



*Etec Jacinto Ferreira de Sá - 066 – Ourinhos*

## Ensino Médio Integrado ao Técnico em Automação Industrial

### ROSQUEADOR DE TAMPAS DE GARRAFAS

#### BOTTLE CAP SCREWING TOOL

Antônio Saturnino Camacho<sup>1</sup>  
Guilherme Pelizari de Souza<sup>2</sup>  
Higor Lourenço Clemente<sup>3</sup>  
João Otavio Antunes<sup>4</sup>  
Lucas Miranda Gurgel De Jesus<sup>5</sup>

**Resumo:** O processo manual de rosqueamento de tampas em embalagens, especialmente em operações de grande escala com garrafas plásticas reutilizáveis, frequentemente resulta em vedação inadequada, comprometendo a segurança do conteúdo e elevando os níveis de desperdício. Com o objetivo de mitigar essas perdas e otimizar a qualidade da vedação, o presente artigo apresenta o desenvolvimento de um dispositivo rosqueador de tampas automatizado e de baixo custo. A solução é detalhada tecnicamente, destacando a utilização de componentes acessíveis como Arduino, motor DC, motor de passo e peças fabricadas via impressão 3D. O projeto inclui a integração com sensores que permitem o controle preciso da força de torque, crucial para garantir a vedação ideal sem danificar as embalagens. Sua arquitetura modular e a capacidade de adaptação a diferentes tamanhos demonstram a versatilidade e a escalabilidade do equipamento. Os testes iniciais de validação do protótipo confirmaram a capacidade do sistema em aplicar o torque padronizado com um desvio mínimo, resultando em uma taxa de falha de vedação significativamente reduzida em comparação ao método manual. Este avanço tecnológico se posiciona como uma ferramenta eficaz para pequenas e médias empresas que buscam automação sem grandes investimentos. Em suma, este trabalho reforça a importância de tecnologias acessíveis para a melhoria de processos produtivos, provando que um rosqueador com controle de torque pode gerar um impacto significativo na qualidade, produtividade e sustentabilidade das operações industriais.

**Palavras-chave:** Rosqueamento Automatizado; Controle de Torque; Rosqueador de Tampas; Sustentabilidade Industrial.

<sup>1</sup>Ensino Fundamental II, Centro Educacional Sesi, Ourinhos /SP, antonio.camacho@etec.sp.gov.br

<sup>2</sup>Ensino Fundamental II, Racanello, em Ourinhos/SP, guilherme.souza696@etec.sp.gov.br

<sup>3</sup>Ensino Fundamental II, Prof. José Alves Martins, em Ourinhos/SP, higor.clemente@etec.gov.sp.br

<sup>4</sup>Ensino Fundamental II, Prof. Homero Calvozo, em Ourinhos/SP, joao.antunes29@etec.sp.gov.br

<sup>5</sup>Ensino Fundamental II, Colégio Bagozzi, em Ourinhos/SP, lucas.jesus160@etec.sp.gov.br

### ***Etec Jacinto Ferreira de Sá - 066 – Ourinhos***

**Abstract :** The manual process of screwing caps onto packaging, especially in large-scale operations involving reusable plastic bottles, frequently results in inadequate sealing, compromising the safety of the contents and raising waste levels. Aiming to mitigate these losses and optimize the quality of the seal, this article presents the development of a low-cost, automated cap-screwing device. The solution is technically detailed, highlighting the use of accessible components such as Arduino, a DC motor, a stepper motor, and parts manufactured via 3D printing. The design includes integration with sensors that allow for precise control of the torque force, which is crucial for ensuring the ideal seal without damaging the packaging. Its modular architecture and adaptability to different sizes demonstrate the equipment's versatility and scalability. Initial prototype validation tests confirmed the system's ability to apply standardized torque with minimal deviation, resulting in a significantly reduced sealing failure rate compared to the manual method. This technological advance is positioned as an effective tool for small and medium-sized enterprises seeking automation without major investment. In summary, this work reinforces the importance of accessible technologies for improving production processes, proving that a torque-controlled capping machine can generate a significant impact on the quality, productivity, and sustainability of industrial operations. **Keywords**

**Keywords:** Automated Capping; Torque Control; Cap-Threading Device; Low-Cost Automation; Industrial Sustainability.

## **1. INTRODUÇÃO**

No contexto industrial, a eficiência dos processos produtivos é algo necessário para garantir a qualidade dos produtos e reduzir desperdícios. Um dos desafios enfrentados por diversas indústrias, como as de bebidas, produtos de limpeza e alimentos, está na etapa de fechamento das embalagens plásticas, especialmente garrafas com tampa rosqueável. O processo manual pode resultar em problemas na integridade da tampa da vedação, comprometendo a segurança do conteúdo e a durabilidade do produto. Diante dessa necessidade, este artigo apresenta o desenvolvimento de um dispositivo inovador: um rosqueador de tampas automatizado, projetado para garantir a padronização e a eficiência desse procedimento.

A principal motivação é buscar por soluções tecnológicas baratas que garantam a automação de processos industriais de forma simples e prática. O problema da pesquisa consiste em avaliar de que maneira a implementação do rosqueador pode melhorar a qualidade da vedação das embalagens, reduzir falhas e aumentar a produtividade das linhas de produção. É fato que a utilização desse dispositivo pode minimizar erros humanos, garantir uma padronização maior no fechamento das



### ***Etec Jacinto Ferreira de Sá - 066 – Ourinhos***

tampas e oferecer maior segurança para o consumidor final.

Para atingir esses objetivos, a pesquisa baseou-se no desenvolvimento de um protótipo, utilizando componentes acessíveis, como Arduino, motores DC e motores de passo, além de peças impressas em 3D. Começou com a experimentação prática para analisar a eficácia do rosqueador, tornando-o compatível com de tampas de garrafas pet.

Espera-se que o dispositivo contribua significativamente para a melhoria dos processos produtivos, permitindo um rosqueamento tenha mais precisão e eficiência, além de possibilitar sua integração em linhas de produção automatizadas. Os resultados finais indicam um avanço na padronização das vedações das embalagens, aumentando a confiabilidade dos produtos e otimizando o tempo de produção.

Os próximos capítulos apresentarão uma análise detalhada da arquitetura do protótipo. Primeiramente, será explorado cada componente fundamental, como o Arduino, o motor DC e as peças impressas em 3D, justificando a escolha e a função de cada um na estrutura física do dispositivo. Em seguida, será detalhada a lógica de programação desenvolvida, que atua como o cérebro do sistema, garantindo a precisão e a repetibilidade do processo.

## **2. PROTÓTIPO DE UM ROSQUEADOR DE TAMPAS DE GARRAFAS**

A vedação adequada das tampas em garrafas plásticas desempenha um papel crucial na preservação da qualidade, segurança e integridade dos produtos armazenados. No ambiente industrial, especialmente quando as garrafas são produzidas em larga escala, um fechamento inconsistente pode gerar uma série de problemas, como vazamentos, contaminação do conteúdo, danos à embalagem e até riscos à saúde do consumidor. Quando o processo de rosqueamento é feito manualmente, as variações humanas, como a força aplicada e a velocidade de rotação, podem comprometer a eficácia da vedação, resultando em tampas mal fixadas ou apertadas com excesso de força. Essas falhas, além de gerarem desperdício e prejuízos, afetam diretamente a reputação das marcas e a satisfação do consumidor.

Diante dessa realidade, o projeto propõe a implementação de uma solução



**SÃO PAULO**  
GOVERNO DO ESTADO  
SÃO PAULO SÃO TODOS

***Etec Jacinto Ferreira de Sá - 066 – Ourinhos***

automatizada para o fechamento das tampas, capaz de garantir a vedação correta e padronizada de todas as embalagens. Para atingir esse objetivo, são aplicados testes de vedação, como ensaios de pressão e imersão. O teste de pressão simula condições de armazenamento e transporte, ao aplicar uma força interna na garrafa para verificar a existência de vazamentos, enquanto o teste de imersão submerge a garrafa em água para identificar falhas no fechamento, observando se surgem bolhas, indicativas de falhas na vedação. Essas metodologias permitem avaliar de maneira precisa e rápida a eficácia do processo de fechamento antes da distribuição em larga escala.

Para o desenvolvimento desse sistema automatizado, foi necessário realizar uma pesquisa teórica e prática, com base em uma abordagem de prototipagem e experimentação. A escolha de componentes e sistemas foi fundamentada em conhecimentos de automação industrial, eletrônica, microcontroladores e manufatura aditiva. O uso de microcontroladores, como o Arduino, é uma parte central do projeto. O Arduino é uma plataforma de prototipagem de código aberto, amplamente utilizada para controlar dispositivos eletrônicos de forma simples e acessível. No projeto, o Arduino recebe os sinais dos sensores, processa as informações e, em seguida, envia comandos para os motores, que realizam o movimento de rosqueamento. Essa flexibilidade permite que o sistema seja facilmente programado para atender diferentes tipos de garrafas e tampas, ajustando os parâmetros conforme necessário.

Os motores usados no sistema são de dois tipos, motores DC e motores de Passo. Os motores DC são utilizados para movimentação rápida e contínua. Eles são controlados por circuitos de controle, que permite inverter a direção de rotação e controlar a velocidade do motor de forma eficiente. Já os motores de passo, por sua vez, oferecem maior precisão e controle sobre o movimento, pois realizam movimentos em etapas discretas. Isso é essencial para garantir que o rosqueamento seja feito de maneira exata e consistente, especialmente quando a precisão no posicionamento da tampa é crítica. Os motores de passo são controlados por driver A4988, que se comunicam com o microcontrolador para ajustar a quantidade de movimento e a precisão do rosqueamento.

O uso de Impressão 3D para a fabricação de peças permite uma flexibilidade notável na criação de componentes personalizados para o sistema. A impressão 3D possibilita a produção de peças com designs complexos, ajustando-se a diferentes

### ***Etec Jacinto Ferreira de Sá - 066 – Ourinhos***

tipos de tampas e garrafas, sem a necessidade de moldes caros ou longos períodos de espera. As peças podem ser feitas de materiais como PLA, que é amplamente utilizado por sua facilidade de impressão e baixo custo. A capacidade de customizar rapidamente as peças e adaptar o sistema a diferentes especificações de garrafas e tampas amplia a versatilidade do projeto, tornando-o aplicável a uma ampla gama de indústrias, como bebidas, produtos de limpeza e cosméticos. Além disso, a impressão 3D reduz os custos de produção e acelera o desenvolvimento do protótipo, permitindo testes rápidos e ajustes no design do sistema.

O sistema de automação desenvolvido integra todas essas tecnologias para criar um processo de rosqueamento eficiente, preciso e adaptável. Com o uso de sensores, motores, microcontroladores e manufatura aditiva, o sistema oferece uma solução robusta para a vedação de tampas em garrafas plásticas, garantindo a qualidade do fechamento e a segurança do produto, além de aumentar a produtividade nas linhas de produção. O controle automatizado não só elimina erros humanos, mas também padroniza o processo, tornando-o mais seguro, rápido e eficaz.

## **2.2 Detalhamento do Projeto**

Este projeto descreve o desenvolvimento e validação de um protótipo de rosqueador de tampas de garrafa, concebido para otimizar produções de pequena escala. Foi feito o desenho do conceito e modelagem das peças mecânicas em software FUSION 360, baseado em um tamanho escolhido das garrafas. Selecionamos os componentes eletrônicos e mecânicos com base nos critérios de custo, acessibilidade e funcionalidade. Foram selecionados um microcontrolador Arduino uno como unidade central de processamento, um motor de passo para o posicionamento de um atuador linear onde está acoplado um motor DC para a aplicação do torque, sensores fim de curso para o posicionado na base e na parte superior do atuador.

Impressão das peças modeladas em uma impressora 3D e montagem da estrutura física do rosqueador, integrando os componentes eletrônicos e , para a estrutura foi usado dois barras lineares de aproximadamente de 210mm e um fuso de 200mm. Programação da lógica de controle no protótipo de desenvolvimento Arduino.

### ***Etec Jacinto Ferreira de Sá - 066 – Ourinhos***

O software foi projetado para controlar a sequência de movimentos, a velocidade e o torque aplicado pelo motor, garantindo a repetibilidade do processo. Foram feitos ensaios práticos para analisar a eficácia do rosqueador. Avaliamos alguns parâmetros como a consistência da vedação e o tempo de ciclo. Os resultados foram comparados aos do processo manual para quantificar as melhorias. Os dados coletados na fase de testes foram tabulados e analisados para discutir o desempenho do protótipo, o avanço na padronização das vedações e o impacto na otimização do tempo de produção.

### **3. POSICIONADOR ROTACIONAL**

O posicionador rotacional é um componente chave do sistema, sendo ativado somente após a conclusão do processo de rosqueamento. Ao finalizar essa etapa, a esteira transportadora emite um sinal que aciona o controle do posicionador, garantindo a sincronia da operação.

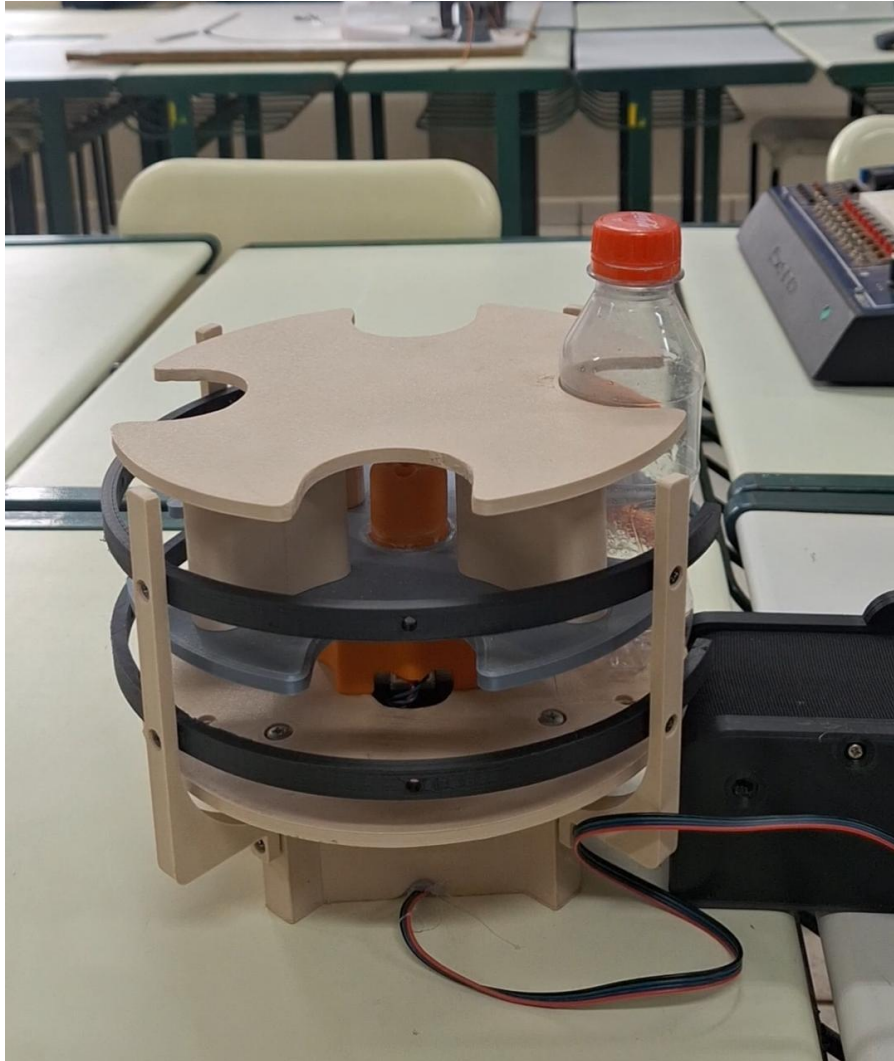
A estrutura do protótipo foi desenvolvida utilizando a tecnologia de Impressão 3D. Para otimizar a segurança durante as fases de movimentação e transporte das garrafas, foram incorporados suportes laterais. Esses suportes têm a função de prevenir a queda dos recipientes da esteira durante o movimento, assegurando a integridade do processo.

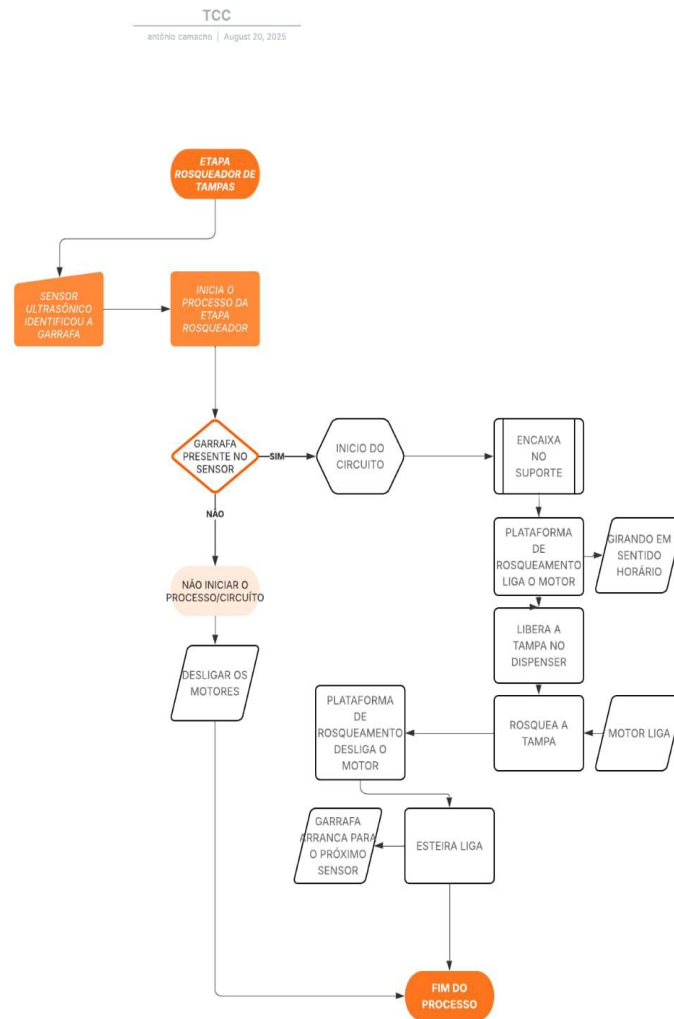
A implementação da lógica de controle exige a definição de uma baixa velocidade de rotação para o motor de passo. Esta não é uma limitação, mas sim uma medida estratégica de projeto visando a otimização do desempenho mecânico. Em aplicações que demandam alta precisão e que envolvem uma carga significativa (considerando o peso e a inércia da garrafa), a operação em velocidade reduzida é essencial.

Essa estratégia garante a manutenção de um alto torque, o qual é crucial para vencer a inércia e o atrito inerentes à carga sem que ocorra a perda de passos. A perda de passos resultaria em um erro cumulativo de posicionamento, o que comprometeria seriamente a sequência de processamento e, conseqüentemente, a qualidade do produto final. Portanto, a baixa velocidade é um recurso intencional que maximiza a confiabilidade e a precisão posicional do protótipo.

A ativação do motor é controlada por um botão de acionamento. A lógica de software foi desenvolvida para detectar apenas o momento da transição (da condição

***Etec Jacinto Ferreira de Sá - 066 – Ourinhos***  
solta para a condição pressionada). Este mecanismo, conhecido como "debounce" de software, é fundamental para evitar que o motor gire continuamente caso o operador mantenha o botão pressionado. Desta forma, fica assegurado que cada toque no botão corresponda a exatamente um avanço posicional do sistema.



**Etec Jacinto Ferreira de Sá - 066 – Ourinhos**


O fluxograma descreve a lógica de controle do protótipo o processo é iniciado pela detecção da garrafa se um sensor verifica se o recipiente está corretamente posicionado. Se a garrafa não for detectada, o sistema aborta a operação, desliga os motores e finaliza o processo para evitar erros. Se a garrafa estiver presente (condição "SIM"), o circuito é acionado. A garrafa é encaixada no suporte e a plataforma de rosqueamento é ativada, ligando o motor e iniciando a rotação no sentido horário. A tampa é liberada do *dispenser* sobre o gargalo e o motor procede ao rosqueamento preciso. Após o aperto, o motor da plataforma é desligado. Em seguida, a esteira é ligada para mover a garrafa para fora da área de trabalho. Assim que a garrafa passa pelo sensor, a esteira é desativada, marcando o FIM DO PROCESSO e preparando o sistema para o próximo ciclo.

*Etec Jacinto Ferreira de Sá - 066 – Ourinhos*

## 4. PROGRAMAÇÃO

### 4.1 Programação posicionar rotacional

```

const int STEP_PIN = 9; // Pino ligado ao STEP do A4988 (Pulso de movimento)
const int DIR_PIN = 8; // Pino ligado ao DIR do A4988 (Direção)
const int ENABLE_PIN = 7; // Pino ligado ao EN do A4988 (LOW para habilitar)

// Pinos de Microstepping (Para configurar 1/16 de passo no A4988)
// MS1=HIGH, MS2=HIGH, MS3=HIGH -> 1/16 Microstepping (3200 passos por volta)

const int MS1_PIN = 6;
const int MS2_PIN = 5;
const int MS3_PIN = 4;

// Pino do Botão
const int BUTTON_PIN = 2; // Pino ligado ao botão (usaremos INPUT_PULLUP)

// --- Parâmetros de Movimento e Torque ---

// O motor precisa de 3200 passos (pulsos) para uma volta completa (360°)
// Portanto, para 90 graus (1/4 de volta): 3200 / 4 = 800 passos
const int STEPS_FOR_90_DEG = 800;

// O atraso entre os pulsos define a velocidade e o torque.
// Um valor MAIOR (mais lento) aumenta o torque e a precisão.
// 2000 us (2ms) é um valor lento e seguro para alto torque e precisão.
const int stepDelayMicroseconds = 2000;

// --- Variáveis de Estado do Botão ---
int lastButtonState = HIGH; // Último estado lido do botão (HIGH porque usamos
INPUT_PULLUP)

// -----
// SETUP
// -----
void setup() {
    // Configurações de Pinos
    pinMode(STEP_PIN, OUTPUT);
    pinMode(DIR_PIN, OUTPUT);
    pinMode(ENABLE_PIN, OUTPUT);

    pinMode(MS1_PIN, OUTPUT);
    pinMode(MS2_PIN, OUTPUT);
    pinMode(MS3_PIN, OUTPUT);

    // Configura o pino do botão como entrada com resistor pull-up interno.
    // Isso significa que o pino estará HIGH (5V) quando o botão estiver solto,

```



**SÃO PAULO**  
GOVERNO DO ESTADO  
SÃO PAULO SÃO TODOS

***Etec Jacinto Ferreira de Sá - 066 – Ourinhos***

```
// e LOW (GND) quando o botão for pressionado.
pinMode(BUTTON_PIN, INPUT_PULLUP);

// --- Configuração do Microstepping (1/16) ---
// HIGH, HIGH, HIGH configura o A4988 para 1/16 de passo (3200 passos/volta).
digitalWrite(MS1_PIN, HIGH);
digitalWrite(MS2_PIN, HIGH);
digitalWrite(MS3_PIN, HIGH);

// Habilita o driver (LOW = Habilitado)
digitalWrite(ENABLE_PIN, LOW);

// Define uma direção INICIAL e FIXA (não será mais alterada no loop)
// HIGH para uma direção, LOW para a direção oposta.
digitalWrite(DIR_PIN, HIGH);
}

// -----
// FUNÇÃO PARA MOVER O MOTOR
// -----
void moveStepper(int steps, long stepDelay) {
  for (int i = 0; i < steps; i++) {
    // 1. Envia o pulso de passo (HIGH)
    digitalWrite(STEP_PIN, HIGH);

    // 2. Atraso para determinar a velocidade (LOW speed = HIGH torque)
    delayMicroseconds(stepDelay);

    // 3. Retira o pulso (LOW)
    digitalWrite(STEP_PIN, LOW);

    // 4. Atraso obrigatório. Este tempo é crucial para o driver processar o pulso.
    delayMicroseconds(stepDelay);
  }
}

// -----
// LOOP PRINCIPAL
// -----
void loop() {
  // 1. Lê o estado atual do botão
  int currentButtonState = digitalRead(BUTTON_PIN);

  // 2. Detecta a transição de HIGH (solto) para LOW (pressionado)
  // Isso garante que a ação ocorra APENAS no momento do clique (detecção de borda).
  if (currentButtonState == LOW && lastButtonState == HIGH) {

    // --- Lógica de Movimento de 90° ---
```



### ***Etec Jacinto Ferreira de Sá - 066 – Ourinhos***

// A direção é FIXA, então a lógica de inversão foi removida.

```
// Chama a função de movimento com os parâmetros calculados
  moveStepper(STEPS_FOR_90_DEG, stepDelayMicroseconds);
}
```

```
// 3. Salva o estado atual para a próxima iteração
lastButtonState = currentButtonState;
```

```
// Pequeno atraso para "debounce" (evitar múltiplos cliques devido a ruído)
delay(5);
}
```

Este código serve para controlar um motor de passo, fazendo com que ele gire exatamente 90 graus em um sentido fixo toda vez que um botão for pressionado. Ele é projetado para ser usado com um driver A4988, que é configurado em modo de 1/16 de microstepping (3200 passos por volta) para garantir alta precisão, por isso calcula 1600 passos para o movimento de 180°. O código também define uma velocidade de rotação lenta para assegurar um alto torque, evitando que o motor perca passos. A lógica principal detecta apenas o momento do clique (transição de solto para pressionado), garantindo que o motor gire 90° apenas uma vez por toque, e não continuamente se o botão for mantido pressionado.

## **4.2 Atuador Linear Com Rosqueador De Tampas**

```
// ----- PARÂMETROS DO MOTOR DE PASSOS -----
int stepDelayMicros = 800; // ajuste conforme necessário
```

```
// Definição de sentidos (ajuste se seu motor girar invertido)
// Vamos considerar:
// - true = DESCER (sentido horário, por exemplo)
// - false = SUBIR (sentido anti-horário, por exemplo)
const bool DIR_DOWN = true;
const bool DIR_UP = false;
```

```
// ----- VARIÁVEIS DE ESTADO -----
bool lastButtonState = HIGH; // para detectar borda de descida no botão
```

```
void singleStep() {
  digitalWrite(STEP_PIN, HIGH);
  delayMicroseconds(stepDelayMicros);
  digitalWrite(STEP_PIN, LOW);
  delayMicroseconds(stepDelayMicros);
}
```

**Etec Jacinto Ferreira de Sá - 066 – Ourinhos**

```

}

void moveUntilLimit(bool direction, int limitPin) {
    digitalWrite(DIR_PIN, direction);

    // Como usamos INPUT_PULLUP:
    // - HIGH = não acionado
    // - LOW = acionado
    while (digitalRead(limitPin) == LOW) {
        singleStep();
    }
}

void moveSteps(long steps, bool direction, bool useLimit = false, int limitPin = -1) {
    digitalWrite(DIR_PIN, direction);

    for (long i = 0; i < steps; i++) {
        if (useLimit && limitPin != -1) {
            if (digitalRead(limitPin) == HIGH) {
                // Fim de curso acionado, interrompe movimento
                break;
            }
        }
        singleStep();
    }
}

void executaCiclo() {
    const long TOTAL_PASSOS_DESCIDA = 2000;
    const long PASSOS_LIGA_DC = 2000;
    const unsigned long TEMPO_DC_EXTRA = 1500; // 3 segundos em ms

    bool dcLigado = false;

    // Garante que o motor DC esteja desligado no início do ciclo
    digitalWrite(DC_MOTOR_PIN, HIGH);

    // ----- DESCER 2500 PASSOS -----
    digitalWrite(DIR_PIN, DIR_DOWN); // DIR_DOWN DIR_UP

    for (long passo = 1; passo <= TOTAL_PASSOS_DESCIDA; passo++) {
        // Segurança: se fim de curso inferior for acionado, para a descida
        if (digitalRead(LIMIT_BOTTOM_PIN) == HIGH) {
            Serial.println("");
            Serial.println("");
            Serial.print("LIMIT_BOTTOM_PIN = ");
            Serial.println(digitalRead(LIMIT_BOTTOM_PIN));
            break;
        }
    }
}

```



**Etec Jacinto Ferreira de Sá - 066 – Ourinhos**

```

singleStep();

// Liga o motor DC quando atingir 2000 passos
if (!dcLigado && passo >= PASSOS_LIGA_DC) {
    digitalWrite(DC_MOTOR_PIN, LOW);
    dcLigado = true;
}
}

// Mantém o motor DC ligado por mais 3 segundos após a descida
delay(TEMPO_DC_EXTRA);
digitalWrite(DC_MOTOR_PIN, HIGH);

// ----- SUBIR ATÉ O PONTO ZERO (FIM DE CURSO SUPERIOR) -----
moveUntilLimit(DIR_UP, LIMIT_TOP_PIN);

moveSteps(200, DIR_DOWN, true, LIMIT_BOTTOM_PIN);
Serial.println("Posicao inicial: 200 passos abaixo do zero.");

}

```

O sistema de automação é concebido para executar um ciclo de trabalho preciso coordenando um motor de passos com um motor de corrente contínua dc. o motor de passos acionado por um driver a4988 é responsável pela movimentação linear ou vertical do atuador com sua velocidade definida pelo intervalo de tempo *stepdelaymicros* entre os pulsos de passo enquanto a direção é controlada pelos estados lógicos *dir up* e *dir down*. a segurança e o referenciamento absoluto da posição são cruciais sendo garantidos pela utilização de dois fins de curso *limit top pin* e *limit bottom pin* onde o *limit top pin* é utilizado para executar o homing na inicialização do sistema *setup* definindo o ponto zero para todos os ciclos. a lógica principal do sistema é encapsulada na função *executaciclo* que estabelece a sequência de trabalho o ciclo é iniciado por um botão *button pin* com detecção de borda e começa com uma descida controlada de 2000 passos *total passos descida*. ao atingir o marco predefinido de 2000 passos *passos liga dc* o motor dc *dc motor pin* é ativado sincronizando a movimentação com uma tarefa secundária sendo mantido ligado por um tempo adicional *tempo dc extra* para finalização do processo. finalmente o ciclo é concluído com o retorno seguro do motor de passos ao ponto zero *limit top pin* utilizando a função *moveuntillimit* para garantir que o sistema esteja corretamente posicionado para o próximo comando.



*Etec Jacinto Ferreira de Sá - 066 – Ourinhos*

## **5. METODOLOGIA**

O desenvolvimento deste projeto foi conduzido por meio de uma abordagem de prototipagem experimental, dividida em quatro fases principais: fundamentação teórica e seleção de componentes, montagem do protótipo, desenvolvimento de software e validação funcional. Para a construção do protótipo, foram utilizados os seguintes recursos de hardware e software: Placa microcontroladora Arduino MEGA 2560 R3, sensores de fins de curso, sensores ultrassônicos, motores nema 17 (motor de passos). Ambiente de Desenvolvimento Integrado (IDE) do Arduino, versão Arduino IDE 2.3 Recursos Instrucionais: Documentação técnica oficial disponível no portal Arduino com e material audiovisual o material audiovisual foi passado pelo o orientador nesse vídeo mostra um rosqueador de tampa em miniatura isso serviu de base para a montagem do protótipo. A primeira etapa consistiu na apresentação dos componentes e na explanação teórica da arquitetura do sistema. O vídeo instruiu o diagrama esquemático e as conexões necessárias para a montagem. A segunda etapa envolveu a montagem física do circuito eletrônico, seguindo as especificações técnicas e o diagrama predefinido. A programação do microcontrolador foi realizada na linguagem C/C++ padrão do Arduino. Para a validação do código e a correção de erros (depuração/bugs), foi empregada a ferramenta de assistência baseada em Inteligência Artificial nativa da plataforma Arduino. O processo de correção foi iterativo: o código era compilado e testado; em caso de falhas ou comportamento inesperado, a ferramenta de IA era consultada para analisar o código, identificar erros de sintaxe ou lógica, e sugerir otimizações. Esta abordagem visou acelerar o ciclo de desenvolvimento e garantir a robustez do software embarcado.

## **6. CONSIDERAÇÃO FINAIS.**

O desenvolvimento deste projeto demonstrou a viabilidade técnica e econômica da automação de processos de vedação em linhas de produção de pequena escala. A integração entre a modelagem mecânica via impressão 3D, o controle eletrônico por meio da plataforma Arduino e a utilização combinada de motores de passo e DC permitiu a criação de um protótipo funcional, capaz de sanar as deficiências inerentes ao rosqueamento manual. Os resultados obtidos validaram a hipótese inicial de que é



**SÃO PAULO**  
GOVERNO DO ESTADO  
SÃO PAULO SÃO TODOS

***Etec Jacinto Ferreira de Sá - 066 – Ourinhos***

possível garantir a padronização do torque e a integridade da vedação com baixo investimento financeiro. O sistema de posicionamento rotacional mostrou-se eficaz na sincronia com o atuador linear, garantindo que o processo ocorresse apenas quando a garrafa estivesse devidamente alinhada, mitigando desperdícios e riscos de acidentes. A escolha por baixas velocidades no motor de passo, em detrimento da rapidez excessiva, provou ser uma estratégia acertada para manter o torque elevado e a precisão do posicionamento sem a necessidade de drivers industriais de alto custo. Além do aspecto técnico, este trabalho reforça a importância da interdisciplinaridade no Ensino Médio Integrado ao Técnico, unindo conhecimentos de mecânica, eletrônica e programação para resolver problemas reais da indústria. A solução apresentada não apenas oferece uma alternativa acessível para pequenas empresas aumentarem sua competitividade e segurança, mas também serve como base para estudos futuros.

## **7. REFERÊNCIAS BIBLIOGRÁFICAS**

- BANZI, Massimo. Primeiros Passos com o Arduino. 1. ed. São Paulo: Novatec, 2011.
- GROOVER, Mikell P. Automação Industrial e Sistemas de Manufatura. 3. ed. São Paulo: Pearson Prentice Hall, 2011.
- THOMAZINI, Daniel; ALBUQUERQUE, Pedro Urbano B. Sensores Industriais: Fundamentos e Aplicações. 8. ed. São Paulo: Érica, 2011.
- VOLPATO, Neri (Org.). Manufatura Aditiva: Tecnologias e Aplicações da Impressão 3D. São Paulo: Blucher, 2017.