

UNIVERSIDADE FEDERAL FLUMINENSE

CLEYTON DA CUNHA GOMES

**TELEMETRIA DE EQUIPAMENTOS
INDUSTRIAIS PARA A INDÚSTRIA 4.0**

NITERÓI

2022

CLEYTON DA CUNHA GOMES

TELEMETRIA DE EQUIPAMENTOS INDUSTRIAIS PARA A INDÚSTRIA 4.0

Trabalho de Conclusão de Curso submetido ao Curso de Tecnologia em Sistemas de Computação da Universidade Federal Fluminense como requisito parcial para obtenção do título de Tecnólogo em Sistemas de Computação.

Orientador:
MARCOS FIALHO DE CARVALHO

NITERÓI

2022

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

G633t Gomes, Cleyton da Cunha
TELEMETRIA DE EQUIPAMENTOS INDUSTRIAIS PARA A INDÚSTRIA 4.0
/ Cleyton da Cunha Gomes ; Marcos Fialho de Carvalho,
orientador. Niterói, 2022.
40 f.

Trabalho de Conclusão de Curso (Graduação em Tecnologia
de Sistemas de Computação)-Universidade Federal Fluminense,
Instituto de Computação, Niterói, 2022.

1. Indústria 4.0. 2. Telemetria. 3. Automação. 4.
Produção intelectual. I. Carvalho, Marcos Fialho de,
orientador. II. Universidade Federal Fluminense. Instituto de
Computação. III. Título.

CDD -

Bibliotecário responsável: Debora do Nascimento - CRB7/6368

CLEYTON DA CUNHA GOMES

TELEMETRIA DE EQUIPAMENTOS INDUSTRIAIS PARA A INDÚSTRIA 4.0

Trabalho de Conclusão de Curso submetido ao Curso de Tecnologia em Sistemas de Computação da Universidade Federal Fluminense como requisito parcial para obtenção do título de Tecnólogo em Sistemas de Computação.

Aprovado em 07 de julho de 2022.

BANCA EXAMINADORA:

Prof. Marcos Fialho de Carvalho - Orientador, UFRJ

Prof. Rafael Elias de Lima Escalfoni, CEFET/RJ

Niterói

2022

“Tudo o que temos de decidir é o que fazer com o tempo que nos é dado.” (Gandalf)

Homenagem a J. R. R. Tolkien

Agradecimentos

Agradeço a Deus por ser meu guia, me ajudar nos momentos difíceis e sempre colocar pessoas boas no meu caminho. Aos meus amigos que me apoiaram e motivaram nessa jornada. Aos meus familiares que acreditaram nos meus sonhos e depositaram em mim confiança. Aos meus professores, em especial a Rafaela Moreira por me ajudar tanto na minha jornada, e ao meu orientador Marcos Fialho pelo apoio neste trabalho. A todos meu muito obrigado!

Resumo

Com a Indústria 4.0, ou 4ª revolução industrial, muitos paradigmas da indústria estão sendo alterados, no entanto, para que ela se concretize, os maquinários das indústrias devem estar de acordo com esse novo modelo, possibilitando principalmente a coleta de informações em tempo real, auxiliando os gestores na tomada de decisões produtivas. No entanto, nem todas as máquinas da indústria são capazes de enviar suas informações de operação em tempo real e a troca de todos os equipamentos do parque industrial é inviável financeiramente, visto que o custo para essa troca seria elevado demais para a maioria das empresas. Desta forma, um dos caminhos possíveis é a atualização das máquinas já existentes, adicionando novas funcionalidades através de módulos, permitindo que elas se integrem no modelo da indústria 4.0. O presente trabalho apresenta uma ferramenta que pode ser acoplada a aparelhos legados, permitindo a coleta e disponibilização de dados operacionais. Para verificar a validade da proposta, realizamos um exemplo de aplicação com uma máquina de usinagem de eletroerosão a fio, modelo Agie Charmilles CUT 30 P. Os resultados trazem indícios de que nossa solução viabiliza a adaptação de maquinários antigos para os novos desafios da quarta revolução industrial.

Palavras-chave: Indústria 4.0, Telemetria, Automação.

Abstract

With Industry 4.0, or 4th industrial revolution, many industry paradigms are being changed, however, for it to materialize, industry machinery must comply with this new model, mainly enabling the collection of information in real time, helping managers in the decision making process. However, not all machines in the industry are able to send their operating information in real time, and the exchange of all equipment in the industrial park is financially unfeasible, since the cost for this exchange would be too high for most companies. In this way, one of the possible paths is to update existing machines, adding new functionalities through modules, allowing them to be integrated into the industry 4.0 model. The present work proposes a tool that can be coupled to legacy devices, allowing the collection and availability of operational data. We conducted a case study with a wire erosion machine, model Agie Charmilles CUT 30 P, to validate our proposal. The results show that our solution makes it possible to adapt old machines to the new challenges of the fourth industrial revolution.

Keywords: Industry 4.0, Telemetry, Automation.

Lista de Figuras

1	Agile Charmilles CUT 30 P	15
2	TCRT5000	18
3	Funcionamento TCRT5000	19
4	Exemplo MQTT	20
5	Modularização Inicial	22
6	Modulo Coletor	23
7	Sinalizador da polia	25
8	Circuito	26
9	Modulo Apresentação	28
10	Modularização completa	29
11	Protótipo de polia	30
12	Braço articulado	30
13	Consumo <i>broker</i> Mosquitto	32
14	Tabelas do banco de dados	33
15	Grafico pizza	35
16	Interface <i>web</i>	35
17	Códigos do Projeto	40

Lista de Tabelas

1	Transição sensor infravermelho	25
---	--	----

Lista de Abreviaturas e Siglas

API *Application Programming Interface*

CNI Confederação Nacional da Indústria

GND *Ground*

HTML *Hypertext Markup Language*

IIoT *Industrial Internet of Things*

IoT *Internet of Things*

M2M *Machine to Machine*

MQTT *Message Queuing Telemetry Transport*

OEE *Overall Equipment Effectiveness*

OSs Ordens de Serviço

PCP Planejamento e Controle da Produção

PyPI *Python Package Index*

QoS *Quality of Service*

SGBD Sistema Gerenciador de Banco de Dados

SSL *Secure Sockets Layer*

Sumário

1	Introdução	12
1.1	Justificativa	12
1.2	Importância	13
1.2.1	Gerência Produtiva	13
1.2.2	Gerência da Manutenção	14
1.3	Objetivos	14
2	Revisão Teórica	16
2.1	Internet of Things	16
2.2	Indústria 4.0	17
2.3	Microcontroladores	17
2.3.1	ESP32	17
2.4	Sensores	18
2.4.1	TCRT5000	18
2.5	MQTT	19
2.5.1	Mosquitto	20
2.5.2	Paho-MQTT	20
2.6	Trabalhos Relacionados	21
2.6.1	Aquisição de dados de uma estamperia mecânica	21
2.6.2	IIoT utilizando o protocolo MQTT	21
3	Metodologia	22

3.1	Coletor	23
3.1.1	Sensor	23
3.1.1.1	Polias	23
3.1.1.1.1	Câmeras	24
3.1.1.1.2	Sensor Infravermelho	24
3.1.2	Controller	25
3.1.2.1	Circuito	26
3.1.2.2	Comunicação	27
3.2	Monitor	27
3.3	Banco de Dados	27
3.4	Apresentação	28
3.4.1	Server	28
3.4.2	Interface	28
4	Desenvolvimento	29
4.1	Protótipo de polia	29
4.2	Sensor	30
4.3	Controller	31
4.4	Broker MQTT	31
4.5	Monitor	32
4.6	Banco de dados	32
4.7	Server	33
4.8	Interface	34
4.9	Trabalhos Futuros	36
5	Conclusão	37
	REFERÊNCIAS	38

Apêndice A - Fontes**40**

1 Introdução

A 4ª revolução industrial é um novo marco na história. Segundo a Confederação Nacional da Indústria (CNI), essa revolução simboliza a inserção de novas tecnologias físicas e virtuais, como por exemplo inteligência artificial, *Internet of Things* (IoT), *Big Data*, robótica avançada, computação em nuvem e impressão 3D no cenário industrial ([CONFEDERAÇÃO NACIONAL DA INDÚSTRIA, 2016](#)).

Com a implementação dessas tecnologias mencionadas, os processos fabris tendem a se tornar mais eficientes e autônomos, tendo como reflexo o a melhora dos custos e customização dos serviços oferecidos, garantindo as empresas um melhor posicionamento de mercado.

Um dos pilares desta 4ª revolução é a IoT ([COELHO, 2016](#)), que segundo o professor e economista Klaus Schwab é uma das principais pontes entre as aplicações físicas e virtuais ([SCHWAB, 2016](#)), permitindo que os elementos conectados à rede possam se comunicar.

Na indústria, a implementação da IoT tem o grande benefício de possibilitar a coleta de dados dos processos fabris, servindo como base para a tomada de decisões, pois sem dados é impossível controlar ([DEMING, 1992](#)).

1.1 Justificativa

A geração de dados na indústria 4.0 parte principalmente dos maquinários, visto que eles são as fontes dos indicadores produtivos, como disponibilidade e *Overall Equipment Effectiveness* (OEE) ([OLIVEIRA, 2021](#)). Para que se chegue a um indicador preciso, é necessário que as informações de funcionamento, como tempo de produção e peças produzidas sejam apontadas corretamente, no entanto não são todos os equipamentos que disponibilizam essas informações, tendo em vista que, na indústria brasileira, por exemplo, os equipamentos têm uma idade média de cerca de 17 anos ([CONFEDERAÇÃO](#)

[NACIONAL DA INDÚSTRIA, 2012](#)).

Os equipamentos mais antigos na manufatura brasileira são resultados do alto custo para a compra de um equipamento novo, impossibilitando que as máquinas sejam trocadas em um intervalo mais curto.

Por maquinários mais antigos não apresentarem suas informações produtivas, ou mesmo não conseguirem transmiti-las, a obtenção dos dados para os indicadores fabris depende unicamente do apontamento manual, realizado pelos próprios operadores do equipamento.

O apontamento manual, além de conter um atraso entre a origem (maquinário) e destino (gerência), pela existência de um intermediador (operador), pode conter erros operacionais, como a falta de apontamento de eventos ou o tempo incorreto da ocorrência ([TOTVS, 2019](#)).

1.2 Importância

Informações oriundas dos equipamentos industriais são fundamentais para a gerência e controle produtivo, pois são informações fidedignas e atuais sobre o equipamento, podendo impactar em tempo real a decisão dos gestores do Planejamento e Controle da Produção (PCP), como também da gerência de manutenção.

1.2.1 Gerência Produtiva

Os responsáveis pelo PCP podem se beneficiar das informações provenientes do equipamento para embasar suas tomadas de decisões. Por exemplo, monitorar o tempo de funcionamento de uma máquina, ou seja, se ela está parada ou trabalhando, serve de fonte direta para o cálculo de disponibilidade do equipamento, que por sua vez impacta diretamente na eficiência do equipamento nos cálculos de *Overall Equipment Effectiveness* (OEE) ([OLIVEIRA, 2021](#)).

Com essa informação, comparando a eficiência de dois equipamentos, por exemplo, pode ser inferido que uma máquina injetora A tem um desempenho melhor com uma determinada ferramenta do que uma máquina injetora B. Essa relação pode ser relevante na decisão de qual máquina terá maior preferência produtiva.

1.2.2 Gerência da Manutenção

Para a gerência de manutenção, informações provenientes dos equipamentos podem servir de base para abertura e preenchimento das Ordens de Serviço (OSs). Pois entre os dados monitorados podem estar os alertas da máquina, como problemas de pressão, temperatura, falta de óleo, entre outros.

Essas informações, além de terem relevância no momento que são geradas para solução do problema em tempo real, também podem servir como uma fonte para análise e detecção de futuros problemas no equipamento, evitando assim que falhas maiores paralise a produção da máquina por um tempo elevado ([BALDISSARELLI; FABRO, 2019](#)).

1.3 Objetivos

Com base no contexto exposto, o objetivo deste trabalho é o desenvolvimento de um *hardware* simples, que possa ser acoplado à máquina que se deseja monitorar, proporcionando a coleta indireta do tempo de funcionamento do aparelho, garantindo assim o fluxo dessa informação para o cálculo de disponibilidade do equipamento.

Além da criação do *hardware*, pretende-se criar um *software* que receba as informações coletadas e as armazene em um banco de dados, de forma a permitir a obtenção de futuros relatórios do equipamento. Será feita também uma interface *web* que mostrará o estado da máquina em tempo real e o tempo de produção *versus* parada através de um gráfico.

Para embasar este trabalho, será utilizada como exemplo uma máquina de usinagem de erosão a fio Agie Charmilles CUT 30 P (Figura 1). Pois como observado em uma empresa do setor metal mecânico de Nova Friburgo - RJ, esta máquina pode demorar vários dias realizando trabalho em apenas uma peça e não possui a necessidade de um operador para isso. Essas peculiaridades são interessantes para o trabalho proposto, pois mesmo com o apontamento manual dos operadores, não é garantido que ele acontecerá durante todo o período de trabalho do equipamento.

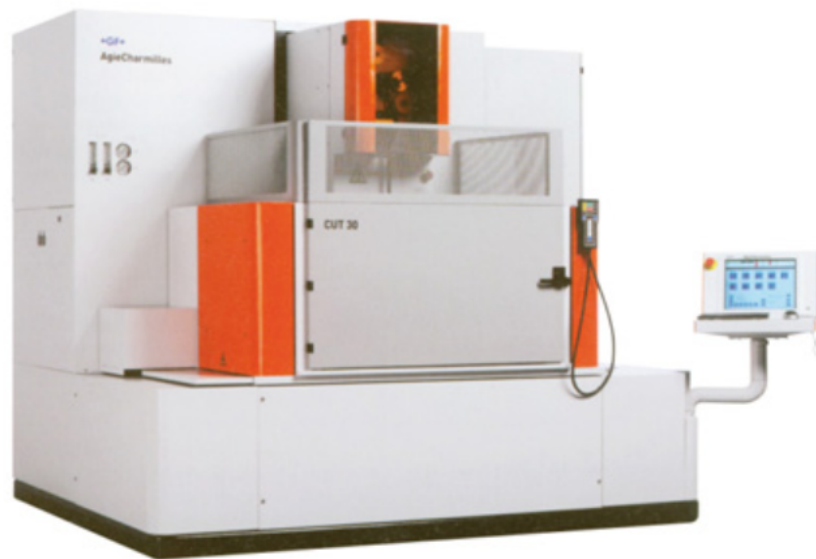


Figura 1: Agile Charmilles CUT 30 P
Fonte: [AgieCharmilles \(2009\)](#).

2 Revisão Teórica

Neste capítulo serão abordados alguns conceitos que são a base do trabalho desenvolvido, como *Internet of Things* (IoT), Indústria 4.0, Microcontroladores, Sensores e *Message Queuing Telemetry Transport* (MQTT). Além dos trabalhos acadêmicos que serviram como base para o desenvolvimento deste artigo.

2.1 Internet of Things

Internet of Things (IoT) diz respeito a elementos físicos (coisas) interligados através de uma rede, permitindo assim que os dispositivos possam se comunicar, possibilitando um melhor desempenho em suas funções. Segundo ([COELHO, 2016](#)), a IoT pode ser entendida como uma maneira de conectar dispositivos de forma a realizarem transmissão e recepção de dados sem a intervenção humana.

Essas “coisas”, a que a sigla IoT se refere, podem ser lâmpadas, tomadas, cafeteiras, roupas e até mesmo carros e geladeiras, dependendo da necessidade e da criatividade.

Quando esta tecnologia é aplicada no ambiente industrial, temos a formação de outro termo, a *Industrial Internet of Things* (IIoT), tratando-se de equipamentos, máquinas, locais e ativos conectados, gerando dados que são transmitidos para a rede, para poderem ser processados e analisados, gerando como consequência a melhoria da eficiência produtiva ([RED HAT, 2021](#)).

Embora a sigla IIoT diga a respeito da indústria, essa inovação não somente é aplicada a ela, pois pode ser utilizada também em outros ramos, como a agricultura, medicina e turismo, por exemplo.

2.2 Indústria 4.0

Também conhecida como a 4ª revolução industrial, a indústria 4.0 é um movimento de transformação na indústria, onde tecnologias como IoT, realidade aumentada, *Big Data*, inteligência artificial e *machine learning* são mais integradas ao ambiente fabril, de forma a compreender melhor os processos industriais e assim poder melhorar o desempenho, garantindo assim um melhor posicionamento de mercado.

2.3 Microcontroladores

Microcontroladores podem ser entendidos como um microcomputador em um circuito integrado, pois possuem todos os componentes necessários e comuns a um computador, como processador, memória e periféricos de entrada e saída.

Um dos microcontroladores mais conhecidos da atualidade é o Arduino UNO, uma plataforma aberta que ficou muito conhecida pela sua facilidade de uso e gama de aplicações possíveis. Atualmente a empresa alcançou a marca de 10 milhões de placas vendidas ([ARDUINO, 2021](#)), isso apenas considerando as placas originais da marca, desconsiderando as placas similares que usam o mesmo *hardware*, como o Franzino e o Seeduino.

Um dos grandes benefícios do Arduino é a comunidade extremamente rica em conteúdo, possuindo um fórum bem colaborativo e com exemplos bem didáticos até mesmo para pessoas que nunca viram ou possuem pouco conhecimento em programação ou eletrônica.

2.3.1 ESP32

O ESP32 é um microcontrolador desenvolvido pela Expressif Systems® que conta com um *hardware* bem robusto quando comparado com outros microcontroladores. Este microcontrolador possui nativamente integração ao *wi-fi* e *bluetooth*, além de um processador *dual core*. A programação deste dispositivo é bem fácil, utilizando o ambiente de desenvolvimento Arduino, sem necessitar de muitas configurações.

Outra característica importante deste microcontrolador, garantida pela própria fabricante, é a sua robustez, facilitando a sua implementação em um ambiente industrial, visto que as variáveis de ambiente, como temperatura, tendem a prejudicar dispositivos eletrônicos. Além disso, ele permite a atualização do *software* e *firmware* remotamente,

proporcionando, assim, a possibilidade de posicioná-lo em ambientes de difícil acesso.

2.4 Sensores

Sensores são dispositivos capazes de captar informações do meio, como luminosidade, temperatura e velocidade; a partir dessa coleta, é possível converter essas informações em sinais elétricos, podendo, com isso, serem interpretadas dentro de um circuito eletrônico.

2.4.1 TCRT5000

O sensor TCRT5000 (Figura 2) é um sensor reflexivo que possui um emissor infravermelho posicionado ao lado de um receptor infravermelho. Quando um material é posicionado à frente deste sensor, o infravermelho é refletido, e, dependendo da cor do material, a quantidade refletida é diferente. Se o material for claro, o infravermelho consegue ser refletido com maior facilidade, diferentemente de superfícies escuras, onde ele é mais absorvido (Figura 3).



Figura 2: TCRT5000
Fonte: [Euler Oliveira \(2019\)](#).

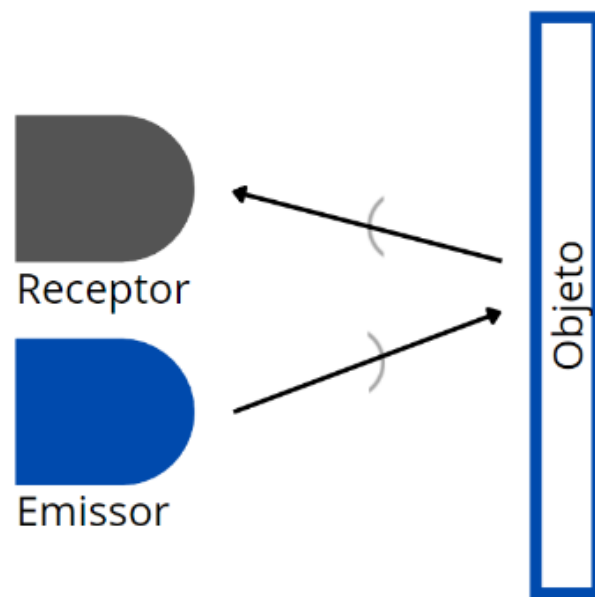


Figura 3: Funcionamento TCRT5000

2.5 MQTT

O *Message Queuing Telemetry Transport* (MQTT) é um protocolo *Machine to Machine* (M2M) desenvolvido pela IBM® na década de 90. Este protocolo faz uso da pilha TCP/IP¹ e possui como fundamentação o modelo de *Publishers* ou Publicadores (Tradução livre) e *Subscribers* ou Inscritos (Tradução livre). Neste modelo, o publicador faz a publicação de uma informação, como a temperatura de um sensor, por exemplo, em um tópico, que pode ser entendido como um título. Dessa forma, todos os dispositivos que estão inscritos neste tópico recebem a informação publicada (NERI; LOMBA; BULHÕES, 2019).

Entre o publicador e os inscritos existe o *broker*, um servidor que gerencia o tráfego das informações entre os dispositivos (Figura 4).

Este protocolo é muito útil para criação de sistemas IoT, pois é um protocolo leve, aberto e simples, não sobrecarregando a rede e nem os dispositivos. Além disso, outra característica importante é que este protocolo possui uma documentação extensa e muitos exemplos de aplicação para diferentes linguagens de programação.

¹Conjunto de protocolos que permitem aos dispositivos de uma rede se comunicarem.

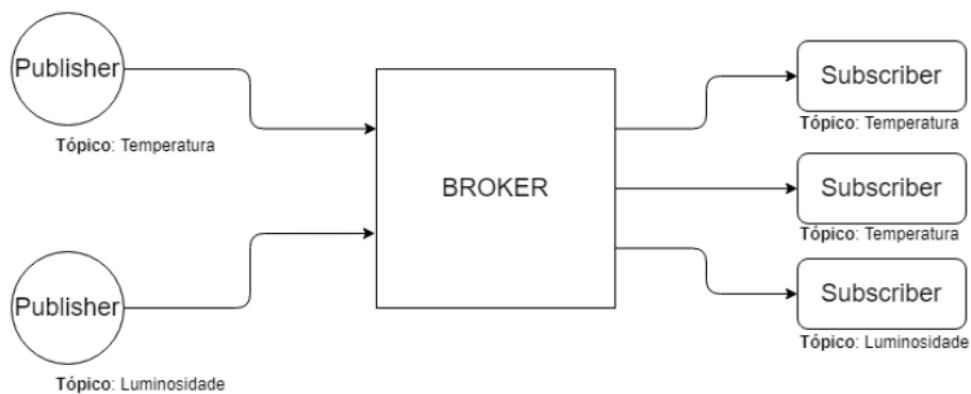


Figura 4: Exemplo MQTT

Uma das possibilidades deste protocolo é a garantia de recebimento das mensagens, acessível através do parâmetro de *Quality of Service* (QoS). Este parâmetro possui três estados possíveis (0, 1 e 2). Em caso de QoS igual a 0, a mensagem não tem confirmação de entrega, ou seja, se assemelha ao protocolo UDP², pois corre o risco de a mensagem não chegar aos dispositivos inscritos, podendo no máximo chegar uma só vez. No caso do QoS igual a 1 a informação chegará com certeza ao menos uma vez, ou seja, pode chegar mais de uma vez devido a demora na confirmação de recebimento. E quando o QoS for 2, é certo que a mensagem chegará exatamente uma vez ([GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO, 2018](#)).

2.5.1 Mosquitto

O Mosquitto é um *broker* de *software* aberto que implementa o MQTT. Ele está disponível para os sistemas operacionais Linux, Windows® e Mac®. Sua instalação e configuração é simples e pode ser feita através do arquivo `.config` na raiz do programa. O *broker* Mosquitto, através de seu arquivo de configuração, permite entre muitas coisas ajustar a porta que será usada para comunicação (por padrão é a porta 1883), configurar um usuário e senha e criptografar a comunicação com *Secure Sockets Layer* (SSL)³.

2.5.2 Paho-MQTT

Paho-MQTT é uma biblioteca *open-source* disponível para a linguagem de programação Python, ela permite a conexão com *brokers* MQTT, como o Mosquitto, possibilitando assim que aplicações possam publicar ou receber informações nos tópicos de interesse.

²Protocolo para transmissão de arquivos na rede sem a confirmação de recebimento

³Protocolo que garante mais segurança na troca de informações na rede.

A instalação desta biblioteca é fácil e pode ser feita através do *Python Package Index* (PyPI), o gerenciador de pacotes do Python.

2.6 Trabalhos Relacionados

O desenvolvimento de equipamentos que se integrem aos maquinários industriais objetivando a melhora produtiva não é recente, outros trabalhos já abordaram esse tema. Dois destes trabalhos serão apresentados nesta seção. O primeiro deles apresenta um sistema de coleta de dados em tempo real de uma estamperia mecânica. Já o segundo fala sobre a coleta de dados em ambiente industrial e a sua transmissão com o protocolo MQTT.

2.6.1 Aquisição de dados de uma estamperia mecânica

Neste trabalho, Oliveira (2016), propõe um sistema para aquisição de dados em uma Estamperia Mecânica. Neste sistema, foi utilizado para coleta dos dados o microcontrolador Arduino Uno, o qual adquire os dados produtivos e alertas da máquina de estampar e envia para um servidor, onde a informação é armazenada no banco de dados. Além disso, neste trabalho foi construída uma interface para a *web*, onde os dados podem ser analisados pelos responsáveis do setor de Planejamento e Controle da Produção (PCP).

2.6.2 IIoT utilizando o protocolo MQTT

Neste trabalho, realizado pelos alunos Fábio Cassio de Souza, Ricardo Donizeti Soares e Roberson José Campos (2021), foi proposta a coleta de dados provenientes do chão de fábrica e seu tratamento, com a finalidade de obter-se mais informações provenientes dos setores produtivos, facilitando assim a tomada de decisões e garantindo a melhoria na eficiência dos processos.

De forma a demonstrar um sistema de coleta de dados, neste trabalho é construído um sistema que coleta a temperatura do ambiente e transmite essa informação através do protocolo MQTT para um servidor, onde essa informação é armazenada em um banco de dados.

3 Metodologia

O desenvolvimento do trabalho possui duas etapas, a primeira é de natureza exploratória, buscando entender a melhor forma de obtenção do estado da máquina (trabalhando/parada) e as tecnologias necessárias para a construção do sistema, a outra é de natureza prática, mostrando a construção do sistema proposto.

De forma a entender melhor o objetivo deste trabalho, pode-se modularizar o sistema, desde a coleta das informações na máquina, a inserção no banco de dados, até a apresentação para o usuário em uma aplicação *web*. Desta maneira podemos dividir o sistema nos seguintes módulos (Figura 5).

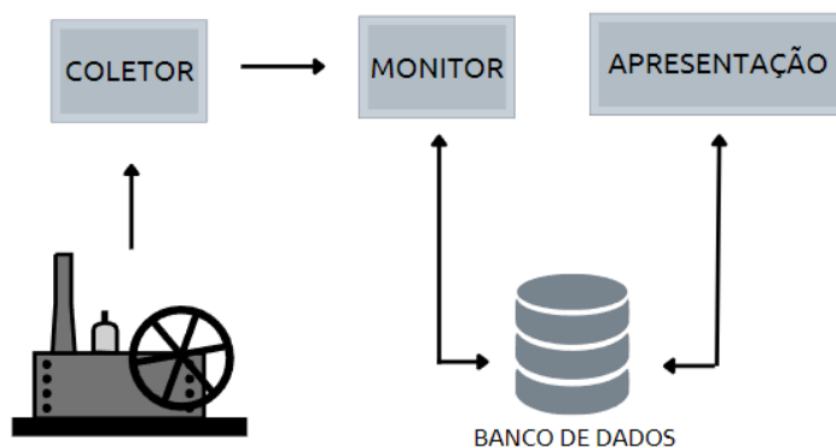


Figura 5: Modularização Inicial

Cada um dos módulos apresentados acima (Figura 5) terão as seguintes responsabilidades no sistema:

Coletor : Responsável por coletar a informação de funcionamento da máquina (trabalhando ou parada) e enviar para o Monitor.

Monitor : Responsável por receber a informação vinda do Coletor e armazená-la no banco de dados.

Apresentação : Responsável por pegar as informações armazenadas no banco de dados e apresentar para o usuário em uma aplicação web.

3.1 Coletor

O Coletor, como já explicitado, tem como responsabilidade coletar as informações da máquina e enviar para o próximo estágio, que será o Monitor. Sendo assim, podemos subdividir o módulo Coletor em duas partes (Figura 6).

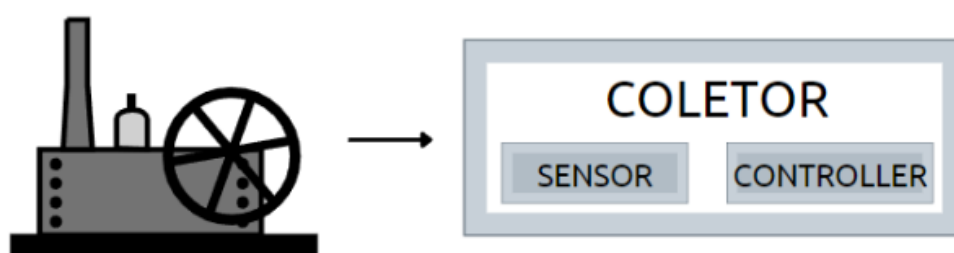


Figura 6: Módulo Coletor

Nessa divisão do Coletor, teremos o Sensor que será o módulo responsável unicamente por coletar a informação de trabalho da máquina, e teremos o módulo Controller, que será responsável por receber a informação coletada pelo módulo Sensor, processar, e enviar para o módulo Monitor.

3.1.1 Sensor

O primeiro passo para o desenvolvimento deste módulo foi analisar a máquina de erosão a fio CUT 30 P(Figura 1), com o objetivo de descobrir qual seria a melhor forma de obter-se indiretamente o estado de funcionamento do aparelho (trabalhando/parada).

Pretende-se realizar esta leitura de forma indireta, buscando não interferir no funcionamento do aparelho, posto que, caso fosse implementado um sistema mais dependente do *hardware* da máquina, isso poderia acarretar problemas, principalmente para os responsáveis pela sua manutenção.

3.1.1.1 Polias

Realizando-se uma análise dos equipamentos de usinagem na empresa na qual teve-se acesso, foi possível verificar que as máquinas de erosão a fio usadas pela empresa,

independente do modelo, possuem um sistema de polias, as quais têm como objetivo permitir a passagem do fio responsável pela erosão da peça modelada.

Ao entrevistar os operadores do equipamento, foi sinalizado que os movimentos das polias serviriam como métrica para verificar o estado do equipamento, pois quando a máquina está parada/desligada elas não giram, voltando a girar somente quando o aparelho está em funcionamento.

Desta maneira foi definido que o movimento da polia seria a fonte de informação do estado da máquina. Mas, para isso, torna-se necessário também definir qual seria o sensor responsável para se verificar o giro das polias.

Foram levantadas, nesse intuito, duas soluções possíveis: uma delas através do uso de uma câmera e outra através de um sensor infravermelho.

3.1.1.1.1 Câmeras

O uso de uma câmera, embora seja eficaz, impacta diretamente na complexidade do projeto, visto que, para isso, seria necessário um *hardware* potente para realizar a análise de um arquivo de vídeo em tempo real, além desse método também abrir brechas para falhas, como a definição da luminosidade do ambiente, por exemplo.

3.1.1.1.2 Sensor Infravermelho

O sensor infravermelho torna-se uma solução viável quando comparado com a câmera, visto que a maioria dos microcontroladores convencionais podem ler uma informação analógica/digital, além de não depender da luminosidade do ambiente. No entanto, assim como a câmera, ele depende da existência de alguma marcação na polia que permita ao sistema analisar que houve alteração entre uma coleta de informação no momento t_1 e em um momento seguinte t_2 .

Para este sinalizador, foi desenvolvida a lógica de impressão de um círculo, metade preto e metade branco, que cobriria a parte superior da polia. Dessa forma, quando o sensor infravermelho estiver posicionado na parte branca, o valor lido no sensor será 1. E quando estiver sobre a parte preta o valor lido será 0 (Figura 7).

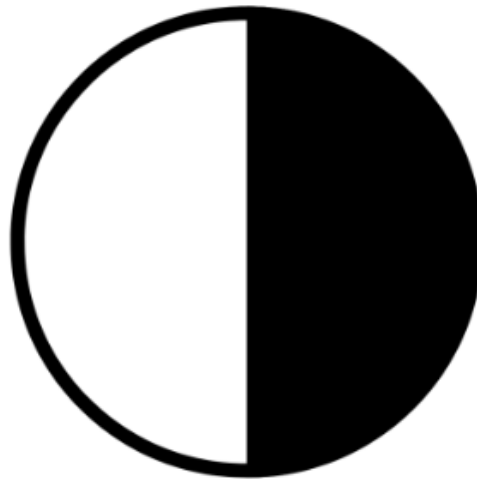


Figura 7: Sinalizador da polia

O estado da máquina (trabalhando/parada) será obtido através da leitura consecutiva dos valores do sensor, coletando o estado através das transições de leitura. Caso o valor lido no sensor mantenha-se o mesmo do lido anteriormente, o estado da máquina será definido como "parada", mas caso seja detectada uma transição no valor, ou seja, o sensor estava posicionado em cima do preto e foi para o branco, ou vice-versa, o estado da máquina será alterado para "em trabalho" (Tabela 1).

Tabela 1: Transição sensor infravermelho

LEITURA	ESTADO
0 \rightarrow 0	PARADA
0 \rightarrow 1	EM TRABALHO
1 \rightarrow 0	EM TRABALHO
1 \rightarrow 1	PARADA

3.1.2 Controller

O Controller será o dispositivo receptor da variável que queremos analisar do meio, neste caso a rotação da polia da máquina. Ele irá coletar essa informação e processá-la, de forma a determinar se a máquina está trabalhando ou parada, e, depois desse processo, enviar para o módulo Monitor.

O microcontrolador escolhido para desempenhar esta função foi o ESP32 (Seção 2.3.1), visto que ele possui nativamente embutido o módulo *wi-fi*, beneficiando o projeto, já que por ser uma conexão sem fio, na prática da implementação deste módulo em uma máquina, não será necessário a instalação de cabeamento de rede até o local, somente nas

proximidades até um ponto de acesso sem fio¹.

Como dito anteriormente, para verificar o movimento da polia, necessita-se monitorar quando ocorre uma transição na leitura do sensor. Para o nosso microcontrolador, isso será entendido como uma transição na leitura do pino digital, alternando do estado alto (HIGH) para baixo (LOW) ou baixo (LOW) para alto (HIGH).

Para detectar a transição, foi definido usar a função `'attachInterrupt()'`, que monitora uma porta digital, e, assim que uma transição for detectada, realiza a chamada da função que é passada por parâmetro.

3.1.2.1 Circuito

O circuito para verificar a rotação da polia da máquina monitorada é bem simples, pois é composto somente por dois elementos, o microcontrolador ESP32 e o módulo infravermelho com o sensor TCRT500 (Figura 8).

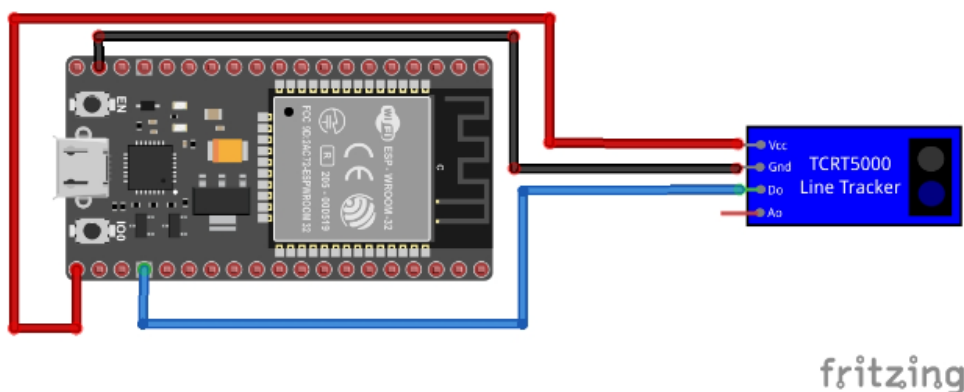


Figura 8: Circuito

O microcontrolador usado possui ao todo 24 pinos disponíveis para entrada de dados, e deles só será usado a porta digital 2 para conectar a saída digital do módulo TCRT5000. Além do pino de dados, o módulo faz uso também do pino de alimentação de 3.3 *volts* e do pino *Ground* (GND) do microcontrolador.

Para realizar a alimentação do microcontrolador ESP32, faz-se necessária a utilização de uma fonte de 5v, por isso optou-se pela saída USB do computador, suficiente para essa aplicação, visto o consumo baixo de corrente (40mA).

¹Dispositivo que permite a conexão de dispositivos móveis a rede sem a necessidade de um cabo

3.1.2.2 Comunicação

Para enviar as informações de trabalho da máquina para o Monitor, optou-se por utilizar o protocolo *Message Queuing Telemetry Transport* (MQTT), devido aos seus benefícios de ser um protocolo leve e não depender de *hardwares* robustos (Seção 2.5). Além disso, é de grande auxílio este protocolo possuir a biblioteca PubSubClient (LEARY, s.d.) para o microcontrolador escolhido, com bons exemplos de uso.

Com a escolha do protocolo MQTT, será necessária a instalação de um servidor *broker*, o qual será responsável por gerir as informações que trafegam entre *Subscribers* e *Publishers*. Para a função do *broker* foi escolhido o Mosquitto, pois, como já mencionado neste trabalho (Seção 2.5.1) é software livre, que não necessita de um *hardware* robusto para ser executado e é disponibilizado para os principais sistemas operacionais do mercado.

3.2 Monitor

O Monitor, por sua vez, será responsável por receber as informações vindas do microcontrolador ESP32 através do protocolo MQTT, além de armazená-las no banco de dados. Dessa forma, para desenvolvimento deste *script* utilizou-se a linguagem de programação Python, em virtude de esta ter sido a principal linguagem trabalhada durante o curso de Computação. Além disso, o fato desta linguagem possuir a biblioteca Paho-MQTT facilita a implementação do protocolo, possuindo, também, a biblioteca MySQL para facilitar a inserção de dados no banco de dados.

3.3 Banco de Dados

Para o banco de dados foi escolhido o Sistema Gerenciador de Banco de Dados (SGBD) MySQL versão 10.4.24, visto que é um *software open source* e suficiente para esta aplicação.

3.4 Apresentação

A função do módulo de Apresentação será coletar a informação do banco de dados e mostrar para o usuário através do navegador em uma interface *web*. Tendo isso em vista, é possível separá-lo em dois novos módulos: o módulo Server, que irá coletar a informação do banco de dados quando for requisitada, e o módulo Interface, que será responsável por requisitar ao servidor as informações das máquinas e apresentá-las para os usuários no navegador (Figura 9).

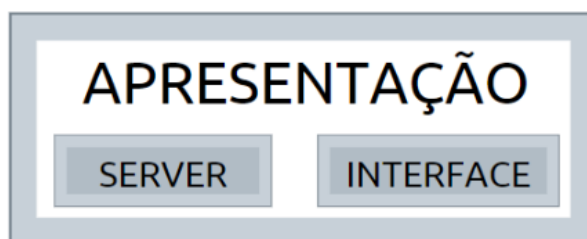


Figura 9: Módulo Apresentação

3.4.1 Server

Para o Server foi escolhida a linguagem Python em conjunto com a biblioteca Flask, responsável por criar a *Application Programming Interface* (API) que será consumida pelo módulo Interface, e a biblioteca MySQL, responsável por obter os dados do banco de dados.

3.4.2 Interface

Para a interface, utilizou-se o *Hypertext Markup Language* (HTML), que será responsável pela apresentação das informações. Em adição, também aplicou-se *framework* Bootstrap para realizar a estilização da tela, e a biblioteca ChartJs para a criação dos gráficos de tempo de trabalhos das máquinas. Para a requisição dos dados da API do Server, será utilizada a linguagem de programação Javascript, em conjunto com o *framework* jQuery.

4 Desenvolvimento

Neste capítulo será demonstrado o desenvolvimento de cada um dos módulos supracitados. De forma a melhorar a representação e o entendimento, pode-se expandir o diagrama exposto na Figura 5, apresentando os módulos de Interface, Server, além do servidor *broker*.

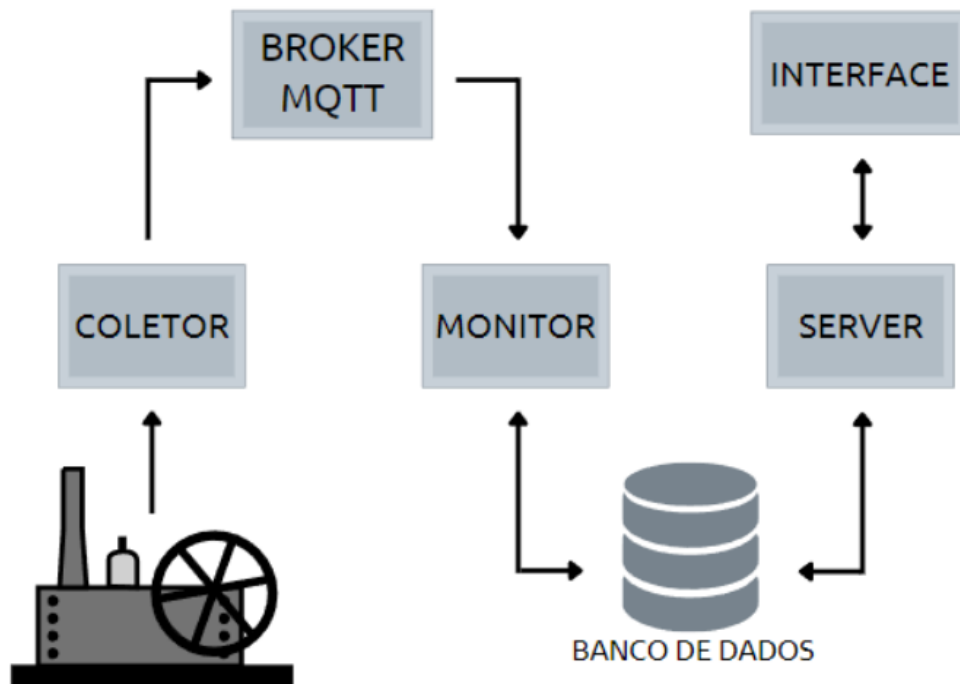


Figura 10: Modularização completa

4.1 Protótipo de polia

Para desenvolver os módulos e realizar os testes, foi primariamente necessário desenvolver um protótipo de polia. Para isso, foi utilizado um disco de papelão, metade coberto com uma folha de papel A4 e a outra metade com papel E.V.A preto. Para simular o movimento de giro da polia, no centro do disco foi acoplado um micro servo 9g SG90 TowerPro™(Figura 11).

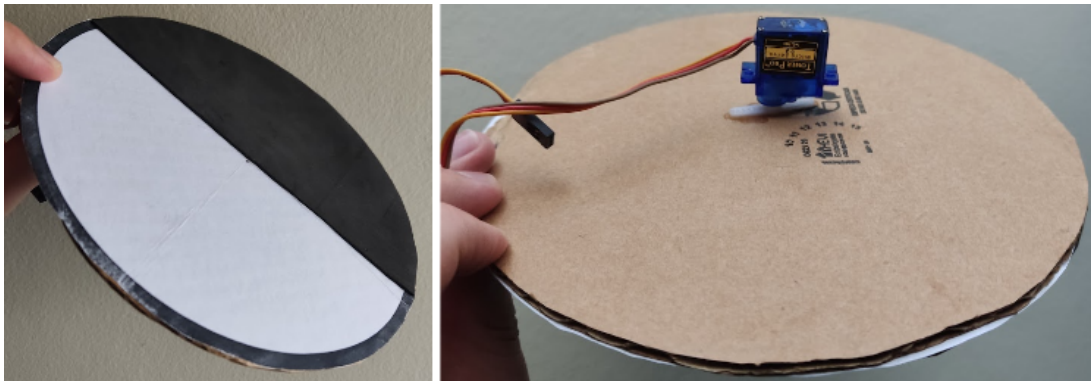


Figura 11: Protótipo de polia

4.2 Sensor

Para posicionar o módulo TCRT5000 sobre o disco, foi construído, utilizando uma impressora 3D, um braço regulável, composto de 4 partes acopladas com parafusos de 5mm. O sensor foi fixado na ponta do braço, permitindo uma maior proximidade em relação ao disco (Figura 12).

A modelagem foi realizada no software Tinkercad da empresa Autodesk®[®], acessível na internet através de qualquer navegador *web*.

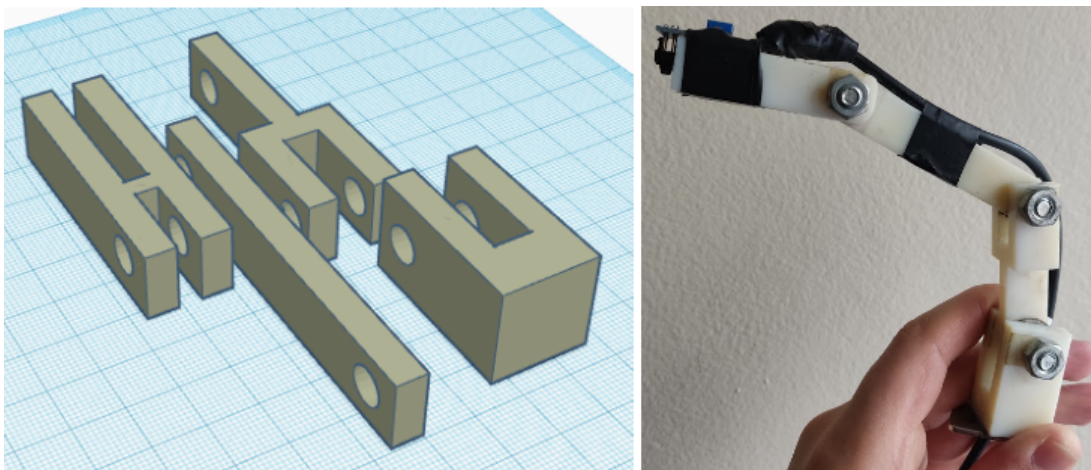


Figura 12: Braço articulado

Na base do braço articulado foi fixado um ímã de neodímio, proporcionando a fixação do braço em uma superfície metálica, necessária para a implementação deste módulo em uma máquina industrial. Além disso, no módulo TCRT5000 foi acoplado um cabo de 4 vias para permitir ao módulo conectar-se ao microcontrolador.

4.3 Controller

O Controller irá publicar a informação sobre o estado da máquina monitorada no tópico “MAQUINA/ID”, sendo ID o identificador da máquina cadastrado no banco de dados (Seção 4.6). Isso irá permitir que todos os aparelhos que publicam neste tópico sejam monitorados pelo módulo Monitor.

Para o envio das informações oriundas do Controller para o Monitor, foi designado o formato json¹, pois é leve, simples e fácil de ser trabalhado na linguagem Python. Quando a máquina estiver parada, será enviada a mensagem "{ESTADO : PARADA}", e quando estiver em trabalho, será enviada a mensagem "{ESTADO : EM PRODUÇÃO}".

O intervalo para o envio das informações para o *broker* foi convencionado em 30 segundos, e o *Quality of Service* (QoS), ou seja, a garantia do envio como ‘0’, desta forma a mensagem enviada não terá garantia de entrega.

Assim que o Controller é iniciado, o sistema considera que a máquina monitorada está desligada, mas quando é detectada uma transição na leitura do sensor (Seção 3.1.1.1.2), o estado da máquina é modificado para “Em Trabalho”. Caso não exista uma nova transição na leitura dentro de 5 minutos, a máquina será considerada como “Parada”.

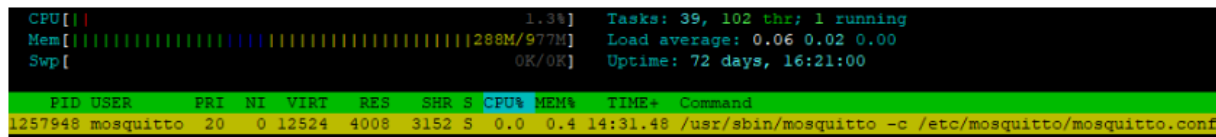
4.4 Broker MQTT

Para o *broker* MQTT, como dito anteriormente (Seção 3.1.2.2), foi utilizado o *broker* Mosquitto. Nesta aplicação, foi realizado o *upload* do Mosquitto em um servidor Linux com Ubuntu® versão 20.04. A instalação foi bem simples, bastando apenas um comando via console para o programa já estar rodando na máquina.

Para garantir a segurança do servidor, foi configurado um usuário e senha, de forma a proteger o *broker* contra a entrada de dispositivos que poderiam se inscrever ou publicar nos tópicos usados, além disso foi bloqueada a conexão de usuários anônimos ao *broker*.

Depois da instalação e configuração, foi verificado o quanto estava sendo consumido do servidor. Para isso, foi utilizado o comando “htop”. Pela imagem abaixo podemos ver um baixo consumo da aplicação após sua instalação.

¹Formato simples para a troca de dados

Figura 13: Consumo *broker* Mosquitto

4.5 Monitor

Para o módulo Monitor foi criado um arquivo de configuração "config.ini", onde as credenciais do banco de dados e do *broker* são armazenadas para a conexão do Monitor. Além das credenciais, são especificadas as portas de conexão e o tópico em que o Monitor será inscrito.

O Monitor foi desenvolvido para se inscrever no tópico "maquinas/+". Dessa forma, múltiplas máquinas poderão ser monitoradas simultaneamente. Quando uma mensagem é publicada neste tópico e é recebida pelo Monitor, o primeiro passo é verificar se a origem desta publicação é de uma máquina cadastrada no Banco de Dados.

Caso a máquina em questão esteja cadastrada, o Monitor verifica qual era o estado anterior da máquina no banco de dados, se a mensagem recebida informar que o estado atual é diferente do anterior, o estado no banco de dados é alterado para recebido.

4.6 Banco de dados

Para o banco de dados foram criadas duas tabelas, a tabela "máquinas", responsável pelo cadastro das máquinas monitoradas pelo sistema, e a tabela "históricos", que mantém os registros de estados das máquinas cadastradas.

Na tabela de máquinas devem estar cadastradas todas as máquinas que serão monitoradas. Nela temos os campos de nome e identificador da máquina.

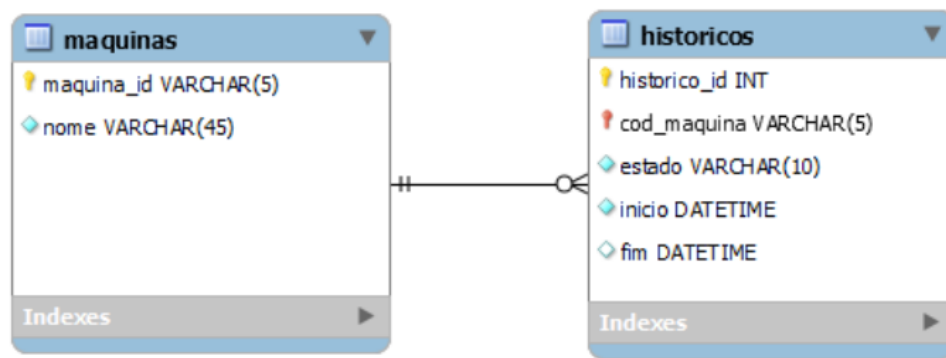


Figura 14: Tabelas do banco de dados

nome : Campo de texto para receber nome da máquina ou modelo.

máquina_id : Campo identificador da máquina, o mesmo usado no microcontrolador e Coletor para identificar a máquina monitorada, no máximo 5 caracteres alfanuméricos.

A tabela "históricos" conta com todo o histórico de funcionamento (trabalhando/parada) das máquinas monitoradas, esta tabela ainda possui campos relacionados ao identificador do histórico, código da máquina monitorada, estado do monitoramento, início e fim do estado.

historico_id : Contador auto incremental do histórico.

cod_maquina : Código da máquina que faz referência.

estado : Estado da máquina monitorada (Trabalhando/Parada).

inicio : Data e Hora do início do estado.

fim : Data e Hora do fim do estado.

4.7 Server

O Server foi construído para possibilitar que a Interface requisite as informações de trabalho das máquinas. Para isso, foi construída uma API que permite a requisição de duas solicitações que podem ser feitas pela Interface.

get maquinas : Retorna as máquinas cadastradas no sistema em JSON, contendo somente a máquina e seu identificador.

get estados_maquinas : Retorna o funcionamento dos equipamentos cadastrados em JSON, contendo o estado atual das máquinas cadastradas.

get maquina/<codigo maquina> : Retorna em JSON o tempo em trabalho e o tempo em parada da máquina informada.

Um dos benefícios da construção de uma *Application Programming Interface* (API) é que, através dela, as informações coletadas podem ser usadas em outros sistemas, como por exemplo uma aplicação para o Telegram®², na qual as informações de trabalho das máquinas monitoradas seriam enviadas quando solicitadas através de um *chat*.

4.8 Interface

Para o módulo de Interface, foi construído uma página para a *web* que, ao ser acessada, busca o histórico de todas as máquinas cadastradas no banco de dados. Essa informação chega a ela através de uma requisição assíncrona feita para a API do módulo Server.

De forma a apresentar as informações em um contexto visual, foi construído, com o auxílio da biblioteca ChartJS um gráfico em pizza que apresenta a informação de funcionamento da máquina, com o tempo de trabalho em verde e o tempo de parada em vermelho (Figura 15).

Além do gráfico, a interface informa o estado das máquinas monitoradas em tempo real, sendo estes "Em Trabalho", "Parada", ou "Desconhecido", quando o dispositivo de monitoramento da máquina está desconectado (Figura 16).

As informações da tela são recarregadas automaticamente a cada 5 minutos, sem a necessidade de recarregar a página manualmente. Assim, uma das possíveis utilizações para essa aplicação seria a apresentação em um *Andon*³, que poderia estar localizado nos pontos de interesse, como o setor de Planejamento e Controle da Produção (PCP), manutenção ou mesmo no próprio setor das máquinas monitoradas.

²Aplicativo de mensagens

³Ferramenta para sinalizar de forma visual ou sonora o estado das máquinas

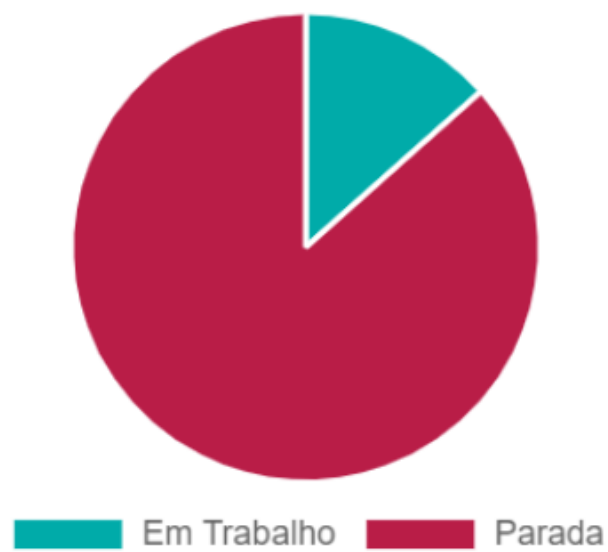
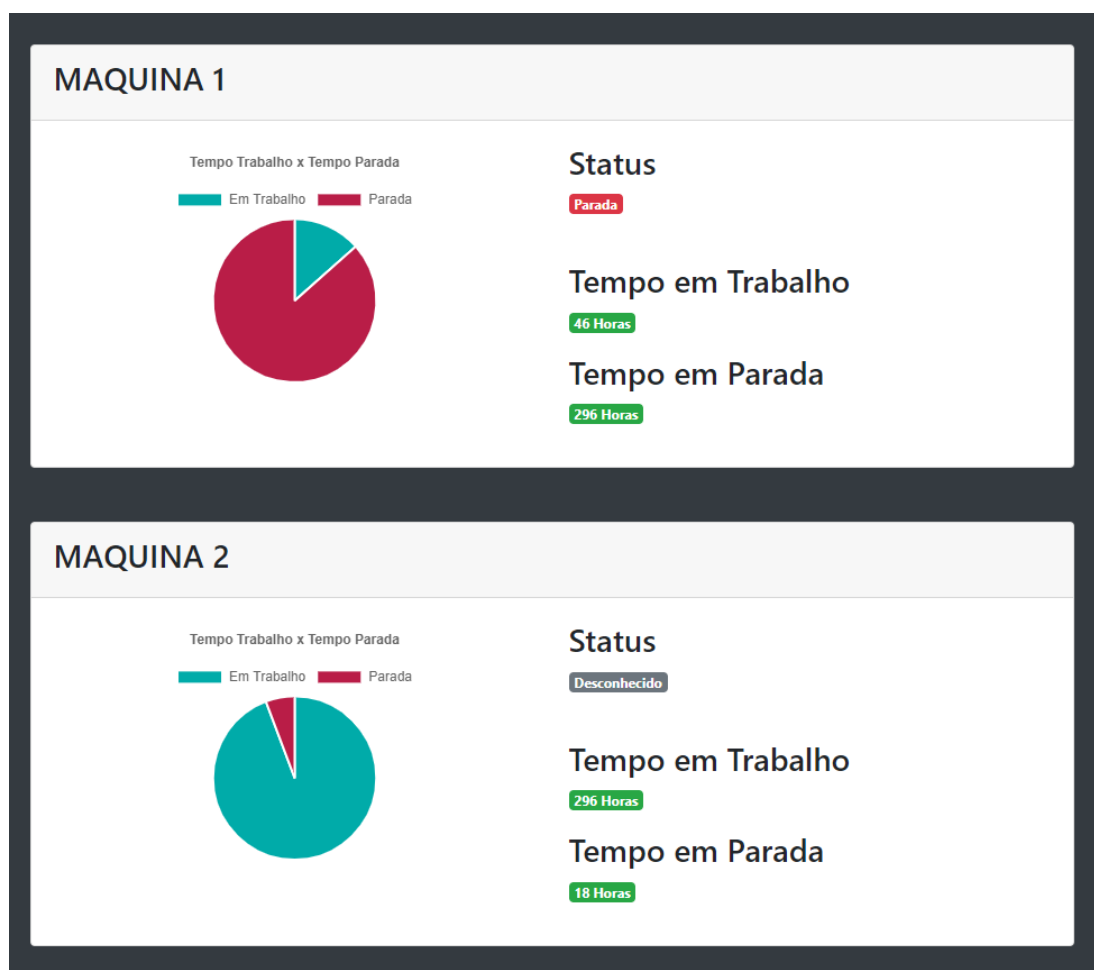


Figura 15: Grafico pizza

Figura 16: Interface *web*

4.9 Trabalhos Futuros

Uma das observações feitas durante o desenvolvimento deste trabalho é pautada no fato do sistema desenvolvido poder permitir que o operador da máquina aponte os motivos de parada do equipamento. Isto é, após o módulo Monitor receber um estado de parada na máquina, seria habilitada uma caixa de entrada na interface *web*, na qual o operador poderia apontar o motivo de ocorrência da parada.

Com o apontamento do motivo de parada, cresce o volume de dados fornecidos aos gestores a fim de embasar suas tomadas de decisões, podendo, dessa forma, calcular os indicadores produtivos de forma automática e em tempo real.

É válido salientar que, em uma aplicação real do sistema apresentado, torna-se necessária a criação de uma estrutura para proteção do microcontrolador, evitando danos externos. Além da estrutura, faz-se essencial também a compra de uma fonte de alimentação, visto que a utilizada neste trabalho foi a porta USB do computador.

5 Conclusão

A indústria 4.0 trouxe muitas possibilidades para os processos produtivos, mas com ela também surgiram alguns desafios, como é o caso da renovação dos maquinários no Brasil. Como forma de contornar este problema, o presente trabalho apresentou o desenvolvimento de um sistema simples que permite o monitoramento remoto de uma máquina de eletroerosão a fio.

Sobre o protótipo, nos testes realizados fora do ambiente industrial o aparato respondeu segundo as expectativas, não foram verificadas falhas de comunicação com o *broker* ou com o *wi-fi*. Além disso, as requisições feitas através da página *web* para a *Application Programming Interface* (API) funcionaram conforme o esperado.

De maneira geral o sistema pode ser utilizado para aferir o tempo de trabalho do equipamento, garantindo a base necessária para o cálculo de disponibilidade e *Overall Equipment Effectiveness* (OEE).

Acerca das comunicações, um ambiente industrial pode proporcionar alguns problemas, já que podem existir vários dispositivos conectados simultaneamente na rede, acarretando até mesmo a perda da conexão via *wi-fi*.

REFERÊNCIAS

AGIECHARMILLES. **AgieCharmilles lança máquinas de eletroerosão**. Disponível em https://www.cimm.com.br/portal/noticia/exibir_noticia/4777-agiecharmilles-lanca-maquinas-de-eletroerosao, acessado em 20/05/2022. fev 2009.

ARDUINO, TEAM. **Introducing the Arduino UNO Mini Limited Edition**. Disponível em <https://blog.arduino.cc/2021/11/24/introducing-the-arduino-uno-mini-limited-edition-pre-orders-now-open/>, acessado em 10/05/2022. 2021.

BALDISSARELLI, Luciano; FABRO, Elton. Manutenção Preditiva na indústria 4.0. **SCIENTIA CUM INDUSTRIA**, v. 7, n. 2, p. 12–22, 2019. Disponível em <https://eg.uc.pt/handle/10316/36992>, acessado em 10/05/2022.

COELHO, Pedro Miguel Nogueira. Rumo a Indústria 4.0, 2016. Dissertação (Mestrado em Engenharia e Gestão Industrial), Universidade de Coimbra, Coimbra, Portugal, Disponível em <https://eg.uc.pt/handle/10316/36992>, acessado em 10/05/2022.

CONFEDERAÇÃO NACIONAL DA INDÚSTRIA. **A Indústria de máquinas na era da economia verde**. Disponível em <https://www.portaldaindustria.com.br/publicacoes/2012/9/a-industria-de-maquinas-na-era-da-economia-verde/>, acessado em 28/06/2022. 2012.

_____. **Desafios para Indústria 4.0 no Brasil**. Disponível em <https://www.portaldaindustria.com.br/publicacoes/2016/8/desafios-para-industria-40-no-brasil/>, acessado em 28/06/2022. 2016.

DEMING, William Edwards. **Qualidade: A Revolução da Administração**. Editora Saraiva, 1992.

GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO. **O protocolo MQTT**. Disponível em <https://www.gta.ufrj.br/ensino/eel878/redes1-2018-1/trabalhos-vf/mqtt/>, acessado em 10/05/2022. 2018.

LEARY, Nick. **Arduino Client for MQTT**. Disponível em <https://pubsubclient.knolleary.net/>, acessado em 20/05/2022.

NERI, R.; LOMBA, M.; BULHÕES, G. MQTT, 2019. Disponível em <https://www.gta.ufrj.br/ensino/eel878/redes1-2019-1/vf/mqtt/>, acessado em 15/05/2022.

OLIVEIRA, Euler. **Como usar com Arduino – Sensor Óptico Reflexivo TCRT5000**. Disponível em <https://blogmasterwalkershop.com.br/arduino/arduino-utilizando-o-sensor-reflexivo-tcrt5000>, acessado em 18/05/2022. 2019.

OLIVEIRA, Marcelo Monteiro Mateus de. Desenvolvimento de uma Plataforma para aquisição de dados e monitoramento do processo produtivo de uma indústria de estamparia mecânica, 2016. Trabalho de Conclusão do Curso de Pós Graduação em Ciência da Computação, Universidade Federal de Pernambuco, Pernambuco, Brasil, Disponível em <https://repositorio.ufpe.br/handle/123456789/25076>, acessado em 10/03/2022.

OLIVEIRA, phillipe mineiro ferro de. Utilização Do Método PDCA Para Melhoria da Eficiência De Equipamentos em Uma Indústria Do Setor Metal Mecânico, 2021. Universidade Federal Fluminense, Niterói, Brasil.

RED HAT. **O que é IIoT?** Disponível em <https://www.redhat.com/pt-br/topics/internet-of-things/what-is-iiot>, acessado em 29/06/2022. 2021.

SCHWAB, Klaus. A quarta revolução industrial. EDIPRO, 2016.

SOUSA, Fábio Cassio de; SOARES, Ricardo Donizeti; CAMPOS, Roberson José. IIoT utilizando o protocolo MQTT. **Centro Universitário Una, Pouso Alegre**, 2021. Disponível em <https://repositorio.animaeducacao.com.br/handle/ANIMA/18962>, acessado em 15/03/2022.

TOTVS. **Apontamento de produção: como otimizar o chão de fábrica da sua indústria?** Disponível em <https://www.totvs.com/blog/gestao-industrial/apontamento-de-producao/>, acessado em 25/05/2022. 2019.

APÊNDICE A - Fontes

Os códigos do módulo Controller, Monitor, Server e Interface podem ser acessados na plataforma GitHub. Acessível através do QR Code abaixo.



Figura 17: Códigos do Projeto

<https://github.com/cleytongomes/tcc-2022-telemetria-industrial>