

PROCESSO DE DOSAGEM MICROCONTROLADA EM LINHA

Alexandre Kuhn, Daniel Khede Dourado Villa

RESUMO

O presente trabalho apresenta a construção de um sistema de dosagem microcontrolado de substâncias químicas. Para isso, foi desenvolvido um método para leitura e medida da vazão de forma digital, assim como um processo para o acionamento de uma bomba dosadora por ação digital. Para integrar os dois sistemas, uma interface homem-máquina foi gerada armazenando dados em nuvem. Os resultados obtidos para a ação dosadora foram satisfatórios, com o sistema apresentando estabilidade e confiabilidade no processo de dosagem, bem como na comunicação com o servidor de nuvem.

1. INTRODUÇÃO

As atividades industriais infelizmente vem encolhendo sua representação no PIB do Brasil de 35,9% em 1985, 17,9% em 2004 e hoje 10,9% em 2014 (FIESP, 2015), onde são utilizados em larga escala sistemas de dosagem sejam esses automatizados ou não. Outro setor onde há larga demanda pela utilização de sistemas de dosagem é no setor de saneamento onde apenas 83,6% da população é atendido por água tratada, esse número aumenta para 92,8% em áreas urbanas (SNIS, 2018), e observa-se que as populações não atendidas são as mais isoladas. As duas situações expostas podem ser melhoradas diretamente com a utilização da Indústria 4.0 ou de controles em nuvem para controle da automação industrial (Santos, 2018).

Em processos industriais químicos, agrícolas, agroindustriais, de saneamento, entre outros, a dosagem de substâncias químicas é prática largamente utilizada. Bombas dosadoras são equipamentos capazes de dosar quantidades precisas e com variações capazes de serem ajustadas. Essas dosadoras estão presentes em diversas linhas de processo, e, comumente, são utilizadas bombas de êmbolo, pistão ou diafragma, além de outras bombas com volume controlado (STURARO, 2009).

Controladores lógicos programáveis, ou microcontroladores lógicos, são controladores que, adequadamente programados, dão “inteligência” ao sistema. Sua história se inicia na década de 1970 com plataformas INTEL e TEXAS INSTRUMENTS, e vem até os dias atuais, com a popularização de plataformas abertas como ARDUINO e RASPBERRY PI. Independente da plataforma, esses controladores são dotados de entradas e saídas tanto analógicas quanto

digitais, possibilitando a execução dos processos de forma independente, precisa, e com redução no consumo de energia (OLIVEIRA, 2017). As ações de controle podem ser por *feedback* ou *feedforward*, isso é, com métrica baseada no erro do processo ou no conhecimento prévio do projetista sobre o comportamento do processo. Em geral, ações de controle em *feedback* são mais estáveis do que as em *feedforward*, a não ser que o processo seja conhecido em profundidade (LUYBEN, 1989).

A possibilidade de comunicação entre microcontroladores e servidores de internet, vem desenvolvendo um novo conceito de tecnologia chamado de Internet das Coisas, do inglês, *Internet of Things* (IoT). Utilizando a internet e servidores capazes de serem operados remotamente, pode-se expandir esse conceito no que é chamado de computação em nuvem. Assim, microcontroladores ligados a uma rede ethernet, física ou WiFi, podem se conectar a servidores em nuvem, com os quais, por meio de protocolos específicos, é possível enviar e receber dados de um determinado processo (OLIVEIRA, 2017).

As possibilidades econômicas da aplicação de microcontroladores vinculados à internet ampliam significativamente a aplicação de tais instrumentos, possibilitando monitoramentos e ações remotas. A lógica empregada na programação de nuvem pode, inclusive, monitorar eventuais erros de sistema, como, por exemplo, o disparo de alarmes por falhas de comunicação, de forma que seja possível o deslocamento de um técnico ou operador ao campo, minimizando a mão de obra humana, mas não eliminando-a em locais remotos por nuvem e IoT.

Neste trabalho, objetiva-se (1) a construção de um sistema de dosagem de substâncias químicas por meio de uma bomba dosadora tipo diafragma com acionamento eletromagnético via bobina, (2) verificar a precisão na dosagem do sistema, (3) integrar o sistema de dosagem com informações em nuvem, inclusive (4) podendo atuar de maneira automatizada na dosagem e (5) com precisão em termos de concentração.

2. DESENVOLVIMENTO

O sistema de dosagem pode ser visto na Figura 1, onde é possível visualizar o experimento como um todo: o microcontrolador (MKR 1000), o sensor de vazão (hidrômetro e o contador de pulsos tipo encoder (detalhe na Figura 2)) e a bomba dosadora. Nas Figuras 2 e 3 é possível verificar o sistema de dosagem em detalhes, com a adaptação no hidrômetro, inserção de um ponteiro de relógio no hidrômetro, que ao medir 1 litro, faz o ponteiro passar pelo encoder de infravermelho, gerando um pulso que pode ser medido. Na Figura 3 há um desenho técnico do encoder, com especial detalhe para a passagem do ponteiro.

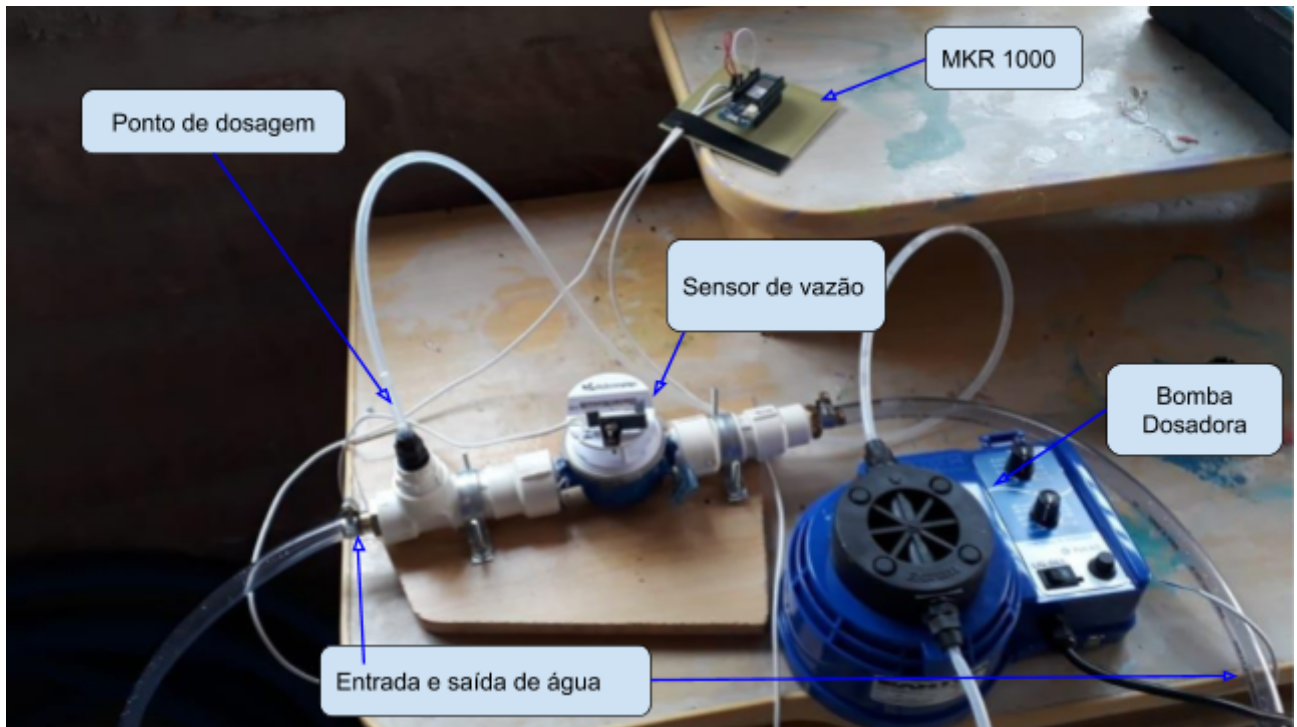


Figura 1: sistema de dosagem

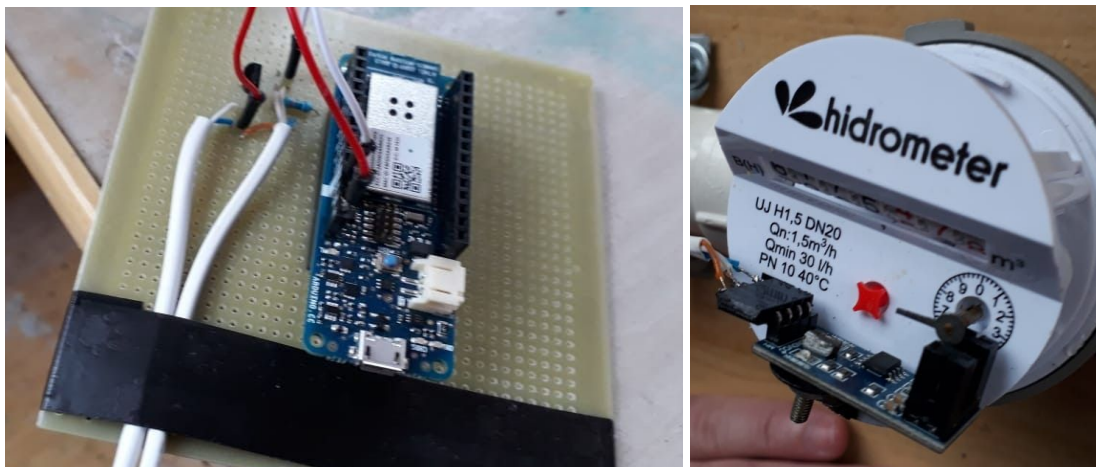


Figura 2: detalhes do sistema de medida de vazão

O sistema utiliza um mecanismo de medida de vazão e, a partir dessa vazão, é dosada uma vazão de solução em linha (Figura 1). Para isso, o sistema envia as informações de vazão para um servidor na nuvem, e esse terá os parâmetros de dosagem do produto em partes por milhão (PPM), enviando este parâmetro a um microcontrolador ARDUINO dotado de WiFi.

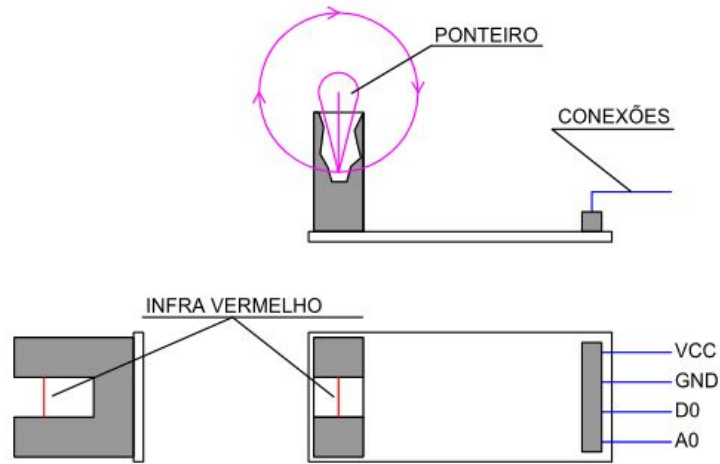


Figura 3: sensor de pulsos para medida de vazão

É possível observar a arquitetura do sistema de dosagem proposto na Figura 4, com o recebimento de água da rede, a leitura da vazão por medidor de vazão, a interpretação em um sistema lógico, no caso o ARDUINO, e integração com a nuvem, no caso o dashboard da marca BLYNK, sendo possível setar condições do processo e ler as condições do processo. A dosagem é feita em linha por uma bomba de diafragma.

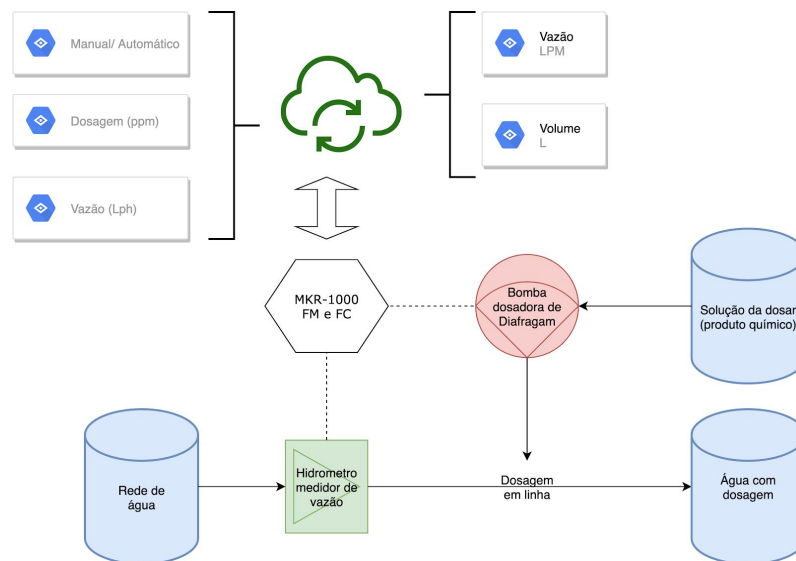


Figura 4 - Arquitetura do sistema dosador

As subseções a seguir apresentam em detalhes cada etapa do projeto.

2.1. MONTAGEM DO SISTEMA DOSADOR

Para realização das ações que controlam a dosagem, foi utilizada uma placa [ARDUINO MKR-1000](#) onde há um microcontrolador integrado nativamente com uma placa de WiFi, e, como todos os módulos da família MKR, esses são capazes de aplicação em internet das coisas. O código usado na programação do módulo se encontra anexada ao final deste artigo. A tensão de alimentação é de 5V, com no mínimo 700mA, e as portas de entrada e saída têm tensão de 3,3V e 20 mA, respectivamente.

Para dosagem, foi utilizada uma bomba dosadora de diafragma com acionamento por bobina elétrica da marca [EXXATA EX1DPlusAV1002](#). Essa é comercializada para dosar de 0 a 10 L/h utilizando pulsos, em que quanto maior a frequência dos pulsos, maior a vazão. O controle, da versão comercial, dos pulsos dessa bomba era feito com um resistor tipo potenciômetro, que variava a corrente de alimentação de um capacitor, em DC, de forma que a corrente era proporcional a frequência de pulsos gerada pela bomba. Esse sistema foi adaptado para que a geração de pulsos fosse feita pelo ARDUINO. Para isso, foi instalado um opto-controlador que é acionado com corrente de 5V, gerando os pulsos que são amplificados antes de chegar ao ARDUINO por um transistor pnp.

Como sensor de vazão, foi feita a adaptação de um hidrômetro comumente utilizado em residências com selo de calibração do INMetro. No ponteiro indicador, onde há a contagem de litros, foi adaptado um ponteiro de modo a possibilitar a instalação de um sensor de infravermelho do tipo [MOCH22A](#), capaz de detectar a passagem do ponteiro pelo infravermelho, gerando um pulso que é contado pela placa ARDUINO (vide Figuras 2 e 3).

2.2. LÓGICA DE PROGRAMAÇÃO

Para implementação do sistema de dosagem de substâncias químicas, o microcontrolador realiza a leitura da vazão e calcula, utilizando a equação em (1), a frequência da bomba dosadora (f) a fim de cumprir com o setpoint da concentração (C) estabelecido na nuvem; a vazão (Q) é medida pelo sistema e o fator 60.000 é um fator de ajuste relativo às unidades. O sistema permite ainda setar a vazão da bomba dosadora na condição “manual”. A hospedagem do sistema de controle foi realizada no aplicativo para a dashboard Blynk. Nesse aplicativo também são expostos dos valores de vazão no momento e do volume acumulado durante um período de 24 horas.

$$f [Hz] = Q [LPM].C[ppm].60.000 \quad (1)$$

Quanto ao tratamento dos sinais, tanto os sinais de entrada no contador de litros, como na saída para a geração e manutenção dos pulsos da bobina, foi necessário o uso de processos de *anti-bouncing*. O efeito de *bouncing*, ou de sinalização falsa, é um efeito comum no uso de sensores microcontrolados, como usados em sistemas ARDUINO, muito em função da sensibilidade desses equipamentos. Para o condicionamento desses sinais falsos, é possível utilizar algoritmos de tratamento e álgebra booleana, criando condições de falso e verdadeiro para que um sinal seja aceito e, após atendido o filtro lógico, considerado pelo microcontrolador. No código em anexo, é possível verificar a implementação desse filtro lógico, aplicado à contagem do pulso de sinal do ponteiro do hidrômetro. No algoritmo, a contagem de “PULL_UP” somente ocorre se o sensor estiver de fato em “PULL_DOWN”.

A respeito dos pulsos de saída da bomba, o tempo de leitura do microcontrolador ARDUINO MKR-1000 é da ordem de nanosegundos, e o tempo para que o sistema mecânico do diafragma esteja pulsando é de no mínimo 200 milissegundos. Para relacionar a necessidade mecânica do processo com a velocidade dos processamentos é necessário inserir uma “pausa”.

Ocorre que o comando “*delay*” pausaria todo o processamento, e no programa utilizado no projeto existem outros processos que ocorrem em paralelo, como a leitura de pulsos para a medida de vazão, o cálculo de vazão ou de volume acumulado, ou mesmo a comunicação com a nuvem. Assim, a utilização deste comando tornaria instável o processo como um todo.

Dessa maneira, os comandos de pausa, que possibilitam o acionamento da bobina da bomba (processo de pulsar), foram realizados por dois comandos “SE” atuando em “PULL UP” (acionado quando igual a 1) e “PULL DOWN” (não acionado quando igual a 0). O processo de pulsar ocorre mediado pela contagem do tempo, em específico usando o contador de tempo “*millis()*”. Ademais, como controle auxiliar foram introduzidas variáveis booleanas que impedem que o processo “PULL UP” ocorra juntamente com o sistema em “PULL DOWN” (MOURA, 2017).

2.3. ENSAIOS PARA VALIDAÇÃO DO SISTEMA PROJETADO

Para a validação da medida de vazão, foi implementado o algoritmo apresentado na Figura 5. Esse algoritmo recebe a contagem de pulsos, realiza a transformação da variável vazão, Q , utilizando (1), e a compara com a quantidade medida durante o período pelo hidrômetro. Como no algoritmo também há um quantificador de passagem de tempo, “*millis()*”, esse pôde ser comparado utilizando um cronômetro convencional.

O volume de água acumulada para cada teste foi de 50L, e, assim como o tempo contado pelo cronômetro do MKR-1000, os dados dos testes foram impressos no terminal.

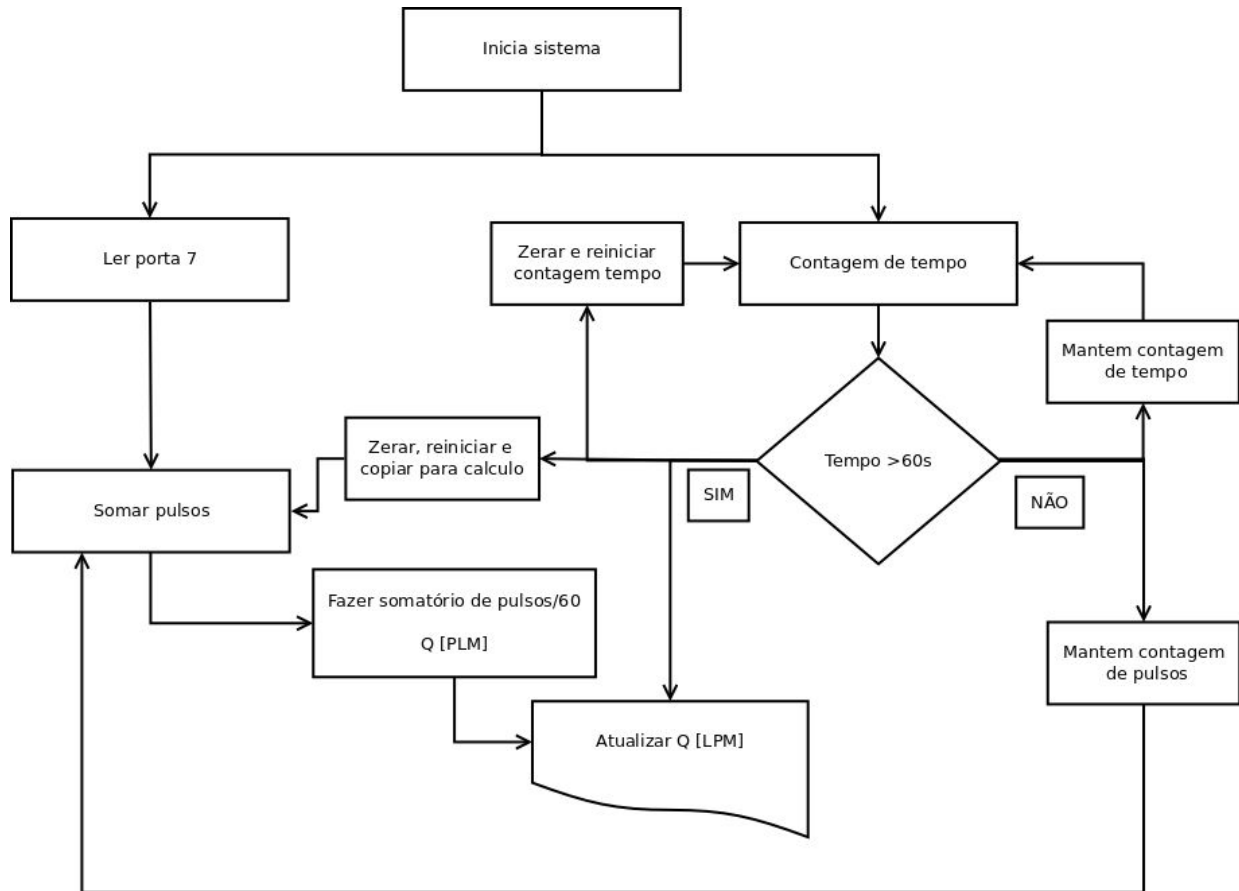


Figura 5: algoritmo de medida de vazão

Para validação do sistema de dosagem, foram comparados os pulsos de acionamento e a vazão gerada pela ação da bomba. A frequência máxima da bomba é de 2 pulsos por segundo (2 Hz), com a vazão sendo diretamente proporcional à frequência dos pulsos. Neste trabalho, utilizaremos como frequências de trabalho o intervalo entre 0,2 e 2 Hz. Durante a validação, a bomba foi *setada* em frequências aleatórias dentro das frequências de trabalho, sendo realizadas medidas de vazão para serem comparadas com a vazão determinada pelo fabricante. As medidas de vazão para validação foram feitas utilizando o cronômetro interno do MKR-1000 e uma proveta.

Na Figura 6 está representado o algoritmo de acionamento da bomba dosadora, onde é possível verificar a condição booleana impedindo que o sinal de acionamento cesse antes de transcorrido o tempo de atraso, sendo este setado em função do processo .

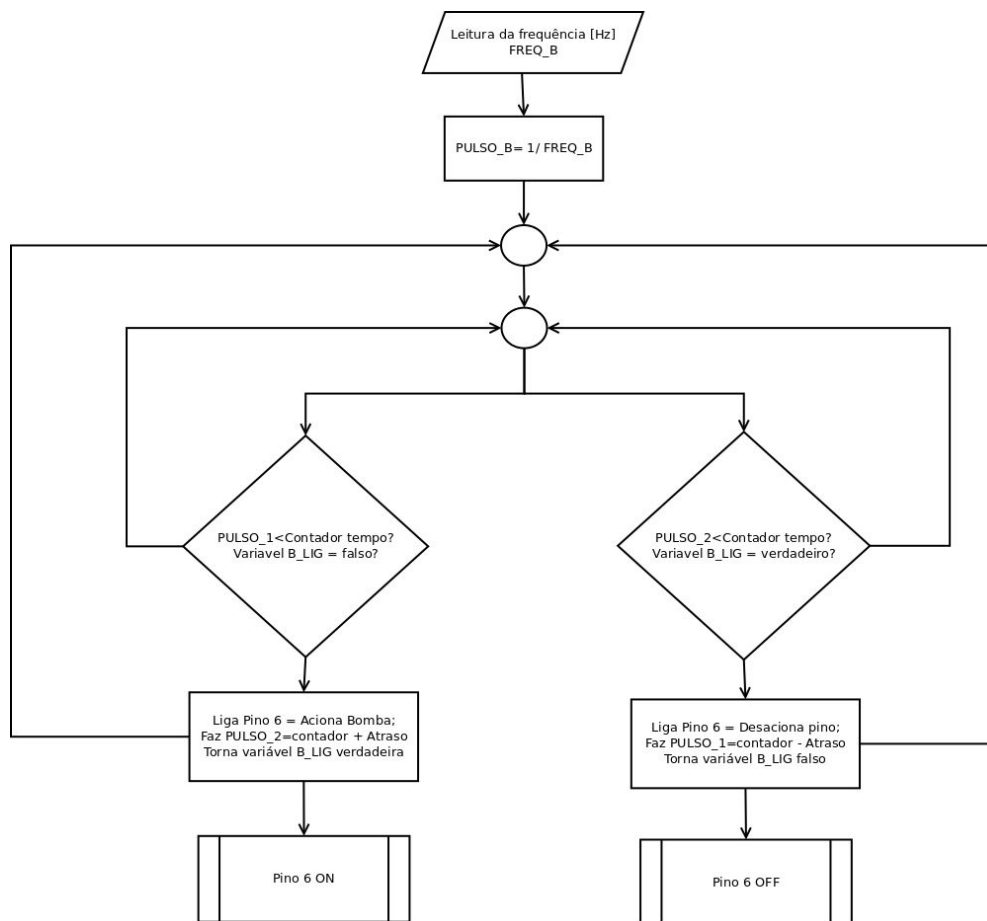


Figura 6: diagrama do algoritmo de acionamento da bomba dosadora

Para integrar os sistemas de medida da vazão com o da bomba, foi implementado um algoritmo, apresentado na Figura 7, capaz de, além de integrá-los, permitir o controle manual pelo operador. Operando no modo automático (não manual), o sistema estabelece a vazão de dosagem em função do setpoint de dosagem, em partes por milhão (PPM) conforme em (1), e da vazão medida.

Como dito anteriormente, existe também a possibilidade de regulagem manual da vazão da bomba dosadora. Na operação manual, a frequência é transformada em frequência para o acionamento da bomba de diafragma, como descrito em (2). Onde a frequência de pulso da bomba (f) é função da vazão (Q) e o fator 5,94 é um fator de ajuste obtido experimentalmente conforme se observa na tabela 2.

$$f[Hz] = Q[LPH].5,94 \quad (2)$$

Para a validação do sistema automático, foi dosada uma solução de cloreto de sódio (sal de cozinha) na concentração de 100g/L, a fim de se verificar a concentração de sal na mistura após a dosagem.

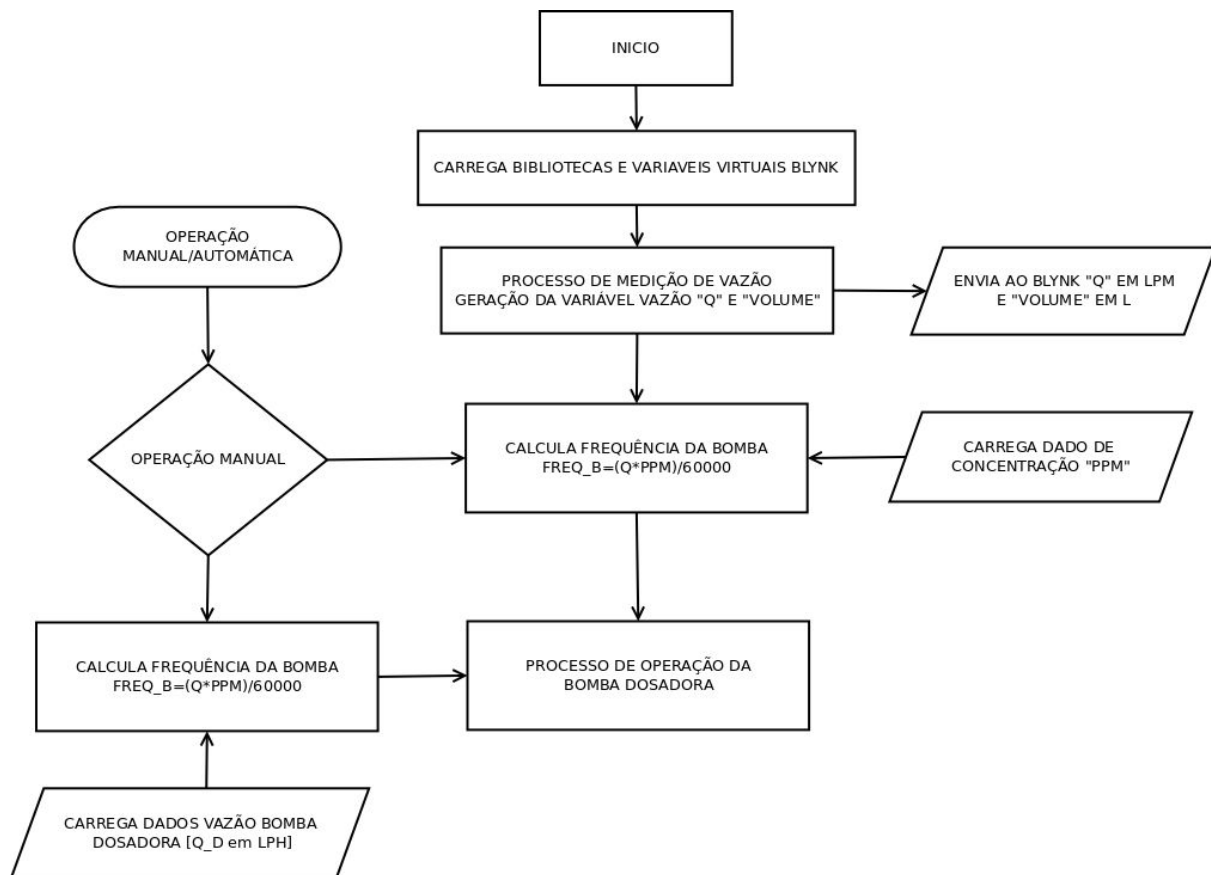


Figura 7: algoritmo de funcionamento da bomba dosadora integrado com a medida de vazão

2.4. INTEGRAÇÃO COM SISTEMA DE NUVEM

A integração da dashboard Blynk em nuvem com o ARDUINO MKR-1000 foi feita utilizando sua integração WiFi nativa, de forma que apenas foram utilizadas bibliotecas de integração entre as plataformas via internet. Nessa integração, tornou-se possível unir as informações e variáveis do microcontrolador com as variáveis no servidor em nuvem. Além disso, pelo ambiente criado é possível ler e enviar informações diretamente ao microcontrolador.

Nessa aplicação, a biblioteca utilizada foi a biblioteca "Blynk versão 0.6.0", existente no banco de dados do ARDUINO, bem como as bibliotecas específicas de WiFi do Arduino. Para estabelecer a comunicação entre os dispositivos, é necessário que sejam utilizados os dados de comunicação da rede à qual se está conectado e um token específico gerado pelo site do dashboard Blynk.

3. RESULTADOS E DISCUSSÃO

A seguir estão apresentados os resultados para as validações realizadas. No experimento para validar a medida da vazão, é possível verificar na Tabela 1 que os valores calculados são muito próximos dos valores esperados, em que as diferenças apresentadas podem ter ocorrido em função de imprecisões nas medidas.

Tabela 1: calibração da leitura de vazão em função do tempo e do volume

tempo [s]	Volume [L]		Q [LPM]	
	Esperado	Medido	Esperado	Medido
1473,0	25,0	24,5	1,0	0,99
510,0	25,0	25,3	3,0	2,98
258,0	25,0	24,8	6,0	5,77
182,8	25,0	25,7	8,0	8,44
120,0	25,0	25,0	12,0	12,5

Já no experimento para validar o sistema de dosagem, foi verificada a linearidade e a correta calibração para a operação da bomba dosadora, submetida a diferentes frequências. Os dados da linearização estão apresentados na Tabela 2, e o resultado pode ser verificado na Figura 8, onde é possível perceber a linearidade da resposta, bem como o bom ajuste da equação.

Tabela 2: frequência de pulso da bomba de diafragma e a vazão

Freq (Hz)	tempo (s)	Volume	Q (LPH)
2	50	165	11,88
1,5	50	120	8,64
1	50	85	6,12
0,5	50	40	2,88
0,2	50	15	1,08

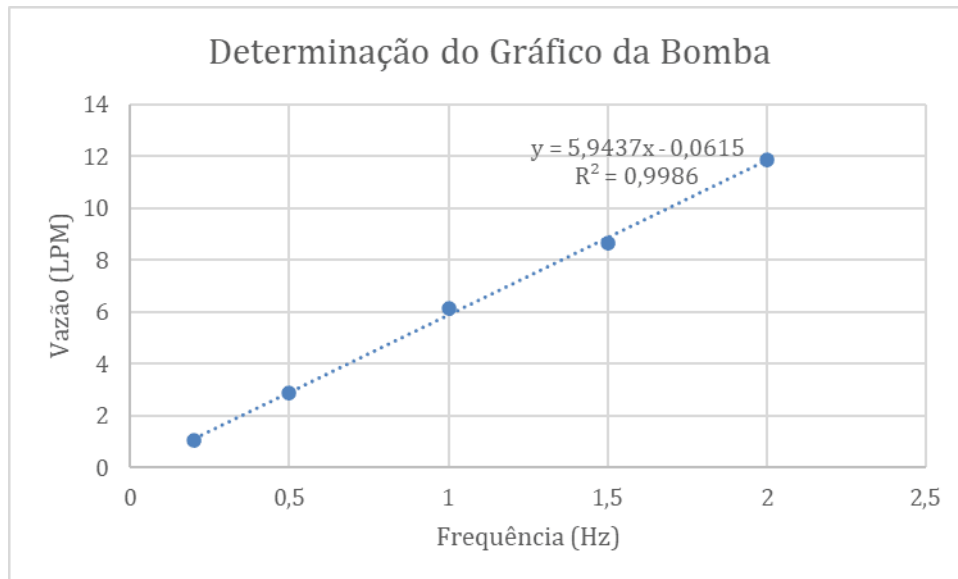


Figura 8: vazão em função de pulsos e linearidade em bomba de diafragma

Por fim, a integração dos sistemas de vazão e dosagem foi validada usando os testes com a solução concentrada de cloreto de sódio, escolhido em função da facilidade de medida e do fato que o produto químico não se degrada, com concentrações de 100g/L foram testados vazões de 3, 6 e 9 (LPM) litros por minuto, e dosagens de 5.000, 7.500 e 10.000, como pode ser observado na Tabela 3. Foi estimada a concentração de cloreto de sódio após a dosagem e comparado com o valor real, medido pelo método de análise de cloretos. Nesse teste, o sistema foi operado no modo automático. A diferença observada entre as colunas “dosagem-setada” e “dosagem-estimada” ocorre em função do efeito de diluição pois a solução a ser dosada tem concentração inferior a 100%. Quanto a diferença entre as colunas “dosagem-estimada” e “dosagem-medida” se observa o efeito da capacidade de dosagem do sistema.

Avaliando os resultados, é possível perceber que as medidas de cloretos verificadas são superiores estimadas. Isso provavelmente ocorreu pois a concentração comum de cloreto de sódio em água para consumo oscila entre 5 e 20 ppm.

Durante os experimentos, a comunicação entre dashboard e ARDUINO foi efetiva e ocorreu sem problemas. Uma imagem instantânea da interface gráfica durante a operação do sistema pode ser vista na Figura 9. No sistema implementado não há acúmulo de dados, porém existem servidores em nuvem que possibilitaram tais ações.

Tabela 3: comparação da dosagem real e da dosagem estimada

Vazão		Frequência	Dosagem		
Rede	Dosadora		Setada	Estimada	Medida
LPM	LPH	Hz	PPM	PPM	PPM
3	1,485	0,25	5000	825	820
6	2,97	0,5	5000	825	815
9	4,455	0,75	5000	825	829
3	2,2275	0,375	7500	1237,5	1238
6	4,455	0,75	7500	1237,5	1235
9	6,6825	1,125	7500	1237,5	1234
3	2,97	0,5	10000	1650	1700
6	5,94	1	10000	1650	1620
9	8,91	1,5	10000	1650	1655



Figura 9: dashboard em operação

4. CONCLUSÃO

Neste trabalho, pôde-se mostrar que o sistema construído foi capaz de realizar a dosagem autônoma de substâncias químicas utilizando os microcontroladores, com a dosagem apresentando uma precisão adequada conforme os requisitos técnicos e de operação.

O sistema de dosagem apresentou integração com a nuvem, permitindo sua atuação de forma automatizada, bem como uma interface com o usuário via smartphone para facilitar sua operação.

O sistema proposto pode colaborar com a aplicação de sistemas vinculados à indústria 4.0, contribuindo com a competitividade dos processos industriais e também com a implantação de sistemas de saneamento em localidades remotas, uma vez que essa é a principal característica do sistema testado.

Em trabalhos futuros, espera-se replicar o resultado em diversas outras atividades do ramo industrial, como por exemplo:

- preparação de misturas e massas para alimentos;
- processos de blendagem em linha;
- processos de dosagem de produtos químicos em processos diversos em linha;
- mistura de componentes para extrusão;
- misturas para síntese de termofixos, dentre outros.

5. REFERÊNCIAS BIBLIOGRÁFICAS

PARSEKIAN, M. P. S. Análise e proposta de forma de gerenciamento de estações de tratamento de águas de abastecimento completo em cidades porte médio no Estado de São Paulo. Dissertação de Mestrado. USP São carlos. 1998;

STURARO, A.Z. Automação de bombas dosadoras com auxílio de controlador lógico programável. Trabalho de Conclusão de Curso. Universidade São Francisco. Campinas. 2009.

LUYBEN, W.L. Process, modeling, simulation and control for chemical engineer. Mc Gran Hill, NY. 725p. 1989.

OLIVEIRA, S. Internet das coisas com ESP8266, ARDUINO e RASPBERRY PI. Novatec Editora, 255p. São Paulo, SP. 2017.

MOURA, C.C. Câmara térmica microcontrolada. Trabalho de Conclusão de Curso. UFTPR. Curitiba, PR, 2017.

SANTOS, B. P.; ALBERTO, A; LIMA T.D.F.; CHARRUA-SANTOS, F.M.B. Indústria 4.0: Desafios e Oportunidades. Revista Produção e Desenvolvimento, v.4; n.4; p 111 - 124, 2018.

SNIS, Sistema Nacional de informações sobre Saneamento, 24 Diagnóstico de Serviços de Água e Esgoto, Brasília, DF, 2018.

FIESP Federação das Indústrias do Estado de São Paulo. Perda de Participação da Indústria de transformação no PIB. São Paulo, SP, maio/2015.

BLYNK. Blynk IoT plataforma for business and developers. <<https://blynk.io>> . Acessado em 20.08.2020.

ANEXO PROGRAMA:

```

/*****
Programação de buscar Dash Blynk
*****/

#define BLYNK_PRINT Serial      //DETERMINAÇÃO DO USO DA SERIAL BLINK

#include <SPI.h>                 //INCLUIR SPI-SERIA PERIPHERAL INTERFACE

#include <WiFi101.h>            //INCLUIR BIBLIOTECA DE WIFI

#include <BlynkSimpleMKR1000.h> //INCLUI USO DE BIBLIOTECA BLINK

/*****INCLUSÃO DO TOKEN GERADO PELO BLYNK*****/

char auth[] = "qYiauomCc_oN3g1TiakB2QiYTFp6Qdo_";

char ssid[] = "KE";             //DIGITAR NOME DA REDE

char pass[] = "92781243";       // DIGITAR SENHA

/*****DETERMINAR PINOS*****/

#define PULSO 7                 //DETERMINAÇÃO DO PINO PARA ENTRADA (PULSO)

#define BOMBA 6                 //DETERMINAÇÃO DO PINO PARA SAÍDA (BOMBA)

#define ATRASO 200              //DETERMINAÇÃO DO DELAY (ATRASO)

/*****DETERMINAÇÃO DE VARIÁVEIS - MEDIR VAZÃO*****/

float Q = 0;                    //VARIÁVEL DE VAZÃO (LITROS POR MINUTO - LPM)

float Q_1 = 0;                  //CONTADOR DE PULSOS PARA DETERMINAÇÃO DE VAZÃO

```

```

float VOLUME = 0;           //VOLUME ACUMULADO DE ÁGUA (LITROS)

unsigned long FREQ = 60000; //DETERMINAÇÃO DO INTERVALO DE MEDIÇÃO (mS)

unsigned long DIA = 0;      //VALOR INICIAL DA VARIÁVEL DIA

unsigned long FREQ_1 = 60000; //TEMPO DE FREQUENCIA DE MEDIÇÃO EM MILISEGUNDOS

unsigned long FREQ_P = 100; //TEMPO DE ESPERA PARA PRINT

bool flag = 0;             //DETERMINAR A VARIÁVEL FLAG, VARIÁVEL BOOLEANA

/*****DETERMINAÇÃO DE VARIÁVEIS - DETERMINAR FREQUÊNCIA DA BOMBA*****/

int OP = 0;                //0=MANUAL E 1=AUTOMÁTICA

float Q_D = 1000;          //VAZÃO DA BOMBA EM MANUAL

float FREQ_B = 1;          //TEMPO DE FREQUENCIA DA BOMBA (Hz)

unsigned long PPM = 100;   //DETERMINAÇÃO DA DOSAGEM EM (ppm)

/*****DETERMINAÇÃO DE VARIÁVEIS - ACIONAMENTO DE BOMBA*****/

float PULSO_B;            //COMANDO QUE GERA O PULSO NA BOMBA

float PULSO_1;            //COMANDO QUE PULSA PO PULL-UP

float PULSO_2;            //COMANDO QUE PULSA O PULL-DOWN

bool B_LIG=false;        //VARIÁVEL BOOLEANA IMPEDIR "LIGUE LIGADO"

BLYNK_WRITE(V2) //DETERMINAÇÃO DE VARIÁVEL VIRTUAL DO BLYNK (PPM) DOSAGEM

{

PPM = param.asInt();

}

```



```

BLYNK_WRITE(V3) //DETERMINAÇÃO VAR VIRTUAL DO BLYNK (Q_D)DOSADORA MANUAL
{
Q_D = param.asInt();
}

BLYNK_WRITE(V4) //DETERMINAÇÃO VAR VIRTUAL DO BLYNK (OP)DOSADORA MANUAL
{
OP = param.asInt();
}

void setup()          //CARREGA ARQUIVO DE INICIALIZAÇÃO
{
/***** USAR O SERIAL *****/
Serial.begin(9600);    //CONFIGURAÇÃO DO SERIAL

/*****DEFINE ENTRADA DE DADOS*****/
pinMode (PULSO, INPUT_PULLUP);    //LEITURA DO MEDIDOR DE VAZÃO

/*****DEFINE A SAÍDA DE PULSOS*****/
pinMode (BOMBA, OUTPUT);    //SELEÇÃO DO PINO DE OUTPUT
digitalWrite (BOMBA, LOW);    //SISTEMA INICIA COM O PINO "LOW"

PULSO_1=0;    //FAZ PULSO_1 SER MENOR QUE MILLIS

/*****INICIAR BLYNK*****/
Blynk.begin(auth, ssid, pass);    //ATIVA
}          //FINALIZA LAÇO SETUP

void loop()          //CARREGA ARQUIVO DE OPERAÇÃO
{
/*****INICIAR BLYNK*****/

```

```

Blynk.virtualWrite(V5, String(VOLUME));      //VOLUME EM LITROS

Blynk.virtualWrite(V6, String(Q));          //Q EM LPM

Blynk.run();                                //FAZER BLYNK FUNCIONAR

/*****CALCULO DA VAZAO DE TERMINAÇÃO DOS PULSOS PARA CONTAGEM*****/

if (digitalRead(PULSO))flag = 1; //VERIFICAÇÃO DE flag É POSITIVA
if (!digitalRead(PULSO)&&flag) //VERIFICAÇÃO DE flag É NEGATIVA
{
Q_1 = (Q_1 + 1); //DEFINE O VOLUME DE CONTROLE
VOLUME=(VOLUME+1); //DEFINE O VOLUME DIÁRIO COMO
flag=0; //TORNA flag NEGATIVO, PARA REINICIAR
} //FECHA CONTAGEM DE PULSOS

DIA = millis(); //DEFINE O TEMPO DE UM PULSO;

if (DIA > 86400000) //DELINE O LIMITE DE 1 DIA;
{
VOLUME = 0; //ZERA VOLUME APÓS 1 DIA;
}

if (millis()> FREQ_1) //TEMPO DE ATUALIZAÇÃO (VAZÃO)
{
Q = 60000*Q_1/FREQ; //CALCULO DA VAZÃO [LPM]
}

```

```

Q_1 = 0;           //ZERA CONTAGEM DE PULSOS (Q_1)

FREQ_1 = FREQ_1 + FREQ;   //ATUALIZA O PARÂMETRO FREQ_1

FREQ_B = ((Q*PPM)/60000); //CALCULO FREQUÊNCIA DA BOMBA EM Hz
}           // FECHA O LACO DA MEDIDA

/*****FAZ FUNCIONAR BOMBA NO MANUAL*****/

if (OP=0)
{
FREQ_B = Q_D*(100/165)*(1000/3600); //CALCULO FREQUÊNCIA DA BOMBA EM Hz
}           //FECHA O LAÇO VERDADEIRO (MANUAL)

/*****FAZ FUNCIONAR BOMBA NO AUTOMÁTICO*****/

else
{
FREQ_B = ((Q*PPM)/60000); //CALCULO FREQUÊNCIA DA BOMBA EM Hz
}           //FECHA O LAÇO FALSO (AUTOMÁTICO)

/*****FAZ FUNCIONAR BOMBA*****/

PULSO_B = (1000*(1/FREQ_B)); //TRANSFORMAR FREQUENCIA EM TEMPO (mS)

```

```

if(PULSO_1<millis() &&!B_LIG) //TESTA DE BOMBA ESTÁ DESLIGADA
{
digitalWrite (BOMBA, HIGH); //ACIONA BOMBA
PULSO_2=millis()+ATRASO; //FAZ PULSO_2, SER 200mS > PULSO_1
B_LIG=true; //INFORMA VARIÁVEL BOOLEANA (BOMBA LIGADA)
} //FECHA LAÇO DE ACIONAMENTO DA BOMBA

if(PULSO_2<=millis() && B_LIG) //TESTA DE BOMBA ESTA LIGADA (PULL-UP)
{
digitalWrite (BOMBA, LOW); // TORNA A BOMBA DESLIGADA
PULSO_1=millis()+PULSO_B - ATRASO; //FAZ PULSO_1, SER FUNÇÃO DA FREQUENCIA
Serial.print (" @@ "); //ESCREVE NO SERIA PARA INDICAÇÕES
Serial.println (PULSO_1); //ESCREVE NO SERIA PARA INDICAÇÕES
DB_LIG=false; //BOMBA DESLIGADA NO INDICADOR
} //FECHA O LAÇO DE DESACIONAMENTO DE PULSO

/*****IMPRIME NO SERIAL OS DADOS MEDIDOS*****/

if (millis()>FREQ_P)
{
Serial.print(" Q, LPM = "); //ESCREVE A VARIÁVEL E UN DA Q (LIT P/ SEG
Serial.print(Q,DEC); //ESCREVE O VALOR DE Q EM DEC = DECIMAIS
Serial.print(" TEMPO "); //ESCREVE A VARIÁVEL TEMPO UND = SEG
Serial.print((millis()/1000), DEC); //ESCREVE O VALOR DE TEMPO (SEG)
Serial.print(" Vol,L = "); //ESCREVE A VARIÁVEL VOLUME UND = L
Serial.print( VOLUME, DEC); //ESCREVE O VALOR VOLUME (L)

```

```
Serial.print(" FREQUENCIA, Hz "); //ESCREVE A VARIÁVEL FREQUÊNCIA UN = Hz
Serial.print(FREQ_B, DEC); //ESCREVE O VALOR DE FREQUÊNCIA (Hz)
Serial.print(" ppm "); //ESCREVE A CONCENTRAÇÃO UN = PPM
Serial.print(PPM, DEC); //ESCREVE O VALOR DE CONCENTRAÇÃO (PPM)
Serial.print (" AUTOMÁTICO "); //ESCREVE CONDIÇÃO DE OPERAÇÃO
Serial.print (OP, DEC); //AUTOMÁTICO=1 MANUAL =0
Serial.println(" ");
FREQ_P=FREQ_P+100;
}
}
```