

**FACULDADE DE TECNOLOGIA DE SÃO BERNARDO DO CAMPO
“ADIB MOISÉS DIB”**

Gabriel Lucas Peixoto
Gabriel Martins Sojo
Raphael Araujo Monteiro
William Lima Vieira

**ABASTECIMENTO POR MEIO DE AGV MONITORADO POR SISTEMA
SUPERVISÓRIO**

São Bernardo do Campo - SP
Junho/2023

**GABRIEL MARTINS SOJO
GABRIEL LUCAS PEIXOTO
RAPHAEL ARAUJO MONTEIRO
WILLIAM LIMA VIEIRA**

**ABASTECIMENTO POR MEIO DE AGV MONITORADO POR SISTEMA
SUPERVISÓRIO**

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de São Bernardo do Campo “Adib Moises Dib” como requisito parcial para a obtenção do título de Tecnólogo em Automação Industrial.

Orientador: Prof. Me. Marco Antônio Baptista de Sousa.

São Bernardo do Campo - SP
Junho/2023

**GABRIEL LUCAS PEIXOTO
GABRIEL MARTINS SOJO
RAPHAEL ARAUJO MONTEIRO
WILLIAM LIMA VIEIRA**

**ABASTECIMENTO POR MEIO DE AGV MONITORADO POR SISTEMA
SUPERVISÓRIO**

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de São Bernardo do Campo “Adib Moises Dib” como requisito parcial para a obtenção do título de Tecnólogo em Automação Industrial.

Trabalho de Conclusão de Curso apresentado e aprovado em ____/____/2023.

Banca Examinadora:

Prof. Me. Marco Antônio Baptista de Sousa, FATEC SBC - Orientador

Prof. Me. Marcos Vagner Zamboni, FATEC SBC - Avaliador

Prof. Me. Renan Rodrigues Mendes, FATEC SBC – Avaliador

“Quando realmente todos querem muito alguma coisa, ela acontece.”

ABEL FERREIRA

RESUMO

Este projeto propõe automatizar a reposição de peças em sistemas de abastecimento por meio de AGV (*Automated Guided Vehicle* – Veículo Autoguiado) monitorado por sistema supervisório, para que ocorra de forma eficiente e objetiva. Neste contexto, o presente trabalho visa apresentar como solução um protótipo utilizando uma placa ESP8266 em uma planta, onde será monitorado se há peças nas estações de trabalho e esses dados serão enviados a outra placa ESP32 de um AGV por meio da comunicação wi-fi, para que seja realizado o abastecimento quando necessário. Esse processo será monitorado por um sistema supervisório em tempo real que irá se comunicar com a placa ESP8266 da planta, gerando dados de consumo das estações, status do AGV, gráficos com base nos dados armazenados e através de análises será possível promover melhorias para o processo. Serão aplicados conceitos em redes industriais, programação e técnicas em melhorias contínuas de processos que vem sendo implantadas no mercado nos últimos anos. As indústrias passam muitas vezes por mudanças atreladas ao seu processo produtivo, porém alguns indicadores como custo, flexibilidade, confiabilidade, qualidade e rapidez na linha de produção são de extrema relevância, fazendo necessário que processos sejam automatizados.

Palavras-chave: Abastecimento, AGV, Sistema Supervisório, Automação Industrial, Microcontrolador.

ABSTRACT

This project proposes to automate the replacement of parts in supply systems through AGV (Automated Guided Vehicle) monitored by a supervisory system, so that it occurs efficiently and objectively. In this context, the present work aims to present a prototype solution using an ESP8266 board in a plant, where it will be monitored if there are parts in the workstations and this data will be sent to another ESP32 board of an AGV through wi-fi communication, for refueling when needed. This process will be monitored by a real-time supervisory system that will communicate with the plant's ESP8266 board, generating station consumption data, AGV status, graphics based on stored data and through analysis it will be possible to promote improvements to the process. Concepts will be applied in industrial networks, programming and techniques in continuous process improvements that have been implemented in the market in recent years. Industries often go through changes linked to their production process, but some indicators such as cost, flexibility, reliability, quality, and speed in the production line are extremely relevant, making it necessary for processes to be automated.

Keywords: Supply, AGV, Supervisory System, Industrial Automation, Microcontroller.

LISTA DE FIGURAS

Figura 1.1 – ESP8266 Modelo ESP-01	18
Figura 1.2 – Pinagem ESP8266	19
Figura 1.3 – NodeMCU 32-S	20
Figura 1.4 – Pinagem ESP32	21
Figura 1.5 – Tabela Especificações e Comparação ESP8266 e ESP32	22
Figura 1.6 – Módulo sensor obstáculo.....	23
Figura 1.7 – Motor Tower Pro SG-5010	25
Figura 1.8 – Processo do Sistema Supervisório de Produção	26
Figura 1.9 – Pirâmide de automação industrial	28
Figura 1.10 – Processo de Sensores e Atuadores	29
Figura 1.11 – Rede com o protocolo Modbus.....	31
Figura 2.1 – Fluxograma de funcionamento do processo	34
Figura 3.1 – Projeto finalizado.....	35
Figura 3.2 – Início da montagem do AGV	37
Figura 3.3 – Montagem do chassi, motores e rodas	39
Figura 3.4 – Montagem do AGV finalizada.....	40
Figura 3.5 – Esquema elétrico do AGV	41
Figura 3.6 – Montagem parcial da estação de trabalho	42
Figura 3.7 – Montagem da estação de trabalho finalizada	42
Figura 3.8 – Produção das peças.....	43
Figura 3.9 – Preferências da IDE do Arduino	44
Figura 3.10 – Adicionando URLs para Gerenciador de Placas da IDE	45
Figura 3.11 – Gerenciador de placas da IDE	45
Figura 3.12 – Instalando pacotes ESP na IDE	46
Figura 3.13 – Selecionando placa ESP para programação.....	46
Figura 3.14 – Tela Principal do sistema supervisório	48
Figura 3.15 – Tela Manual do sistema supervisório	49
Figura 3.16 – Comunicação do ESP8266 em seu código	50
Figura 3.17 – Comunicação do ESP32 em seu código	50
Figura 3.18 – Configurações de comunicação do ScadaBR	51

LISTA DE ABREVIATURAS E SIGLAS

AGV	Automated Guided Vehicle
RTU	Remote Terminal Unit
CPU	Central Processing Unit
WMS	Warehouse Management System
IoT	Internet of Things
IDE	Integrated Development Environment
URL	Uniform Resource Locator
SCADA	Supervisory Control and Data Acquisition

SUMÁRIO

INTRODUÇÃO	11
1 FUNDAMENTAÇÃO TEORICA	13
1.1 A linha de produção industrial.....	13
1.2 O veículo autoguiado (AGV)	15
1.2.1 Placa ESP e suas funcionalidades.....	17
1.2.2 Sensor obstáculo.....	23
1.2.3 Servo Motor Tower Pro SG-5010	24
1.3 O Sistema supervisório	26
1.3.1 Sensores e atuadores	28
1.3.2 Estação remota	29
1.3.3 Rede de comunicação.....	30
1.3.4 Estações de monitoramento central	31
2 METODOLOGIA	32
2.1 Metodologia aplicada.....	32
2.2 O tema problema, justificativa e diagrama de funcionamento	33
3 DESENVOLVIMENTO DO PROJETO	35
3.1 Projeto finalizado	35
3.2 Montagem da estrutura mecânica e elétrica do protótipo.....	36
3.2.1 Montagem do AGV	37
3.2.2 Montagem da planta industrial.....	41
3.3 Programação dos microcontroladores ESP	43
3.4 Desenvolvimento do sistema supervisório	47
3.5 Integração do sistema supervisório com as placas ESP	49
3.6 Problemas e soluções	51

CONSIDERAÇÕES FINAIS	54
REFERÊNCIAS.....	56
APÊNDICES	58

INTRODUÇÃO

Nos dias atuais, os processos industriais utilizam cada vez mais sistemas de controle, interligando suas atividades com softwares e dispositivos que realizam as tarefas, relacionando o operador com uma interface gráfica, de modo visual e amigável, permitindo-o monitorar e realizar comandos.

A realização deste projeto é de extrema importância, pois será utilizado na prática, uma boa parte dos conhecimentos adquiridos no decorrer do curso de Automação, além de ser um projeto relevante para qualquer linha industrial que queira otimizar os seus métodos. O objetivo é projetar um sistema de abastecimento automatizado, que manterá a linha de produção abastecida através de um AGV (*Automated Guided Vehicle* – Veículo Autoguiado) de acordo com a necessidade de cada estação de trabalho, buscando peças em um estoque, aumentando assim a eficiência e evitando gargalos no processo de produção. Será desenvolvido para atender à necessidade cada vez maior de manter esses processos em um fluxo contínuo de produção.

A proposta é automatizar o processo de abastecimento da linha de produção com o auxílio de um AGV, que será controlado pela placa microprocessada ESP32. A planta será monitorada por uma placa microprocessada ESP8266 que receberá os sinais de peça em cada estação de trabalho através de sensores infravermelhos, enviando estas informações para o AGV através da comunicação Wi-Fi. Desta forma, o AGV irá se deslocar seguindo uma linha demarcada no piso, coletar a peça no estoque utilizando uma garra acionada por um servo motor e levará até a estação de trabalho solicitante. Todo este processo será monitorado em tempo real por um sistema supervisor que irá se comunicar através de Wi-Fi com a placa ESP8266 da planta, além de poder enviar comandos remotamente em um modo de operação manual ao AGV, identificar e antecipar possíveis falhas, armazenar dados e gerar gráficos, dando mais eficiência e qualidade ao processo.

O projeto tem a seguinte estrutura:

Capítulo 1 – Fundamentação teórica: em que será apresentada a teoria, objetivos e conceito que irão atender o projeto e suas referências para desenvolvimento, sobre o Abastecimento da linha de produção, AGV e o sistema supervisorio.

Capítulo 2 – Metodologia: em que se mostra os métodos e ferramentas utilizados no desenvolvimento do trabalho e construção do projeto.

Capítulo 3 – Desenvolvimento do projeto: em que se descreve a construção, seus circuitos e suas características físicas.

Considerações finais: são descritos os objetivos e as justificativas do projeto, trazendo as relações entre os fatos verificados, teorias, objetivos alcançados, pontos a serem destacados e possíveis sugestões para melhorias futuras.

1 FUNDAMENTAÇÃO TEORICA

Neste capítulo são abordados os artigos técnicos, as referências e os estudos relacionados ao tema do projeto, os quais colaboram com o desenvolvimento do projeto referente ao Abastecimento eficiente da linha de produção através do AGV.

1.1 A linha de produção industrial

De acordo com Ford (1913) a introdução do formato de linha na produção em massa é considerada uma grande inovação tecnológica, estando presente em diferentes setores, sejam eles industriais, em pequenas, médias e grandes empresas, ou seja, no mundo inteiro.

Conforme Kalatec (2020) destaca-se que a eficiência da linha de produção está no fato de se produzir em etapas, de forma padronizada e com tempo determinado, reduzindo o tempo e custos, aumentando consideravelmente a produção industrial.

A produção em massa resultou em inúmeros benefícios para as indústrias, tais como:

- ✓ produção rápida em larga escala;
- ✓ a diminuição da probabilidade de falhas através do auxílio de máquinas cada vez mais modernas;
- ✓ realização de tarefas específicas divididas, trazendo assim agilidade à produção;
- ✓ padronização dos processos, o que significa um produto com mais qualidade;
- ✓ organização do processo industrial, com cada etapa da linha de produção sendo sempre pré-definida;
- ✓ baixa necessidade de reconfiguração;
- ✓ barateamento dos produtos, que se tornam mais acessíveis ao público consumidor.

A linha de produção é composta por esteiras rolantes, máquinas, robôs e mão de obra humana, que opera, abastece e monitora todas as ações e cada operário ou equipe é responsável por uma etapa, sendo abastecida com a matéria-prima, que segue por diversos processos de transformação e passo a passo, até o seu acabamento, armazenamento e estocagem.

Todos os processos a serem implementados na linha de produção devem ser pensados para obtenção do produto final, além de separá-los em etapas, como: envase, acabamento, embalagem e entre outros, tendo em mente o quanto se quer produzir e em que prazo se deseja entregar, para que isso direcione a tomadas de decisões assertivas. Além disso, otimizar a produção é essencial, investindo em maquinário, equipamentos específicos para cada setor, ferramentas necessárias para manutenções, mão de obra qualificada e treinamentos para os colaboradores.

Deve-se instalar sistemas de automação que são úteis justamente para controlar e monitorar cada processo, facilitando a identificação de falhas e aumentando sua eficiência. Em primeiro lugar deve-se definir a ordem de cada etapa do processo, o que cada um fará e criar um layout de sua linha de produção, ou seja, o “desenho” de onde cada operador e máquina irá ficar. Esse layout deve ser pensado de acordo com a demanda e a variedade de mercadorias.

Na linha de produção existem muitos tipos de layout, como: linear, celular, funcional, posicional, e entre outros, lembrando que deve ser escolhido de acordo com o seu modelo de negócio. Isso tem grande influência no resultado, deixando o trabalho mais fluido e dinâmico.

Com a linha de produção em andamento, é importante acompanhar diariamente o seu funcionamento, levantando dados de seu desempenho, monitorando cada ação a ser desenvolvida ali e identificando gargalos, pontos vulneráveis etc.

Ao longo da era industrial, foram surgindo diferentes formas para se organizar o trabalho realizado. Foi criado na primeira metade do século XX por Henry Ford, a linha de produção fordista, sendo que o fordismo pregava a superprodução, então a

ideia em si era produzir em massa e vender muito, por isso uma das estratégias desse modelo de produção era contratar uma numerosa mão de obra com baixa qualificação e para acolher com essa quantidade numerosa de colaboradores e todas as etapas da linha de produção, foram criados grandes complexos industriais e as linhas foram implementadas para facilitar e organizar muito mais o trabalho.

Havia grandes estoques, o que garantia entrega imediata e fidelização do cliente. Porém, o fordismo demonstrou falhas, como a baixa variedade de produtos ofertada. Então decaiu, não suportando crises de mercado que se apresentaram na época. Houve um acúmulo de mercadorias estocadas e queda nas vendas para a Europa, além de outros fatores. A linha de produção fordista foi adotada até meados de 1970.

Na segunda metade do século XX, Eiji Toyoda desenvolveu a linha de produção Toyotista com seu método de produção enxuta e flexível, sendo que nesse modelo a mão de obra é reduzida e muito mais qualificada, as indústrias são menores e estando espalhadas pelo mundo a fabricação de mercadorias acompanha o consumo, reduzindo assim os custos de produção e elevando os lucros.

Esse tipo de linha de produção Toyotista usa equipamentos automatizados para otimizar o processo industrial. A tecnologia é empregada para acelerar operações e aumentar a qualidade com a padronização das funções, além disso, problemas podem ser diagnosticados em tempo real.

Equipamentos automatizados são capazes de realizar tarefas repetitivas com precisão. Esse sistema de produção gera economia e é muito utilizado em empresas de pequeno a grande porte.

1.2 O veículo autoguiado (AGV)

Conforme Delage (2022), o AGV (*Automated Guided Vehicle* – Veículo Autoguiado) existe desde a década de 1950, quando foi lançado pela empresa Barrett

um rebocador que seguia um fio no piso, e a partir dessa primeira versão, foi onde surgiu um novo tipo de AGV, o qual seguia marcadores de UV invisíveis no chão, e por dali em diante, foram lançados novos modelos e avançando de acordo com a tecnologia empregada. Para se ter uma ideia do quanto têm sido cada vez mais empregados nos armazéns de logística, estima-se que o mercado de AGV's atinja cerca de US \$ 3,6 bilhões até 2025, de acordo com o relatório da MarketsandMarkets, tem a justificativa para o uso crescente dos AGVs que está na busca pela segurança do local de trabalho e na necessidade de reduzir custos operacionais e aumentar a produtividade.

O objetivo principal desse equipamento é integrar processos industriais automatizados, sendo controlado por um sistema de software específico. Os AGV's têm sido a aposta de muitas empresas para aprimorar o fluxo de materiais e reduzir os custos logísticos. Constitui-se um robô portátil que segue ao longo de linhas demarcadas no piso, podendo também utilizar ondas de rádio, câmeras de visão, ímãs e lasers para orientar o seu deslocamento, com isso otimizando o fluxo e trazendo mais precisão ao processo. Portanto esse tipo de veículo transporta desde pequenas até grandes peças, podendo trabalhar em integração com um WMS (*Warehouse Management System* - Sistema de Gerenciamento de Armazéns), que transmite as ondas de movimentação para que o AGV possa fazer o armazenamento de materiais em um estoque e com sua utilização abrange desde linhas de produção e montagem até os centros de distribuição. Sendo assim, é capaz de aumentar a eficiência da operação em ganho de tempo, ou seja, maior produtividade.

Torna-se mais eficiente a sua implementação quando é feito um estudo detalhado, através de simulações e testes que possam ajudar a definir as melhores configurações, isso antes da sua utilização, de modo que o equipamento consiga atender aos objetivos e expectativas pré-estabelecidos.

Delage (2022) ainda enfatiza que através de alto retorno e competitividade, os AGV's trazem benefícios que superam muito os seus custos. Em um cenário de alta competitividade, escassez de mão de obra e uma pressão contínua para atender prazos de entrega cada vez menores, contar com AGV's na operação automatizada

pode ajudar bastante a superar tais desafios. Os veículos autônomos prometem um alto nível de precisão, maior velocidade e menos custos.

Sendo assim, o presente estudo visou desenvolver a implantação de um sistema de movimentação de materiais a partir de AGV's para abastecer uma linha de produção com eficiência.

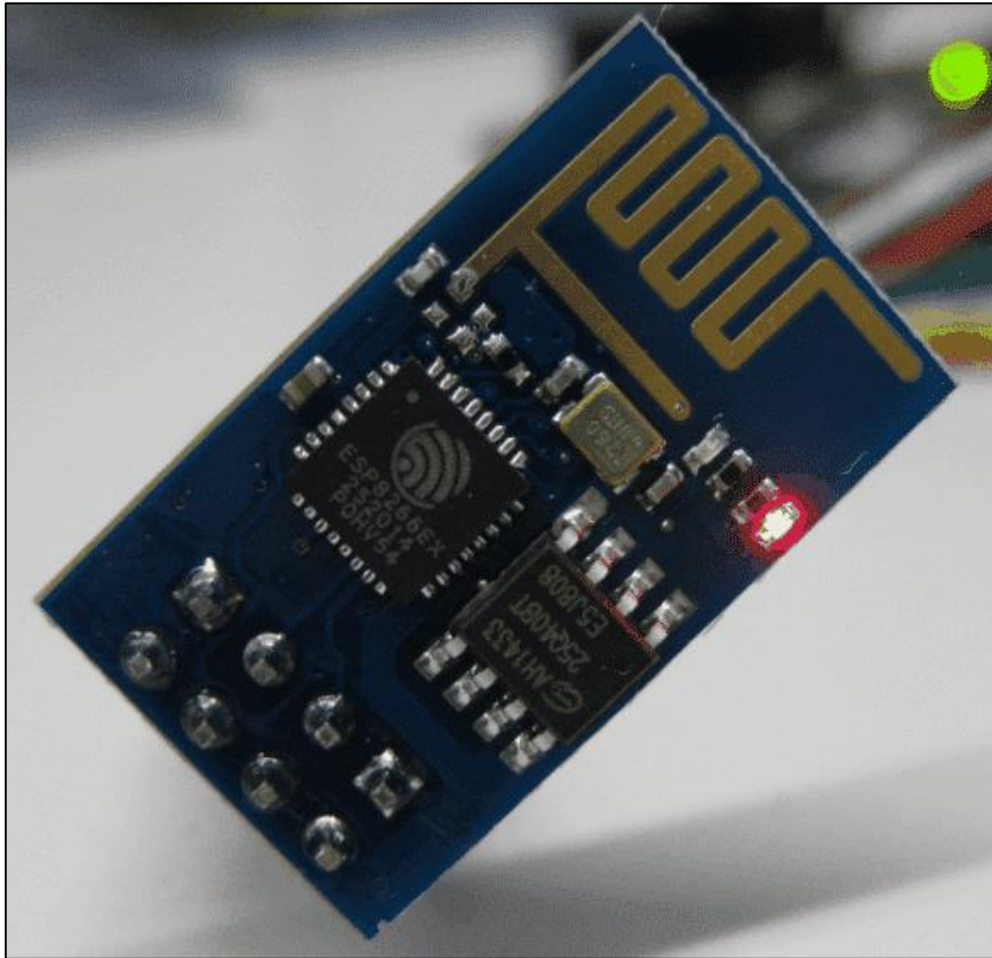
1.2.1 Placa ESP e suas funcionalidades

Conforme Vida de Silício (2018), as placas ESP's tem seu desenvolvimento com microcontroladores, pois além das funcionalidades encontradas em sistemas embarcados deste tipo, contam também com tudo que é necessário para conectar-se à internet via Wi-Fi.

De acordo com Blog Eletrogate (2022), o ESP8266 é um chip microprocessado desenvolvido pela empresa chinesa Espressif em 2015. Possui uma CPU de 32 Bits rodando a 80 MHz e suporta internet nos padrões 802.11 b/g/n, além de vários protocolos de segurança como WEP, WPA, WPA2, etc. O produto foi uma inovação tecnológica na época, pois já vinha com os circuitos de WIFI embutidos no próprio chip e com preço acessível. Para a utilização dos chips ESP8266 foram desenvolvidos vários módulos, cada um com suas características. Através deles você terá disponíveis interfaces I2C, SPI e PWM, além da interface serial. Além disso, a alimentação de todos eles é de 3.3 V, assim como o nível de sinal nos pinos. Esses módulos são ideais para projetos de IoT (*Internet of Things* – Internet das Coisas), pois possuem pouquíssimo consumo de energia em modo sleep. O uso do ESP8266 foi expandido devido à multifuncionalidades.

A seguir a Figura 1.1 mostra a placa ESP8266 Modelo ESP-01:

Figura 1.1 – ESP8266 Modelo ESP-01

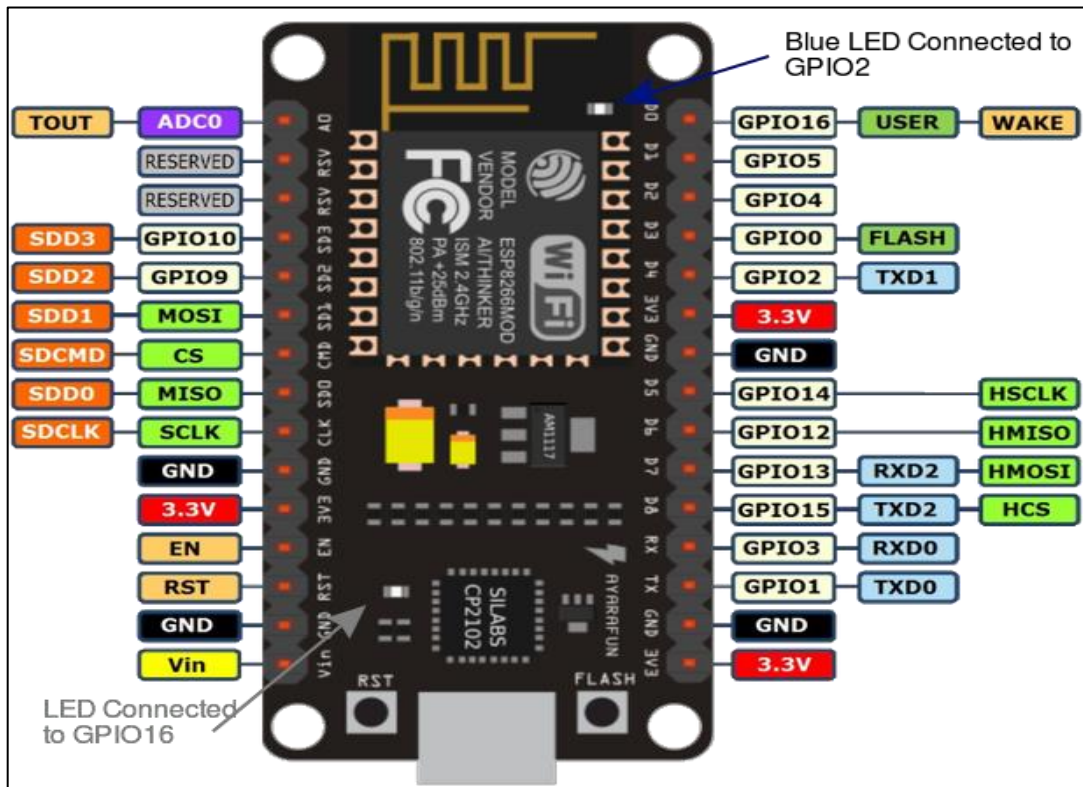


Fonte: <https://embarcados.com.br/modulo-esp8266/> [s.d.]

Este módulo tem como vantagens o tamanho muito reduzido e o custo baixo, podendo ser integrado facilmente a outras soluções através de uma comunicação serial UART.

A seguir a Figura 1.2 demonstra a disposição e função da pinagem do ESP8266.

Figura 1.2 – Pinagem ESP8266



Fonte: <https://curtocircuito.com.br/blog/Categoria%20IoT/conhecendo-esp32>, 2018

O ESP32 apresenta-se como um meio inovador no desenvolvimento de projetos automatizados. Esse componente demonstra maior versatilidade do que seu antecessor, pois além do clássico módulo de comunicação Wi-Fi, apresenta um sistema com processador Dual Core, Bluetooth híbrido e múltiplos sensores embutidos, tornando a construção de sistema como internet das coisas (IoT) muito mais simples e compacto.

Na placa é possível notar que o chip ESP32 com antena embutida, uma interface usb-serial e regulador de tensão 3.3V. A programação pode ser feita em LUA ou usando a IDE do Arduino através de um cabo micro-usb.

Essa placa conta com 4 MB de memória flash e permite criar variadas aplicações para projetos de IoT, acesso remoto, webserver e dataloggers tornando-se um grande aliado do maker IoT devido a facilidade na programação, hardware embutido. É um pequeno microcontrolador desenvolvido com a capacidade de proporcionar comunicação sem fio através do Wi-Fi e através do próprio sistema Bluetooth.

A seguir a Figura 1.3 mostra este novo microcontrolador, o ESP32:

Figura 1.3 – NodeMCU 32-S



Fonte: <https://portal.vidadesilicio.com.br/conhecendo-o-esp32/>, 2018

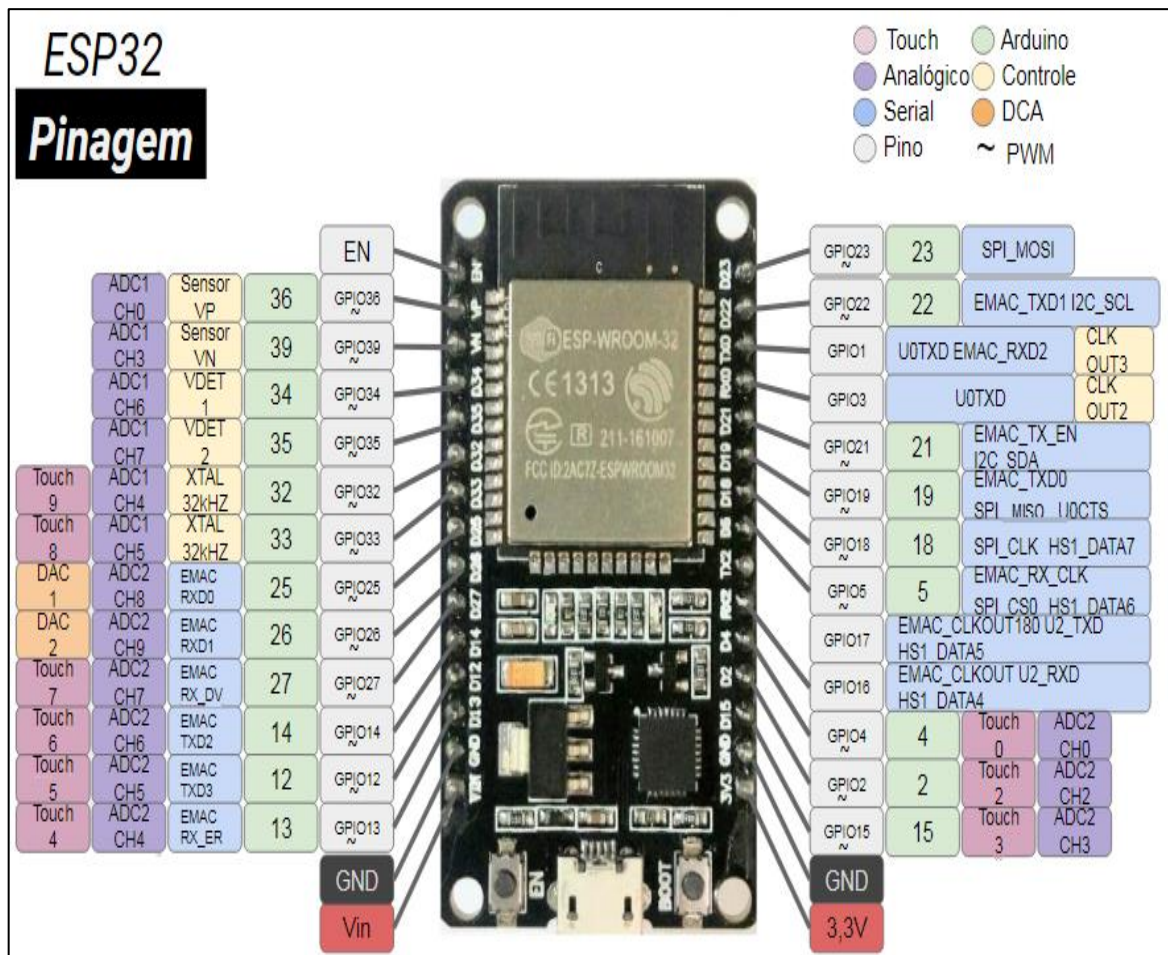
Características do ESP-WROOM32:

- ✓ Processador principal: LX6 32-bit Dual-core, operando 2-240 MHz;
- ✓ Processador secundário: ULP (Ultra Low Power coprocessor) 8MHz e consome 150uA;
- ✓ FLASH: 4MB;
- ✓ Processador principal: LX6 32-bit Dual-core, operando 2-240 MHz;
- ✓ Processador secundário: ULP (Ultra Low Power coprocessor) 8MHz e consome 150uA;
- ✓ FLASH: 4MB;
- ✓ RAM: 520kB;
- ✓ GPIO: 34, com 3.3V e 12mA;
- ✓ ADC: 18, com resolução de 12-bit;
- ✓ DAC: 2, com resolução 8-bit;
- ✓ WiFi: 2,4 GHz, 802.11 b/g/n;
- ✓ Bluetooth: Bluetooth Low Energy v4.2 (BLE);
- ✓ Acelerador via hardware para encriptações, hash e afins. (AES, RSA, SHA e ECC);

- ✓ True Random Number Generator (TRGN);
- ✓ 4 Timers de 64-bit;
- ✓ 4 Watchdogs;
- ✓ 10 Sensores de Touch Capacitivo;
- ✓ 1 Sensor de temperatura interno;
- ✓ 1 Sensor de efeito Hall.

Conforme Curto-Circuito (2018), menciona que ao analisar a placa ESP32, é visto que há 34 portas GPIO, 3 SPI, 2 I2S, 18 canais ADC, 3 UART, 10 pinos de leitura capacitiva e PWM. A seguir, será demonstrado a disposição dos pinos e suas funções:

Figura 1.4 – Pinagem ESP32



Fonte: <https://curtircircuito.com.br/blog/Categoria%20IoT/conhecendo-esp32>, 2018

Para melhor entendimento, a Figura 1.5 realiza uma comparação entre os sistemas operacionais de um ESP8266 e um ESP32.

Figura 1.5 – Tabela Especificações e Comparação ESP8266 e ESP32.

Especificações	ESP8266	ESP32
MCU	xtensa® single 32-bit 1106	xtensa® dual-core 32-bit 1x6 600 DMIPS
802.11 b/g/n Wi-Fi	HT20	HT40
Bluetooth	Não	Bluetooth 4.2 Le
Frequência	80 MHz	160 MHz
SRAM	160 Kbytes	512 Kbytes
Flash	SPI Flash, 16 Mbytes	SPI Flash, 16 Mbytes
GPIO	17	36
Hardware/ Software PWM	Não/ 8 canais	1 / 16 canais
SPI/ I2C/I2S/UART	2/1/2/2	4/2/2/2
ADC	10-bit	12-bit
CAN	Não	1
Interface Ethernet Mac	Não	1
Sensor Capacitivo	Não	Sim
Sensor de Temperatura	Não	Sim
Temperatura de Trabalho	-40°C a 125°C	-40°C a 125°C

Fonte: <https://curtocircuito.com.br/blog/Categoria%20IoT/conhecendo-esp32>, 2018

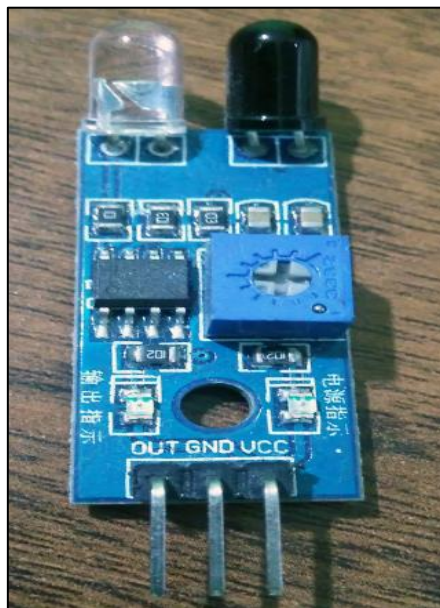
O ESP32 apresenta uma ótima evolução em relação ao ESP8266. Desta forma, percebe-se que esta evolução trouxe flexibilidade operacional e maior segurança para que seja utilizado em projetos de automação.

1.2.2 Sensor obstáculo

Conforme enfatizado em Mundo Projetado (2022), o sensor de obstáculo é um circuito composto por um emissor e um receptor IR, mais o CI comparador LM393, que facilita sua conexão com Arduino, PIC ou Raspberry Pi, visto que sua tensão é de 3,3-5V, e pode ser utilizado para automatizar certos sistemas. Por exemplo, fazer um controle de quantas pessoas saíram ou entraram em um ambiente.

O seu funcionamento é quando algum obstáculo é colocado em frente ao sensor, pois o sinal infravermelho é refletido para o receptor. Quando isso acontece, o pino de saída OUT é colocado em nível baixo (0), e o led verde do modulo é aceso, indicando que algum obstáculo foi detectado. A seguir a figura 1.6 mostra o sensor obstáculo:

Figura 1.6 – Módulo Sensor obstáculo



Fonte: <http://mundoprojetado.com.br/sensor-de-obstaculo-infravermelho>, 2022

O funcionamento por trás desse tipo de sensor é relativamente simples. Basicamente, um LED emissor de luz infravermelha fica aceso o tempo todo. E, paralelo a ele, existe um LED infravermelho receptor. Idealmente, a luz do emissor não chega diretamente ao receptor, já que os dois estão em paralelo. Na prática, a luz que chega lateralmente pode ser desconsiderada.

As aplicações com sensor de obstáculo, sendo o infravermelho ou não, possui algumas aplicações interessantes.

- ✓ Fazer controle de entrada e saída de pessoas em um ambiente;
- ✓ Verificar a chegada de algum objeto;
- ✓ Detectar a invasão de alguém dentro de uma casa;
- ✓ Verificar a ausência de um obstáculo.
- ✓ Alguns robôs limpadores possuem o sensor de obstáculo em sua base e detectam sempre a presença do piso à frente. Quando o piso não for detectado (encontrou uma escada ou buraco por exemplo), ele para de andar e evita quedas.

1.2.3 Servo Motor Tower Pro SG-5010

Conforme Kalatec (S/D), é destacado que o servo motor é uma das tecnologias mais utilizadas no processo produtivo de uma indústria e em muitas outras aplicações, e oferecem diversas vantagens a curto e longo prazo que se tornam quase indispensáveis, dependendo da sua finalidade, sendo que é muito utilizado em sistemas de coordenadas e braços robóticos, drones, automação industrial, máquinas diversas, aeromodelos de helicópteros e aviões, nos ramos aeroespacial, agrícola, defesa, médica e em muitas outras aplicações.

De acordo com Pro Auto Eletric (S/D) menciona que os servos motores são responsáveis por movimentar, rotacionar e alterar a velocidade de outros objetos com grande precisão.

O sistema servo, além de ser um dos mais sofisticados para quem deseja maximizar a sua produção por meio da automação.

São diversos modelos sendo que um deles é Tower Pro SG-5010 é um atuador que possui alta qualidade. Na Figura 1.7 mostra o servo motor Tower Pro SG-5010 e conforme mencionado:

Figura 1.7 – Motor Tower Pro SG-5010



Fonte: <https://www.filipeflop.com/produto/servo-tower-pro-sg-5010/>, [s.d.]

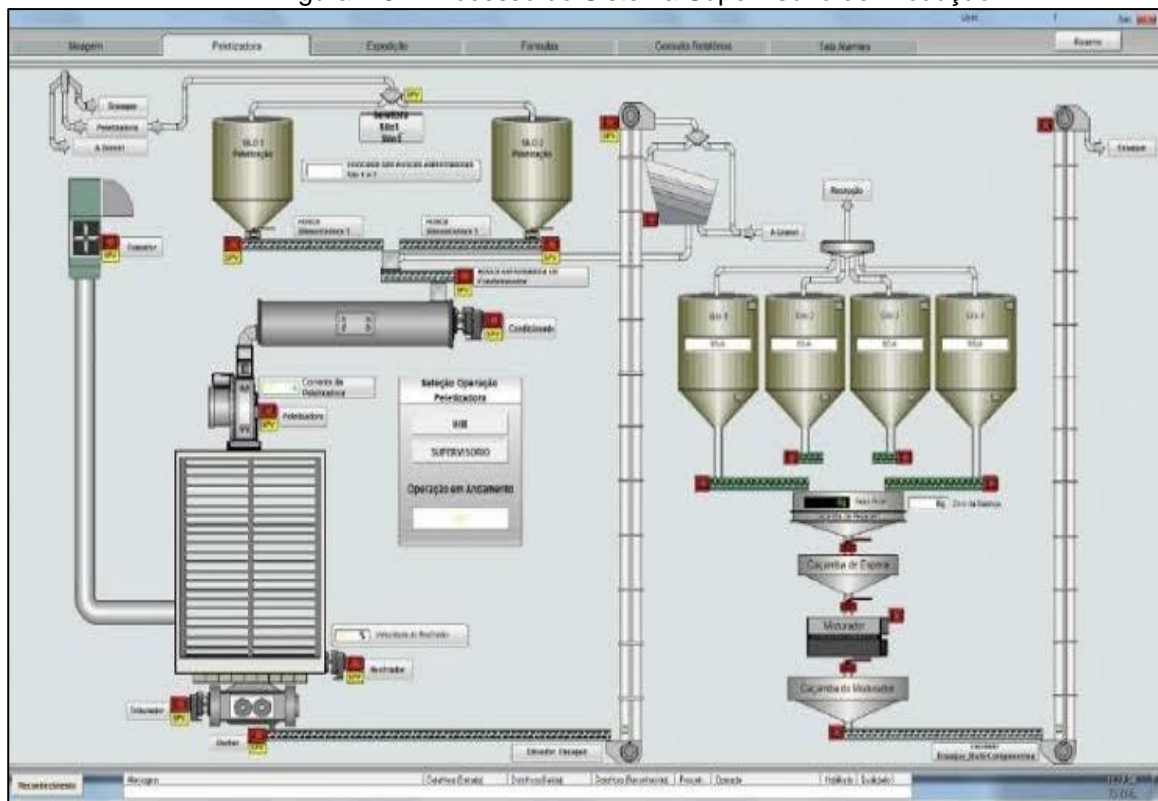
Características do Motor Tower Pro SG 5010:

- ✓ Voltagem de Operação: 4,8 – 6,0v;
- ✓ Modulação: Analógica;
- ✓ Velocidade: 0,17 seg/60° (4,8v) e 0,14 seg/60° (6,0v);
- ✓ Torque máximo: 8,0 kg.cm (4,8v) e 11,0 kg.cm (6,0v);
- ✓ Temperatura de Operação: -20°C ~ 60°C;
- ✓ Dimensões: 40,1×20,3×43,2 mm;
- ✓ Peso: 38g;
- ✓ Faixa de Rotação: 180°;
- ✓ Tamanho cabo: 30 cm;

1.3 O Sistema supervisório

De acordo com Logique (2018), nos meados das décadas de 70 e 80, com o avanço da eletrônica e surgimento dos microprocessadores, o computador se tornou uma peça-chave nos diversos setores industriais. Surgiram também os primeiros sistemas supervisórios, que se destina a capturar e armazenar em um banco de dados, informações sobre um processo de produção. As informações vêm de sensores que capturam dados específicos (conhecidos como variáveis de processo) da planta industrial. A Figura 1.8 mostra um sistema supervisório de produção:

Figura 1.8 – Processo do Sistema Supervisório de Produção



Fonte: <http://automacao.cotanet.com.br/sistemas-supervisórios/sistema-supervisorio-de-producao>, [s.d.]

O sistema supervisório de produção leva para sua corporação a tão sonhada alta da virtude e produtividade, com diminuição de seus custos.

Essa automação promove a execução de tarefas complexas com queda exponencial de erros, o controle pode ser feito de forma remota, sendo que o funcionário não precisa ir até a máquina.

Produz relatórios com os eventos e emergências, desta maneira a corporação cria um protocolo para ações futuras, tudo isso levando em consideração o aumento da virtude de vida dos funcionários de sua corporação.

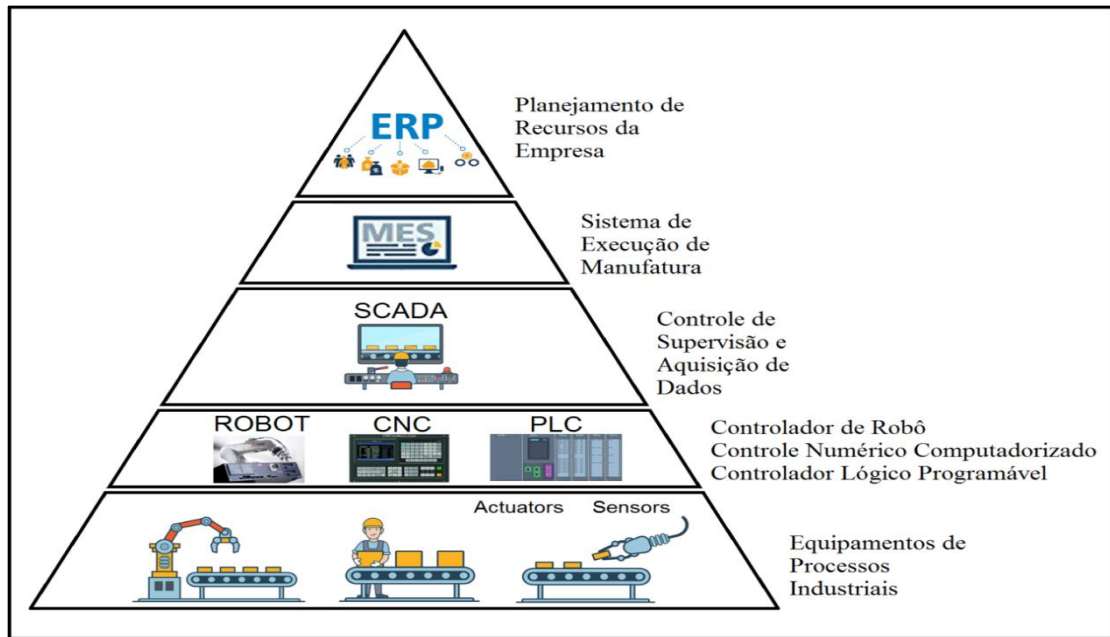
Conforme Logique (2018) os sistemas supervisórios são softwares que têm por finalidade monitorar e muito das vezes comandar dados importantes em processos industriais, prediais etc. Os supervisórios obtêm e organizam as variáveis relevantes adquiridas dos seus processos, apresentando as informações de forma clara em uma tela para o usuário responsável pelo monitoramento, facilitando a interação e controle, visando dar a devida atenção aos fatores obtidos no processo e poder agir de acordo com a necessidade, realizando todo esse controle em tempo real.

E para facilitar a visualização e diminuir os erros de operação, é possível a criação de telas, para aplicações e sistemas diferentes; usuários de acessos, para que seja possível também criar níveis de acesso para determinados cargos e funções; é possível inserir inúmeros objetos que tornem a interface mais intuitiva (botões, texto, lâmpadas, gráficos, tabelas, entre outros).

O bom funcionamento de um sistema supervisório é dado através dos dispositivos que são atrelados ao seu sistema, que são sensores, atuadores, PLC's, entre outros.

A seguir a Figura 1.9 mostra os elementos da pirâmide da automação industrial que interagem entre si:

Figura 1.9 – Pirâmide de automação industrial



Fonte: <https://www.redeindustria40.com.br>, 2022

A funcionalidade de um sistema supervisorio pode ser explicada através de suas divisões de seus principais componentes que serão mencionados nos capítulos a seguir suas descrições.

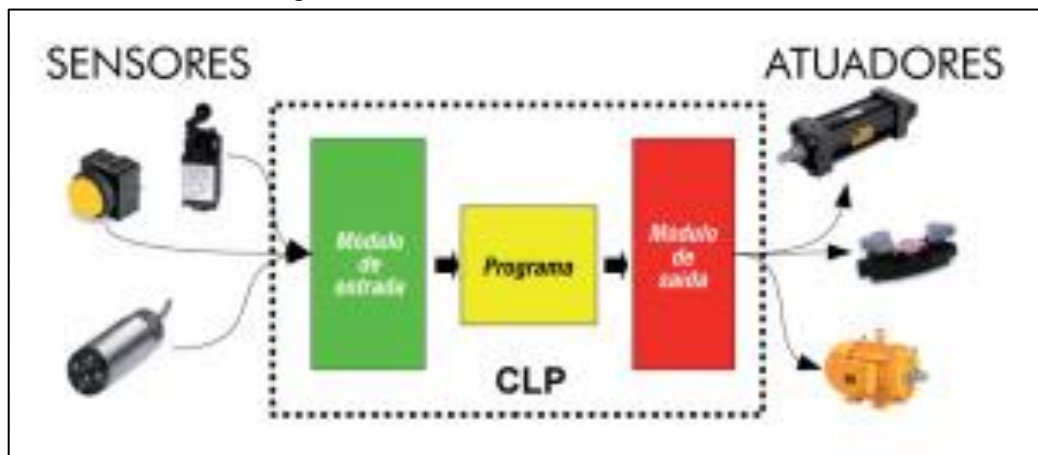
1.3.1 Sensores e atuadores

De acordo com Logique (2018) os dispositivos sensores e atuadores são ligados aos equipamentos supervisionados pelo sistema e são responsáveis por converter variáveis físicas em sinais elétricos.

Os sensores são responsáveis por converter parâmetros físicos em analógicos e digitais, que são enviados para um controlador. Estes parâmetros podem ser temperaturas, pressões, níveis e status de uma determinada máquina.

Os atuadores exercem por sua vez ações sobre o processo que são capazes de ligar, desligar, movimentar e modificar o status de determinado equipamento, conforme ilustra a Figura 1.10:

Figura 1.10 – Processo de Sensores e Atuadores



Fonte: <https://www.tecniar.com.br>, 2022

De acordo com Borges e Dores (2010), os sensores quando realizam medições de forma direta, transformando uma forma de energia em outra são chamados de transdutores. Já as medições que ocorrem de forma indireta e alteram suas propriedades, como a resistência, capacitância ou indutância por exemplo, sob a ação da grandeza de forma que essa alteração ocorre mais ou menos proporcional.

Um atuador assim como um sensor é um transdutor que transforma uma forma de energia em outra, porém este faz o caminho inverso, ao invés de transformar partes de uma grandeza física em um sinal elétrico, ele transforma um sinal elétrico em uma grandeza física, como movimento, magnetismo, calor entre outros.

De acordo com Brugnari e Maestrelli (2010), os atuadores atendem comandos que podem ser manuais ou automáticos, ou seja, qualquer elemento que realize um comando recebido de outro dispositivo, com base em uma entrada ou critério a ser seguido.

1.3.2 Estação remota

De acordo com Weblink (2020), enfatiza que onde o processo de aquisição dos dados se inicia. Para isso são utilizados os PLC's ou as RTU's – Remote Terminal Units (ou Unidade Terminal Remota).

Os PLC's e as RTU's são dispositivos eletrônicos que permitem a comunicação entre a estação central e os equipamentos monitorados. Através destes dispositivos é possível a obtenção dos dados informados pelos sensores, execução de seus cálculos e apresentação de saídas.

A RTU é um componente independente do PLC pois ele processa os próprios dados e envia as informações para o PLC através do protocolo de comunicação determinado no projeto.

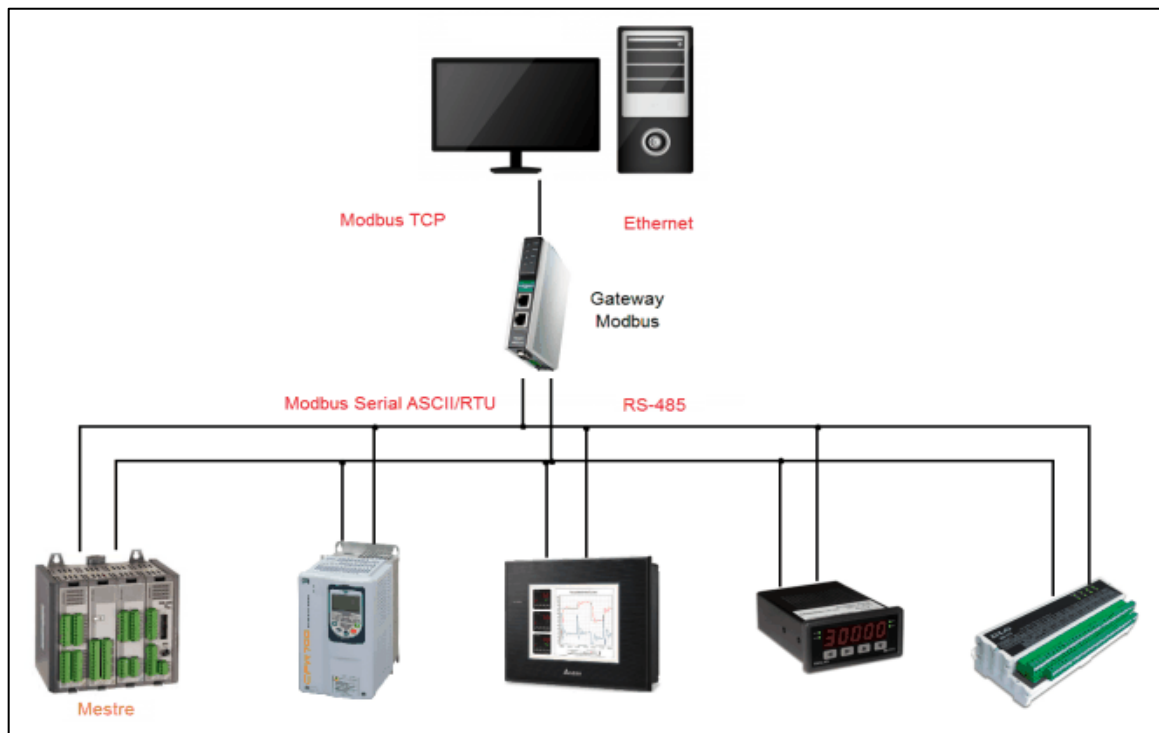
1.3.3 Rede de comunicação

Conforme Weblink (2020) explica-se que através destas redes de comunicação que todos os dispositivos são integrados e se comunicam entre si. Nestas redes existem diferentes protocolos de comunicação que são escolhidos de acordo com a necessidade do usuário.

O protocolo Modbus é um dos mais utilizados em automação industrial, graças à sua simplicidade e facilidade de implementação, podendo ser utilizado em diversos padrões de meio físico, como: RS-232; RS-485; Ethernet TCP/IP (MODBUS TCP).

Um exemplo de rede com o protocolo Modbus, com um gateway fazendo a conexão entre os dois tipos de Modbus, o serial em RS-485 e o TCP/IP em ethernet. Ainda existe no mercado a opção do gateway Modbus wireless. O mestre da rede, que nesse caso é um PLC envia e recebe dados dos escravos, que são posteriormente um inversor de frequência, uma IHM (Interface Homem Máquina), um controlador de temperatura e uma interface de I/O remota Modbus, conforme ilustra a Figura 1.11.

Figura 1.11 – Exemplo de Rede com o protocolo Modbus



Fonte: <https://embarcados.com.br>, [s.d.]

1.3.4 Estações de monitoramento central

Logique (2018), destaca que as estações de monitoramento central são responsáveis pelo monitoramento e supervisão do sistema, elas recebem as informações encaminhadas pelas estações remotas e agem conforme os eventos apresentados, essas estações podem ser apresentadas em um único computador como também distribuídas entre vários através de uma rede que permita o compartilhamento de dados.

Através do sistema supervisório com todos esses fatores integrados, que possibilitam a otimização, melhoria e qualidade do processo e conseqüentemente ganhos financeiros gerados pelo melhor aproveitamento da capacidade produtiva.

2 METODOLOGIA

Neste capítulo encontra-se a trajetória para o desenvolvimento, construção e realização deste projeto, o método de pesquisa científica usando o conceito, métodos para o desenvolvimento para a criação de veículo automatizada para abastecimento com eficiência na linha de produção em uma estação.

Dessa forma, a partir da união dos membros do grupo com os conhecimentos obtidos com curso de Automação Industrial para buscar a melhor solução para o desenvolvimento do projeto sistema automatizado para abastecimento eficiente na linha de produção.

2.1 Metodologia aplicada

De acordo com Conceito (2010) a metodologia tem a finalidade de percorrer para o desenvolvimento de pesquisa mediante um processo de investigação de método específico que enfocam que os procedimentos amplos do raciocínio e as técnicas que são procedimentos que operacionalizam os métodos mediante instrumentos adequados, conjunto de métodos que regem uma investigação científica.

De acordo com Lima (2022) a preparação da metodologia deve-se ser planejada trabalho científico supõe uma sequência de várias etapas que compreende: determinação do tema-problema e justificativa, levantamento da bibliografia referente ao tema, leitura e documentação dessa bibliografia após seleção, construção lógica do trabalho e redação do texto.

A construção da redação do TCC tem como base o Manual de Normalização de Projeto de Trabalho de Graduação da FATEC - SBC (2021) que se encontra embasado nas normas da ABNT.

2.2 O tema problema, justificativa e diagrama de funcionamento

O tema-problema surgiu através de pesquisas e conhecimento adquiridos em aulas e experiência profissional dos integrantes do grupo.

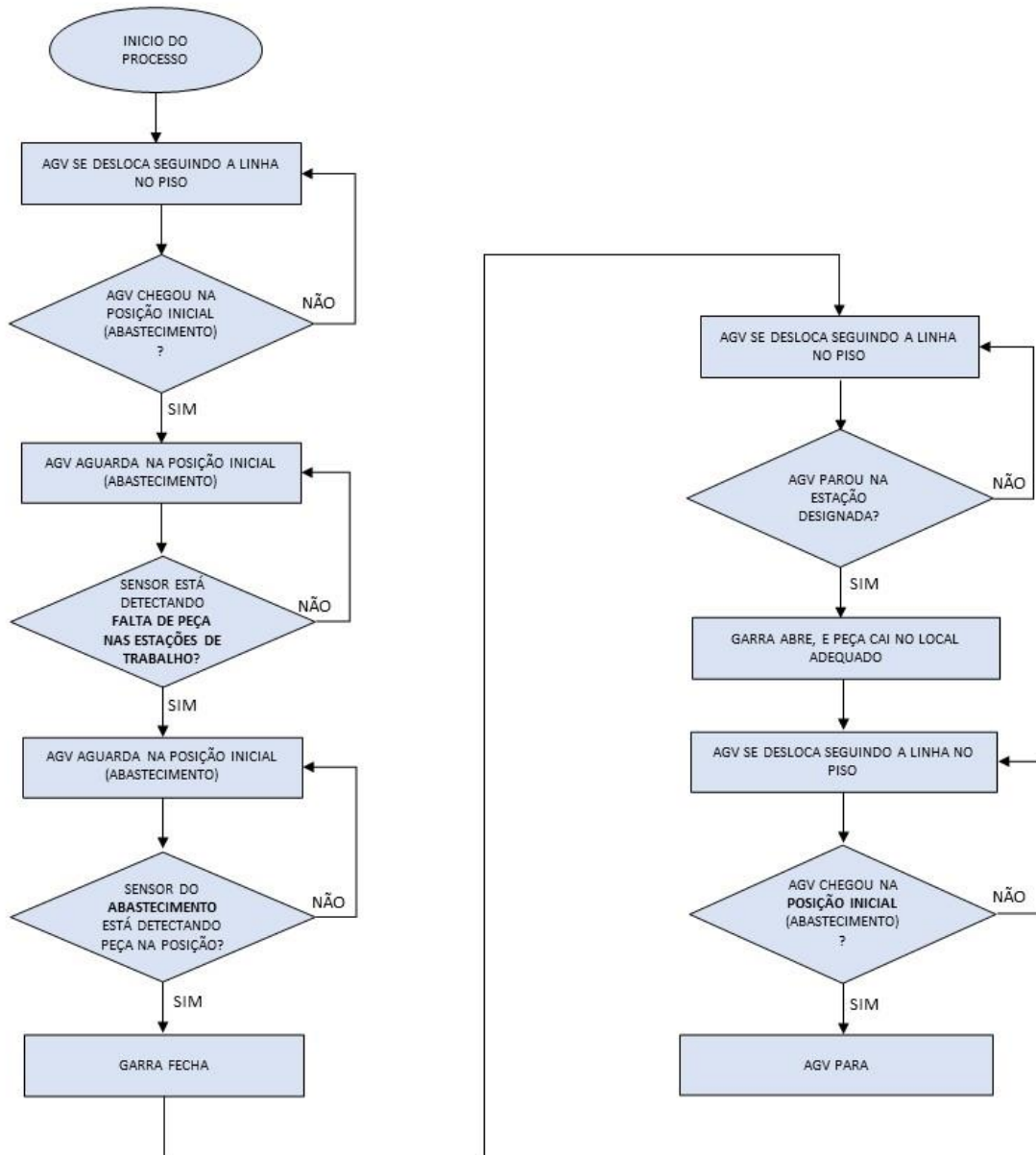
O projeto sobre o abastecimento com a utilização do AGV surgiu com a necessidade e visão de que no mundo industrial há gargalos imensos referente ao abastecimento de linhas de produção, sejam quais foram os nichos de mercado. Porém com um protótipo será testado de forma empírica como pode-se ter uma redução nos custos e tempo dentro de um processo de abastecimento de uma linha industrial.

Muitos projetos de AGV apresentados anteriormente não tinham todos os conceitos que o grupo via como necessários para que o processo fosse realmente eficiente, como uma interface gráfica para demonstrar os níveis do processo, uma comunicação sem necessidade de cabos para envio de comandos de rotas e prioridades ou mesmo um projeto realmente satisfatório com os requisitos estabelecidos. Inclusive, um dos integrantes do grupo já participou de outro projeto de AGV e não foi bem-sucedido como esperado, então para que o projeto alcançasse o maior êxito possível houve muito empenho da equipe.

Pensando também na questão acadêmica, foi levado em consideração a importância de agregar os conhecimentos adquiridos ao longo do curso de Tecnologia em Automação Industrial, então está sendo utilizado o maior número possível de princípios e ideias que foram passadas no decorrer dos anos, como a criação de rotas para que o processo aconteça da forma mais fluida possível, o desenvolvimento de um sistema de supervisão, programação, eletrônica, princípios de mecânica, entre outros conceitos.

Para facilitar o entendimento do projeto foi construído um fluxograma de funcionamento do processo, conforme ilustra a Figura 2.1.

Figura 2.1 – Fluxograma de funcionamento do processo



Fonte: Autoria própria, 2023

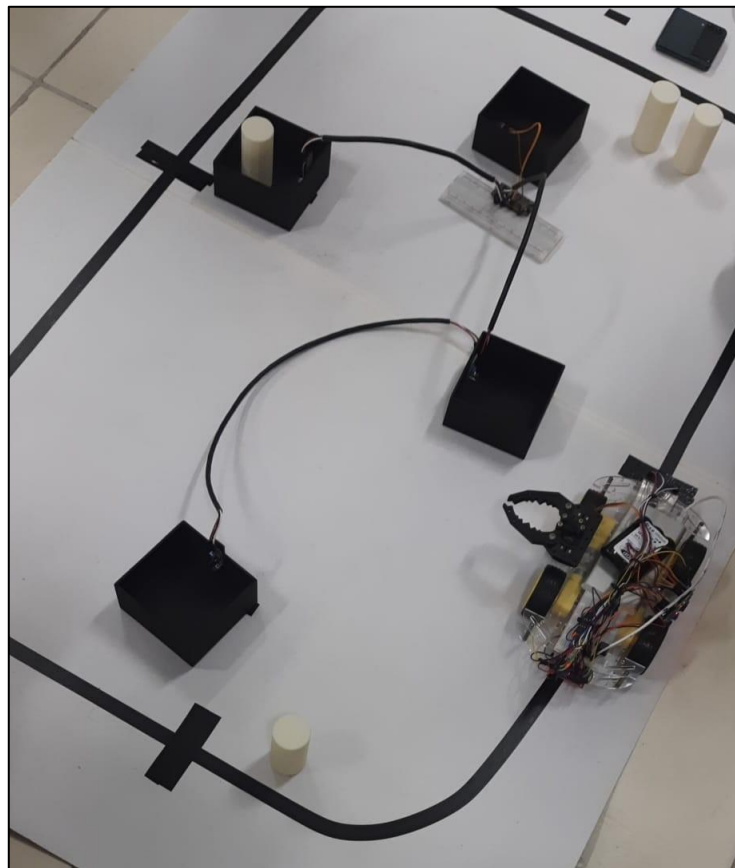
3 DESENVOLVIMENTO DO PROJETO

Neste capítulo encontra-se o passo a passo do desenvolvimento e construção lógica do projeto intitulado Abastecimento por meio de AGV Monitorado por Sistema Supervisório.

3.1 Projeto finalizado

O projeto trata-se do protótipo de um abastecimento por meio de AGV, tendo os processos controlados e monitorados um sistema supervisório. Este abastecimento pode ser aplicado a qualquer processo produtivo industrial. A seguir, a Figura 3.1 ilustrado o projeto finalizado.

Figura 3.1 – Projeto finalizado.



Fonte: Autoria própria, 2023

No AGV há dois sensores infravermelho que fazem a leitura de uma faixa no piso que o guiam pelo caminho designado, além de fazer a leitura das estações em que as peças serão depositadas. Outro sensor do mesmo tipo é montado na parte lateral para fazer a leitura apenas do estoque onde as peças serão coletadas. Na lateral oposta ao sensor, está a garra que coletará as peças, acionada por um servomotor.

Todas as estações possuem uma caixa em que as peças serão depositadas, identificando a presença delas através de um sensor infravermelho. Caso não haja peça em alguma das estações, o AGV deverá seguir até o estoque e aguardar que o operador coloque a peça na posição definida. Um sensor infravermelho identificará que a peça está na posição, só então o AGV deve fechar a garra e seguir para a estação que está solicitando a peça.

Todo este processo descrito anteriormente, será monitorado em tempo real por um sistema supervisório, gerando dados que poderão ser analisados e assim melhorar o processo cada vez mais, aplicando as ferramentas da qualidade. Será possível também enviar comandos em um modo manual enviando peças a qualquer estação desejada, desabilitar alguma estação caso esteja fora de operação, trabalhar com níveis de prioridade para o abastecimento das estações, evitando assim possíveis gargalos no sistema de produção.

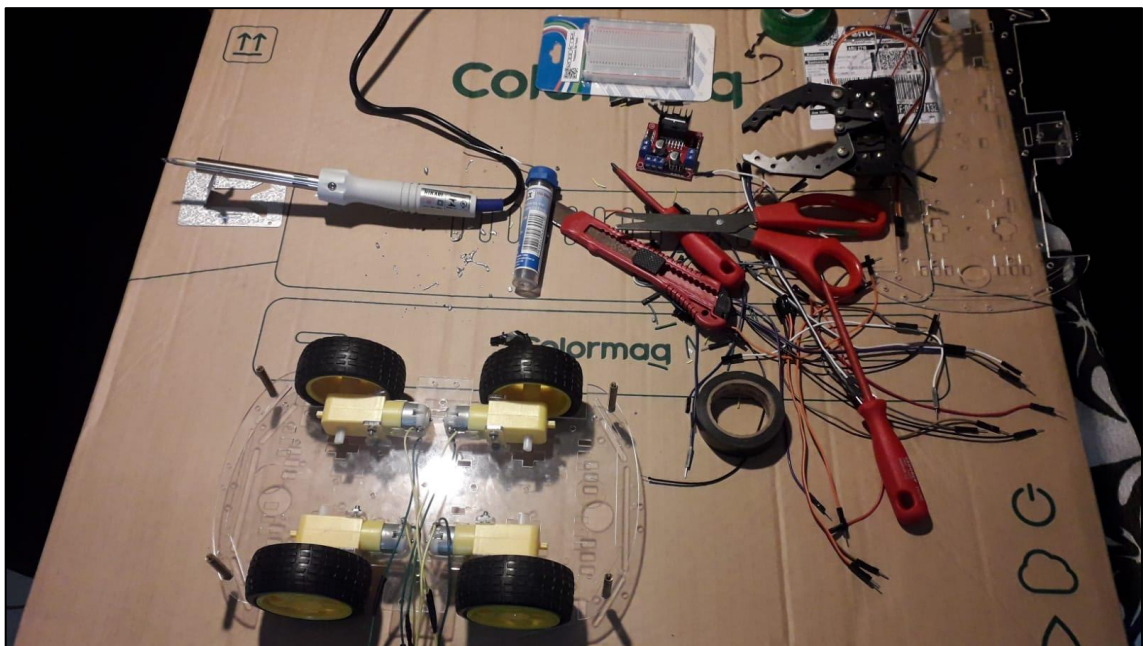
3.2 Montagem da estrutura mecânica e elétrica do protótipo

A montagem do protótipo foi dividido em duas partes: o AGV e a planta industrial, onde serão apresentados nos tópicos a seguir detalhadamente seu desenvolvimento.

3.2.1 Montagem do AGV

O AGV foi adquirido pronto, sendo necessário realizar somente a montagem de suas partes (chassi, rodas e motores das rodas) conforme ilustrado na Figura 3.2. Os outros componentes foram adquiridos posteriormente de acordo com a necessidade do projeto.

Figura 3.2 – Início da montagem do AGV.



Fonte: Autoria própria, 2023

A seguir, uma descrição de todos os componentes que compõe o AGV:

1 Kit chassi em acrílico com 4 rodas 4WD:

- ✓ Peso: ~478g;
- ✓ Diâmetro dos pneus: 7cm;
- ✓ Largura dos pneus: 2,5cm;
- ✓ Acrílico com espessura de 3mm;
- ✓ Dimensões totais (CxLxA): ~25x15x3cm;
- ✓ Espaçamento entre as placas acrílicas: 2,5cm;

4 Motores:

- ✓ Caixa de redução 48:1;
- ✓ Velocidade de 140 RPM a 4,5 VDC sem carga;
- ✓ Corrente com máxima carga de 250 mA;
- ✓ Tensão de alimentação recomendada 4,5 VDC.

1 Ponte H L298N:

- ✓ Tensão da lógica: 5 VDC;
- ✓ Tensão de operação: 7 a 30 VDC;
- ✓ Corrente de operação de até 2 A por canal;

1 Garra robótica Croc v2:

- ✓ Abertura de 12,1 cm.

1 Servo motor TowerPro SG-5010:

- ✓ Tensão de 4,8 a 6 VDC;
- ✓ Engrenagens em plástico;
- ✓ Torque (stall): 5,5 kgf.cm (4,8 VDC) / 6,5 kgf.cm (6 VDC);
- ✓ Velocidade de operação: 0,19 seg/60 graus (4,8 V sem carga);
- ✓ Velocidade de operação: 0,15 seg/60 graus (6,0 V sem carga).

3 Sensores infravermelho (obstáculo):

- ✓ Sinal infravermelho;
- ✓ Tensão de operação de 3,3 a 5 VDC;
- ✓ Range de detecção de 2 a 10 cm (ajustado via trimpot);

1 Placa ESP32:

- ✓ Processador Xtensa 32-Bit LX6 Dual Core;
- ✓ 34 x Portas programáveis GPIO's;
- ✓ Clock de 80 a 240 MHz (Ajustável);
- ✓ Memória ROM de 448 kB;
- ✓ Memória SRAM de 520 kB;
- ✓ Memória Flash Externa de 32-Bit de acesso e 4 Mb;
- ✓ Compatível com a IDE do Arduino;

- ✓ Tensão de nível lógico de 3,3 VDC;
- ✓ Tensão de Alimentação de 4,5 a 12 VDC (Pino Vin);
- ✓ Corrente de consumo de 80 mA (típica) e 500 mA (máxima);
- ✓ WiFi 802.11 b/g/n: 2,4 a 2,5 GHz;
- ✓ Segurança WiFi: WPA / WPA2 / WPA2-Enterprise / WPS;
- ✓ Bluetooth 4.2 BR / EDR e BLE (Bluetooth Low Energy).

1 Bateria Li-ion 18650:

- ✓ Recarregável;
- ✓ Capacidade de 2500 mAh;
- ✓ Tensão nominal de 12 VDC.

1 Power bank:

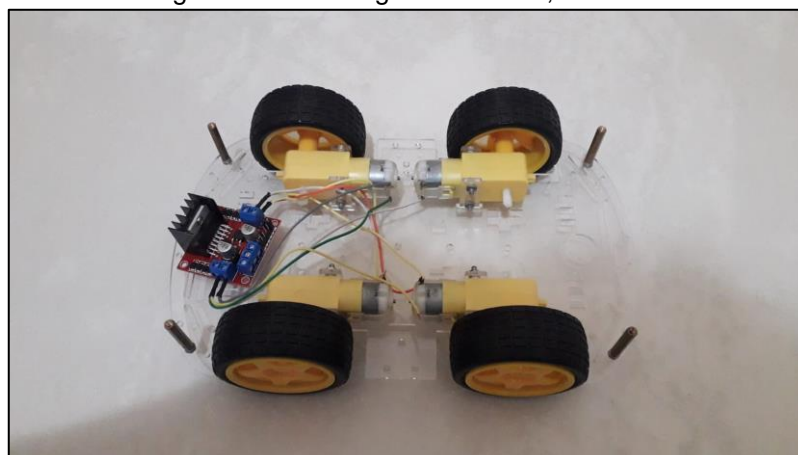
- ✓ Recarregável;
- ✓ Capacidade de 10000 mAh;
- ✓ Tensão nominal de 5 VDC.

1 Protoboard de 400 pontos.

Cabos de ligação.

Foi realizado a montagem do chassi, motores e rodas, unindo as partes através de parafusos. Os motores foram instalados entre os chassis inferior e superior, juntamente com a ponte H que irá alimentá-los, conforme ilustrado na Figura 3.3.

Figura 3.3 – Montagem do chassi, motores e rodas.



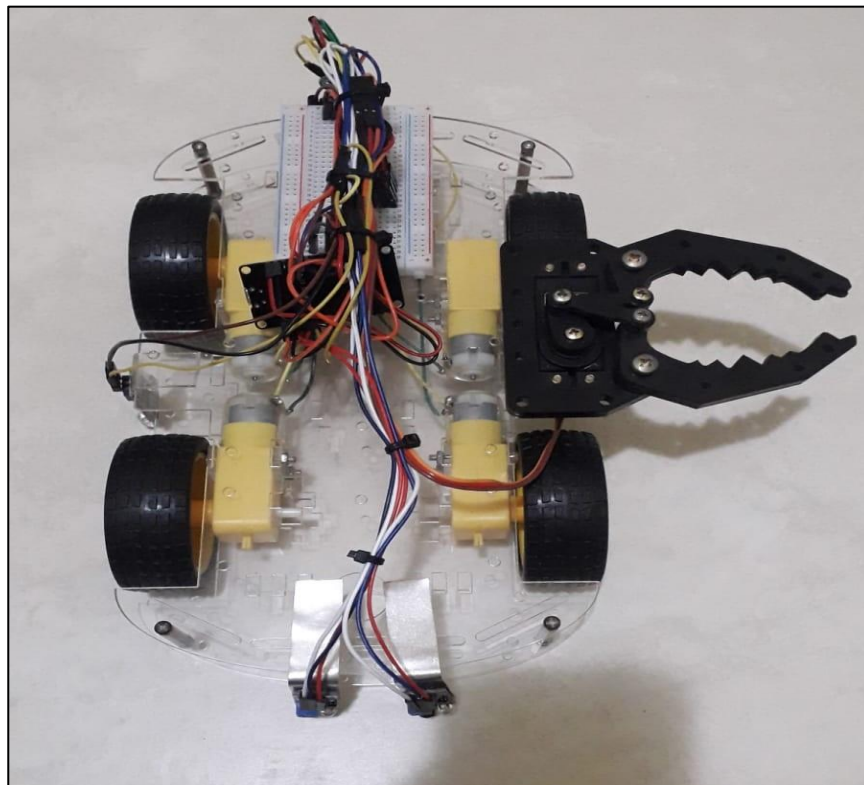
Fonte: Autoria própria, 2023

Os sensores do tipo infravermelho que serão utilizados para identificar a linha no piso e as estações de trabalho foram fixados na parte frontal em um suporte de alumínio, utilizando fita dupla face. Já o terceiro sensor que terá como função somente identificar o estoque onde as peças serão coletadas, foi fixado na lateral esquerda entre as rodas, tendo como referência a vista frontal do AGV.

A garra do tipo croc, que é acionada por um servomotor e foi instalada na lateral direita entre as rodas, na parte superior do chassi com fita dupla face, tendo como referência a vista frontal do AGV.

Após finalizar a montagem de todos os componentes, foi fixado uma protoboard de 400 pontos na parte superior do chassi para que fosse realizado as conexões elétricas entre o ESP32, sensores infravermelhos, ponte H, servo motor e as fontes de alimentação que está sendo utilizado um power bank 5 VDC para o servo motor e a placa ESP32 e uma bateria de Li-Ion 12 VDC para a ponte H, instalados na parte superior do chassi. A seguir, a Figura 3.4 ilustra o AGV com a montagem finalizada.

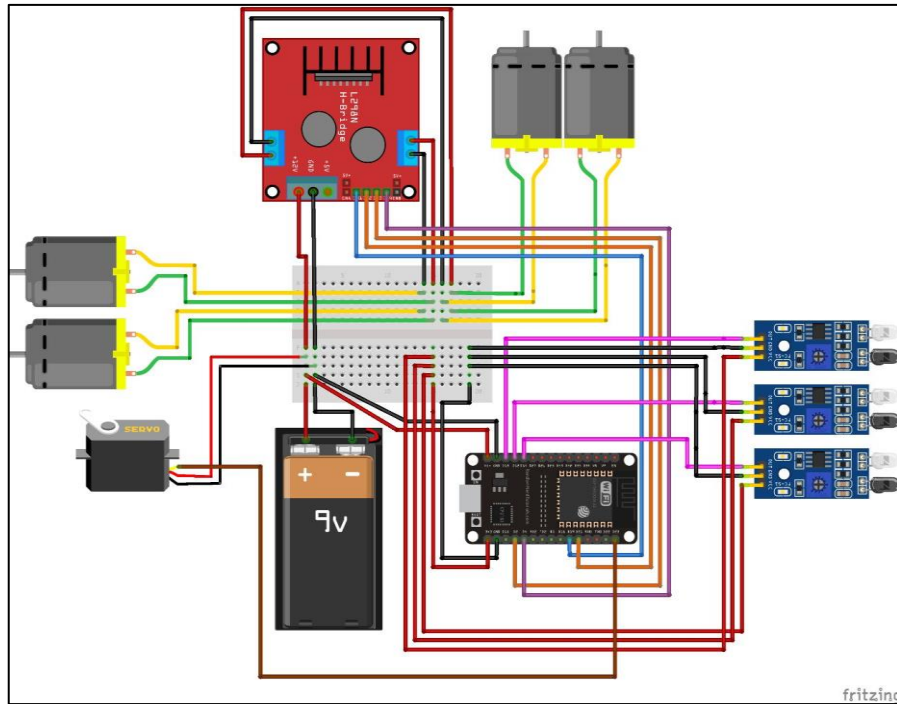
Figura 3.4 – Montagem do AGV finalizada.



Fonte: Autoria própria, 2023

Foi desenvolvido através da plataforma Fritzing, o esquema elétrico do AGV, contendo todos os componentes utilizados, conforme ilustrado na Figura 3.5.

Figura 3.5 – Esquema elétrico do AGV.



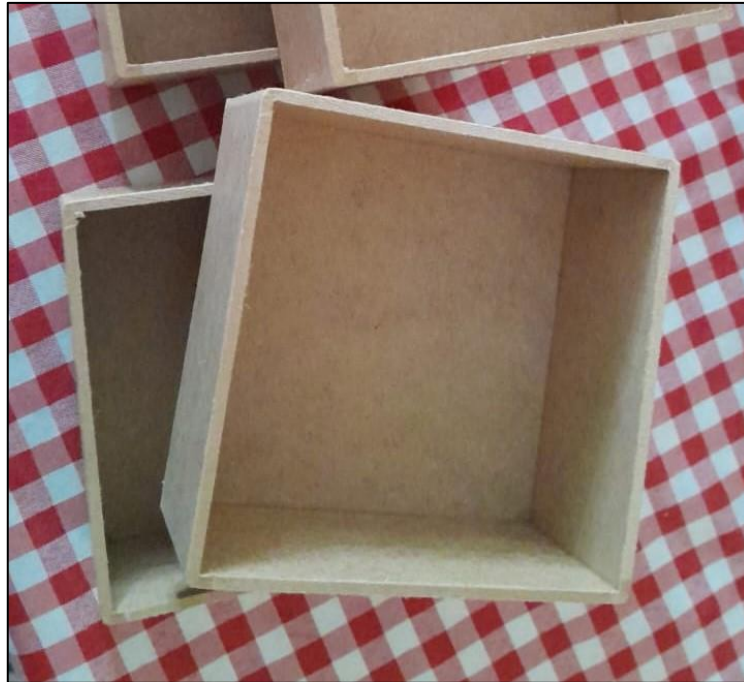
Fonte: Autoria própria, 2023

3.2.2 Montagem da planta industrial

A planta industrial é composta por 3 estações de trabalho, 1 estoque e o caminho demarcado no piso com fita pelo qual o AGV seguirá.

Para as estações de trabalho que receberão as peças, foi adquirido caixas de madeira do tipo MDF, são de formato quadrado e medem 10cmx10cmx5cm, conforme ilustrado na Figura 3.6.

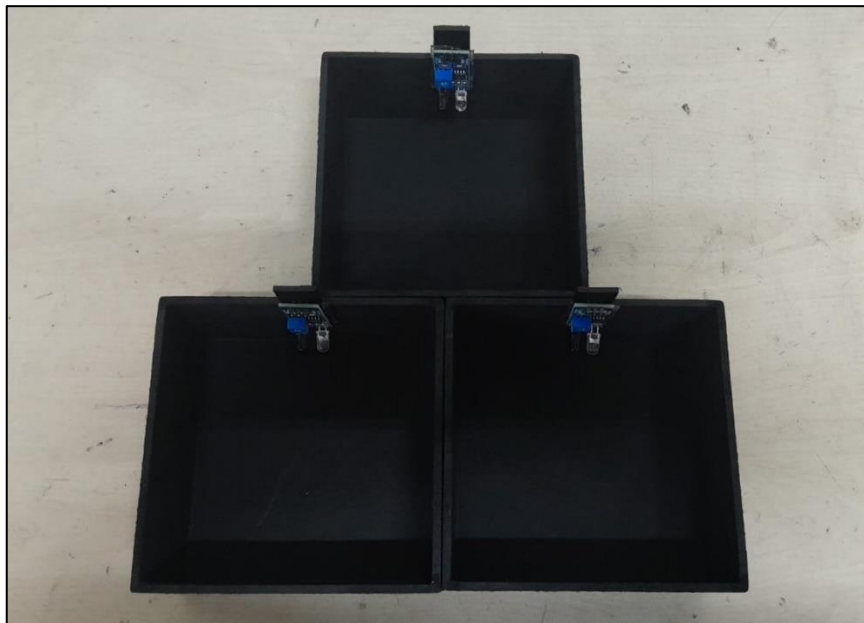
Figura 3.6 – Montagem parcial da estação de trabalho.



Fonte: Autoria própria, 2023

Após a montagem, foi realizado a pintura com spray preto, pois as peças utilizadas no processo serão na cor branca, para que assim, o sensor possa identificá-las com facilidade. Por fim, foi fixado internamente na parte lateral o sensor infravermelho que irá detectar a presença de peças, conforme ilustrado na Figura 3.7.

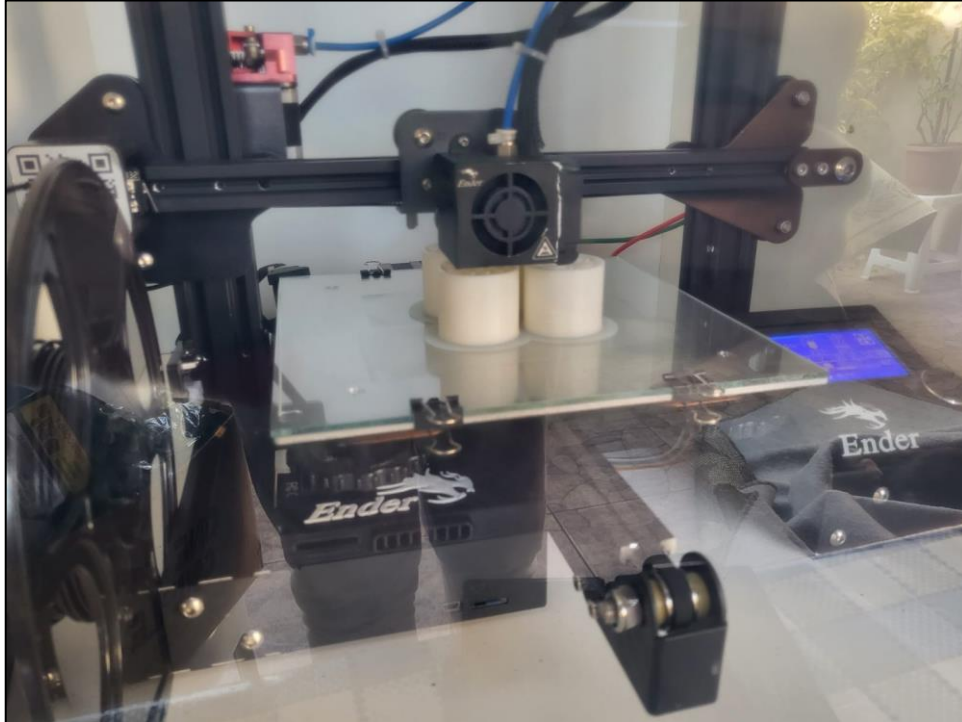
Figura 3.7 – Montagem da estação de trabalho finalizada.



Fonte: Autoria própria, 2023

As peças foram produzidas em plástico através de uma impressora 3D, são de formato cilíndrico e medem 4cm de diâmetro por 5cm de altura, conforme ilustrado na Figura 3.8.

Figura 3.8 – Produção das peças.



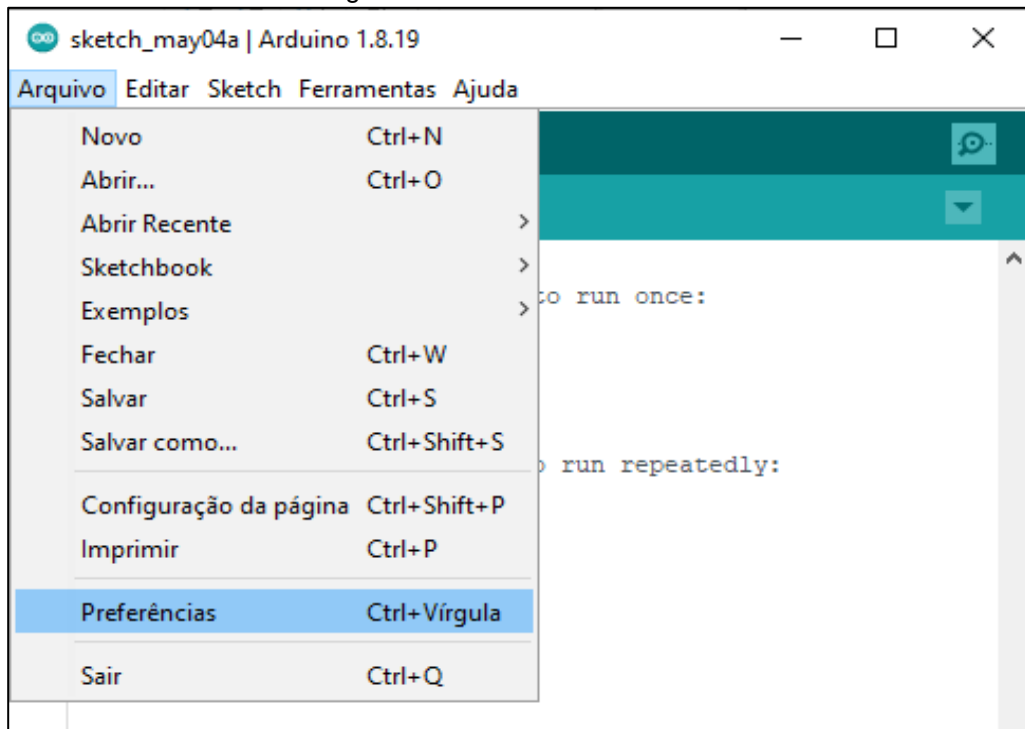
Fonte: Autoria própria, 2023

3.3 Programação dos microcontroladores ESP

Foi utilizado dois microcontroladores ESP no projeto, um ESP32 para controlar e monitorar o AGV e o outro ESP8266 para monitorar e enviar os status dos sensores da planta. A escolha do modelo de placa ESP foi determinada de acordo com a aplicação, já que o AGV tem uma quantidade maior de sensores e atuadores, a placa ESP32 é adequada para a sua aplicação por ter 36 pinos para entradas e saídas e a placa ESP8266 somente 17 pinos. A programação foi desenvolvida na linguagem C, que é a mais comum na programação de protótipos que utilizam microprocessadores, através da IDE (*Integrated Development Environment* – Ambiente de Desenvolvimento Integrado) do Arduino instalado em um computador onde o código desenvolvido foi verificado, compilado e carregado em cada placa ESP.

Para a programação do ESP através da IDE do Arduino foi necessário adicionar as placas referentes a cada microprocessador (ESP32 e ESP8266) da seguinte forma: com a IDE do Arduino aberta clicar em **Arquivo > Preferências >**, conforme ilustra a Figura 3.9.

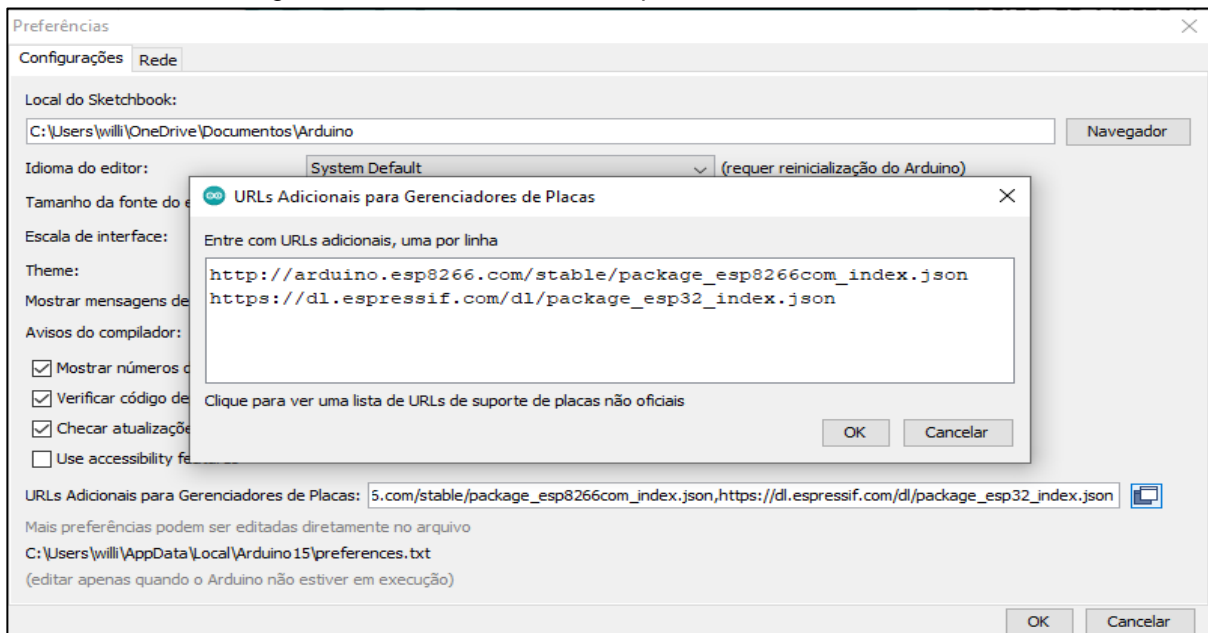
Figura 3.9 – Preferências da IDE do Arduino.



Fonte: Autoria própria, 2023

Adicionar ao campo **URLs Adicionais para Gerenciadores de Placas** os link's http://arduino.esp8266.com/stable/package_esp8266com_index.json para programação do ESP8266 e https://dl.espressif.com/dl/package_esp32_index.json para programação do ESP32, conforme ilustra a Figura 3.10 a seguir.

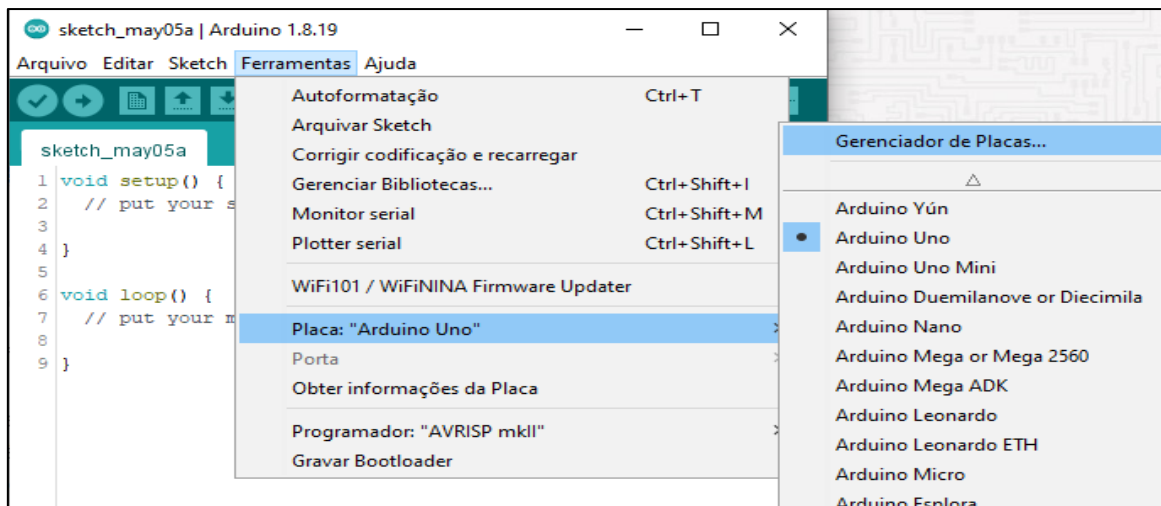
Figura 3.10 – Adicionando URLs para Gerenciador de Placas da IDE.



Fonte: Autoria própria, 2023

Desta forma, a IDE encontra as bibliotecas para que sejam realizadas as instalações em **Ferramentas > Placa > Gerenciador de Placas**, conforme ilustra a Figura 3.11.

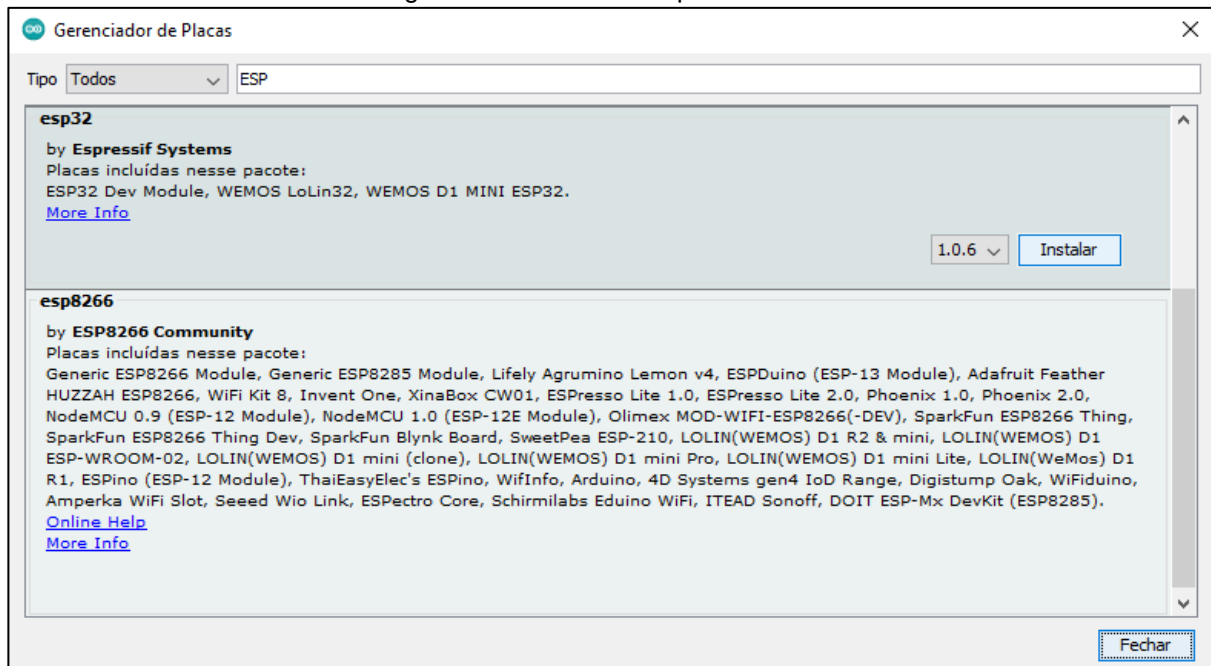
Figura 3.11 – Gerenciador de placas da IDE.



Fonte: Autoria própria, 2023

Em **Gerenciador de Placas** deve-se pesquisar pelo nome ESP para encontrar e instalar os pacotes, conforme ilustra a Figura 3.12 a seguir:

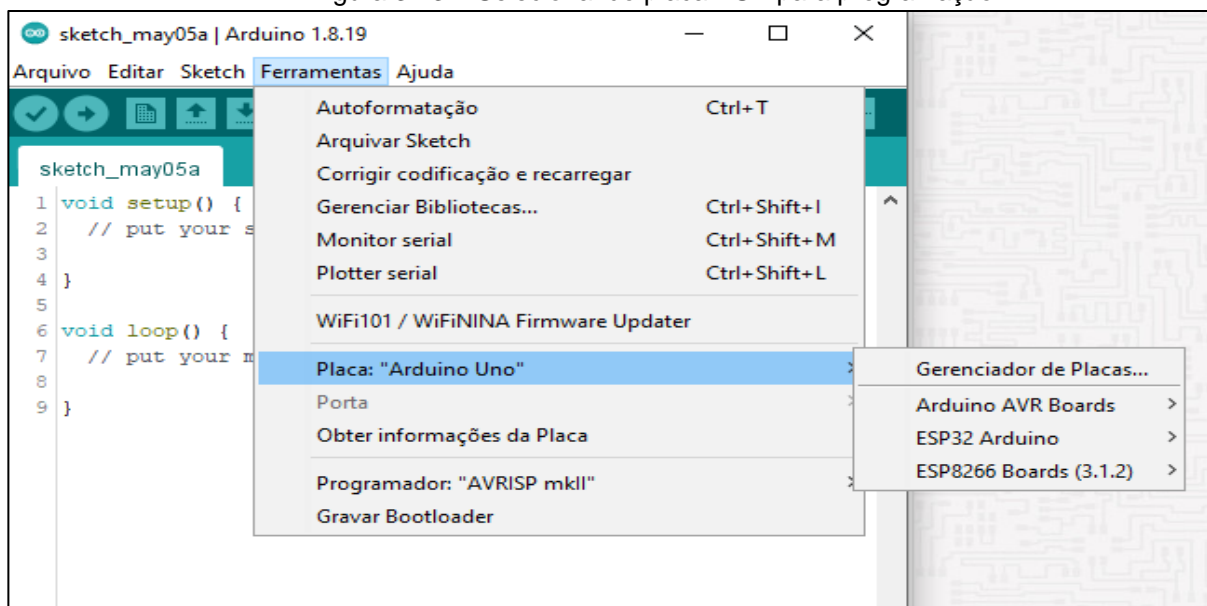
Figura 3.12 – Instalando pacotes ESP na IDE.



Fonte: Autoria própria, 2023

Com as bibliotecas instaladas a IDE estará configurada, bastando somente selecionar qual a placa irá programar em **Ferramentas > Placa >**, conforme ilustrado na Figura 3.13.

Figura 3.13 – Selecionando placa ESP para programação.



Fonte: Autoria própria, 2023

Além da instalação do pacote de cada placa ESP, também foi utilizado bibliotecas específicas no desenvolvimento do programa, de acordo com a necessidade da utilização de funções.

3.4 Desenvolvimento do sistema supervisório

O sistema supervisório foi desenvolvido através da plataforma ScadaBR, que se trata de um software nacional e gratuito, com o objetivo de monitorar o processo em tempo real, realizar a contagem de peças consumidas pelas estações, analisar os dados adquiridos para que seja possível implementar melhorias no processo, além de ser possível enviar comandos, alternando o modo de operação entre manual e automático de acordo com a necessidade.

O ScadaBR é um sistema supervisório que oferece as seguintes vantagens: monitoramento e controle remoto de processos industriais, integração com diferentes dispositivos e protocolos de comunicação, interface intuitiva e personalizável, capacidade de visualização em tempo real, histórico de dados para análise e tomada de decisões, suporte a alarmes e notificações, e código-fonte aberto para personalização e desenvolvimento.

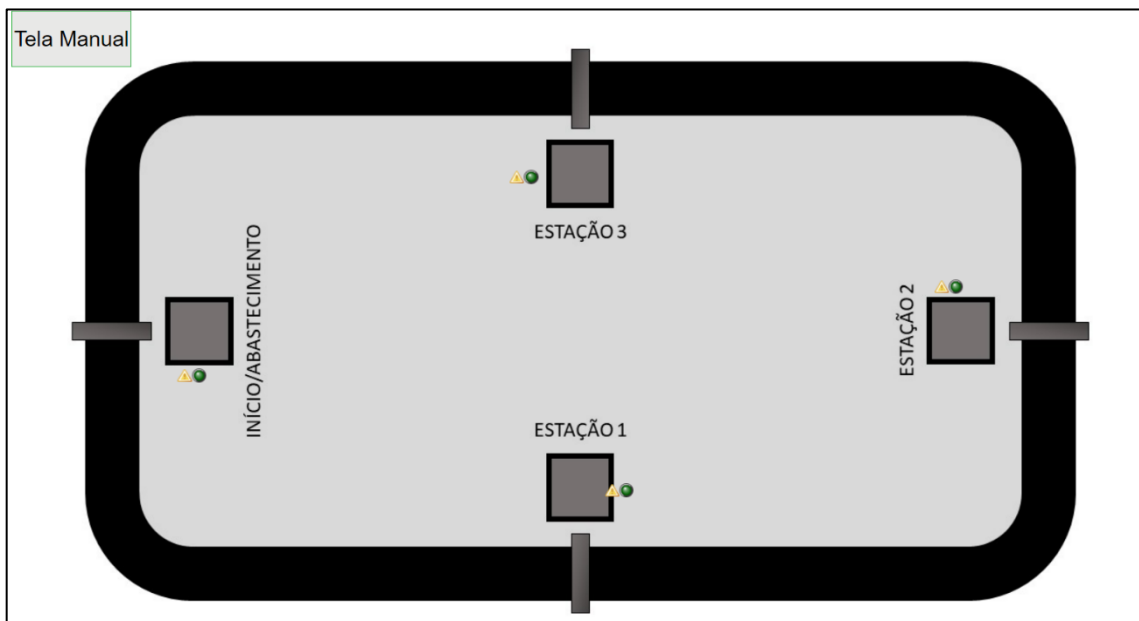
Principais Funcionalidades:

- ✓ Engine de Alarmes e Eventos parametrizáveis;
- ✓ Alarmes Sonoros;
- ✓ Sistema de Permissões por Usuários;
- ✓ Relatórios de Variáveis, Alarmes e Eventos;
- ✓ Engine de "scripts" para controle, automação e "batches", entre outros;
- ✓ Ambiente de Desenvolvimento para Aplicações SCADA (*Supervisory Control And Data Acquisition* – Sistema de Supervisão e Aquisição de Dados);
- ✓ Acesso em Browser (Desktop, mobile);
- ✓ API com exemplos em Java, Visual Basic, PHP e Python, entre outros;

- ✓ Aquisição de dados em mais de 20 protocolos como: Modbus TCP/IP e Serial, OPC, DNP3, IEC, Serial ASCII, HTTP;
- ✓ Variáveis calculadas (com funções matemáticas, estatísticas, e lógica em geral);
- ✓ Visualização de dados (variáveis ou "tags") em tempo real;
- ✓ Registro contínuo das variáveis em Banco de Dados (Historian/ Datalogger);
- ✓ Construção de telas gráficas (HMI ou Sinópticos);

Foram desenvolvidas duas telas com um botão que navega entre elas. A primeira é a de monitoramento principal, com o circuito disposto conforme a montagem física. Os sensores de monitoramento das peças são representados pelos sinaleiros verdes, que ficam acionados quando os sensores estão detectando peças nas posições, conforme ilustrado na Figura 3.14.

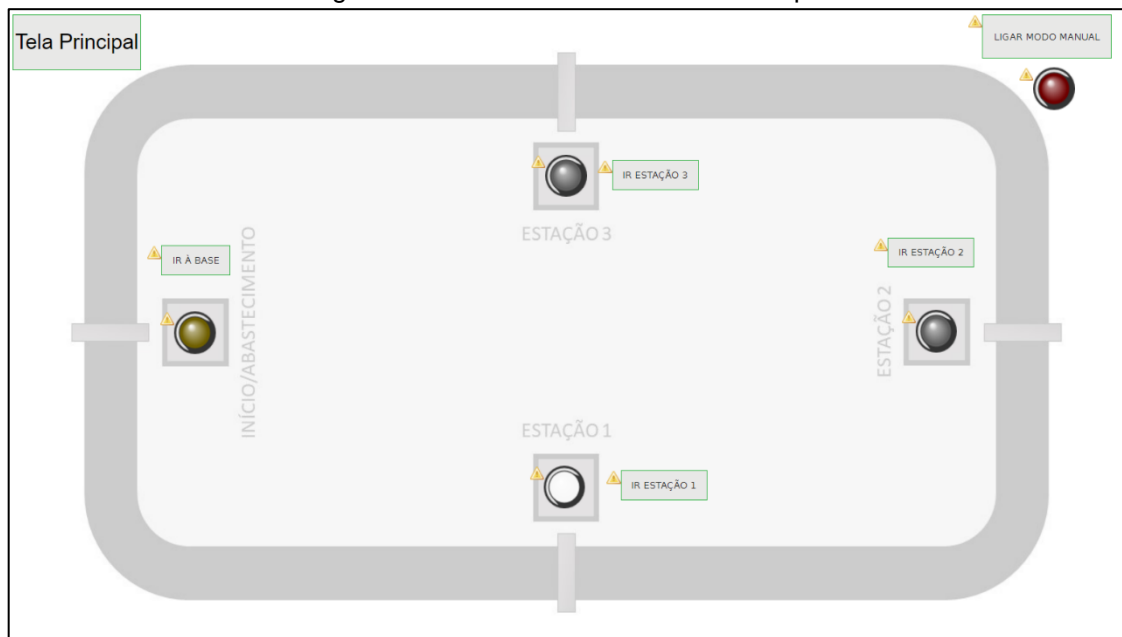
Figura 3.14 – Tela Principal do sistema supervisório.



Fonte: Autoria própria, 2023

A segunda tela é a que executa comandos manuais. Disposta com um botão na parte lateral que habilita os comandos manuais e 4 botões na figura do circuito, que forçam o AGV a se deslocar até o local designado. Todos os comandos manuais possuem sinaleiros, que indicam o comando acionado, conforme ilustra a Figura 3.15.

Figura 3.15 – Tela Manual do sistema supervisório.



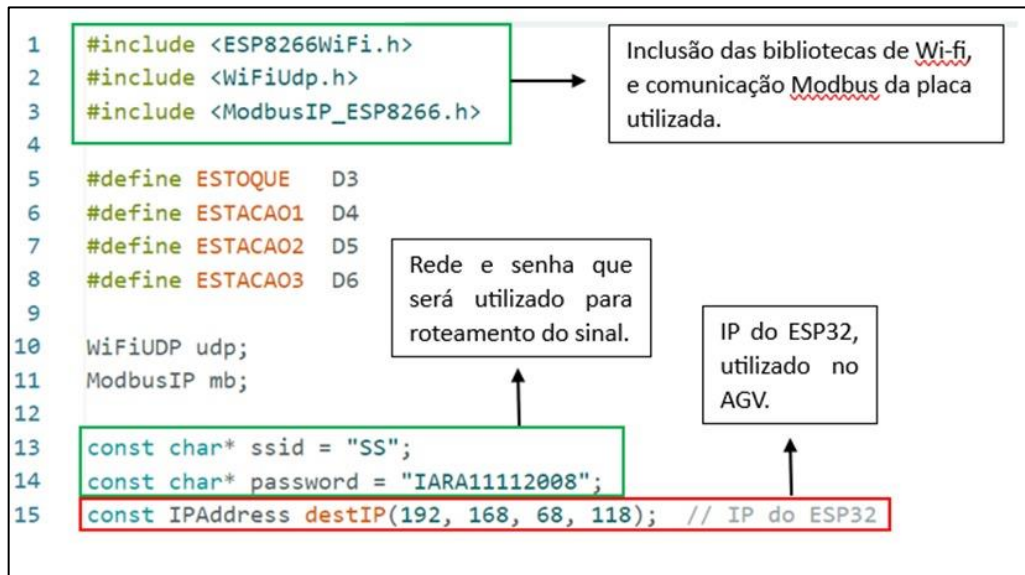
Fonte: Autoria própria, 2023

3.5 Integração do sistema supervisório com as placas ESP

As placas ESP32 que controla o AGV e a ESP8266 que monitora a planta estão se comunicando através de Wi-Fi, onde o AGV irá somente receber informações e comandos enviadas pela planta, não enviando nenhuma informação a planta devido o sistema de comunicação. O sistema supervisório se comunica somente com a placa ESP8266 também através de Wi-Fi utilizando o protocolo Modbus TCP/IP.

A seguir, a Figura 3.16 ilustra a parte do código em que é configurado a comunicação do ESP8266.

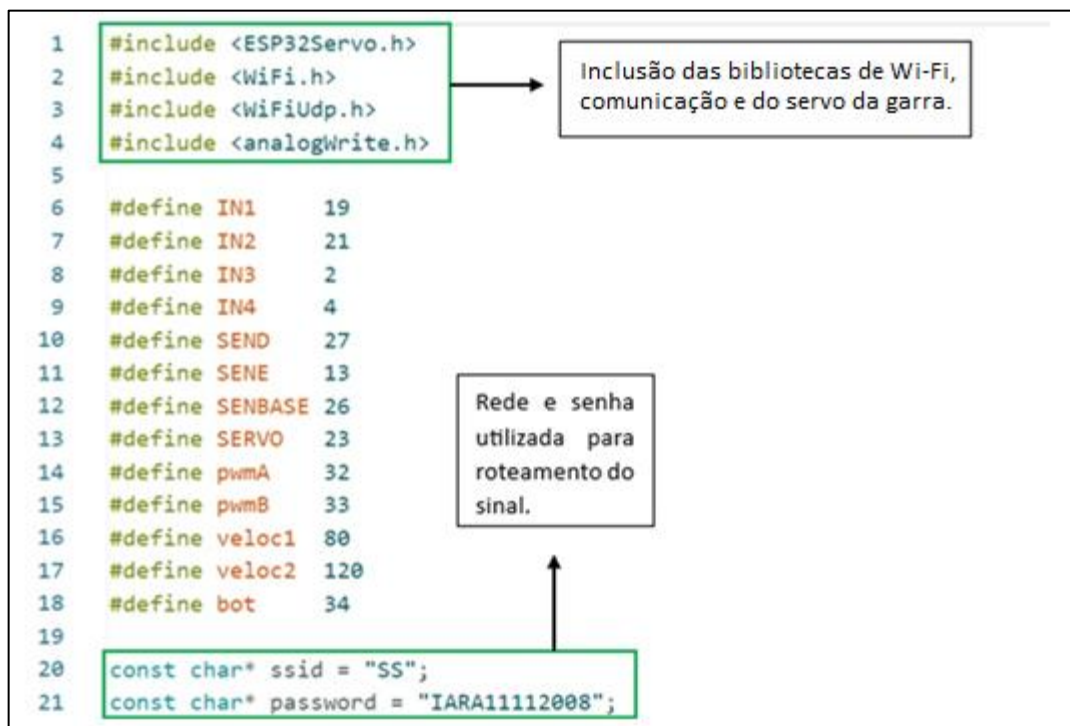
Figura 3.16 – Comunicação do ESP8266 em seu código.



Fonte: Autoria própria, 2023

A seguir, a Figura 3.17 ilustra a parte do código em que é configurado a comunicação do ESP32.

Figura 3.17 – Comunicação do ESP32 em seu código.



Fonte: Autoria própria, 2023

A comunicação dos sistemas é realizada por meio do protocolo Modbus TCP/IP e utiliza as camadas de registros Input Register e Coils. O supervisor se comunica

com o ESP8266 que fica localizado no ponto central do protótipo, e controla a parte da linha de processo. O supervisor utiliza apenas um data source (local do software que armazenas os data bases) e é configurado com o IP da placa, os dados das variáveis do processo são atualizados a cada 500 milissegundos. A seguir, a Figura 3.18 ilustra as configurações de comunicação do ScadaBR.

Figura 3.18 – Configurações de comunicação do ScadaBR.

The screenshot shows the 'Propriedades do Modbus IP' configuration window. The settings are as follows:

Nome	ESP8266
Export ID (XID)	DS_523544
Período de atualização	500 milissegundo(s)
Quantização	<input type="checkbox"/>
Timeout (ms)	500
Retentativas	2
Apenas quantidades contíguas	<input type="checkbox"/>
Criar pontos de monitor de escravo	<input type="checkbox"/>
Máxima contagem de leitura de bits	2000
Máxima contagem de leitura de registradores	125
Máxima contagem de escrita de registradores	120
Tipo de transporte	TCP
Host	192.168.68.116
Porta	502
Encapsulado	<input type="checkbox"/>
Criar ponto monitor de conexão	<input type="checkbox"/>
Níveis de alarme de eventos	
Exceção de data source	Urgente
Exceção de leitura de data point	Urgente
Exceção de escrita em data point	Urgente

Fonte: Autoria própria, 2023

3.6 Problemas e soluções

Este subcapítulo foi escrito com o objetivo expor problemas onde foram tomadas decisões que determinaram o rumo do projeto, para que assim também seja possível ajudar futuros projetos semelhantes a serem desenvolvidos. A seguir estão listados os principais problemas e soluções.

Problema 1 – Bateria do AGV sem capacidade:

Durante testes após a montagem final do AGV, com o servo motor que aciona o mecanismo da garra instalado, o power bank que alimentava todo o AGV estava não conseguiu fornecer energia suficiente para que os 4 motores que movimentam as rodas fossem acionados. Após algumas análises, foi identificado que devido a ponte H estar acionando 2 motores em cada canal e cada motor necessitar de uma alimentação entre 3 e 6 VDC, deveria ser alimentado por uma tensão de pelo menos 12 VDC para suportar os motores, porém o power bank alimentava com 5 VDC. Este problema passou a ser notado após a instalação do servo motor e a garra, talvez o fato do peso do AGV ter aumentado fez com que a carga nos motores aumentasse e causasse este problema.

Para solução deste problema foi definido pelo grupo e pensando na viabilidade do projeto que houvesse uma fonte de alimentação dedicada para a ponte H. Foi utilizado primeiramente baterias convencionais de 9 VDC, porém descarregavam muito rápido, tornando seu uso inviável. Foi pesquisado qual tipo de bateria atenderia e decidiu-se utilizar a bateria de Li-Ion 18650, que fornece tensão de 12 VDC, é recarregável e tem capacidade de 2500 mAh, solucionando o problema.

Problema 2 – Comunicação ente placas ESP:

Durante os testes de funcionamento da comunicação entre os ESPs, constataram-se dois obstáculos significativos. O primeiro está relacionado à maneira como a comunicação UDP envia e lê as informações no pacote. Tornou-se evidente que enviar uma mensagem no formato string era mais conveniente, uma vez que, ao enviar uma variável do tipo inteiro, o ESP8266 e o ESP32 interpretavam essa variável de forma diferente. Conseqüentemente, as mensagens enviadas pelas estações eram interpretadas incorretamente pelo AGV, invalidando a comunicação. Por outro lado, ao utilizar uma variável do tipo string, foi necessário apenas limitar o tamanho da mensagem. Ao enviar, por exemplo, o valor "2", a informação recebida continha mais do que apenas o valor "2"; vinha acompanhada de uma mensagem adicional, como "2jdsasdn", tornando impossível a interpretação correta pelo ESP. Para resolver esse problema, foi necessário eliminar todo o excesso presente na mensagem.

Outro problema ocorreu em relação à forma de envio das mensagens. Inicialmente, acreditava-se que as mensagens eram enviadas e, caso não fossem lidas, seriam perdidas. No entanto, descobriu-se que todas as mensagens enviadas seriam lidas, independentemente do momento. Isso levou o AGV a interpretar informações equivocadas. Por exemplo, quando solicitado a ir para a estação 2, ele ficava preso em um loop, indo apenas para a estação 1. Essa situação ocorria devido ao envio prévio de várias requisições para que ele fosse para a estação 1. Para solucionar esse problema, foi necessário implementar um mecanismo para impedir o envio repetido da mensagem em uma única requisição, permitindo que o ESP recebesse apenas uma requisição por vez.

Problema 3 – Mecanismo da garra espanado:

Devido o mecanismo que prende a garra ao servo motor ser de material plástico, e a engrenagem do servo motor de metal, houve um desgaste após repetições da garra abrindo e fechando para coletar e soltar as peças durante os testes finais.

Para solução deste problema, foi necessário comprar uma garra nova, pois não é vendido separadamente somente a peça que espanou. Entretanto, foi visto que se trata de um problema recorrente que aconteceu com o modelo de garra adquirido, pois o novo modelo adquirido veio com uma melhoria onde a parte do mecanismo que encaixa no servo motor está com uma bucha de metal, para que não haja desgaste precoce da peça.

CONSIDERAÇÕES FINAIS

O projeto que se intitula Abastecimento por meio de AGV Monitorado por Sistema Supervisório tem como o objetivo automatizar a reposição de peças em sistemas de abastecimento por meio de AGV, monitorando todo o processo em tempo real através de um sistema supervisório. O objetivo foi analisar os benefícios e desafios dessa abordagem, avaliando sua viabilidade e eficiência no contexto do abastecimento de materiais em ambientes industriais.

Durante a pesquisa, podemos constatar que o uso de AGV's no abastecimento traz uma série de vantagens significativas. Esses veículos autônomos, equipados com sensores e sistemas de navegação avançados, são capazes de se mover de forma autônoma e segura, evitando obstáculos e otimizando o tempo de transporte. Além disso, a flexibilidade operacional desses veículos permite a adaptação rápida a diferentes demandas e fluxos de trabalho.

Ao integrar um sistema supervisório ao processo, foi possível monitorar em tempo real o desempenho do AGV, garantindo a eficiência das operações e a detecção de possíveis falhas ou problemas. O sistema supervisório fornece informações valiosas sobre a localização do veículo, o status do abastecimento e a disponibilidade de materiais nas estações de trabalho, permitindo uma gestão mais precisa e assertiva.

No entanto, também foi identificado desafios que precisam ser abordados para a implementação bem-sucedida desse sistema. A segurança é uma preocupação fundamental, já que os AGV's compartilham espaços de trabalho com funcionários e outros equipamentos. É essencial estabelecer medidas de segurança adequadas, como sensores de detecção de obstáculos e protocolos de emergência.

Outra questão importante é a integração entre os AGV's e os sistemas de armazenamento e controle de estoque existentes. A comunicação eficiente entre os diferentes componentes do sistema é fundamental para garantir a sincronização adequada do abastecimento e evitar problemas como falta ou excesso de materiais.

Recomenda-se a continuidade dos estudos nessa área, buscando aprimorar a tecnologia dos AGV's e dos sistemas supervisórios, bem como investigar soluções para os desafios identificados, implementar o uso de banco de dados para gerenciar todos os dados obtidos do processo para que possam ser analisados, gerando gráficos, relatórios e promovendo melhorias contínuas nos processos. Além disso, é importante realizar testes e simulações em ambientes reais para validar a eficiência e a segurança desse sistema em diferentes contextos industriais.

Por fim, a implementação do abastecimento por meio de AGV monitorado por sistema supervisório tem o potencial de trazer benefícios significativos, como aumento da produtividade, redução de erros e melhoria nas operações logísticas. Com os devidos cuidados e investimentos, essa abordagem pode revolucionar a forma como o abastecimento é realizado em ambientes industriais, impulsionando a eficiência e a competitividade das empresas.

REFERÊNCIAS

AUTOMAÇÃO COTANET. **Sistema supervisorio de produção**. Disponível em: <<https://automacao.cotanet.com.br/sistemas-supervisorios/sistema-supervisorio-de-producao/>>. Acesso em: 10 set. 2022.

BLOG ELETROGATE. **Conhecendo o esp 32 introdução**. Disponível em: <<https://blog.eletrogate.com/conhecendo-o-esp32-introducao-1/>>. Acesso em: 05 nov. 2022.

CONCEITO. DE. **Metodologia (2010)**. Disponível em: <<https://conceito.de/metodologia/>>. Acesso em: 20 out. 2022.

CURTO-CIRCUITO. **Conhecendo esp 8286 (2017)**. Disponível em: <<https://curtocircuito.com.br/blog/Categoria%20IoT/conhecendo-esp32/>>. Acesso em: 05 nov. 2022.

CURTO-CIRCUITO. **Conhecendo esp 32 (2018)**. Disponível em: <<https://curtocircuito.com.br/blog/Categoria%20IoT/conhecendo-esp32/>>. Acesso em: 05 nov. 2022.

DELAGE. **Como funcionam os agvs**. Disponível em: <<https://delage.com.br/blog/como-funcionam-os-agvs/>>. Acesso em: 22 ago. 2022.

EMBARCADOS. **Modulo ESP 826**. Disponível em: <<https://blog.embarcados.com/modulo-esp-826/>>. Acesso em: 31 out. 2022.

FACULDADE DE TECNOLOGIA DE SÃO BERNARDO DO CAMPO "ADIB MOISÉS DIB". **Manual de Normalização de TCC – Trabalho de Conclusão de Curso**. Material didático para utilização nos projetos de trabalho de graduação, dos cursos de Tecnologia em Automação Industrial, Manufatura Avançada e Tecnologia em Informática para Negócios. 6. ed. São Bernardo do Campo: FATEC, 2021.

KALATEC. **O que é servo motor. Modulo ESP 826**. Disponível em: <<https://blog.kalatec.com.br/o-que-e-servo-motor/>>. Acesso em: 01 nov. 2022.

LIMA, Ana Luiza Lorenzen. **"Método científico"; Brasil Escola**. Disponível em: <<https://brasilecola.uol.com.br/quimica/metodo-cientifico.htm/>>. Acesso em: 18 out. 2022.

LOGIQUE SISTEMAS. **Sistemas Supervisórios.** Disponível em: <<https://www.logiquesistemas.com.br/blog/sistemas-supervisorios/>>. Acesso em: 12 set. 2022.

WEBLINK, Andrei L. **Como Funcionam e Tipos de Protocolos de Internet.** Disponível em: <<https://www.weblink.com.br/blog/tecnologia/conheca-os-principais-protocolos-de-internet/>>. Acesso em 11 out. 2022.

MUNDO PROJETADO. **Sensor de obstáculo infravermelho.** Disponível em: <<https://mundoprojetado.com.br/sensor-de-obstaculo-infravermelho/>>. Acesso em: 29 out. 2022.

PORTAL VIDA DE SILICIO. **Conhecendo o ESP32 (2017).** Disponível em: <<https://portal.vidadesilicio.com.br/conhecendo-o-esp32//>>. Acesso em: 31 out 2022.

REDE INDUSTRIA 40. **Internet das coisas na manufatura industrial uma ferramenta da indústria 40 para interação dos processos.** Disponível em: <<https://redeindustria40.com.br/internet-das-coisas-na-manufatura-industrial-uma-ferramenta-da-industria-4-0-para-interacao-dos-processos//>>. Acesso em: 02 set. 2022.

TECNIAR. **Notícias o que é o clp.** Disponível em: <Fonte: <https://www.tecniar.com.br/noticias/o-que-e-o-clp/>>. Acesso em: 22 ago. 2022.

APÊNDICE A – PROGRAMAÇÃO DO ESP32 (AGV)

```

#include <ESP32Servo.h>
#include <WiFi.h>
#include <WiFiUdp.h>
#include <analogWrite.h>

#define IN1 19
#define IN2 21
#define IN3 2
#define IN4 4
#define SEND 27
#define SENE 13
#define SENBASE 26
#define SERVO 23
#define pwmA 32
#define pwmB 33
#define veloc1 100
#define veloc2 120
#define veloc3 70
#define bot 5

const char* ssid = "TCC";
const char* password = "";
String mensagem;
int estacao = 0, packetSize;
bool sensorDireita, sensorEsquerda, sensorEstacao, sensorBase, base = false,
contandoEstacao = false;

WiFiUDP udp;
Servo myservo;

void setup() {
  Serial.begin(115200);

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Conectando-se à rede Wi-Fi...");
  }
  Serial.println(WiFi.localIP());
  delay(1000);
  Serial.println("Conexão Wi-Fi estabelecida");

  udp.begin(1234);
  Serial.println("Comunicação UDP iniciada");

  myservo.attach(SERVO);

```

```

pinMode(IN1, OUTPUT);
pinMode(IN2, OUTPUT);
pinMode(IN3, OUTPUT);
pinMode(IN4, OUTPUT);
pinMode(SEND, INPUT);
pinMode(SENE, INPUT);
pinMode(SENBASE, INPUT);
pinMode(bot, INPUT);
pinMode(pwmA, OUTPUT);
pinMode(pwmB, OUTPUT);
myservo.write(0);
delay(1000);
myservo.write(90);
}

void loop() {
  receberMensagem();
  movimentar();
}

void movimentar() {
  if (mensagem == "0" && base == false) {
    andarBase();
  } else if (mensagem == "1" && base == true) {
    andarEstacao(1);
  } else if (mensagem == "2" && base == true) {
    andarEstacao(2);
  } else if (mensagem == "3" && base == true) {
    andarEstacao(3);
  } else if (mensagem == "5" && base == true) {
    estacao = 0;
    parar();
  }
}

void receberMensagem() {
  packetSize = udp.parsePacket();
  if (packetSize) {
    char buffer[255];
    udp.read(buffer, 255);
    buffer[packetSize] = '\0';
    mensagem = buffer;
  }
  Serial.println(mensagem);
}

void andarBase() {
  while (!digitalRead(SENBASE)) {
    andar();
  }
}

```

```

    parar();
    estacao = 0;
    base = true;
    abreGarra();
}

void andarEstacao(int quantidadeEstacao) {
    verificaBot();
    delay(2000);
    fechaGarra();
    delay(2000);
    while (estacao < quantidadeEstacao) {
        andar();
    }
    parar();
    delay(2000);
    abreGarra();
    delay(1000);
    fechaGarra();
    delay(2000);
    while (!digitalRead(SENBASE)) {
        andar();
    }
    parar();
    delay(2000);
    abreGarra();
    delay(2000);
    estacao = 0;
}

void contaEstacao() {
    estacao += 1;
    while (digitalRead(SEND) && digitalRead(SENE)){
        delay(10);
    }
}

void andar() {
    sensorDireita = digitalRead(SEND);
    sensorEsquerda = digitalRead(SENE);

    if (!sensorDireita && !sensorEsquerda) {
        analogWrite(pwmA, veloc1);
        analogWrite(pwmB, veloc1);
        digitalWrite(IN1, HIGH);
        digitalWrite(IN2, LOW);
        digitalWrite(IN3, LOW);
        digitalWrite(IN4, HIGH);
    } else if (sensorDireita && !sensorEsquerda) {
        analogWrite(pwmA, veloc2);

```

```
    analogWrite(pwmB, veloc2);
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
} else if (!sensorDireita && sensorEsquerda) {
    analogWrite(pwmA, veloc2);
    analogWrite(pwmB, veloc2);
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);

} else if (sensorDireita && sensorEsquerda) {

    analogWrite(pwmA, veloc3);
    analogWrite(pwmB, veloc3);
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    contaEstacao();
}
}

void parar() {
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, HIGH);
}

void verificaBot() {
    while (digitalRead(bot)) {
        delay(10);
    }
}

void fechaGarra() {
    myservo.write(90);
}

void abreGarra() {
    myservo.write(0);
}
```

APÊNDICE B – PROGRAMAÇÃO DO ESP8266 (PLANTA)

```

#include <ESP8266WiFi.h>
#include <WiFiUdp.h>
#include <ModbusIP_ESP8266.h>

#define ESTOQUE D0
#define ESTACAO1 D7
#define ESTACAO2 D6
#define ESTACAO3 D5

WiFiUDP udp;
ModbusIP mb;

const char* ssid = "TCC";
const char* password = "";
const IPAddress destIP(192, 168, 0, 102); // IP do ESP32
const int destPort = 1234;
int packetSize, quantidadeEstacao1 = 0, quantidadeEstacao2 = 0,
quantidadeEstacao3 = 0;
String mensagem = "", packet, mensagemRecebida = "";

bool base = 0, mensagemEnviada = false, mensagemEnviada1 = false,
mensagemEnviada2 = false, mensagemEnviada3 = false, mensagemEnviada5 =
false;

void setup() {
  Serial.begin(115200);

  // Conectar-se à rede Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Conectando-se à rede Wi-Fi...");
    Serial.println(WiFi.localIP());
    Serial.println(WiFi.macAddress());
  }
  Serial.println("Conexão Wi-Fi estabelecida");

  // Iniciar a comunicação UDP
  udp.begin(8888);
  Serial.println("Comunicação UDP iniciada");

  mb.server();

  mb.addCoil(100); //INICIA MOVIMENTO PARA BASE
  mb.addCoil(101); //ESTAÇÃO 1
  mb.addCoil(102); //ESTAÇÃO 2
  mb.addCoil(103); //ESTAÇÃO 3

```

```

mb.addCoil(104); //MODO AUTOMATICO
mb.addIsts(200); //INDICAÇÃO BASE
mb.addIsts(201); //INDICAÇÃO ESTAÇÃO 1
mb.addIsts(202); //INDICAÇÃO ESTAÇÃO 2
mb.addIsts(203); //INDICAÇÃO ESTAÇÃO 3
mb.addHreg(301);
mb.addHreg(302);
mb.addHreg(303);

pinMode(BUILTIN_LED, OUTPUT);
pinMode(ESTOQUE, INPUT);
pinMode(ESTACAO1, INPUT);
pinMode(ESTACAO2, INPUT);
pinMode(ESTACAO3, INPUT);

digitalWrite(BUILTIN_LED, LOW);
}

void loop() {
  mb.task();
  iniciaBase();
  verificaManualAutomatico();
  verificaMensagemPodeSerEnviada();
}

void iniciaBase() {
  while (!mb.Coil(100) && base == 0) {
    mb.task();
  }
  if (base == 0) {
    mensagem = "0";
    base = 1;
    mb.Ists(200,base);
    enviaMensagem();
    mensagemEnviada = false;
  }
}

void verificaManualAutomatico() {
  if (!mb.Coil(104)) {
    movimentoManual();
  } else {
    movimentoAutomatico();
  }
}

void movimentoManual() {
  if (mb.Coil(101)) {
    mensagem = "1";
  } else if (mb.Coil(102)) {

```

```

    mensagem = "2";
} else if (mb.Coil(103)) {
    mensagem = "3";
} else {
    mensagem = "5";
}
enviaMensagemManual();
}

void movimentoAutomatico() {
    if (digitalRead(ESTACAO1)) { //VERIFICA SE ESTAÇÃO 1 TEM PEÇA
        mensagem = "1";
        enviaMensagemAut(1);
        mb.lsts(201,mensagemEnviada1);
    } else if (digitalRead(ESTACAO2)) { //VERIFICA SE ESTAÇÃO 2 TEM PEÇA
        mensagem = "2";
        enviaMensagemAut(2);
        mb.lsts(202,mensagemEnviada2);
    } else if (digitalRead(ESTACAO3)) { //VERIFICA SE ESTAÇÃO 3 TEM PEÇA
        mensagem = "3";
        enviaMensagemAut(3);
        mb.lsts(203,mensagemEnviada3);
    } else {
        mensagem = "5";
        enviaMensagemAut(5);
    }
}

void verificaMensagemPodeSerEnviada() {
    if (!mb.Coil(101) && !mb.Coil(102) && !mb.Coil(103)) {
        mensagemEnviada = false;
    }
    if (!digitalRead(ESTACAO1)) {
        mensagemEnviada1 = false;
        mb.lsts(201,mensagemEnviada1);
    }
    if (!digitalRead(ESTACAO2)) {
        mensagemEnviada2 = false;
        mb.lsts(202,mensagemEnviada2);
    }
    if (!digitalRead(ESTACAO3)) {
        mensagemEnviada3 = false;
        mb.lsts(203,mensagemEnviada3);
    }
}

void enviaMensagemAut(int valor) {
    if (valor == 1 && !mensagemEnviada1) {
        enviaMensagem();
        mensagemEnviada1 = true;
    }
}

```



```

    mb.lsts(201,mensagemEnviada1);
    contaQuantidadePecas(1);
} else if (valor == 2 && !mensagemEnviada2) {
    enviaMensagem();
    mensagemEnviada2 = true;
    mb.lsts(202,mensagemEnviada2);
    contaQuantidadePecas(2);
} else if (valor == 3 && !mensagemEnviada3) {
    enviaMensagem();
    mensagemEnviada3 = true;
    mb.lsts(203,mensagemEnviada3);
    contaQuantidadePecas(3);
} else if (valor == 5) {
    enviaMensagem();
}
}
delay(500);
}

void enviaMensagem() {
    while (digitalRead(ESTOQUE)) {
        delay(10);
    }
    udp.beginPacket(destIP, destPort);
    udp.print(mensagem);
    udp.endPacket();
    Serial.print("Mensagem enviada: ");
    Serial.println(mensagem);
}

void enviaMensagemManual() {
    if (!mensagemEnviada) {
        udp.beginPacket(destIP, destPort);
        udp.print(mensagem);
        udp.endPacket();
        Serial.print("Mensagem enviada pela Estacao: ");
        Serial.println(mensagem);
        mensagemEnviada = true;
    }
    delay(500);
}

void contaQuantidadePecas(int n_estacao){
    if (n_estacao == 1){
        quantidadeEstacao1 += 1;
        mb.Hreg(301,quantidadeEstacao1);
    }
    else if (n_estacao == 2){
        quantidadeEstacao1 += 1;
        mb.Hreg(302,quantidadeEstacao2);
    }
}

```

```
else if (n_estacao == 3){  
    quantidadeEstacao1 += 1;  
    mb.Hreg(303,quantidadeEstacao3);  
}  
}
```