação está equipada com as máquinas CNC, já que o mercado tem uma grande competitividade. (MECÂNICA INDUSTRIAL, 2017).

2.2. Comunicação de Dados

Conceituando inicialmente a comunicação de dados entre equipamentos/máquinas e a interação que eles possuem, ou podem possuir, com o ambiente em que estão inseridos, pode-se tomar por base a figura 3, que mostra as possíveis interações de equipamentos, o meio onde estão inseridos e possibilidades de contato com o mundo externo. (SACAMONO, JOSÉ BENEDITO, 2018).

Figura 3 Comunicação de dados máquina a máquina.



Fonte: Adaptado SACAMONO, JOSÉ BENEDITO (2018).

Nela pode se perceber que a máquina pode possuir diversas interações que são rotuladas de comunicação, desde a interação entre os dispositivos da máquina com seu controlador central, outros sistemas de serviço e gerenciamento até a interação com pessoas, onde as mesmas podem ser responsáveis por operação da mesma, até a velha forma de coleta de dados manuais, utilizando a interface homem-máquina (IHM), tomando nota das principais informações geradas, tais como, número de peças, tempos de operação, e em alguns casos mais específicos, qualidade das peças produzidas, OEE do equipamento (Eficiência Global do Posto de Trabalho), entre outros. (SACAMONO, JOSÉ BENEDITO, 2018).

2.2.1. Ethernet

A Ethernet é um protocolo de rede que controla o método com o qual computadores e dispositivos se comunicam. O Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE) define Ethernet como protocolo 802.3. É um protocolo de link de dados da camada 2 que é frequentemente usado como parte da pilha TCP/IP. A rede Ethernet é usada para casos em que a velocidade e a confiabilidade da rede são mais importantes que a flexibilidade oferecida por uma solução sem fio, o que a torna a melhor opção para sistemas de missão crítica que precisam manter a transmissão de dados rápida e consistente. (VENTURELLI, MÁRCIO, 2016), (WEISS, OLGA, 2020).

A Ethernet é comumente utilizada pois é uma rede de fácil implementação, que permite diversos protocolos dentro do seu padrão, por ser uma rede padronizada que está em constante evolução, por ser uma rede interoperável e escalar e por poder ser implementada em qualquer ambiente. Caracterizada por usualmente utilizar do cabo par trançado como meio físico e por conectores RJ45, com pinagem definidas pelos padrões T568A e T568B. (VENTURELLI, MÁRCIO, 2016), (WEISS, OLGA, 2020).

Cabe lembrar que de acordo com o modelo OSI (*Open Systems Interconnect*), um sistema de rede de comunicação é dividido em 7 camadas, como se pode ver abaixo:

1. Camada Física
2. Camada de Ligação de Dados ou Enlace de Dados
3. Camada de Rede
4. Camada de Transporte
5. Camada de Sessão
6. Camada de Apresentação
7. Camada de Aplicação

As 4 primeiras camadas são dedicadas ao meio físico, topologias e ao processamento de envio e recebimento de pacotes de comunicação, já as camadas restantes são relacionadas ao protocolo de comunicação utilizado e variam de acordo como tais. (VENTURELLI, MÁRCIO, 2016).

Entrando mais a fundo em cada uma das camadas do modelo OSI tem-se as seguintes definições:

- Camada Física: define especificações elétricas e físicas dos dispositivos. Como características dos cabos, conectores e sua pinagem, temporização, repetidores. Outra função muito importante da camada física é definir se pode haver tráfego de informações em ambos os sentidos acontecendo simultaneamente.
- Camada de Ligação: conhecida como enlace de dados e é responsável por controlar o fluxo de dados, conexão, transmissão, recepção dos dados. Opcionalmente pode corrigir erros de transmissão.
- Camada de Rede: fornece os meios funcionais e procedimentos de transferência de comprimentos de dados variáveis de uma fonte para seu destino em uma rede

diferente. Realiza o roteamento de funções e pode realizar fragmentação, remontagem de erros de entrega.

- Camada de Transporte: responsável pelo controle do fluxo, ordenação e a correção de erros no fluxo de dados, sendo responsável por garantir a entrega e recebimentos dos dados de forma correta. Faz a ligação entre as camadas de aplicação e camadas físicas.
- Camada de Sessão: responsável pela troca de dados e a comunicação entre hosts ela permite que duas aplicações em computadores diferentes estabeleçam uma comunicação, definindo como será feita a transmissão de dados, pondo marcações nos dados que serão transmitidos. Se porventura a rede falhar, os computadores reiniciam a transmissão dos dados a partir da última marcação recebida pelo computador receptor.
- Camada de Apresentação: também conhecida como camada de tradução, é responsável por formatar e converter os dados recebidos em um formato conhecido pelo protocolo para sua transmissão. É nesse nível que pode ser utilizado a criptografia de dados.
- Camada de Aplicação: é a camada de aplicação, onde serão utilizadas as funções especializadas para promover a interação entre máquina e usuário, como por exemplo envio de e-mails, transferência de dados etc.

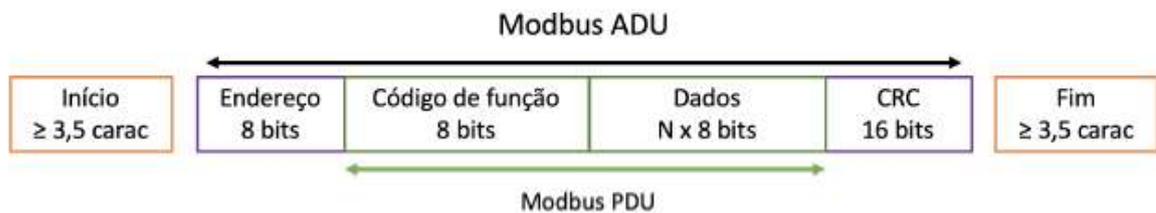
2.2.2. Modbus

O Modbus, criado em 1979 pela Modicon, é um protocolo em nível de aplicação que tem como objetivo comunicar diferentes dispositivos de automação, transferindo dados sob uma camada serial, porém ao longo do tempo recebeu atualizações para ser implementado também em *Transmission Control Protocol/Internet Protocol (TCP/IP)* e *User Datagram Protocol (UDP)* (THOMAS, 2008). O protocolo é aberto desde seu lançamento, e a simplicidade na operação o tornou uma solução de baixo valor, permitindo tornar-se um dos padrões mais utilizados mundialmente (STRACK, 2011).

As atualizações para permitir o uso de redes mais modernas ocasionou a separação da camada única do protocolo para uma camada de protocolo básico, que define a unidade de dados de protocolo (PDU) e uma camada que define a unidade de dados de aplicação (ADU).

O protocolo também possui três diferentes formatos de ADUs: TCP, *American Standard Code for Information Interchange* (ASCII) e unidade de terminal remoto (RTU). Essa última, normalmente utilizada em linhas seriais, possui uma simples estrutura, que pode ser visualizada na Figura 4. Utilizando-se de cabo com um par com malha de baixa impedância e sendo ligada, no lado dos dispositivos de automação a borneiras, e na presença de um computador se utiliza de um conector DB9 com conversor serial. (THOMAS, 2008; NATIONAL INSTRUMENTS, 2019).

Figura 4 Estrutura do protocolo Modbus RTU



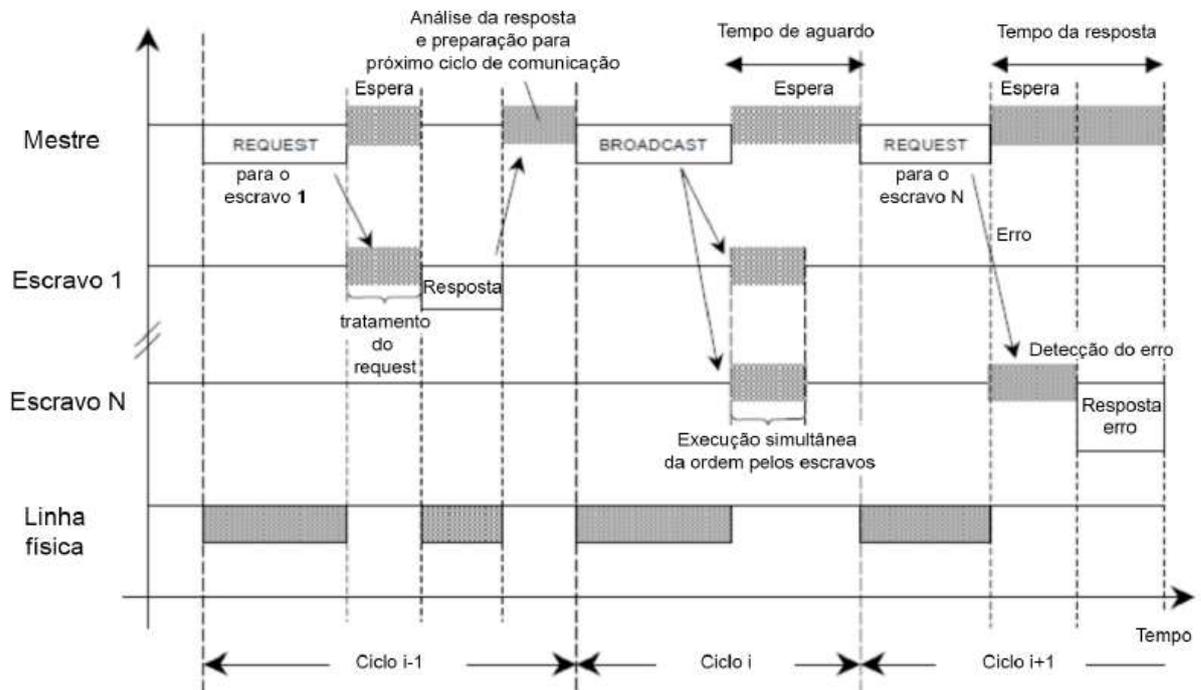
Fonte: Adaptado Strack (2011).

Seu funcionamento baseia-se na troca de mensagens entre um dispositivo mestre e até 247 dispositivos escravos que são identificados por um número de endereço. Na troca de mensagens, o mestre envia uma solicitação a um determinado endereço e o escravo responde (modo *unicast*). No caso do envio de uma requisição com endereço 0, todos os escravos processam o dado, porém nenhum responde (modo *broadcast*). O *Cyclic Redundancy Check* (CRC) são dois bytes verificadores para garantir a integridade dos dados (NATIONAL INSTRUMENTS, 2019).

Cada mensagem pode conter até 256 bytes, sendo 1 byte de endereço, 2 bytes de CRC e até 253 bytes de PDU, que é formado por um byte de código da função e o restante de dados. O protocolo Modbus define 3 aplicações de PDU: *Request PDU* (requerimento); *Response PDU* (resposta) e *Exception Response PDU* (resposta de erro), visualizadas na Figura 5, a qual contém o diagrama de tempo (STRACK, 2011; MODBUS ORGANIZATION, 2012).

É importante mencionar que um ciclo de comunicação (*request e response*) ocorre por vez, sempre aguardando o término do anterior para ir para a seguinte. O tempo de espera entre um dado e outro é de 3,5 caracteres no mínimo e os valores entre os caracteres de uma mensagem não podem ser maiores que 1,5 caracteres (MODBUS ORGANIZATION, 2006).

Figura 5 Diagrama de tempo do protocolo Modbus

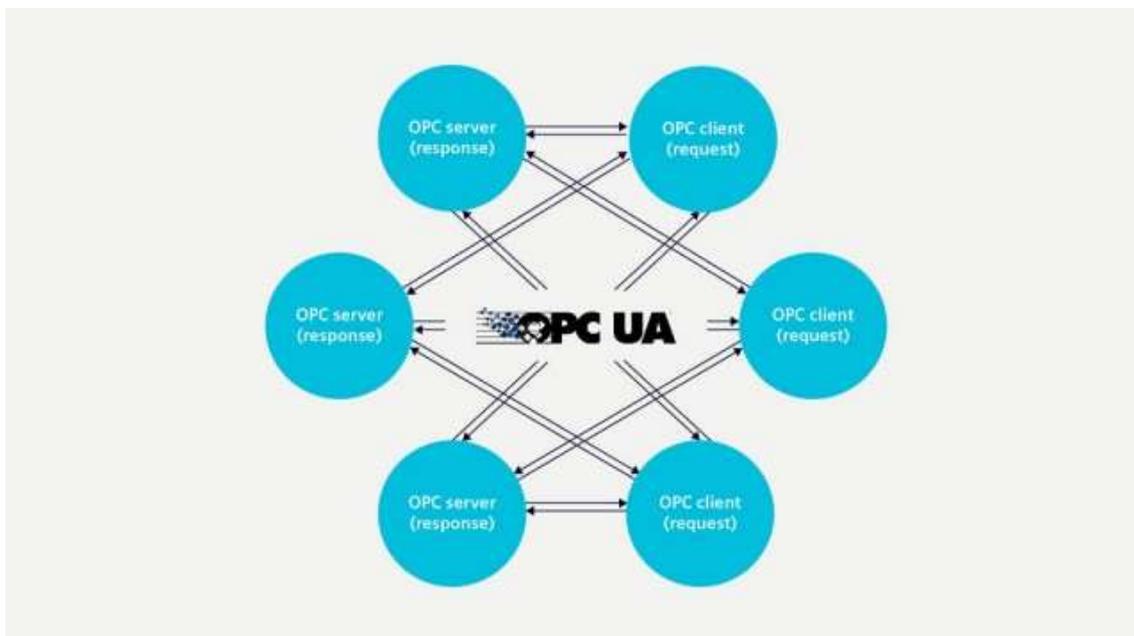


Fonte: Adaptado Modbus Organizatoin (2006)

2.2.3. OPC UA

A primeira versão protocolo OPC foi lançada em 1996, tinha como objetivo fazer uma interface entre dispositivos de controle (PLC's), e as soluções SCADA de mercado, tornando uma solicitação de leitura ou escrita de dados genérica em um solicitação específica de um componente, como mostrado na figura 6, e foi expandindo para um grande número de dispositivos industriais, permitindo que os usuários finais, executassem implementações de sistemas utilizando diferentes produtos de fabricantes diferentes, afim de atingir melhor o objetivo final do projeto, seja ele por reduções de custo ou até mesmo por uma solução específica para determinada necessidade do processo produtivo individual. (LUTH, JIM, 2022).

Figura 6 Esquemático básico da comunicação OPC



Fonte: Adaptado Siemens OPC UA(2022).

O protocolo inicialmente necessitava de uma plataforma Windows para operar para que ela executasse a interpretação do OLE, essas aplicações são conhecidas como OPC clássico, e possui grande desenvolvimento em usuários de diferentes seguimentos, desde automação industrial, automação predial, indústria de processos, aplicações de energia renováveis entre outros. O padrão OPC clássico, é baseado na plataforma Windows utilizando COM/DCOM (Modelo de Objeto Componente Distribuído), para a troca de dados entre componentes e software. As especificações do protocolo fornecem definições separadas para acesso a dados de processo, alarmes, e histórico de dados, principalmente considerando a prioridade das informações. (LUTH, JIM, 2022).

O OPC Clássico, atendeu bem as necessidades por um período de tempo, mas com a evolução da tecnologia, foi requerido que o protocolo também evoluísse, assim em 2008 a *OPC Foundation*, criou o OPC de arquitetura unificada (OPC UA), uma arquitetura independente de plataforma orientada que integra todos os serviços do OPC clássico e agrega mais funcionalidades ao padrão em uma estrutura expansível. (LUTH, JIM, 2022).

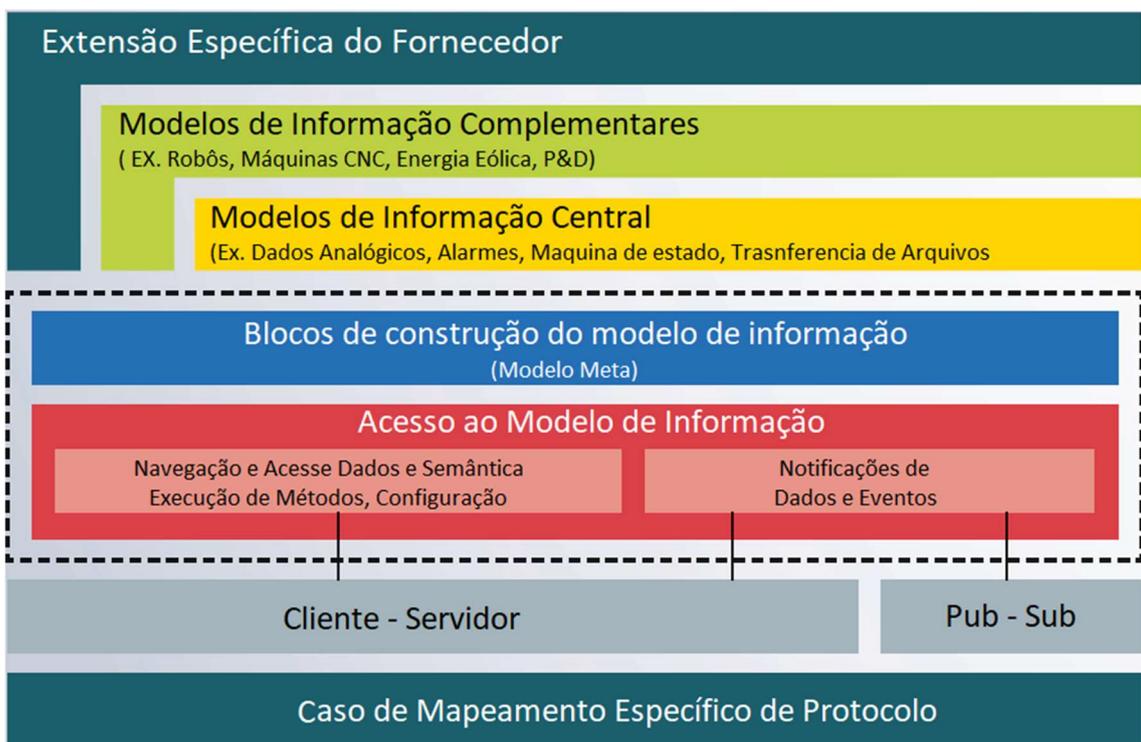
O OPC UA foi projetado para aprimorar e superar as capacidades do OPC clássico, sendo fundamentalmente equivalente a tal, mas com uma série de benefícios. Por ser uma plataforma independente, agora não requer mais uma plataforma Windows para rodar e pode ser utilizada em PLCs, microcontroladores, servidores de nuvem, entre outros, pois pode ser executada em

qualquer sistema, seja ele Windows, *Apple OSX*, Android, Linux etc. O próprio OPC UA fornece a infraestrutura necessária para execução dos serviços máquina para máquina ou máquina para empresa. (OPC FOUNDATION, 2022).

Quando se fala em comunicação com o mundo externo, tem-se a preocupação com a segurança das informações que estão sendo transitadas, e o protocolo OPC UA, também possui essa preocupação possuindo também configurações de conexão, firewalls e restrições de acesso, por controles de usuários, autenticações e encriptação de dados, permitindo que o usuário faça tome as ações necessárias para proteção correta das informações e disponibilização das mesmas para o usuário que irá precisar de tais informações. . (OPC FOUNDATION, 2022).

Pela característica da expansividade o OPC UA torna-se uma plataforma preocupada com atualizações futuras, pensando em novas tecnologias e metodologias inovadoras, como novos protocolos de comunicação, algoritmos de segurança, padrões de codificação ou serviços de aplicativo, podem ser incorporados ao OPC UA, mantendo a compatibilidade com os produtos já existentes. Na figura 7 pode-se ver um esboço do mapeamento estrutural do padrão em OPC UA. (OPC FOUNDATION, 2022).

Figura 7 Mapa do protocolo OPC UA

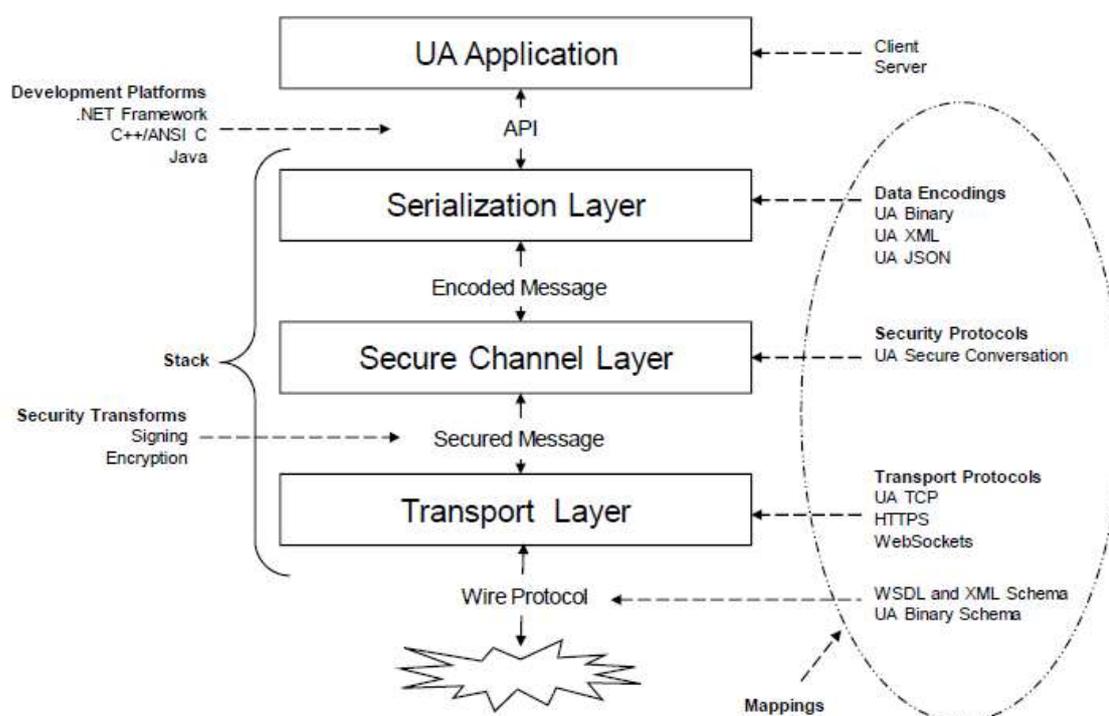


Fonte: Adaptado *OPC Foundation: Unified Architecture* (2022).

Fazendo assim com que os produtos fabricados com OPC UA de hoje, funcionem juntamente com produtos do futuro sem problemas de compatibilidade. Esse modelo de estrutura define as regras e blocos básicos necessários para expor um modelo de informações OPC UA. Permitindo que se utilize de modelos já previamente definidos pela *OPC Foundation*, ou que outras organizações criem seus modelos sobre esses que são previamente definidos para expor suas informações específicas com o OPC UA. (*OPC FOUNDATION*, 2022).

O protocolo OPC UA possui uma estrutura de mensagens, conforme indicado na figura 8, que é mostrada a seguir.

Figura 8 Estrutura Geral do OPC UA



Fonte: Adaptado *OPC Foundation: OPC 10000-6* (2022).

2.2.4. Arquivo XML

O XML é a abreviação de *extensible markup language* e foi criado pelo W3C (*World Wide Web Consortium*) na década de 1990, é um tipo de linguagem de marcação que define regras para codificar diferentes documentos e é entre outras funções, muito utilizado para a criação de Notas Fiscais Eletrônicas, uma vez que essa linguagem de marcação é usada para

estruturar dados de forma hierárquica e legível por máquinas e seres humanos, fornecendo um padrão aos programas e aos programadores. Esse padrão é um formato amplamente aceito para transmissão de dados entre sistemas diferentes. (LOURENTI, A. M., 2022)

Linguagens de marcação são sistemas usados para definir padrões e formatos de exibição dentro de um documento. Em resumo, funcionam para definir como um determinado conteúdo vai ser visualizado na tela ou como os dados serão distribuídos. Essa codificação interna é feita pelo uso de marcadores ou tags. A grande vantagem do XML é facilitar o compartilhamento de dados. Seu armazenamento é feito em texto, permitindo que a leitura seja feita por diferentes aplicativos. Os arquivos nesse formato podem ser atualizados sem perda de informações importantes. (LOURENTI, A. M., 2022)

O XML é composto por elementos, que são as unidades básicas de informação. Cada elemento é definido por uma tag de abertura e uma tag de fechamento. Os elementos podem conter texto, outros elementos aninhados e atributos. Os atributos fornecem informações adicionais sobre os elementos e são definidos dentro das tags de abertura.

A estrutura de um arquivo XML é baseada em um formato hierárquico, o que significa que os elementos são organizados em uma árvore de elementos. Cada elemento pode conter outros elementos, chamados de elementos filhos, formando uma estrutura aninhada. Essa estrutura é fundamental para representar relações e hierarquias de dados. A estrutura do arquivo XML é definida por um conjunto de regras chamadas de *Document Type Definition* (DTD) ou *XML Schema Definition* (XSD). Essas definições especificam quais elementos são permitidos, quais atributos podem ser usados e como eles devem ser estruturados. Com base nessas definições, um analisador XML pode validar se um arquivo XML está em conformidade com a estrutura esperada.

Outra característica importante do XML é que ele é extensível, o que significa que você pode definir suas próprias estruturas e elementos personalizados de acordo com as necessidades do seu domínio. Essa flexibilidade torna o XML uma das mais utilizadas escolhas para representar e compartilhar dados estruturados em diversos campos, desde integração de sistemas até configuração de aplicativos e representação de documentos.

2.2.5. Segurança de Dados

A cibersegurança é um campo que se dedica a proteger sistemas de computadores, redes, dispositivos e dados contra ameaças cibernéticas. Com o aumento da conectividade e dependência da tecnologia, a segurança cibernética tornou-se uma preocupação crítica em várias áreas. A cibersegurança envolve uma variedade de práticas e medidas para mitigar riscos e proteger ativos digitais. Alguns dos principais elementos da cibersegurança incluem:

- **Proteção de dados:** Envolve a implementação de medidas para garantir a confidencialidade, integridade e disponibilidade dos dados. Isso pode incluir criptografia de dados, autenticação de usuários, controle de acesso, backups regulares e proteção contra perda de dados.
- **Proteção de rede:** Envolve a segurança das redes de computadores, como firewalls, detecção e prevenção de intrusões, monitoramento de tráfego, filtragem de pacotes e configurações seguras de rede.
- **Gerenciamento de vulnerabilidades:** Envolve a identificação e correção de vulnerabilidades nos sistemas e aplicativos. Isso pode ser feito por meio de atualizações de segurança, aplicação de patches, configurações adequadas e testes de penetração.
- **Conscientização e treinamento:** A educação dos usuários é fundamental na cibersegurança. Os usuários devem ser treinados para reconhecer e evitar ameaças comuns, como *phishing*, engenharia social e malware. Isso pode incluir práticas de segurança, políticas de senha, uso seguro da Internet e práticas seguras de navegação.
- **Resposta a incidentes:** É importante ter um plano de resposta a incidentes para lidar com possíveis violações de segurança. Isso inclui ações para identificar, conter, investigar e remediar incidentes de segurança, além de restaurar a operação normal dos sistemas o mais rápido possível.

Além disso, a colaboração entre entidades governamentais, empresas, organizações de segurança e usuários individuais desempenha um papel importante na cibersegurança. A troca de informações sobre ameaças e melhores práticas ajuda a fortalecer as defesas e proteger a infraestrutura digital em geral.

É importante ressaltar que a cibersegurança é um campo em constante evolução, pois novas ameaças e vulnerabilidades surgem regularmente. Portanto, é essencial estar atualizado

com as práticas de segurança mais recentes e adotar uma abordagem proativa para proteger os sistemas e dados contra ameaças cibernéticas.

A segurança de dados é uma parte fundamental da cibersegurança e envolve a proteção das informações contra acesso não autorizado, uso indevido, divulgação, modificação ou exclusão não autorizada. A segurança de dados abrange várias medidas técnicas e práticas de gerenciamento de informações para garantir a confidencialidade, integridade e disponibilidade dos dados. Algumas medidas podem ser utilizadas para se obter uma melhor e mais robusta segurança de dados dentre as quais podemos citar: criptografia, controles de acesso, backup de dados, treinamento de usuários entre outros.

A criptografia é um método de codificação dos dados para que somente pessoas autorizadas possam acessá-los. Isso envolve o uso de algoritmos matemáticos para transformar os dados em um formato ilegível e, em seguida, decifrá-los novamente quando necessário. A criptografia é amplamente utilizada para proteger dados confidenciais durante o armazenamento e a transmissão.

O controle de acesso refere-se às medidas que garantem que apenas usuários autorizados possam acessar os dados. Isso envolve a autenticação dos usuários por meio de senhas, autenticação de dois fatores, certificados digitais ou biometria. Além disso, é importante implementar políticas de controle de acesso baseadas em funções e privilégios, garantindo que os usuários tenham acesso apenas aos dados necessários para realizar suas funções.

Realizar backups regulares dos dados é essencial para garantir a disponibilidade e integridade dos mesmos. Os backups devem ser armazenados de forma segura, preferencialmente em locais externos à infraestrutura principal, para protegê-los contra falhas do sistema, desastres naturais ou ataques cibernéticos. Além disso, é importante ter planos de recuperação de dados para restaurar as informações em caso de perda ou corrupção.

A conscientização dos usuários é uma parte crucial da segurança de dados. Os funcionários devem receber treinamento regular sobre práticas seguras de manuseio de dados, reconhecimento de ameaças cibernéticas, *phishing* e engenharia social. Isso ajuda a reduzir os riscos decorrentes de erros humanos e a fortalecer a postura de segurança de toda a organização.

É essencial que as organizações entendam e cumpram as regulamentações aplicáveis para proteger os dados de forma adequada. A segurança de dados é um aspecto crítico para proteger informações confidenciais e garantir a confiança dos clientes e parceiros comerciais. É uma combinação de tecnologia, processos e práticas de gerenciamento que evolui continuamente para enfrentar as ameaças emergentes e proteger os dados de forma eficaz.

2.3. Fabricantes Equipamentos CNC

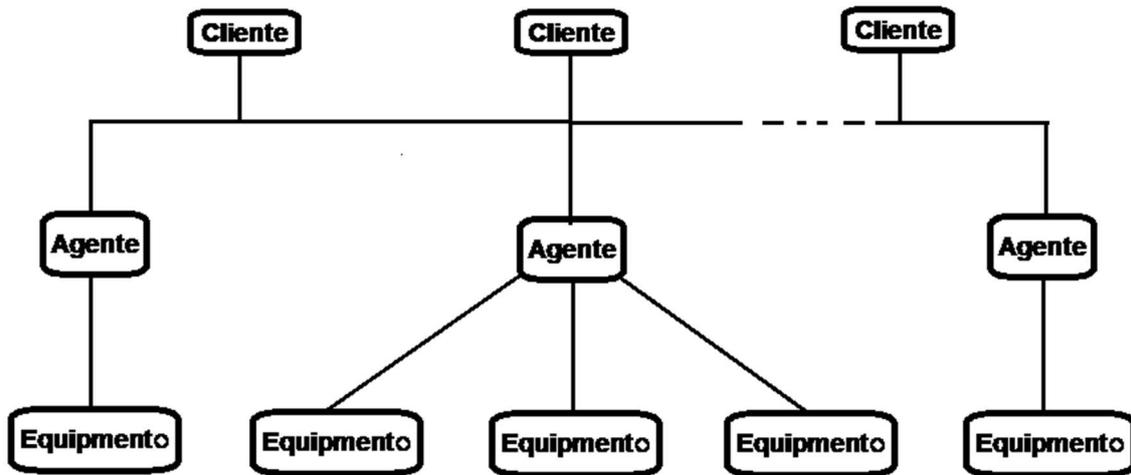
Atualmente no mercado existem diversos fabricantes de máquinas CNC's, e esses fabricantes por sua vez, em sua maioria, utilizam de controladores CNC's de grandes fabricantes mundiais, tal como: Siemens, Fanuc, Mazak, entre outros. Cada um desses controladores possuem características e particularidades específicas de configuração, programação e comunicação de dados, em exemplo a Siemens, possui protocolo de comunicação Profinet I/O para comunicação com demais componentes do maquinário, controlador lógico programável (CLP), I/O's remotos, inversores de frequência, entre outros, e também possui protocolo de comunicação OPC UA (*Open Platform Communications Unified Architecture*) para comunicação com dispositivos de aquisição de dados e comunicação com nuvem.(Siemens, 2022).

Explorando cada um desses fabricantes, protocolos e particularidades, e considerando a utilização pela indústria encontra-se uma certa tendência da utilização de máquinas CNC's com comandos Siemens e Mazak, cujo a qual ira-se aprofundar os conhecimentos para o desenvolvimento da solução proposta no presente trabalho.

2.3.1. Controladores Mazak

Os controladores CNC's da Mazak utilizam como servidor de comunicação o *MTCConnect* que possui uma estrutura bem similar ao OPC UA, tendo um método de comunicação baseado em cliente/servidor, onde temos o cliente fazendo solicitações e aguardando a resposta do servidor. Também um segundo método que *Publish/Subscribe*, onde temos os servidores disponibilizando as informações em HTTP e o cliente interessado coleta as informações necessárias, disponíveis em sua implementação. Na figura 9 tem-se a representação de um possível cenário de aplicação do *MTCConnect*. (*MTCCONNECT INSTITUTE*, 2021)

Figura 9 Cenário de Aplicação MTConnect

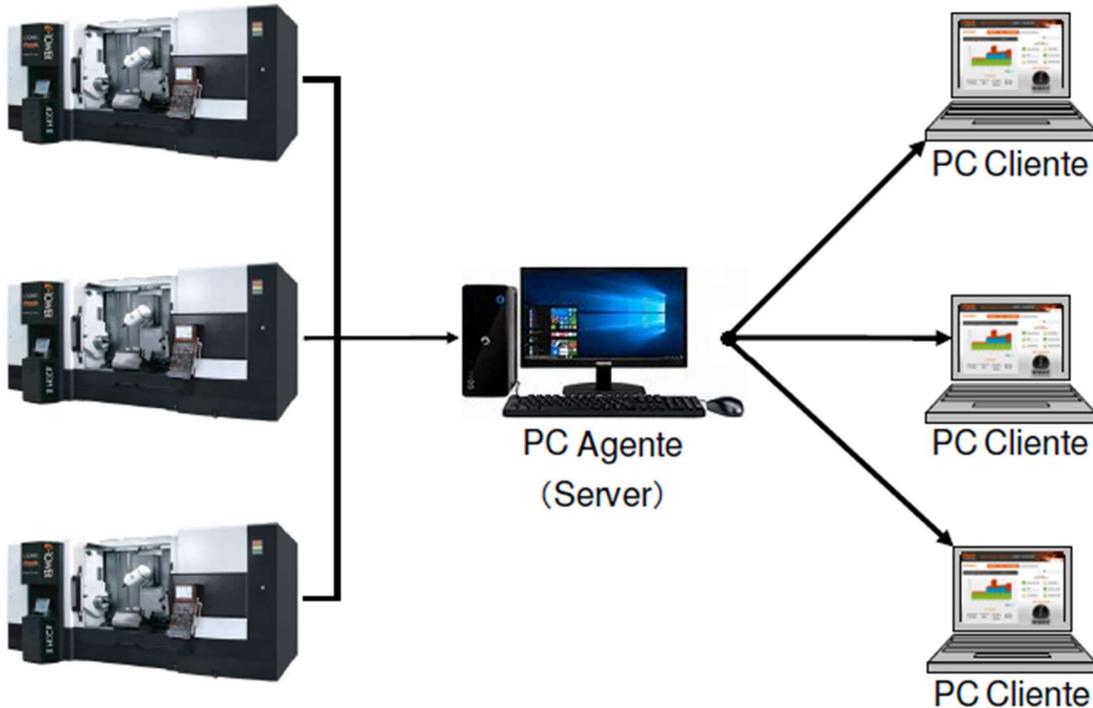


Fonte: Adaptado *MTConnect Institute* 2021

O *MTConnect* é um padrão aberto, ou seja, pode ser utilizado por qualquer pessoa ou entidade, basta fazer o download do software, e é responsável por fazer a conexão entre os endereçamentos de equipamento e dados no agente(servidor). O *MTConnect* é capaz, atualmente, somente entender arquivos XML (*Extensible Markup Language*). (*MTCONNECT INSTITUTE*, 2021)

O *MTConnect*, como mostrado na figura 10, precisa estar em execução em um servidor conectado ao equipamento para que o cliente fazer as solicitações de informações ao *MTConnect* e receber as informações retornadas do mesmo. Aqui funcionando diferente ao protocolo OPC UA que roda diretamente do controlador do CNC Siemens gerando as informações. (*MTCONNECT INSTITUTE*, 2021)

Figura 10 Exemplo de rede com MTConnect



Fonte: Adaptado *MTConnect Institute* 2021

A Mazak é participante do *MTConnect* desde 2007, possuindo arquivos XML padrões pré desenvolvidos que podem ser utilizados de forma fácil para a aquisição de dados do CNC. Esses dados de máquina assim como no caso apresentado com o OPC UA, atingem o estado de máquina, se em funcionamento ou não, se possui alarmes ou não, número e /ou descrição dos alarmes ativos, potência consumida pela máquina, velocidades de giro, velocidade de avanço, potências, ferramentas e uso individual de cada eixo, além de informações gerais de programa e posição atual de cada eixo. (*MTCONNECT INSTITUTE*, 2021)

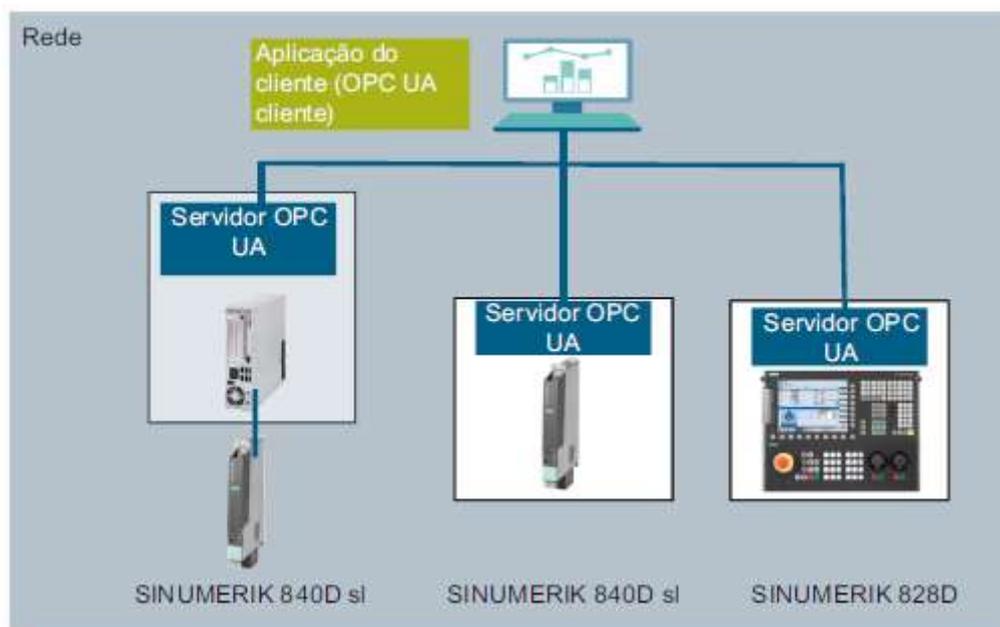
O *MTconnect* pode ser customizado e seu caso sempre estará aliado a um segundo software de visualização onde tem-se desenvolvidas as telas de visualização dos dados coletados, bem como podem fazer todo o gerenciamento da produção, programação de paradas preventivas para manutenção de equipamentos e otimizar o processo de fabricação, através do uso dos dados coletados. (*MTCONNECT INSTITUTE*, 2021)

2.3.2. Controladores Siemens

Os controladores Siemens utilizam-se da comunicação OPC UA para a aquisição de dados, esse protocolo de comunicação é gerenciado pela *OPC Foundation*, que é responsável pela padronização, inserção de funções e correções de protocolo caso sejam necessárias. Esse padrão, nada mais é do que uma série de especificações para fabricantes de equipamentos que foi desenvolvido por esses mesmos fabricantes em parceria com grandes indústrias, considerados usuários finais e desenvolvedores de software, onde se define uma interface padrão para comunicação entre cliente e servidor, ou entre servidores, que incluem acesso em tempo real, monitoração de dados, alarmes, acesso a dados históricos entre outros.

O servidor *SINUMERIK OPC UA* oferece uma interface de comunicação com o padrão independente do fabricante da máquina CNC, onde as informações do controle do *SINUMERIK* podem ser trocadas com um cliente OPC UA. O padrão de dados da comunicação é fornecido pela *OPC Foundation* podendo ser montado de forma simples, uma estação de trabalho para um cliente OPC, várias estações de trabalho para um cliente OPC ou até mesmo em uma ligação multiplataformas tendo variadas estações de trabalho e várias clientes coletando as informações necessárias de interesse. Na Figura 11, apresenta-se um possível cenário de ligação de rede utilizando-se do *SINUMERIK*. (SIEMENS,2022)

Figura 11 Cenário de Aplicação SINUMERIK



Fonte: Adaptado *Sinumerik Access machine* (2022)

O OPC UA é um padrão aberto para comunicações horizontais entre máquinas (M2M) e para comunicações verticais de máquinas para a nuvem. É independente de fornecedor ou plataforma, suporta mecanismos estendidos de segurança e pode ser combinado de forma otimizada com outro protocolo de comunicação em uma rede Ethernet Industrial compartilhada. Um padrão de comunicação aberto e neutro em relação à plataforma utilizada, que permite comunicações contínuas com aplicativos de terceiros e pode ser configurado de forma flexível para atender a requisitos específicos. Mecanismos de *ciber* segurança oriundos de TI, tais como autenticação, autorização e criptografia garantem trocas de dados de forma segura. (SIEMENS,2022)

O OPC UA cliente/servidor é amplamente utilizado na automação. Nesse conceito a comunicação é baseada no conceito um-para-um, os clientes OPC UA acessam aos dados no servidor OPC UA por meio de comunicações ponto a ponto. Os clientes enviam solicitações ao servidor, do qual recebem respostas dos dados solicitados. Essa forma de comunicação permite trocas de dados criptografadas, confiáveis, seguras, com consumo de banda reduzido e sem perdas de dados, mesmo se o ambiente e a qualidade da rede estiverem abaixo do ideal, apresentando distúrbios. A comunicação cliente/servidor é sempre baseada em TCP/IP. (SIEMENS,2022)

No OPC UA com *PubSub* (*publish/subscribe*), a comunicação é baseada no conceito um-para-muitos ou muitos-para-um é usado: um *publisher* disponibiliza dados que podem ser recebidos por uma quantidade ilimitada de *subscribers* na rede. Dependendo da tecnologia usada isso pode resultar em ciclos de comunicação muito rápidos. Em conjunto com o *Time-Sensitive Networking* (TSN), OPC UA *PubSub*, pode permitir a comunicação em tempo real no nível de controle, atendendo às demandas das aplicações de tempo crítico. (SIEMENS,2022)

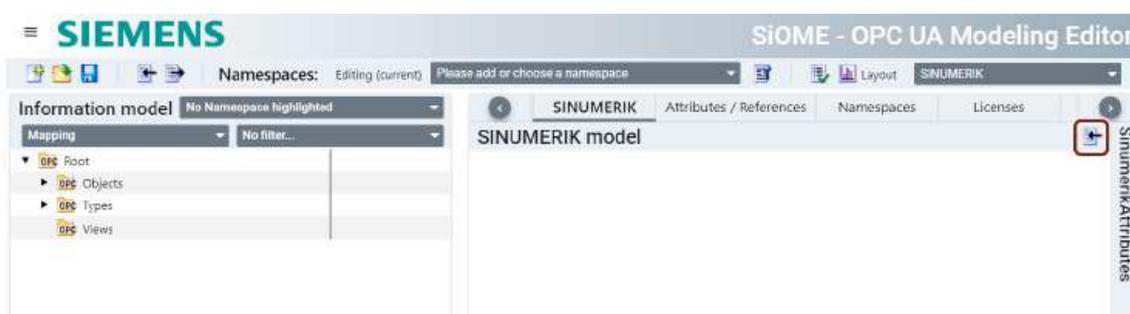
O servidor *SINUMERIK* OPC UA fornece a possibilidade de comunicar com o *SINUMERIK* via OPC UA, e oferece uma série de funcionalidades como ler e escrever variáveis *SINUMERIK*, fazer transferência de programas de peças, visualização e gerenciamento de alarmes e falhas, escolha do programa a ser executado, além de suportar modelos de objetos customizados pelo usuário (CSOM). (SIEMENS,2022)

Para que seja habilitada a comunicação OPC UA no *SINUMERIK*, é necessária uma licença de utilização, que deve ser adquirida junto a Siemens, para que ela seja validada para a utilização, a nível de teste a comunicação funciona sem a licença, mas tem um tempo de utilização de máximo de 30 minutos para tais testes. A habilitação da comunicação se dá por

meio de configuração na IHM, onde são carregados todos os dados de configuração Ethernet como os dados de endereçamento da máquina e a habilitação do OPC. (SIEMENS,2022)

Para que haja a troca de dados entre servidor (CNC) e o cliente é necessário que se carregue um modelo de variáveis, que se tem interesse para que elas sejam informadas pelo *SINUMERIK*. Esse modelo pode ser um modelo padrão ou pode ser um modelo customizado pelo próprio usuário. A Siemens possui um software chamado *SiOME*, a figura 12 mostra uma tela inicial do software, que é um software livre e possui a capacidade de criar um modelo CSOM. Nele podem ser exploradas todas as variáveis disponíveis para configuração do CSOM, e posterior execução dela no *SINUMERIK*. (SIEMENS,2022)

Figura 12 Software SiOME página inicial



Fonte: Adaptado *Sinumerik Access machine* (2022)

Para a Criação do modelo temos um arranjo de variáveis como mostrado na tabela 1:

Tabela 1 Tabela de Variáveis SINUMERIK

Prefixo	Range	Descrição dos dados
\$MM	9000 - 9999	Dados de Visualização de Máquina
\$MMS	51000 - 51299	Dados Gerais de Configuração de Máquina
	51300 - 51999	Dados Gerais de Ciclo de Máquina
\$MCS	52000 - 52299	Dados Específicos de Configuração de Canais de Máquina
	52300 - 52999	Dados Específicos de Ciclo de Canais de Máquina
\$MAS	53000 - 53299	Dados Específicos de Configuração Eixos de Máquina
	53300 - 53999	Dados Específicos de Ciclo de Eixos de Máquina
\$SNS	54000 - 54299	Dados Gerais de Set-up de configuração

	54300 - 54999	Dados Gerais de Set-up de ciclo
\$SCS	55000 - 55299	Dados Específicos de Configuração de Set-up de Canais
	55300 - 55999	Dados Específicos de Ciclo de Set-up de Canais
\$SAS	56000 - 56299	Dados Específicos de Configuração de Set-up de Eixos
	56300 - 56999	Dados Específicos de Ciclo de Set-up de Eixos

A partir das variáveis apresentadas dentro de cada um dos subgrupos se explora dados como estado de máquina, se em funcionamento ou não, se possui alarmes ou não, número e /ou descrição dos alarmes ativos, potência consumida pela máquina, velocidades de giro, velocidade de avanço, potências, ferramentas e uso individual de cada eixo, além de informações gerais de programa e posição atual de cada eixo. (SIEMENS,2019)

3. METODOLOGIA

Nesse trabalho explorou-se bibliograficamente as possibilidades e conceitos apresentados nos capítulos anteriores, com o objetivo de adquirir conhecimento para se realizar uma implementação de solução para o problema apresentado, que consiste em executar uma comunicação de dados com controladores CNC para aquisição de dados de máquina.

Analisando inicialmente as redes de comunicação e seus protocolos, utilizando o OPC UA sobre a Ethernet e a Modbus via serial, elaborou-se a seguinte tabela de comparação de características, representada na tabela 2.

Tabela 2 Tabela de Comparação OPC UA e Modbus

Característica	Ethernet	Modbus
Meio Físico	Cabo par trançado	Cabo um par com malha
Conector	RJ45	Borneira/ DB9
Porta	Rede Ethernet	Serial (COM)
Distância Máx.	100 m	1200 m
Velocidade	10 Mbps à 1 Gbps	9,6 kbps à 115 kbps
Configuração	Endereço IP, Máscara, Subrede, Gateway, DNS e porta	Serial, Velocidade, paridade, bits parada, time out, tempo ciclo
Dados	Endereçamento de objeto	Endereço físico de memória

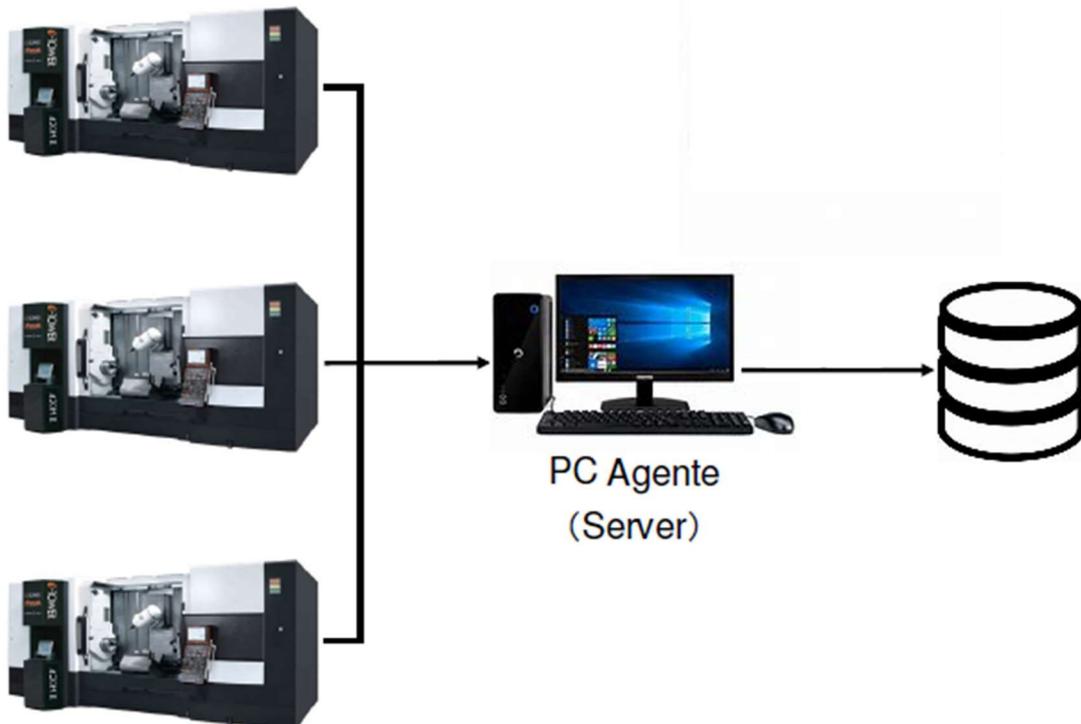
Através de um levantamento inicial com os fabricantes de equipamentos CNC's, definiu-se como principais objetivos o estudo e levantamento de necessidades para execução do trabalho com dois fabricantes Mazak e Siemens, que através da pesquisa inicial demonstraram viabilidade de implementação. Dando sequência aos estudos iniciou a busca documental das possibilidades e necessidades que cada um dos fabricantes teria como requisitos para garantir que a execução da implementação se tornasse possível de ser aplicada.

Essa pesquisa baseou-se nos manuais técnicos dos fabricantes dos equipamentos e das instituições que regem os protocolos de comunicação utilizados por eles, uma vez que é de premissa, que não será desenvolvido um protocolo de comunicação e sim utilizado o que é disponibilizado pelo fabricante, e que esse protocolo, seja baseado em ethernet para que se tenha um melhor desempenho de troca de dados.

Desta forma chegou-se ao entendimento que para a comunicação com controladores Siemens terão de ser executadas algumas etapas de desenvolvimento onde se é necessário que se possua uma licença de operação de servidor OPC no próprio controlador, que atuará também

como servidor de comunicação OPC UA, havendo a necessidade de utilização de um arquivo XML binarizado, que pode ser padrão de fornecimento do fabricante do CNC ou customizado por algum desenvolvedor, essa customização deve ser executada através de um software específico, *SiOME*, que faz a interpretação do protocolo, linkando os endereçamentos das variáveis desejadas com objeto de comunicação. O controlador deve ser configurado para que se estabeleça uma conexão ethernet, com endereçamento e portas específicas e o OPC UA deve ser habilitado para uso e direcionado para executar o arquivo XML carregado no controlador. Após esses passos executados e com a comunicação estabelecida entre servidor, controlador CNC e o usuário (cliente), os dados ficarão disponíveis por sistema de broadcast. Por sua vez o desenvolvimento de script no cliente será responsável por buscar os dados e direcioná-los para o armazenamento em banco de dados, como indicado na Figura 13.

Figura 13 Comunicação e armazenamento em banco de dados.



Fonte: Adaptado *MTCConnect Institute* 2021

Com o Controlador Mazak, o processo é parecido, mas a comunicação é regida pela *MTCConnect* e o servidor é um computador e não o próprio controlador do CNC. Esse servidor também deve rodar um arquivo XML, que por sua vez é fornecido pela *MTCConnect* para determinado modelo de equipamento, ou assim como no caso do controlador Siemens, pode ser editado por software de edição. Como o computador é o servidor nesse caso, pode-se fazer

comunicação com o controlador Mazak e disponibilizar as variáveis disponíveis de máquina com o uso desse mesmo computador, tendo esse uma segunda função de cliente que irá executar o armazenamento dos dados, ou utilizá-lo somente como servidor de comunicação, configurando-o para que as informações possam ser acessadas por outros equipamentos.

3.1. Esquemático desenvolvimento

Definindo inicialmente que será utilizado um CNC Mazak para a aquisição de dados, iniciou-se o desenvolvimento do projeto. A figura 14, mostra um esquemático explicativo da sequência do desenvolvimento, indicando uma sequência lógica dos passos a serem implementados para que se alcance o objetivo desejado

Figura 14 Esquemático do desenvolvimento.



3.2. Variáveis Disponíveis

Por padrão com a comunicação com o servidor Mazak ficam disponíveis as seguintes variáveis para a coleta de dados, que estão dispostas na Tabela 3 a seguir:

Tabela 3 Relação de Variáveis Controlador Mazak

Nome	Descrição
<i>Power</i>	Máquina Ligada ou Desligada
<i>Spindle Speed</i>	Velocidade em RPM do Spindle
<i>Axis Position</i>	Posição do Eixo
<i>Independent Axis Feed</i>	Velocidade de Avanço do Eixo
<i>Path Feed</i>	Trajectoria de Execução
<i>Block Number</i>	Número do bloco do Programa em Execução
<i>Status</i>	Status Geral de Máquina
<i>CNC Mode</i>	Modo de Operação
<i>Work Number</i>	Número do Programa em Execução
<i>Current Alarm</i>	Número do Alarme Ativo
<i>Spindle Load</i>	Carga do Spindle (%)
<i>Axis Load</i>	Carga do Eixo (%)
<i>Spindle Override</i>	Sobrecarga do Spindle (%)
<i>Axis Override</i>	Sobrecarga do Eixo (%)
<i>Tool Number</i>	Número da Ferramenta
<i>Tool Suffix</i>	Sufixo da Ferramenta
<i>Tool Group Number</i>	Número do Grupo de Ferramentas
<i>Spindle Temperature</i>	Temperatura do Spindle
<i>Parts Count</i>	Contagem de peças
<i>Rapid Override</i>	Taxa de sobrecarga Máquina (%)
<i>Sub Pro Work Number</i>	Número Subgrupo do Programa
<i>Auto Operation Time</i>	Tempo de Operação em Automático
<i>Cutting Time</i>	Tempo de Execução de Corte
<i>Total Time</i>	Tempo total de Operação
<i>Sequence Number</i>	Número de Peças Executadas
<i>Unit Number (MAS)</i>	Número serial da Máquina

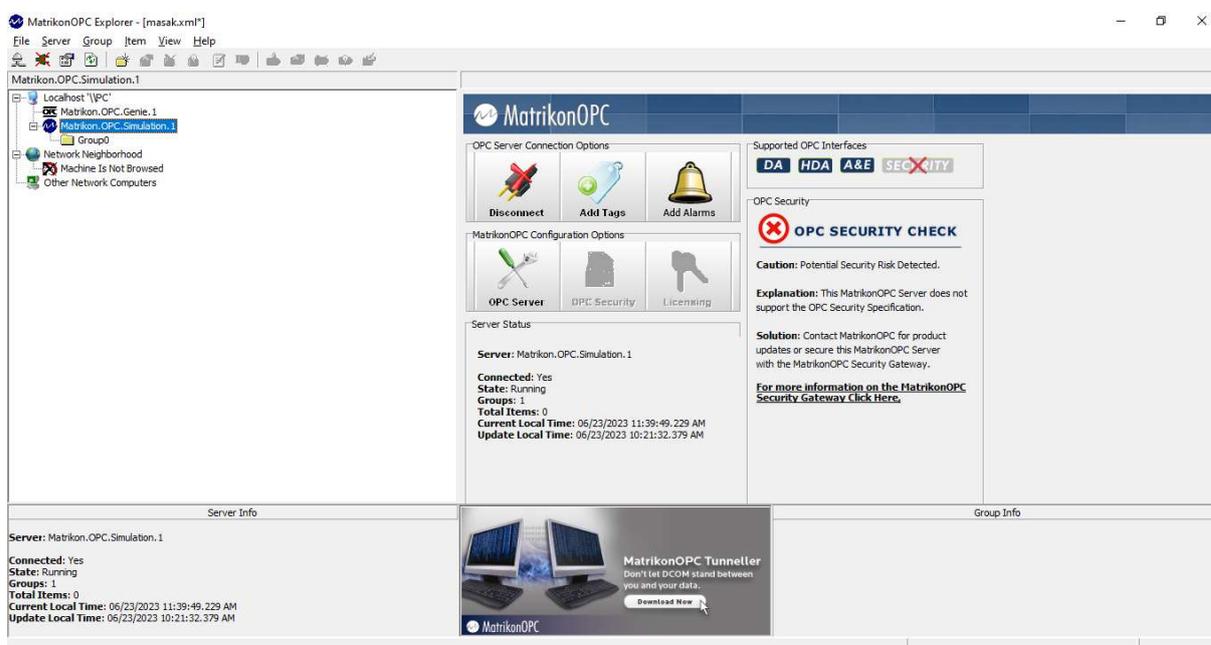
A partir das variáveis disponíveis em nossa comunicação, executou-se uma análise dos dados que possuem um maior interesse em serem estudados e optou-se pela coleta das seguintes variáveis: máquina ligada, velocidade do *spindle*, carga do *spindle*. Cujas as quais nos mostrarão alguns dos dados importantes do processo de fabricação, podendo-se expandir essa relação de variáveis para que no futuro se possam adquirir mais dados de interesse para uma melhor análise de processo.

3.3. Configuração Servidor OPC

Neste ponto de implementação decidiu-se pela utilização de um servidor OPC, que será responsável por carregar as variáveis disponíveis no controlador da Mazak, desta forma, pode-se fazer a implementação, configurações e testes sem parada de produção, não tendo assim custos com parada de máquina e atraso de produção. Para tal função optou-se pela utilização do Servidor Matrikon OPC, que é um dos servidores OPC de Mercado e possui uma grande disponibilidade e características que permitem que a execução da comunicação, para que a aquisição de dados ocorra de forma com que se permita a configuração e adição das variáveis de forma idêntica ao processo que seria encontrado com a comunicação real com o CNC.

Para a configuração do servidor OPC, existem duas possibilidades de conexão: local ou remota. Para estabelecer a configuração do servidor de forma remota, inicialmente, tem-se a necessidade de configurar tanto servidor Matrikon quanto a própria rede Windows, adicionando endereço, máscara, *gateway* padrão de rede colocando-se na mesma rede de comunicação em que o dispositivo com o servidor OPC se encontra. Para estabelecer a configuração local, basta adicionar um servidor na aba *Localhost*, como pode-se observar na figura 15 abaixo.

Figura 15 Adição servidor OPC Local Matrikon.



Fonte: Elaborado por Autor (2023).

Em nossa aplicação optou-se pelo uso de um servidor local e como próximo passo, pode-se configurar as variáveis escolhidas para a aquisição de dados, configurando-as com tipo de

dados que cada uma delas possui, vinculando, assim, a mesma com o mesmo tipo de dados que seria gerado pelo CNC. A partir dessa adição de variáveis basta conectar o servidor OPC, fazendo com que o mesmo entre em execução, liberando os dados cadastrados na rede para coleta dos dados.

3.3.1. Script para Comunicação OPC

Fazendo o uso do Visual Studio como compilador C# (C Sharp), o firmware deve primeiramente criar uma instância para configurar e iniciar a comunicação com o servidor OPC. Para executar essa configuração faz-se necessário a utilização da biblioteca Standart da OPC Foundation, *OPC Foundation.NET Standard Library*, que pode ser adicionada ao script através do gerenciamento de pacotes da *Nuget* no Visual Studio, utilizando-se processo apresentado no diagrama da figura 16, tem-se a configuração do servidor OPC Local que será utilizado no presente estudo.

Figura 16 Configuração do Servidor OPC Local



Seguindo o processo de configuração devesse incluir as variáveis que desejamos obter os valores através da comunicação OPC. Para tal necessita-se fazer a configuração e inserção dessas variáveis utilizando-se exato nome utilizado no servidor OPC, como mostrado na figura 16, fazendo as devidas configurações de tipo necessárias para cada uma das variáveis.

Figura 17 Script Leitura de Variáveis do Servidor OPC

```
List<ReadValueId> nodesToRead = new List<ReadValueId>
{
    new ReadValueId
    {
        NodeId = (NodeId)ExpandedNodeId.Parse("ns=2;s=Power"),
        AttributeId = Attributes.Value
    },
    new ReadValueId
    {
        NodeId = (NodeId)ExpandedNodeId.Parse("ns=2;s=SpindleSpeed"),
        AttributeId = Attributes.Value
    },
    new ReadValueId
    {
        NodeId = (NodeId)ExpandedNodeId.Parse("ns=2;s=SpindleLoad"),
        AttributeId = Attributes.Value
    },
};

var readRequest = new ReadRequest
{
    NodesToRead = nodesToRead.ToArray()
};

var readResponse = session.Read(null, 0, TimestampsToReturn.Both, readRequest);

if (readResponse != null && readResponse.Results.Count > 0)
{
    // Verificação dos valores lidos
    //POWER
    DataValue value0 = readResponse.Results[0];
    if (StatusCode.IsGood(value0.StatusCode))
    {
        Boolean Power = value0.GetValue<Boolean>(false);
    }

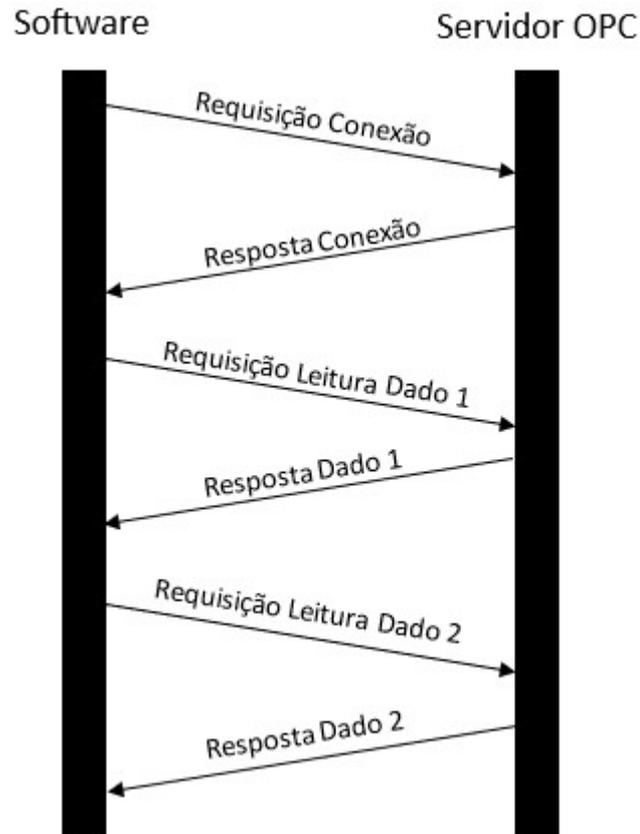
    //SPINDLE SPEED
    DataValue value1 = readResponse.Results[1];
    if (StatusCode.IsGood(value1.StatusCode))
    {
        // Obtenha o valor da variável lida
        UInt16 Spindle_speed = value1.GetValue<UInt16>(0);
    }

    //SPINDLE LOAD
    DataValue value2 = readResponse.Results[2];
    if (StatusCode.IsGood(value2.StatusCode))
    {
        // Obtenha o valor da variável lida
        float Spindle_speed = value2.GetValue<float>(0);
    }
}
```

Fonte: Elaborado por Autor (2023).

A comunicação com o servidor OPC funciona como uma comunicação ponto a ponto, onde é feita uma solicitação de dado e aguarda-se a resposta dessa solicitação, na Figura 18 pode-se observar um modelo de barras dessa comunicação, onde se apresenta um modelo visual desse processo de transmissão de dados.

Figura 18 Diagrama Barras Comunicação



Fonte: Elaborado por Autor (2023).

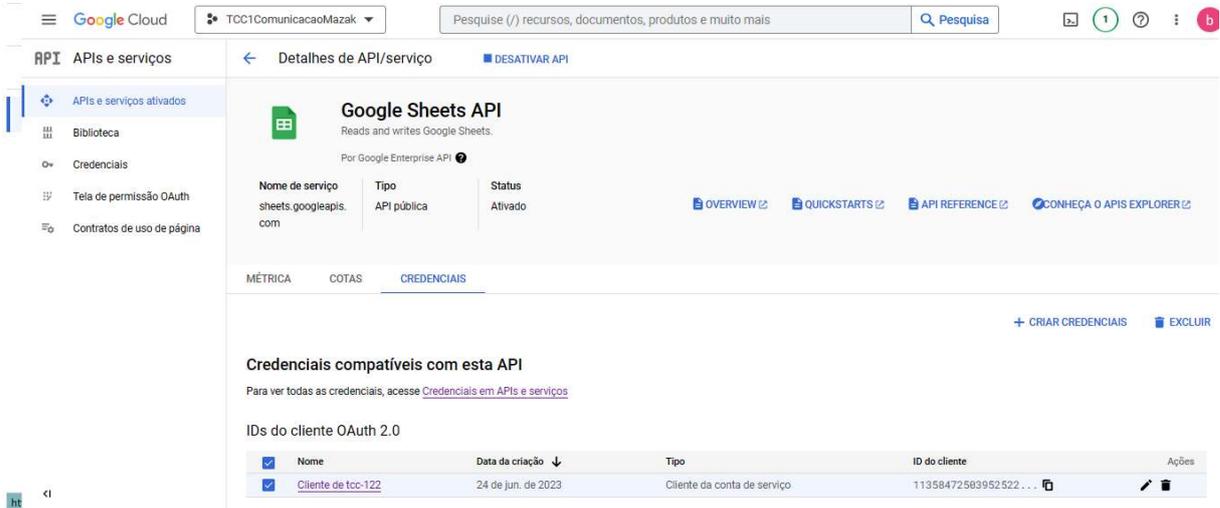
3.4. Armazenamento de dados

Para o armazenamento de dados usou-se da plataforma do *Google Sheets* onde podem ser armazenadas as informações e se ter um gerenciamento de acessos a tais informações de acordo com a necessidade. Para tal faz-se necessário o uso de uma conta no Google Docs, bem como a configuração de um projeto com controle de acesso e credenciais para APIs do Google Cloud. Para desenvolvimento do firmware foi utilizada a API “*Google.Apis.Sheets.V4*” disponibilizada pela própria Google e que possui as funções necessárias para desenvolver o script de forma a atender a necessidade do projeto.

Primeiramente precisa-se gerar as credenciais de autenticação (chave de API) no *Google Cloud Console* para autenticar a aplicação e posteriormente obter acesso à API do *Google Sheets*. Junto as credenciais, é necessário a captação do arquivo JSON que é o arquivo de criptografia e através dele que se é permitido o acesso a planilha. As configurações de

permissões de acesso de usuário para a leitura e escrita são necessárias para o correto funcionamento, como mostrado na figura 19.

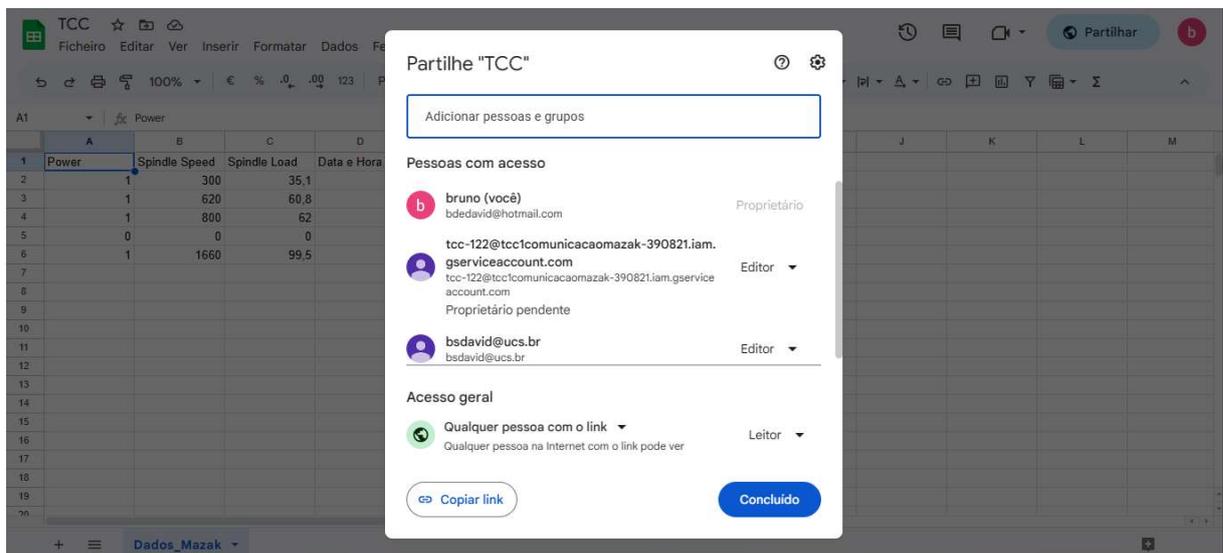
Figura 19 Visualização configuração Google Sheets API



Fonte: Elaborado por Autor (2023).

Executadas a configuração da API no Google Cloud, executa-se a criação e configuração da planilha no *Google Sheets*, que necessita que se adicione como editor a credencial da API gerada anteriormente, permitindo assim que o programa consiga fazer as alterações devidas na tabela, como mostrado na Figura 20. Terminado esses passos segue-se para o desenvolvimento do script que fará a escrita dos dados lidos anteriormente através da comunicação OPC para o banco de armazenamento de dados.

Figura 20 Compartilhamento Planilha Google Sheet



Fonte: Elaborado por Autor (2023).

4. FUNCIONAMENTO E TESTE

O software desenvolvido tem como cerne funcionamento o armazenamento de dados oriundos de uma comunicação OPC UA, para que esse objetivo seja alcançado faz-se necessário que sejam atingidas certas etapas de desenvolvimento.

Inicialmente os esforços são direcionados ao bom funcionamento do servidor OPC, nessa etapa faz uso de ferramentas para criação e configuração do servidor com as devidas variáveis disponíveis para a comunicação de dados. Configurações de endereçamentos de rede do servidor, liberação de porta de comunicação e a correta implementação das variáveis com suas características específicas de tipo, limites. Na figura 21, encontra-se o servidor utilizado para a execução da comunicação em execução, onde podemos observar as variáveis adicionadas e alguns dos dados do servidor.

Figura 21 Servidor OPC Conectado

The screenshot shows the MatrikonOPC Explorer interface. The left pane displays a tree view of the local network, including 'Localhost "\BRUNO"', 'Matrikon.OPC.Genie.1', 'Matrikon.OPC.Simulation.1', and 'Dados'. The main pane shows the 'Contents of "Dados"' table with the following data:

Item ID	Access Path	Value	Quality	Timestamp	Status
Power		True	Good, non-specific	06/30/2023...	Active
SpindleLoad		88,5	Good, non-specific	06/30/2023...	Active
SpindleSpeed		1680	Good, non-specific	06/30/2023...	Active

Below the table, there is a 'Server Info' section showing: Server: Matrikon.OPC.Simulation.1, Connected: Yes, State: Running, Groups: 1, Total Items: 3, Current Local Time: 06/30/2023 10:21:12.236 PM, Update Local Time: 06/30/2023 1:46:42.434 PM. To the right, a 'Group Info' section shows: Group: Dados, Connected (Async I/O): No, Active: Yes, Items: 3, Current Update Rate: 1000 ms, Percent Deadband: 0,00%, Data Change Rate: 0,00 Items/Sec. A 'Did you know?' tip is also visible, stating 'OPC Explorer can filter your browsable OPC items.' and 'Click For Details'.

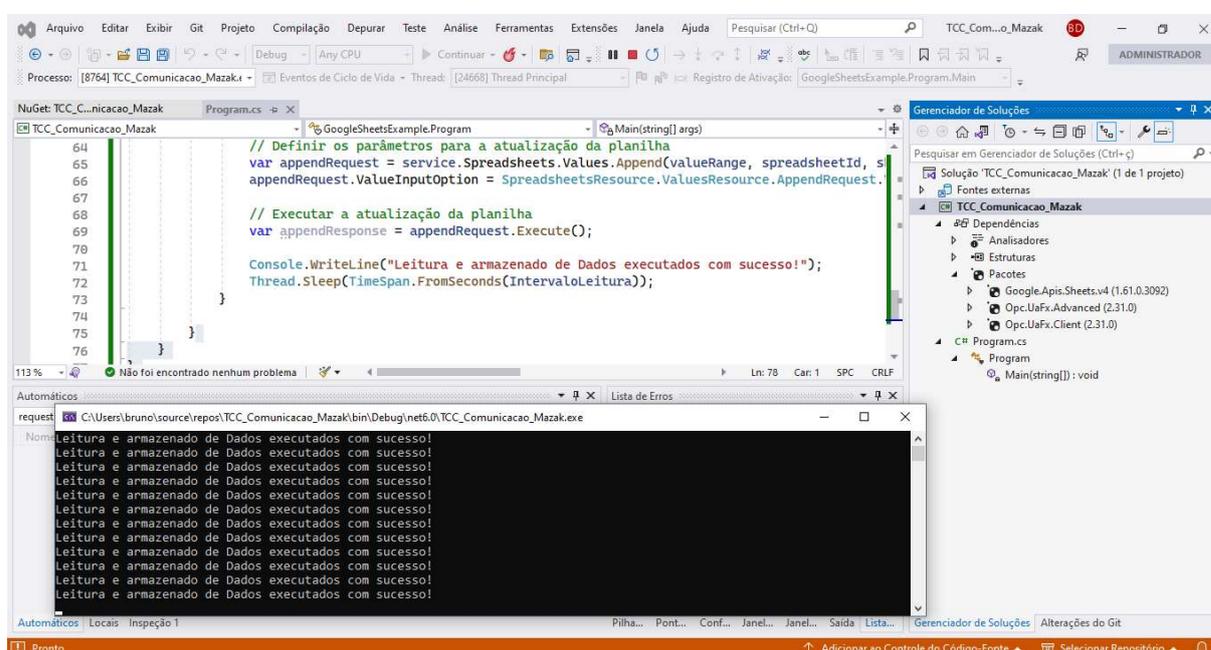
Fonte: Elaborado por Autor (2023).

Dando sequência é executada toda e qualquer ação necessária para a vinculação de dados ao *Google Sheets*. Através do *Google Console Cloud* temos que habilitar o uso da API, criando credenciais de usuário e chaves de acesso que permitam que o nosso software consiga executar as operações de leitura e edição. Após as configurações cria-se o JSON, que é o arquivo que carrega as informações de endereço, identidade e acesso á nossa planilha do *Google Sheets* através da liberação da API, e que permite que seja executada as funções de troca de dados. Ainda assim faz-se necessário que a tabela seja compartilhada com o aplicativo, utilizando para

isso as credenciais de usuário, e que no compartilhamento sejam aceitas que o determinado usuário seja capaz de executar a leitura e edição de dados na referenciada planilha.

Por fim, o script é o agente integrador que irá guiar os processos, fazendo a abertura da comunicação OPC com o servidor, coletando os dados estipulados de coleta, após essas etapas desenvolvidas e com os dados em variáveis locais ele executa a conexão com o banco de dados, em nosso caso a planilha criada no *Google Sheets*, executando a inserção dos novos dados obtidos. Na Figura 22, apresenta-se o programa em execução gerando uma mensagem de leitura e armazenamento de dados executados.

Figura 22 Software em execução



Fonte: Elaborado por Autor (2023)

A nossa planilha irá armazenar os dados novos enviados, tendo os dados novos sendo inseridos na primeira linha vazia encontrada. E por sua vez com a planilha podendo ser compartilhada com mais usuários que possuem permissões somente de visualização dos dados, esses usuários podem através de análises específicas de tempo, funcionamento, ou operações estar utilizando tais dados para execução de tarefas, definições de projetos, tomada de decisões etc.

Na figura 23 apresenta-se os dados enviados a planilha através da execução programa bem como os usuários que alteraram os dados atribuindo uma cor específica para cada usuário, percebam que se apresentam dados grifados em roxo, que foram gerados pelo software e dados em azul que foram gerados por outro usuário da planilha com permissão de edição.

Figura 23 Tabela de Dados Armazenados no Google Sheets

The screenshot shows a Google Sheets spreadsheet with a table of data. The table has columns for date, time, and user. The data is color-coded: purple for software-generated data and blue for user-generated data. A version history sidebar is visible on the right.

Row	Col 1	Col 2	Col 3	Col 4	Col 5	Col 6	Col 7	Col 8	Col 9	Col 10
309	0	1989	56,28	01/07/2023 17:57	Mazak_01					
310	1	2480	58,55	01/07/2023 17:58	Mazak_01					
311	1	623	6,04	01/07/2023 17:58	Mazak_01					
312	1	1369	72,14	01/07/2023 17:58	Mazak_01					
313	1	466	46,20	01/07/2023 17:58	Mazak_01					
314	1	3062	76,10	01/07/2023 17:58	Mazak_01					
315	0	375	37,18	01/07/2023 17:58	Mazak_01					
316	0	3047	82,85	01/07/2023 17:59	Mazak_01					
317	0	1953	36,34	01/07/2023 17:59	Mazak_01					
318	1	2406	5,87	01/07/2023 17:59	Mazak_01					
319	0	1990	29,08	01/07/2023 17:59	Mazak_01					
320	1	2708	89,55	01/07/2023 17:59	Mazak_01					
321	1	1379	95,41	01/07/2023 18:00	Mazak_01					
322	1	2745	25,47	01/07/2023 18:00	Mazak_01					
323	1	3260	47,00	01/07/2023 18:00	Mazak_01					
324	1	2704	19,48	01/07/2023 18:00	Mazak_01					
325	1	493	13,36	01/07/2023 18:00	Mazak_01					
326	1	802	49,21	01/07/2023 18:00	Mazak_01					
327	0	384	20,03	01/07/2023 18:01	Mazak_01					
328	0	453	28,05	01/07/2023 18:01	Mazak_01					
329	0	1852	24,12	01/07/2023 18:01	Mazak_01					
330	1	2127	77,93	01/07/2023 18:01	Mazak_01					

Version History (Historico de versões):

- 1 de julho, 18:05 (Versão atual)
 - bruno
 - tcc-122@tcc1comunicacaoamazak-390821.lam.gserviceaccount.com
- 30 de junho, 22:16
 - bruno
- 28 de junho, 20:03
 - bruno
- 24 de junho, 22:25
 - bruno

Mostrar alterações:

Fonte: Elaborado por Autor (2023)

5. CONCLUSÃO

Após interpretação de diversos documento explorando as possibilidades nos controladores CNC's Siemens e Mazak, tem-se definido que as possibilidades de aquisição de dados podem ser implementadas, em cada um dos casos com suas particularidades e necessidades para viabilizar o desenvolvimento.

Em ambos os casos a partir das variáveis disponibilizadas, tem-se uma série de informações que vão desde status simples de funcionamento da máquina, como máquina em operação até dados complexos de posição atual de cada eixo dentro do movimento, consumos de potência e até mesmo posições relativas ou offsets de ferramenta. Com o acesso a essa gama de dados gerados pode-se executar diversos tipos de análises para definição de processos, ordens de produção para cada equipamento baseado no desempenho para a execução de cada peça finalizada em cada máquina presente do sistema de aquisição.

Cabe lembrar também que são possíveis alterações de variáveis, e não somente a leitura delas, assim podendo por exemplo alterar parte das programações, por exemplo, torque de operação ou um offset de determinada ferramenta, a partir da sua utilização. Assim como é possível com o auxílio do servidor enviar programação de peças, podendo se ter um backup de tais programas no servidor e de acordo com a necessidade de carregue tal programa com o intuito de reduzir tempo de setup de máquina uma vez que o programador/operador de máquina não precisará executar essa função de forma direta na IHM do CNC.

As comunicações para as aquisições de dados pesquisadas são de implementações similares, e em ambos os casos pelo evidenciado no estudo, fazem o uso de modelos padrões de objetos, e utilizando o XML como meio de interpolação entre dado de máquina e dado disponibilizado para o cliente final em OPC UA.

Pode-se adicionalmente até especular que ambos as implementações trabalham sobre o mesmo protocolo, o que não necessariamente é de se correto afirmar, pois como ambos trabalham em cima do meio físico ethernet é de se esperar que possuam similaridades e meios comuns, com suas particularidades para a execução.

Entre as particularidades encontradas estão as variáveis disponíveis, que variam de fabricante para fabricante, também podendo variar de um modelo de equipamento para outro, uma vez que um pode possuir mais ou menos eixos e/ou spindles que outro modelo de mesmo

fabricante. Outro ponto importante é que em ambos os fabricantes os arquivos XML podem ser editados o que pode implicar em um aumento ou redução dos dados disponibilizados.

Na implementação apresentada fez-se a coleta de apenas alguns dados utilizados como método comprobatório do funcionamento da viabilidade da aplicação do projeto em usuários de máquinas CNC, que possuem interesse em possuir mais dados relacionados em processos controlados por comandos numéricos.

Tais dados coletados foram disponibilizados através da ferramenta do *Google Sheets*, essa por sua vez é de larga utilização e pode ser acessada, e tratada pelos usuários permitidos, para que sejam efetuadas as análises de processo. O que torna essa ferramenta atraente é que ela é de fácil utilização e gerenciamento, podendo ser definidos novos acessos e bloqueios a qualquer momento, por parte do proprietário além de não necessitar algum software específico para o acesso aos dados, uma vez que a plataforma roda on-line.

Conclui-se por tanto que após a finalização desse desenvolvimento fica como projeto futuro a implementação física para leitura de dados reais, que deve ocorrer quando houver a disponibilidade de máquina. Podendo-se fazer para essa etapa a adição de mais dados disponíveis nos equipamentos, para que sejam gerados mais dados uteis, permitindo por sua vez, cada vez mais melhores análises e/ou com mais ênfase em determinada característica/aspecto de funcionamento. Atingindo assim os propósitos da indústria 4.0, que são não somente possuir os dados, mas possuir os dados corretos, para a tomada da decisão mais efetiva para cada determinada situação.

Permitindo assim que em posteriores aplicações com a comunicação coletando dados reais de máquina possa-se executar análises de dados de processo fabril, podendo-se assim optar pela tomada de determinadas decisões a partir de tais dados de máquina, desconsiderando assim a vivência do operador ou líder de produção, que por indeterminada ocasião ou situação possa de alguma forma estar prevalecendo a utilização tendenciosa de um determinado equipamento por um desempenho melhor de fabricação ou por um determinado operador com que possui uma melhor afinidade de trabalho.

REFERÊNCIAS

- LEITÃO, M. A. Implementação de um servidor opc ua em linguagem c# para comunicação com dispositivos através do protocolo modbus/ethernet em tempo real. , [S.l.], 2010.
- LOURENTI, A. M. O que é e para que serve o arquivo XML. 24/01/2020. Canal Tech. Disponível em < [O que é e para que serve o arquivo XML - Canaltech](#) > Acesso em: 09/12/2022
- LUTH, J; OPC 10000-1 - **OPC Unified Architecture: Part 1: Overview and Concepts**. OPC Foundation. 01/11/2022, revisão 1.05.22
- MECÂNICA INDUSTRIAL. **Máquinas CNC**. 10/07/2017. Disponível em:< [Máquinas CNC | Mecânica Industrial \(mecanicaindustrial.com.br\)](#) > Acesso em: 06/12/2022.
- MODBUS ORGANIZATION. **Modbus over serial line: specification and implementation guide**. 2006.
- MODBUS ORGANIZATION. **Modbus application protocol specification v1.1b3**. 2012.
- MTCONNECT INSTITUTE. **MTConnect Part 1.0: Overview and Fundamentals** 06/09/2021; Version 1.8.0
- NATIONAL INSTRUMENTS. **O protocolo Modbus em detalhes**. 2019. Disponível em: <<http://www.ni.com/pt-br/innovations/white-papers/14/the-modbus-protocol-in-depth.html>> . Acesso em: 19/11/2022.
- OPC FOUNDATION. **OPC 10000-6 - OPC Unified Architecture: Part 1: Mappings**. OPC Foundation. 21/02/2022, revisão 1.05.22. Technical Manual.
- OPC FOUNDATION. **OPC 40502 – OPC UA for Computerized Numerical Control (CNC) Systems**. OPC Foundation. 05/07/2017, revisão 1. Technical Manual.
- OPC FOUNDATION. **Unified Architecture**. OPC Foundation. Disponível em:< [Unified Architecture - OPC Foundation](#) > Acesso em: 05/12/2022.
- PINHEIRO, J. M. S.; **O Modelo OSI- Projeto de Redes**. 22/11/2004. Disponível em:<| http://www.projetoderedes.com.br/artigos/artigo_modelo_osi.php > Acessado em 18/12/2022
- POPP, M.; **Comunicação industrial com Profinet**. São Paulo: PI Brasil, 2018.
- SACAMONO, J.; [et al.]; **Industria 4.0: Conceitos e fundamentos**. São Paulo: Bucher, 2018
- SIEMENS. **OPC UA – Open Platform Communications Unified Architecture**. Disponível em: < [OPC UA | Comunicação Industrial | Siemens Siemens Brasil](#) > Acesso em: 20/11/2022
- SIEMENS. **Sinumerik Acess Machine/ OPC UA: Manual de projeto**; Siemens AG. Regensburg, 05/2022. Versão 3.1 SP2 Technical Manual.
- SIEMENS. **Sinumerik 840D sl/ NC variables: List manual**; Siemens AG. Regensburg, 06/2019 Technical Manual.
- STRACK, G. **Módulo de I/O remoto MODBUS**. 2011. Trabalho de Conclusão de Curso

(Graduação em Engenharia Elétrica) — Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, 2011.

THOMAS, G. Introduction to the modbus protocol. **The extension — a technical supplement to control network**, [S.l.], v. 9, n. 4, p. 1–4, 2008.

VENTURELLI, M.; **Redes Ethernet Industrial: Conceito e aplicação na Automação e Controle Industrial**, 31/05/2016. Automação Industrial. Disponível em:< [Redes Ethernet Industrial: Conceito e aplicação na Automação e Controle Industrial \(automacaoindustrial.info\)](http://Redes Ethernet Industrial: Conceito e aplicação na Automação e Controle Industrial (automacaoindustrial.info))

> Acesso em:07/10/22

VILARROYA, A.; REMIGIO, B. Servidor opc-ua sobre beaglebone black., [S.l.], 2015.

WEISS, OLGA. O Que é Ethernet e como funciona. 05/06/2020. USB Over Network. Disponível em <[O que é Ethernet | Como Funciona a Ethernet \(net-usb.com\)](http://O que é Ethernet | Como Funciona a Ethernet (net-usb.com))> Acesso em :07/10/2022

APENDICE A- SCRIPT SOFTWARE

```

using Google.Apis.Auth.OAuth2;
using Google.Apis.Sheets.v4;
using Google.Apis.Sheets.v4.Data;
using Google.Apis.Services;
using System;
using System.Collections.Generic;
using Opc.UaFx.Client;
using Opc.Ua;

namespace GoogleSheetsExample
{
    class Program
    {
        static void Main(string[] args)
        {
            //while (true)
            // {
                int IntervaloLeitura = 10; // tempo de leitura em segundos
                // Defina o endereço URL do servidor
                OPC Matrikon

                string serverUrl =
                "opc.tcp://localhost:4840/Matrikon.OPC.Simulation.1";

                // Crie um cliente OPC
                using (var client = new OpcClient(serverUrl))
                {
                    try
                    {
                        // Conecte-se ao servidor OPC
                        client.Connect();

                        // Verifique se a conexão foi estabelecida
                        if (client.State == OpcClientState.Connected)
                        {
                            // Leitura das variáveis do sistema

                            UInt16 Power=0;
                            UInt16 Spindle_speed=0;
                            float Spindle_Load=0;

                            List<ReadValueId> nodesToRead = new List<ReadValueId>
                            {
                                new ReadValueId
                                {
                                    NodeId =
                                    (NodeId)ExpandedNodeId.Parse("ns=2;s=Power"),
                                    AttributeId = Attributes.Value
                                },
                                new ReadValueId
                                {
                                    NodeId =
                                    (NodeId)ExpandedNodeId.Parse("ns=2;s=SpindleSpeed"),
                                    AttributeId = Attributes.Value
                                },
                                new ReadValueId

```

```

        {
            NodeId =
(NodeId)ExpandedNodeId.Parse("ns=2;s=SpindleLoad"),
            AttributeId = Attributes.Value
        },

};

var readRequest = new ReadRequest
{
    NodesToRead = nodesToRead.ToArray()
};

var readResponse = session.Read(null, 0,
TimestampsToReturn.Both, readRequest);

0) if (readResponse != null && readResponse.Results.Count >
    {
        // Verificação dos valores lidos
        //POWER
        DataValue value0 = readResponse.Results[0];
        if (StatusCode.IsGood(value0.StatusCode))
        {
            Power = value0.GetValue<UInt16>(0);
        }

        //SPINDLE SPEED
        DataValue value1 = readResponse.Results[1];
        if (StatusCode.IsGood(value1.StatusCode))
        {
            Spindle_speed = value1.GetValue<UInt16>(0);
        }

        //SPINDLE LOAD
        DataValue value2 = readResponse.Results[2];
        if (StatusCode.IsGood(value2.StatusCode))
        {
            Spindle_Load = value2.GetValue<float>(0);
        }
    }

    // ///////////ESCRITA GOOGLE SHEETS/////////
    //Gera lista de objetos de valores para a escrita

    DateTime dataHoraAtual = DateTime.Now;
    string dataatual = string.Format("{0:dd/MM/yyyy HH:mm}",
dataHoraAtual);
    List<object> rowData = new List<object>() { Power,
Spindle_speed, Spindle_Load, dataatual, "Mazak_01" };

    string spreadsheetId =
"1BmqPdNzmbceeJmsd8S_uKPY2PyFTWs8NnqgVZXUu0lo"; // ID da sua planilha do Google
Sheets
    string sheetName = "Dados_Mazak"; // Nome da planilha

```

```

        // Carregar o arquivo JSON
        GoogleCredential credential;
        using (var stream = new
System.IO.FileStream("C:\\Users\\bruno\\Documents\\Bruno\\UCS\\TCC\\tcc1comunicac
aomazak-390821-988ade2df8ca.json", System.IO.FileMode.Open,
System.IO.FileAccess.Read))
        {
            credential = GoogleCredential.FromStream(stream)
                .CreateScoped(new[] {
SheetsService.Scope.Spreadsheets });
        }

        // Criar o serviço Sheets
        var service = new SheetsService(new
BaseClientService.Initializer()
        {
            HttpClientInitializer = credential,
            ApplicationName = "Google Sheets Example"
        });

        // Criar uma lista contendo os valores a serem
armazenados
        var row = new List<IList<object>> { rowData };

        // Criar um objeto ValueRange contendo as linhas de
valores
        var valueRange = new ValueRange
        {
            Values = row
        };

        // Definir os parâmetros para a atualização da planilha
        var appendRequest =
service.Spreadsheets.Values.Append(valueRange, spreadsheetId, sheetName);
        appendRequest.ValueInputOption =
SpreadsheetsResource.ValuesResource.AppendRequest.ValueInputOptionEnum.USERENTERE
D;

        // Executar a atualização da planilha
        var appendResponse = appendRequest.Execute();

        //Console.WriteLine("Leitura e armazenado de Dados
executados com sucesso!");

        // Entra em Hibernação até tempo de próxima leitura
        Thread.Sleep(TimeSpan.FromSeconds(IntervaloLeitura));
    }
    else
    {
        Console.WriteLine("Não foi possível estabelecer a conexão
com o servidor OPC");
    }
}
catch (Exception ex)
{
    Console.WriteLine($"Erro ao conectar ao servidor OPC:
{ex.Message}");
}
}

```

```
}  
  }  
}
```