

UNIVERSIDADE FEDERAL DO MARANHÃO

SISTEMAS OPERACIONAIS

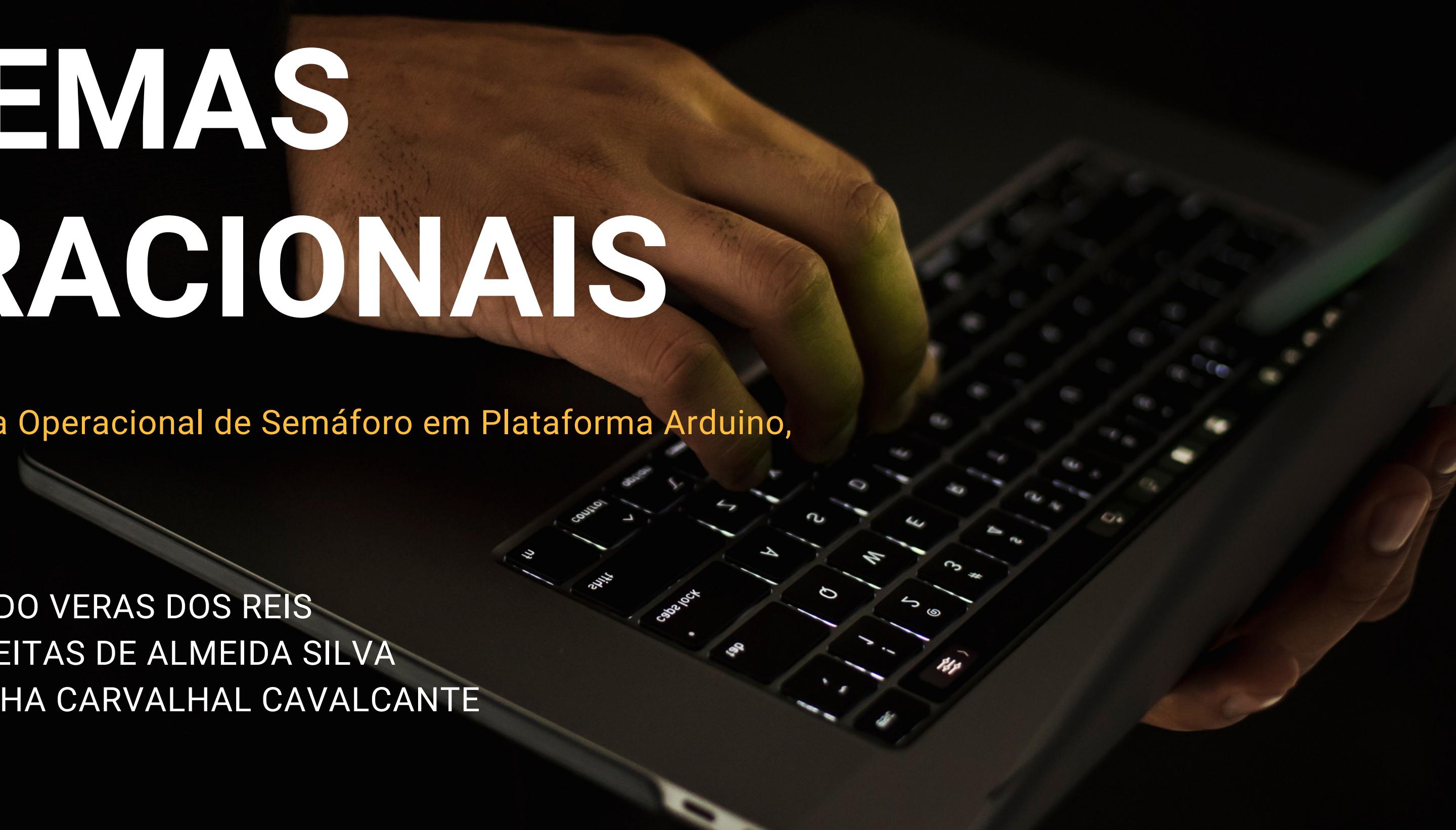
Aplicação de Sistema Operacional de Semáforo em Plataforma Arduino,
utilizando FreeRTOS

DISCENTES:

JOÃO PEDRO AZEVEDO VERAS DOS REIS

LUIS GUILHERME FREITAS DE ALMEIDA SILVA

RENATA COSTA ROCHA CARVALHAL CAVALCANTE



O QUE VEM POR AÍ

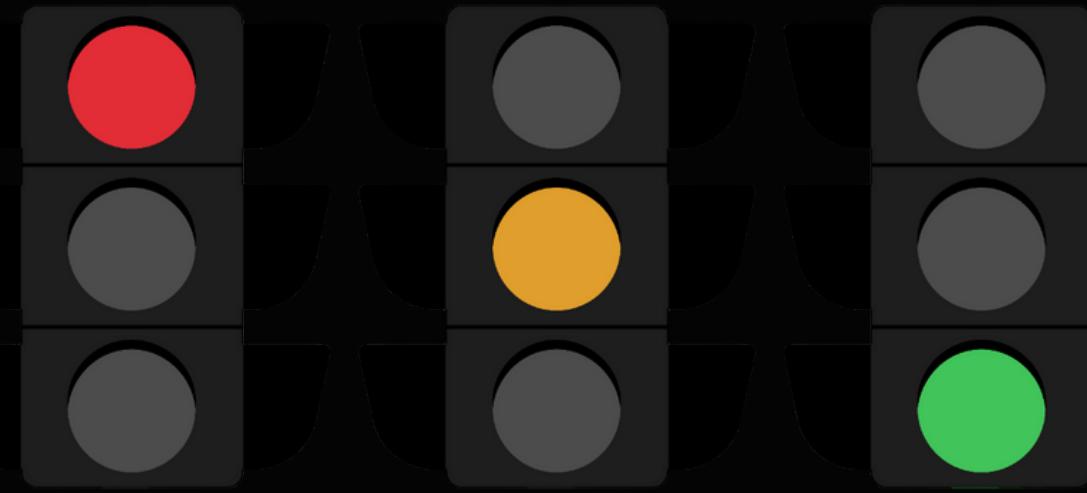
INTRODUÇÃO ÀS TECNOLOGIAS E APRENDIZAGEM

- CONTEXTUALIZAÇÃO DO PROBLEMA
- ALGORITMO UTILIZADO
- PRIORIDADES
- FLUXOGRAMAS
- CASOS FUNCIONAIS

CONTEXTUALIZAÇÃO DO PROBLEMA



CONTEXTUALIZAÇÃO DO PROBLEMA



ALGORITMO DE SISTEMA DE TEMPO REAL

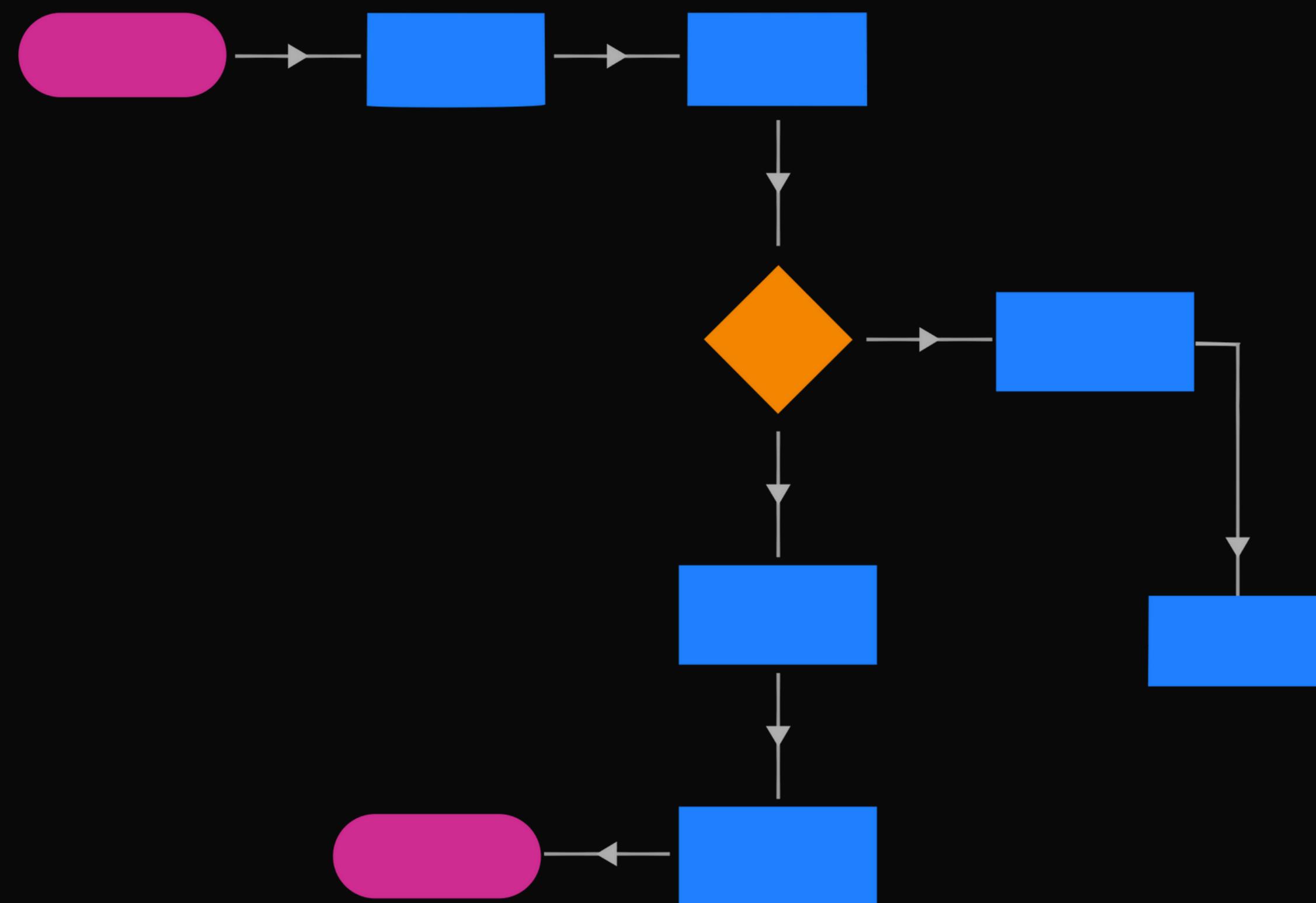


PRIORIDADES DOS PROCESSOS

PROCESSO	PRIORIDADE
AcionarSemafaroTarefa	4
ControlarSemafaroTask	3
VerificarBotaoAzulTask	2
ReiniciarSistemaTask	1

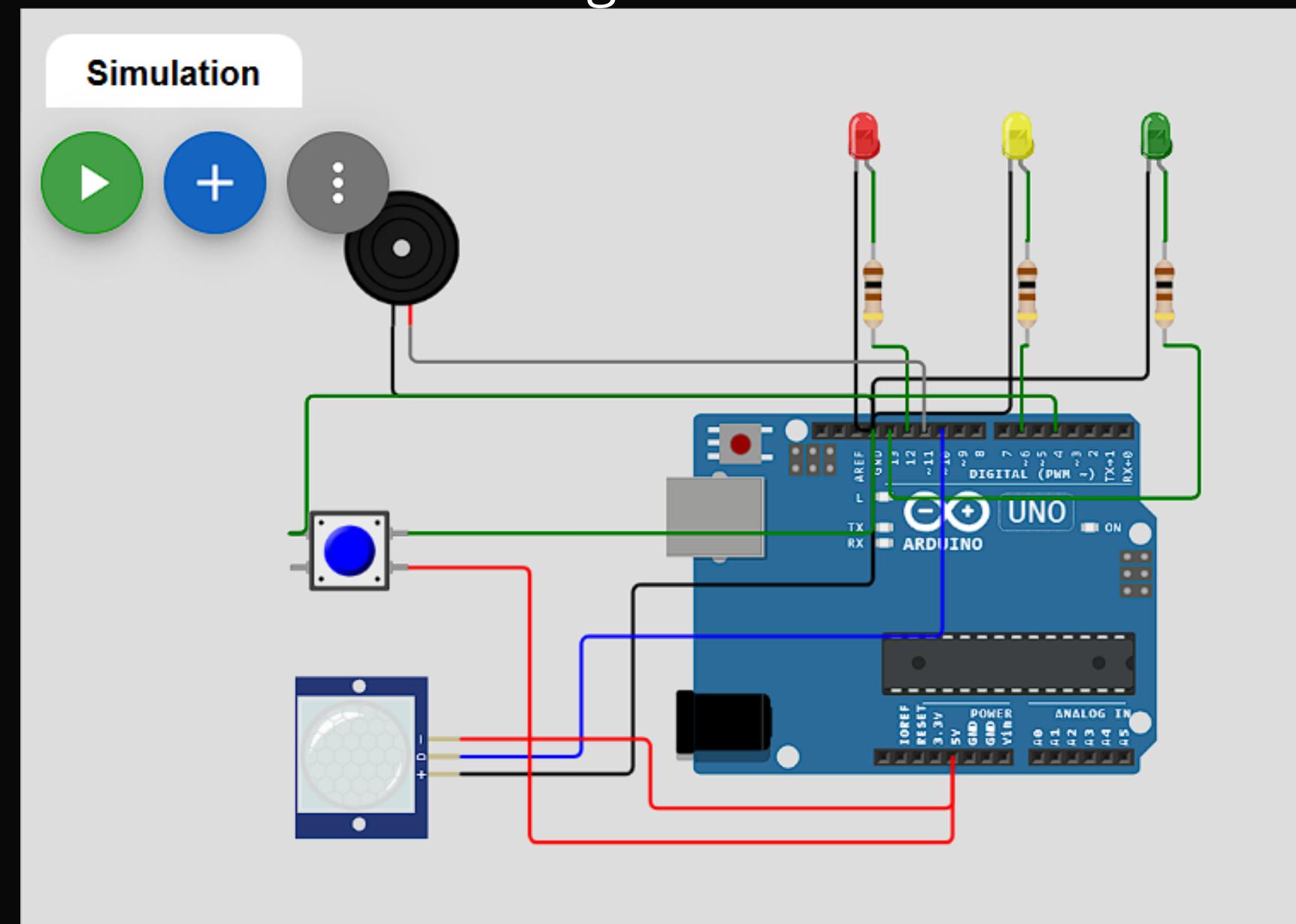
- Tarefa que aciona o semáforo.
- Tarefa principal que controla o semáforo e responde a eventos.
- Tarefa que verifica o botão azul e envia sinais para o fila.
- Tarefa que reinicia o sistema com base nos sinais recebidos pelo fila.

FLUXOGRAMAS



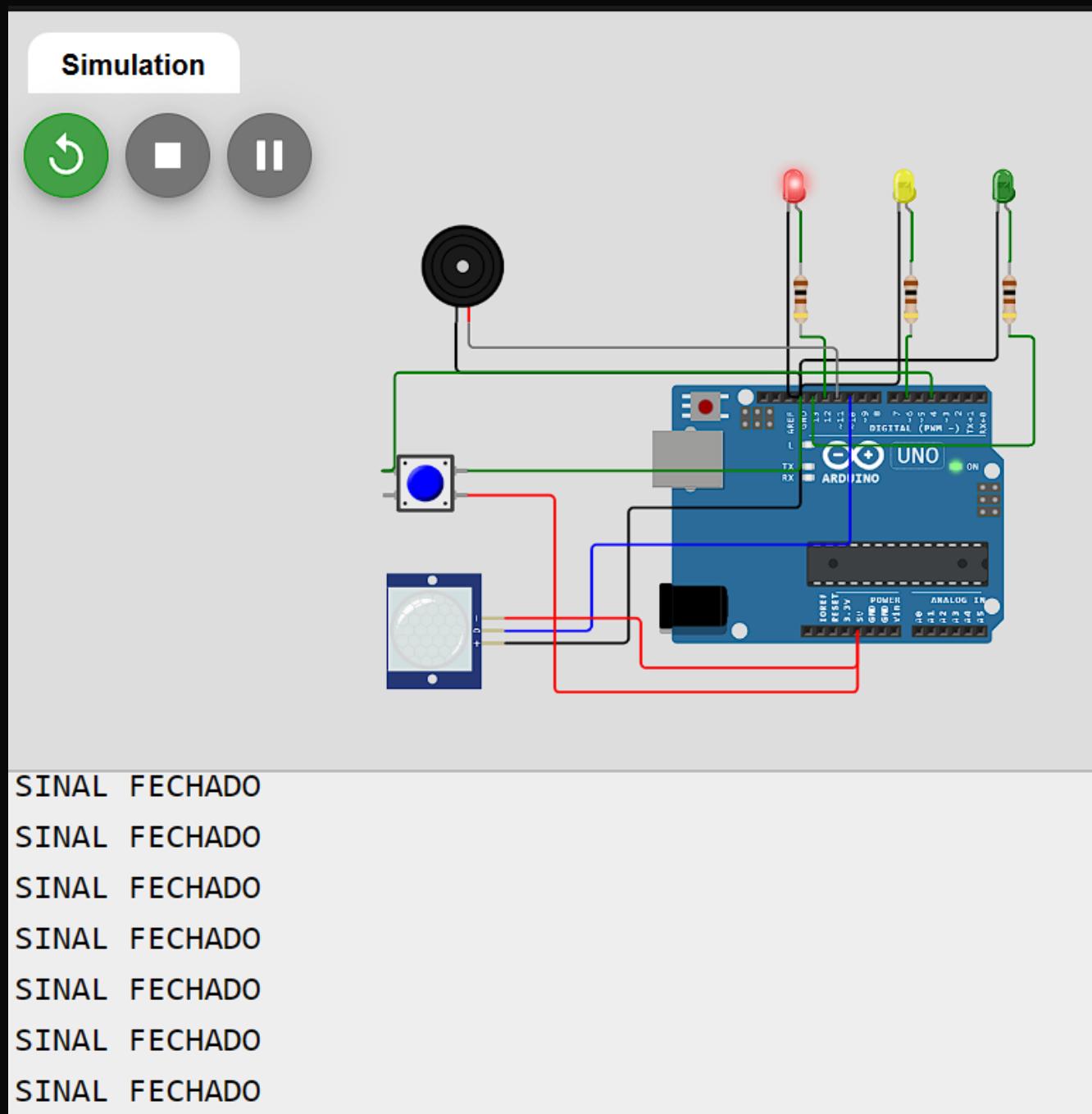
CASOS FUNCIONAIS

Estado desligado do sistema



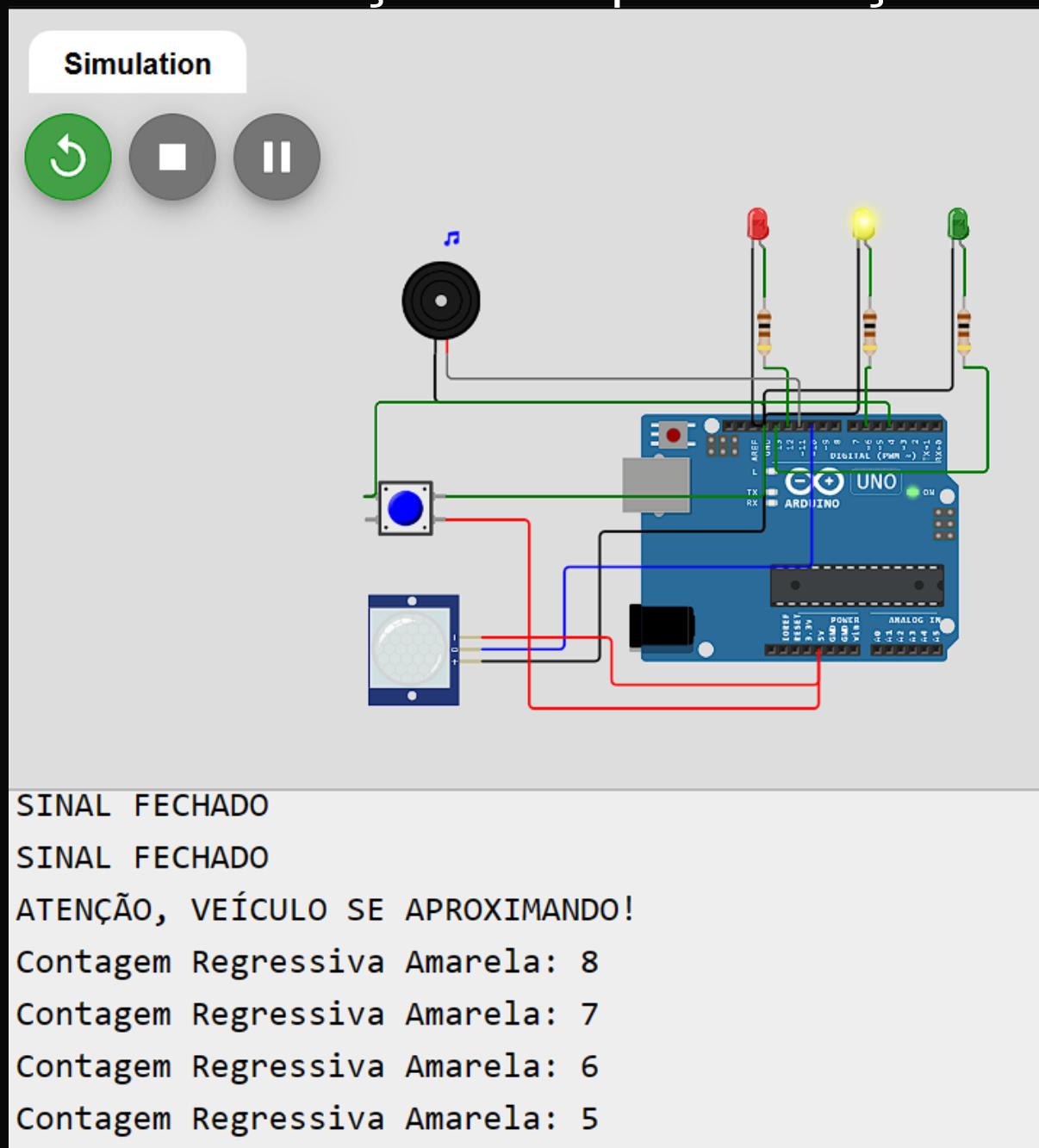
CASOS FUNCIONAIS

Estado inicial do sistema



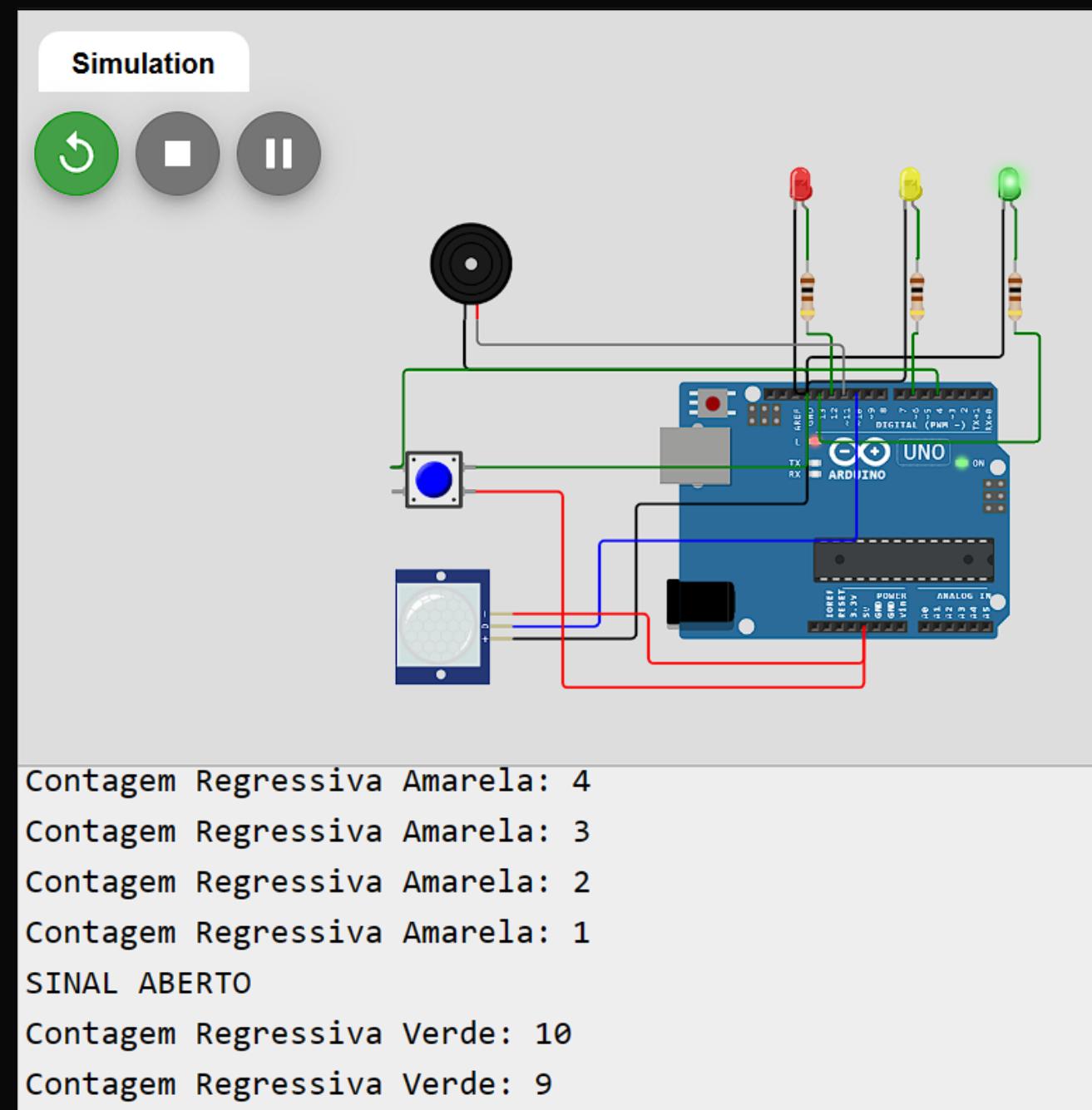
CASOS FUNCIONAIS

Pós-detecção de aproximação



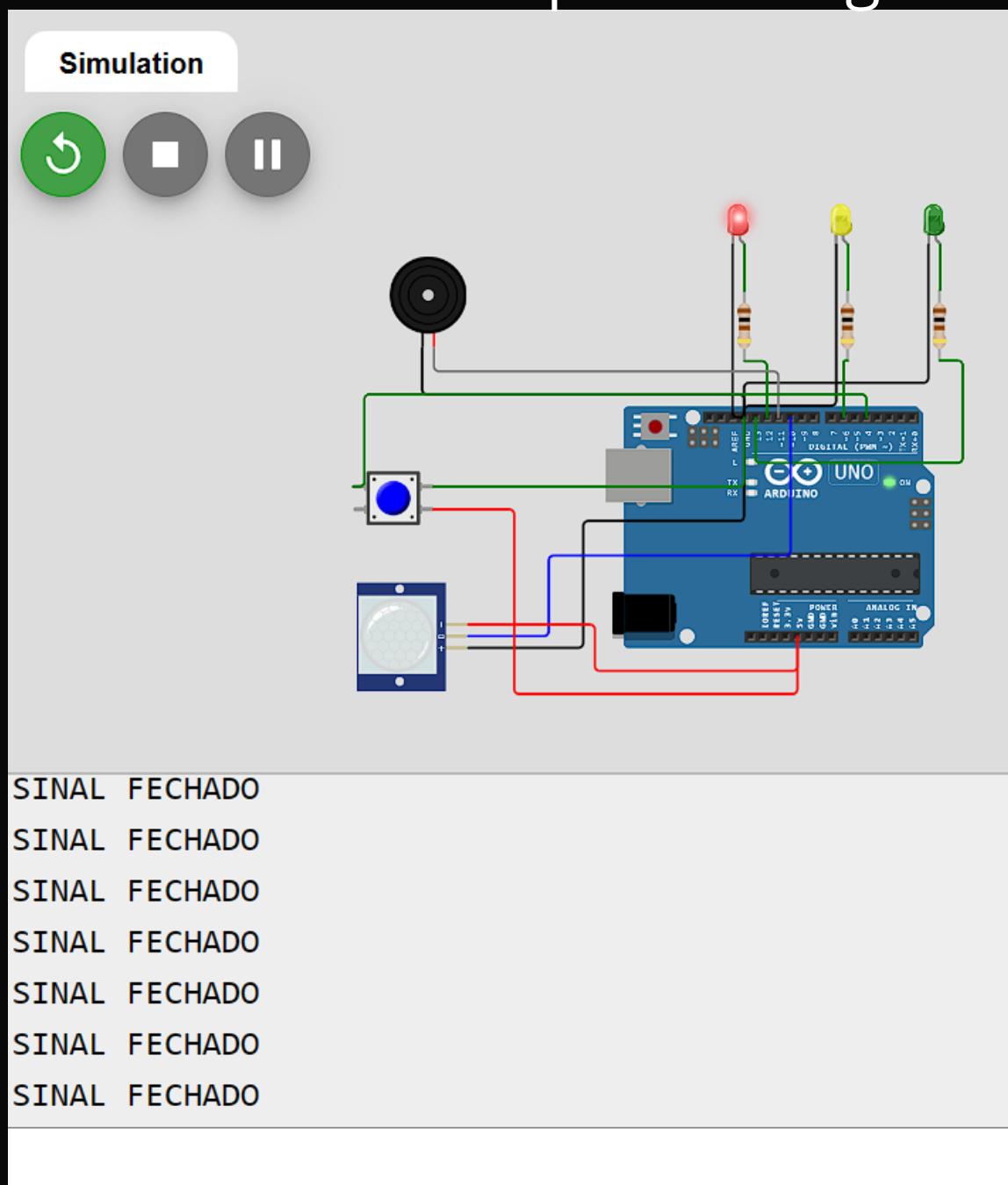
CASOS FUNCIONAIS

Estado de sinal verde



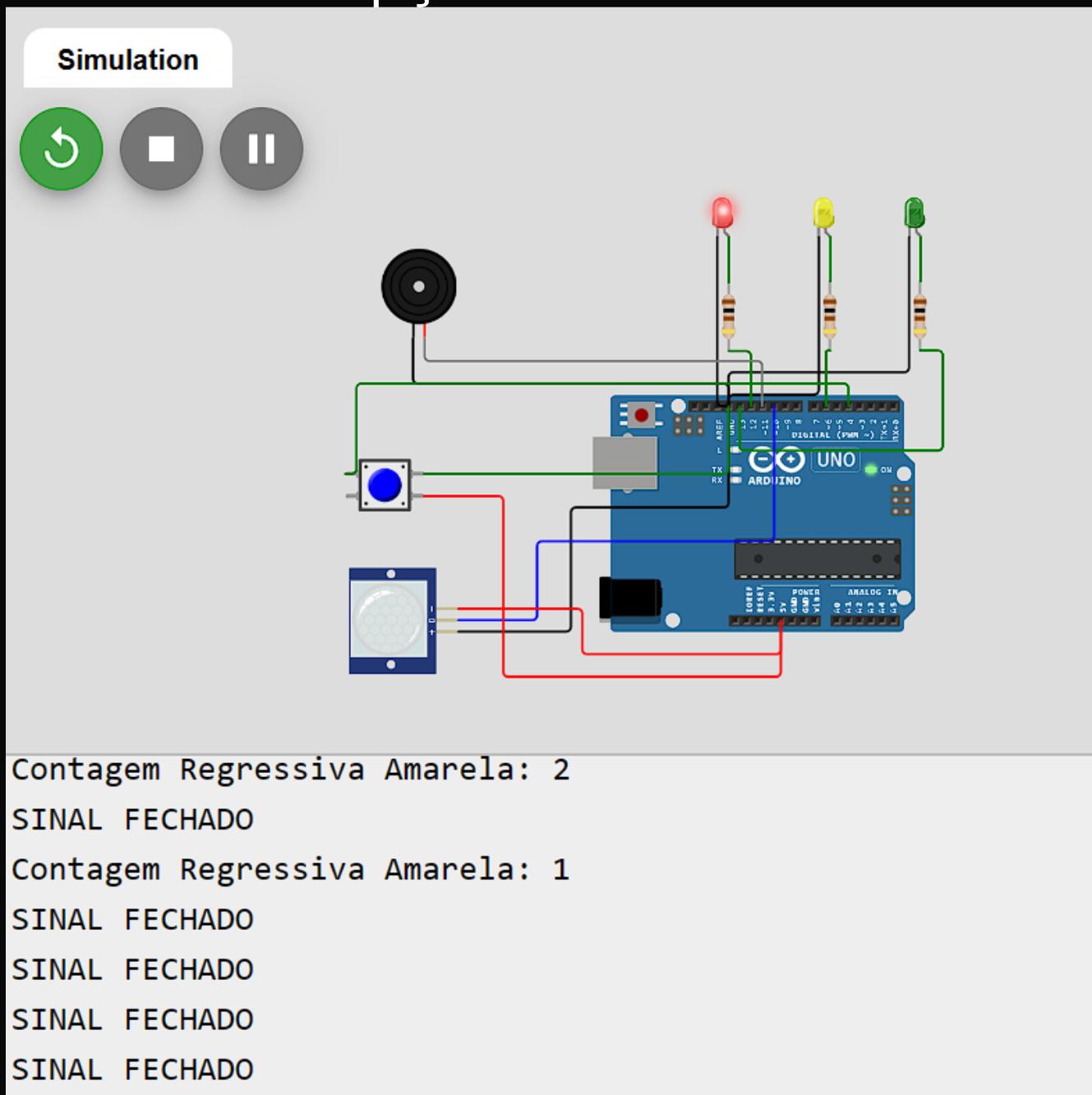
CASOS FUNCIONAIS

Sinal vermelho após contagem



CASOS FUNCIONAIS

Preempção do sistema



CASOS FUNCIONAIS

WOKWi SAVE SHARE Semáforo Copy Docs SIGN UP

```
sketch.ino      diagram.json    libraries.txt    Library Manager
135 }           136 vTaskDelay(pdMS_TO_TICKS(100));
137 }
138 }
139
140 void verificarBotaoAzul(void *pvParameters) {
141     bool botaoAnterior = digitalRead(botaoAzulPin);
142     while (1) {
143         bool botaoAtual = digitalRead(botaoAzulPin);
144         if (botaoAnterior == HIGH && botaoAtual == LOW) {
145             bool botaoPressionado = true;
146             digitalWrite(ledAmareloPin, LOW);
147             digitalWrite(ledVerdePin, LOW);
148             xQueueSend(fila, &botaoPressionado, portMAX_DELAY);
149         }
150         botaoAnterior = botaoAtual;
151         vTaskDelay(pdMS_TO_TICKS(100));
152     }
153 }
154
155 void reiniciarSistemaTask(void *pvParameters) {
156     while (1) {
157         bool botaoPressionado = false;
158         if (xQueueReceive(fila, &botaoPressionado, portMAX_DELAY))
159             reiniciarSistema();
160     }
161 }
162
163
164 void setup() {
165     Serial.begin(9600);
166     pinMode(ledVermelhoPin, OUTPUT);
167     pinMode(ledAmareloPin, OUTPUT);
168     pinMode(ledVerdePin, OUTPUT);
```

Simulation

00:03.686 100%

SINAL FECHADO
SINAL FECHADO
ATENÇÃO, VEÍCULO SE APROXIMANDO!
Contagem Regressiva Amarela: 8
Contagem Regressiva Amarela: 7

CASOS FUNCIONAIS

WOKWi SALVAR COMPARTILHAR Semáforo Documentos R

esboço.ino diagrama.json bibliotecas.txt Gerente de Biblioteca

```
1 #include < Arduino.h >
2 #include < Arduino_FreeRTOS.h >
3 #include < tarefa.h >
4 #include < semphr.h >
5 #include < fila.h >
6
7 const int ledVermelhoPin = 12 ;
8 const int ledAmareloPin = 6 ;
9 const int ledVerdePin = 13 ;
10 const int pirPin = 10 ;
11 const int botaoAzulPin = 4 ;
12 const int buzzerPin = 11 ;
13
14 SemaphoreHandle_t semaforoSinal;
15 SemaphoreHandle_t semaforoBotao;
16 volátil bool sistemaAtivo = true ;
17 volátil bool sinalFechado = true ;
18 volátil int tempoPiscarAmarelo = 8 ;
19 volátil int tempoLigarVerde = 10 ;
20 volátil int contadorInicial = 5 ;
21 volátil bool mensagemExibida = false ;
22
23 QueueHandle_t fila;
24
25 void acionarSemafaro ( void *pvParameters ) {
26     int contador = contadorInicial;
27     enquanto ( contador > 0 ) {
28         if ( sistemaAtivo && sinalFechado ) {
29             xSemaphoreTake ( semaforoSinal, portMAX_DELAY ) ;
30             if ( digitalRead ( ledVermelhoPin ) == ALTO ) {
31                 Série . println ( "SINAL FECHADO" ) ;
32                 contador--;
33             }
34         }
35     }
36 }
```

Simulação 00:50.032 93%

Pare a simulação Simular movimento X

SINAL FECHADO
SINAL FECHADO
SINAL FECHADO
SINAL FECHADO
SINAL FECHADO
SINAL FECHADO
SINAL FECHADO

BIBLIOTECA UTILIZADA

<https://freertos.org/index.html>

Reconhecimentos e Direitos Autorais

@autor: JOÃO PEDRO AZEVEDO VERAS DOS REIS, LUIS GUILHERME FREITAS DE ALMEIDA SILVA, RENATA COSTA ROCHA CARVALHAL CAVALCANTE

@data última versão: 10/12/2023

@versão: 1.0

@Agradecimentos: Universidade Federal do Maranhão (UFMA), Professor Doutor Thales Levi Azevedo Valente, e colegas de curso.

@Copyright/License

Este material é resultado de um trabalho acadêmico para a disciplina SISTEMAS OPERACIONAIS, sobre a orientação do professor Dr. THALES LEVI AZEVEDO VALENTE, semestre letivo 2023.2, curso Engenharia da Computação, na Universidade Federal do Maranhão (UFMA). Todo o material sob esta licença é software livre: pode ser usado para fins acadêmicos e comerciais sem nenhum custo. Não há papelada, nem royalties, nem restrições de "copyleft" do tipo GNU. Ele é licenciado sob os termos da licença MIT reproduzida abaixo e, portanto, é compatível com GPL e também se qualifica como software de código aberto. É de domínio público. Os detalhes legais estão abaixo. O espírito desta licença é que você é livre para usar este material para qualquer finalidade, sem nenhum custo. O único requisito é que, se você usá-lo, nos dê crédito.

Copyright © 2023 Educational Material

Este material está licenciado sob a Licença MIT. É permitido o uso, cópia, modificação, e distribuição deste material para qualquer fim, desde que acompanhado deste aviso de direitos autorais.

O MATERIAL É FORNECIDO "COMO ESTÁ", SEM GARANTIA DE QUALQUER TIPO, EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS NÃO SE LIMITANDO ÀS GARANTIAS DE COMERCIALIZAÇÃO, ADEQUAÇÃO A UM DETERMINADO FIM E NÃO VIOLAÇÃO. EM HIPÓTESE ALGUMA OS AUTORES OU DETENTORES DE DIREITOS AUTORAIS SERÃO RESPONSÁVEIS POR QUALQUER RECLAMAÇÃO, DANOS OU OUTRA RESPONSABILIDADE, SEJA EM UMA AÇÃO DE CONTRATO, ATO ILÍCITO OU DE OUTRA FORMA, DECORRENTE DE, OU EM CONEXÃO COM O MATERIAL OU O USO OU OUTRAS NEGOCIAÇÕES NO MATERIAL.

Para mais informações sobre a Licença MIT: <https://opensource.org/licenses/MIT>.