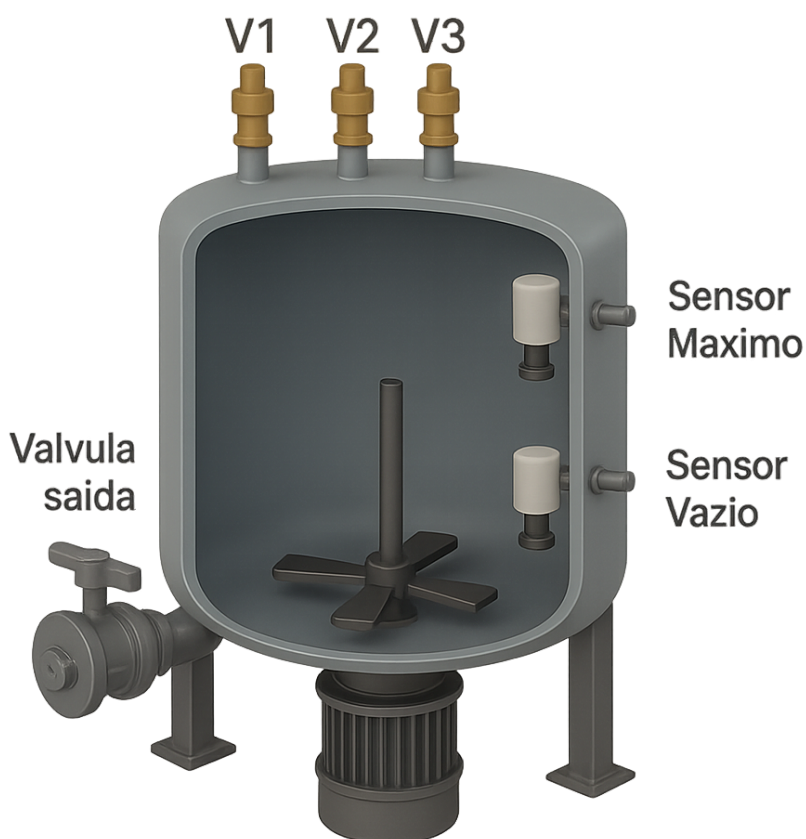
 INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO	Curso: Técnico Integrado em Mecatrônica	Data: 24/04/25	Nota
	Período: Vespertino	Prova: Prova 1	
	Disciplina: Microcontroladores	Valor: 90%	
	Prof.: Marcos Aparecido Chaves Ferreira	Revisão do Aluno:	
Alunos (a): _____ _____			

4) Utilize o código exemplo para realizar a seguinte.

Inicie definindo as entradas com *pull-up* interno ativado e configure os pinos correspondentes às saídas. Utilize a **PORTA D** (registradores PIND, DDRD e PORTD) para especificar todos os pinos de entrada e saída (I/O), conforme a tabela apresentada a seguir.

PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
Vsai	Misturador	V3	V2	V1	S. Vazio	S. Max.	START

Ao pressionar o botão **START**, a **válvula de saída** será acionada para manter o recipiente fechado, enquanto a **válvula V1 (saída)** será aberta por 5 segundos. (1,0) Em seguida, a **válvula V2 (saída)** será ligada por 3 segundos, seguida da ativação do **misturador**, também por 2 segundos, a fim de homogeneizar a mistura. (1,0). Após esse período, a **válvula V3** será aberta e permanecerá acionada até que o **sensor de nível máximo** detecte a presença de líquido, seguida da ativação do **misturador**, também por 2 segundos (1,0). Por fim, a válvula de saída será desligada e o sistema monitorará até que o sensor de nível mínimo (vazio) seja acionado, retornando então ao estado inicial (1,0). **(Total: 5,0pts)**



```
#define V1 PD0
ORG 0x00
```

```
Inicio:
LDI R19, 0b01010101
OUT DDRC, R19
LDI R19, 0b10101010
OUT PORTC, R19
RJMP PRINCIPAL
```

```
MISTURAR:
CBI PORTB,0
SBI PORTB,2
RCALL ATRASO
RCALL ATRASO
CBI PORTB, 2
RJMP Esvaziar
```

```
ESVAZIAR:
SBI PORTD,1
SBIC PINC,1
RJMP Esvaziar
RJMP PRINCIPAL
```

```
ENCHER:
SBI PORTB,0
SBIC PINB,3
RJMP ENCHER
RJMP MISTURAR
```

```
Principal:
SBIC PINB,2
RJMP Principal
RJMP ENCHER
```

OBS: Válvulas V1, V2 e V3 são NF(normalmente fechadas); Válvula de saída NA(normalmente aberta); Cada chamada ATRASO representa um *delay* de 1 segundo.

INSTRUÇÕES BÁSICAS ASSEMBLY - ATMEGA

Lógica e aritmética

ADD	Rd, Rr	Soma dois registradores	$Rd \leftarrow Rd + Rr$	Z, C, N, V, H	1
ADC	Rd, Rr	Soma dois registradores com Carry	$Rd \leftarrow Rd + Rr + C$	Z, C, N, V, H	1
SUB	Rd, Rr	Subtrai dois registradores	$Rd \leftarrow Rd - Rr$	Z, C, N, V, H	1
CLR	Rd	Limpa registrador	$Rd \leftarrow Rd \otimes Rd$	Z, N, V	1
INC	Rd	Incrementa registrador	$Rd \leftarrow Rd + 1$	Z, N, V	1
DEC	Rd	Decrementa registrador	$Rd \leftarrow Rd - 1$	Z, N, V	1
AND	Rd, Rr	Lógica E entre registradores	$Rd \leftarrow Rd \bullet Rr$	Z, N, V	1
ANDI	Rd, K	Lógica E entre registrador e constante	$Rd \leftarrow Rd \bullet K$	Z, N, V	1
OR	Rd, Rr	Lógica OU entre registradores	$Rd \leftarrow Rd \vee Rr$	Z, N, V	1
ORI	Rd, K	Lógica OU entre registrador e constante	$Rd \leftarrow Rd \vee K$	Z, N, V	1

Desvios e chamadas

RJMP	k	Desvio relativo	$PC \leftarrow PC + k + 1$	Nenhum	2
RCALL	k	Chama de sub-rotina	$PC \leftarrow PC + k + 1$	Nenhum	3
RET		Retorno de sub-rotina	$PC \leftarrow STACK$	Nenhum	4
RETI		Retorno de interrupção	$PC \leftarrow STACK$	I	4

Movimentação de dados

MOV	Rd, Rr	Movimento entre registradores	$Rd \leftarrow Rr$	Nenhum	1
LDI	Rd, K	Carrega valor imediato	$Rd \leftarrow K$	Nenhum	1
IN	Rd, P	Leitura de registrador de I/O	$Rd \leftarrow P$	Nenhum	1
OUT	P, Rr	Escrita de registrador de I/O	$P \leftarrow Rr$	Nenhum	1
LDS	Rd, k	Carrega diretamente da SRAM	$Rd \leftarrow (k)$	Nenhum	2
STS	k, Rr	Carrega diretamente para SRAM	$(k) \leftarrow Rr$	Nenhum	2

Manipulação de bits

SBI	P, b	Ativa o bit no registrador de I/O	$I/O(P,b) \leftarrow 1$	Nenhum	2
CBI	P, b	Limpa o bit do registrador de I/O	$I/O(P,b) \leftarrow 0$	Nenhum	2

Teste de bit

BREQ	k	Desvia se igual	if(Z=1) $PC \leftarrow PC + k + 1$	Nenhum	1/ 2
BRNE	k	Desvia se diferente	if(Z=0) $PC \leftarrow PC + k + 1$	Nenhum	1/ 2
SBIC	P, b	Pula se o bit do registrador de I/O estiver limpo (0)	if(P(b)=0) $PC \leftarrow PC + 2$ ou 3	Nenhum	1/ 2/ 3
SBIS	P, b	Pula se o bit do registrador de I/O estiver ativo (1)	if(P(b)=1) $PC \leftarrow PC + 2$ ou 3	Nenhum	1/ 2/ 3