****

ESCOLA SENAI “José Polizotto”

CURSO TÉCNICO DESENVOLVIMENTO DE SISTEMAS

HENRIQUE DA SILVA GUIMARÃES

MARIA LUIZA PELEGRINI PEREIRA DE ANDRADE

VINÍCIUS NACHBAR DE MEDEIROS

YURI MOMESSO DE LIMA

**PROJETO DO SEMÁFORO INTELIGENTE**

**GARÇA**

**2026**

**SUMÁRIO**

[**1. Introdução** 3](#_Toc200737531)

[**2. Objetivos** 4](#_Toc200737532)

[**2.1 Objetivo geral** 4](#_Toc200737533)

[**2.2 Objetivo específico** 4](#_Toc200737534)

[**3. Métodos** 5](#_Toc200737535)

[**3.1 Lista de Materiais** 5](#_Toc200737536)

[**3.2 Linguagem de Programação** 5](#_Toc200737537)

[**3.3 Sistema Operacional** 5](#_Toc200737538)

[**3.3.1 Requisitos mínimos** 5](#_Toc200737539)

[**3.3.2 Configurações usadas** 6](#_Toc200737540)

[**3.4 Hardware** 6](#_Toc200737541)

[**3.5 IDE do Arduino** 6](#_Toc200737542)

[**3.6 Requisitos funcionais** 7](#_Toc200737543)

[**3.7 Requisitos não funcionais** 7](#_Toc200737544)

[**3.8 História de usuário** 8](#_Toc200737545)

[**3.9 Regras de negócio** 10](#_Toc200737546)

[**3.12 Cronograma** 11](#_Toc200737547)

[**3.13 Fluxograma** 12](#_Toc200737548)

[**4. Vulnerabilidades do Sistema** 13](#_Toc200737549)

[**5. Política de Segurança Interna** 14](#_Toc200737550)

[**5.1 Do Objetivo** 15](#_Toc200737551)

[**5.2 Do Sigilo E Confidencialidade** 15](#_Toc200737552)

[**5.3 Das Normas De Contas E Senhas De Usuário** 16](#_Toc200737553)

[**5.4 Das Infrações E Penalidades** 17](#_Toc200737554)

[**6. Resultados obtidos** 19](#_Toc200737555)

[**7. Conclusão** 20](#_Toc200737556)

[**8. Referencias** 21](#_Toc200737557)

**1. Introdução**

O desenvolvimento do presente projeto consiste na implementação de semáforos na rua Barão do Rio Branco, já que essa rua possui um alto fluxo de veículos e de pedestres, e não possui um devido controle de tráfego, o que resulta em uma análise crítica para possíveis melhorias



O projeto visa melhorar o fluxo do tráfego e a segurança dos pedestres, para isso, foi criado um software capaz de controlar o semáforo de melhor maneira para aquela rua, utilizando tecnologias para o controle remoto do dispositivo, de maneira que apenas pessoas autorizadas possam configurá-lo

**2. Objetivos**

**2.1 Objetivo geral**

O Objetivo do projeto é melhorar o tráfego e a segurança da rua Barão do Rio Branco, principalmente em horários muito movimentados, através da implementação de um dispositivo de controle de tráfego inteligente e capaz de ser configurado remotamente e de forma segura.

**2.2 Objetivo específico**

* Tecnologia
* Tráfego
* Remoto
* Inovação
* Segurança

**3. Métodos**

**3.1 Lista de Materiais**

|  |  |
| --- | --- |
| Lista de materiais | |
| Quantidade | Item |
| 2 | Led Difuso 5 mm Vermelho |
| 2 | Led Difuso 5 mm Amarelo |
| 2 | Led Difuso 5 mm Verde |
| 1 | Isopor |
| 20 | Jumpers |
| 1 | Wemos D1 |
| 1 | Tinta amarela |
| 1 | Tinta azul |
| 1 | Tinta Vermelha |

**Fonte:** Os Autores

**3.2 Linguagem de Programação**

No desenvolvimento do projeto, foi usado a linguagem de programação C++, O C++ é utilizado no Arduino IDE porque ele é uma linguagem orientada a objetos e permite utilizar os recursos do C.

## **3.3 Sistema Operacional**

No desenvolvimento do projeto, foi usado o sistema operacional Windows 11, um sistema operacional de código fechado, gerenciado pela Microsoft

O Windows 11 foi escolhido pela sua versatilidade, suporte e confiabilidade. Por ser um sistema operacional de uma grande empresa ele é constantemente atualizado, e, por ser o sistema mais utilizado do mundo, possui uma maior quantidade de aplicativos feitos para ele, além de ser mais fácil de usar

## **3.3.1 Requisitos mínimos**

**Processadores:** Processador com 2 ou mais núcleos

**RAM:** 8 GB

**Armazenamento:** 64 GB

**Firmware do Sistema:** UEFI

**TPM:** [Trusted Platform Module](https://docs.microsoft.com/windows/security/information-protection/tpm/trusted-platform-module-overview) (TPM) versão 2.0

**3.3.2 Configurações usadas**

Especificações do dispositivo

* Processador: 11th Gen Intel(R) Core (TM) i7-11390H
* RAM instalada: 16,0 GB
* Armazenamento: 512 GB

Especificações do sistema

* Edição Windows 11 Education
* Versão 24H2

**3.4 Hardware**

O Hardware utilizado foi a placa WeMos D1, ela oferece WiFi nativo e é uma ótima opção para projetos IoT. A WeMos D1 tem conector micro USB e pode ser programada utilizando a IDE do Arduino

**3.5 IDE do Arduino**

A Arduino IDE foi escolhida, pois é um software muito completo, atualizado e intuitivo, possuindo muito funcionalidades essenciais para a programação do microcontrolador. Além disso, por ser um dos softwares mais utilizados para a sua função, possui muito conteúdo e suporte, o que torna seu uso mais eficiente e rápido,

Foi utilizado a versão 2.3.6, pois era a versão mais atualizada durante a produção do projeto

**3.6 Requisitos funcionais**

|  |  |  |
| --- | --- | --- |
| **REQUISITOS FUNCIONAIS** | | |
| **ID** | **DESCRIÇÃO DO REQUISITO** | **CLASSIFICAÇÃO** |
| **RF01** | O SISTEMA DEVE LIGAR E DESLIGAR OS LEDS, COM UM INTERVALO DE TEMPO ENTRE CADA UM DELES | ESSENCIAL |
| **RF02** | O SISTEMA DEVE PERMITIR QUE OS VALORES DE TEMPO SEJAM ALTERADOS REMOTAMENTE | ESSENCIAL |
| **RF03** | O SISTEMA DEVE PEDIR INFORMAÇÕES A QUEM FOR CONFIGURA-LO REMOTAMENTE | ESSENCIAL |
| **RF04** | O SISTEMA DEVE TER UM LUGAR ONDE AS PESSOAS POSSAM COMENTAR SOBRE A SITUAÇÃO DO TRÁFEGO | IMPORTANTE |

**Fonte:** Os Autores

**3.7 Requisitos não funcionais**

|  |  |  |
| --- | --- | --- |
| **REQUISITOS NÃO FUNCIONAIS** | | |
| **ID** | **DESCRIÇÃO DO REQUISITO** | **CLASSIFICAÇÃO** |
| **RNF01** | O SEMÁFORO DEVE APRESENTAR UM DESIGN MODERNO | DESEJÁVEL |
| **RNF02** | O HARDWARE DEVE SER ESCONDIDO E PROTEGIDO LONGE DE PESSOAS NÃO AUTORIZADAS | DESEJÁVEL |
| **RNF03** | O SOFTWARE DEVE FUNCIONAR ININTERRUPTAMENTE | IMPORTANTE |
| **RNF04** | O SISTEMA DEVE TER UM LUGAR ONDE AS PESSOAS POSSAM COMENTAR SOBRE A SITUAÇÃO DO TRÁFEGO | IMPORTANTE |

**Fonte:** Os Autores

**3.8 História de usuário**

|  |  |  |
| --- | --- | --- |
| **ID REQUISITO** | **ID** | **HISTÓRIAS DE USUARIO** |
| **RF01** | **HU01** | **Como Stakeholder;**  **quero** que o sistema alterne entre os LEDs, com intervalo de tempo entre cada um deles.  **Para que** o tráfego seja gerenciado corretamente |
| **RF02** | **HU02** | **Como Funcionário;**  **Quero** que os valores de tempo do semáforo possam ser alterados remotamente. **Para que** não precise ir pessoalmente para configurar. |
|
|
|
| **RF03** | **HU03** | **Como Stakeholder;**  **Quero** que o sistema peça informações pessoais para quem for configurá-lo remotamente.  **Para que** nenhuma pessoa de fora tenha o acesso ao sistema do semáforo. |
|
|
|
| **RF04** | **HU04** | **Como Pedestre/Motorista;**  **Quero** Comentar sobre a atual situação do tráfego e fazer comentários sobre o semáforo **Para Que** a segurança do trânsito possa ser garantida e melhorado para os demais |
|
|
|
|

**Fonte:** Os Autores

|  |  |  |
| --- | --- | --- |
| **ID REQUISITO** | **ID** | **HISTÓRIAS DE USUARIO** |
| **RNF01** | **HU01** | **Como Stakeholder;**  **Quero** que o semáforo apresente um design moderno. **Para que** se destaque dos demais e torne o local mais agradável |
| **RNF02** | **HU02** | **Como Stakeholder;**  **Quero** que o hardware seja bem escondido e ao seja tão visível para pessoas. **Para que** pessoas erradas não mexam no hardware, e nem corra risco de furtos. |
| **RNF03** | **HU03** | **Como Stakeholder;**  **Quero** que o software funcione ininterruptamente **para que** o semáforo funcione 24/7, garantindo a segurança do trânsito o tempo todo |

**Fonte:** Os Autores

**3.9 Regras de negócio**

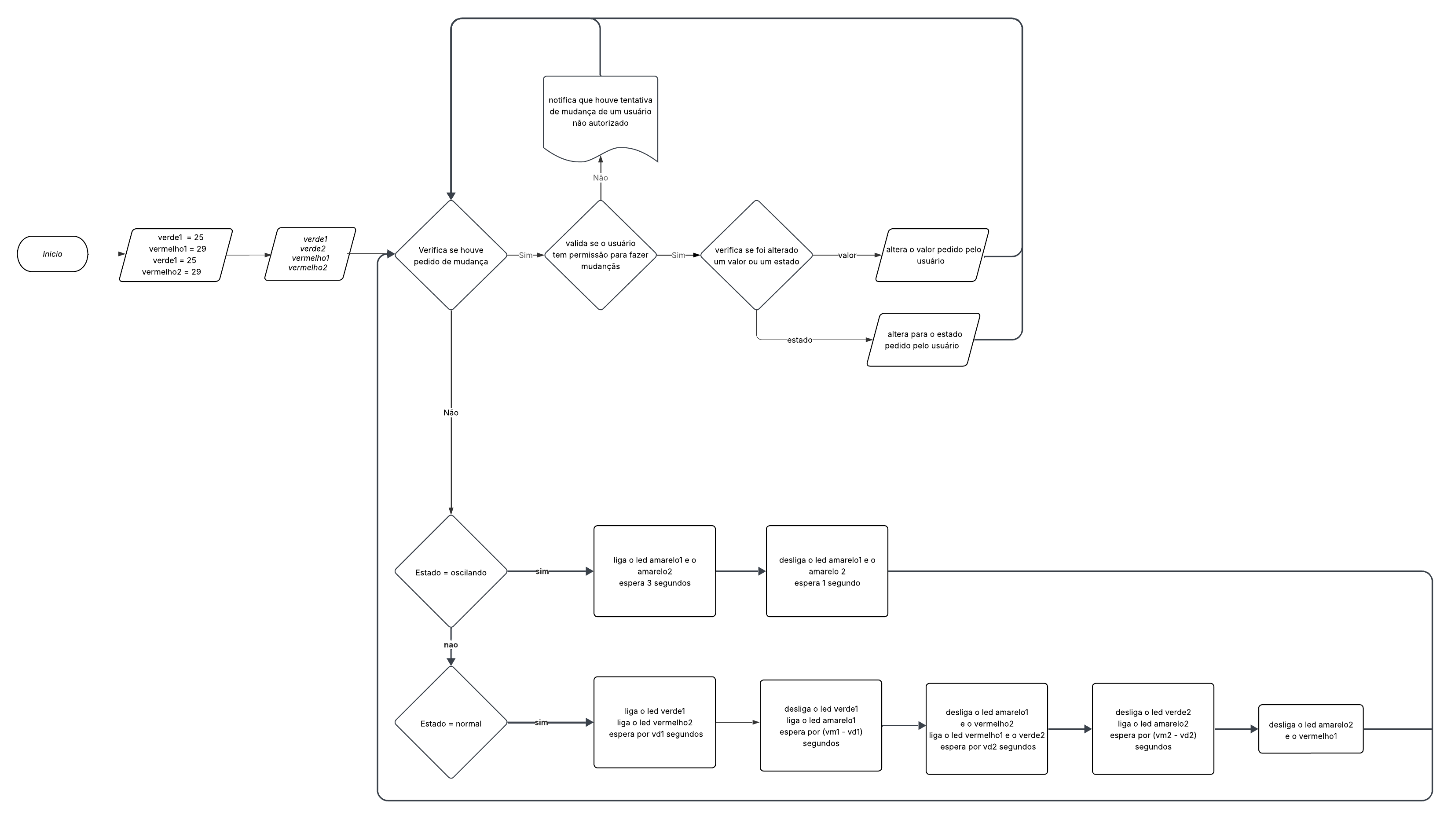
|  |  |
| --- | --- |
| **REGRAS DE NEGÓCIO** | |
| **CADASTRO E ACESSO** | RN01: Somente usuários autenticados podem acessar funcionalidades restritas do sistema. |
| RN02: Funcionários devem fornecer CPF e RF (Registro de Funcionário) e biometria válidos para realizar o cadastro. |
| RN03: Funcionários só podem ser cadastrados pela administração |
| **ACESSO REMOTO** | RN04: Os funcionários autenticados podem alterar os valores de intervalos do semáforo |
| RN05: Os funcionários autenticados podem selecionar modos do semáforo, apenas caso seja necessário para se adaptar a situação do tráfego, como ficar piscando o LED amarelo ou ficar sempre verde |
| RN06: O sistema deve registrar as alterações feitos pelos funcionários, para identificar se um funcionário está exercendo seu trabalho corretamente |
| **CONTROLE DE TRÁFEGO** | RN07: O sistema deve alterar os LEDs de maneira sincronizada para evitar acidentes |
| **CANAL DE**  **COMUNICAÇÕES** | RN08: Caso algum pedestre ou motorista queira relatar algum problema no trânsito, ele poderá enviar uma mensagem no WhatsApp para a empresa |
| **SEGURANÇA** | RN09: O sistema deve bloquear tentativas de registro, após de 3 erros |
| RN10: O software de registro e alteração dos valores deve ser distribuídos apenas para funcionários |
| RN11: Para proteger o hardware do semáforo de vandalismo e condições climáticos, ele deverá estar bem protegido dentro do semáforo, ou em um lugar coberto e de difícil acesso |

**Fonte:** Os Autores

**3.12 Cronograma**

|  |  |  |  |
| --- | --- | --- | --- |
| **Cronograma** | | | |
| **Planejamento de atividades** | **Status** | **%** | **Meta** |
| **Sprint 01** | | | |
| Levantamento de requisitos | Concluído | 100% | 100% |
| Documentação | Concluído | 100% | 100% |
| Programação do Semáforo | Concluído | 100% | 100% |
| Lista de Materiais | Concluído | 100% | 100% |
| Política de Segurança | Concluído | 100% | 100% |
| Vulnerabilidades Detectadas | Concluído | 100% | 100% |
| Cronograma | Concluído | 100% | 100% |
| SO a ser utilizado | Concluído | 100% | 100% |
| Linguagem de Programação a ser utilizada | Concluído | 100% | 100% |
| **Sprint 02** | | | |
| Documento descritivo da solução | Concluído | 100% | 100% |
| Diagrama funcional do projeto físico do semáforo | Concluído | 100% | 100% |
| **Sprint 03** | | | |
| Maquete do ambiente escolhido | Concluído | 100% | 100% |
| Código fonte do programa | Concluído | 100% | 100% |

**3.13 Fluxograma**

****

**4. Vulnerabilidades do Sistema**

* A falta/queda de energia é problema para os motoristas e pedestres que precisam do semáforo para atravessar, podendo causar até mesmo um acidente.
* Desgastes de componentes do semáforo podem levar a danos ao hardware, como curto-circuito e mau funcionamento do circuito
* O vandalismo o Furto de cabos podem ser problemas graves, já que impossibilitam o semáforo de funcionar
* O compartilhamento indevido de senhas e informações confidenciais põem em risco a segurança do trânsito, pois pessoas mal-intencionada podem alteram o funcionamento
* O compartilhamento indevido de senhas também pode resultar em ataques cibernéticos
* Se o sistema não for mantido atualizado, hackers podem encontrar falhas e usá-las para invadir o sistema

**5. Política de Segurança Interna**

**5.1 Do Objetivo**

**Art. 1º -** A instituição desta PSI tem como objetivo prover normas e procedimentos que formalizem as práticas de segurança da informação e assegurem o uso adequado dos equipamentos, softwares, serviços e outros recursos tecnológicos disponibilizados aos usuários para uso profissional, uma vez que a sua má utilização pode afetar a segurança pública. Um dos objetivos é divulgar os princípios de segurança da Informação, descritos abaixo:

**I.** A segurança da informação é responsabilidade de todos e deve ser tratada em um alto nível organizacional.

**II.** Adotar uma abordagem baseada em riscos. A segurança da informação é fundamentada em decisões baseadas em riscos.

**III.** Promover um ambiente positivo de segurança da informação. Para instituir uma cultura positiva, deve-se promover e apoiar a coordenação das atividades das partes interessadas para atingir um direcionamento coerente para a segurança da informação. Isto ajudará na implantação de programas de treinamento, educação e conscientização.

**5.2 Do Sigilo E Confidencialidade**

**Art. 2º -** Todas as informações obtidas durante o processo de trabalho devem ser confidenciais e tratadas apenas com funcionários responsáveis.

**Art. 3º -** É dever dos funcionários manter absoluto sigilo com relação a toda e qualquer informação a que tiver acesso no âmbito da Smart Street, nos seguintes termos:

**I.** manter sigilo, escrito e/ou verbal, ou por qualquer outra forma, de todos os dados, processos, informações, documentos e materiais, dentre outros;

**II.** não revelar, reproduzir, utilizar ou dar conhecimento, em hipótese alguma a terceiros, inclusive funcionários, de dados, informações sem prévia autorização da Smart Street;

**III.** não revelar tais informações a terceiro ou a pessoa alguma que não aquelas que, em razão de suas funções, necessitem conhecê-las.

**Art. 4º -** Em caso de rescisão deixar imediatamente de utilizar qualquer credencial de acesso aos softwares ou áreas da empresa. Devolver quaisquer materiais que contenham informações de propriedade do Smart Street.

**5.3 Das Normas De Contas E Senhas De Usuário**

**Art. 5º -** As normas de contas e senhas estabelecem os procedimentos adequados para a criação e correta utilização dos acessos aos diversos ativos de TI dentro da rede corporativa da Smart Street.

**Art. 6º -** As senhas são sigilosas, individuais e intransferíveis, não podendo ser compartilhadas ou divulgadas a terceiros. Deve-se evitar anotar senhas em papel ou guardá-las de forma digital.

**Art. 7º -** As credenciais de acesso são fornecidas para uso corporativo. A ativação da credencial constitui um acordo onde o usuário declara que compreende e respeitará todos os documentos normativos de segurança da Informação.

**Art. 8º -** A Smart Street poderá revogar o acesso a qualquer momento, caso fique constatado qual foi a credencial de acesso que ocasionou um incidente de segurança.

**Art. 9º -** É recomendado a criação de uma senha forte com difícil decodificação. Desse modo, o usuário deve evitar utilizar data de nascimento, sobrenome, nome de familiares etc.

**Art. 10º -** Para a formação de senhas o usuário deve seguir as seguintes regras de formação:

**I.** Tamanho mínimo de 14 caracteres;

**II.** Utilizar pelo menos um caractere especial (ex: @, #, $, %);

**III.** Utilizar pelo menos uma letra maiúscula, uma letra minúscula;

**IV.** É recomendado para a sua segurança que as senhas sejam trocadas periodicamente.

**5.4 Das Infrações E Penalidades**

**Art.11º -** Constitui infração qualquer ato que viole as normas estabelecidas neste PSI, incluindo, mas não se limitando a:

**I -** Compartilhamento de senhas ou credenciais de acesso com terceiros.

**II -** Uso indevido dos equipamentos, softwares ou outros recursos tecnológicos disponibilizados.

**III -** Acesso não autorizado a informações sigilosas ou confidenciais.

**IV -** Não realizar o log-off de contas pessoais ao término do expediente.

**V -** Manter o uso de credenciais ou informações fora do horário de trabalho, sem autorização de um superior.

**VI -** Alteração ou tentativa de modificação dos sistemas e equipamentos.

**Art. 12º -** A não observância das normas desta PSI, seja isolada ou cumulativamente, implicará ao infrator as seguintes punições: Aviso de Descumprimento, Advertência ou Suspensão, Demissão por justa causa e abertura de processo civil ou criminal, se for o caso.

**Art. 13º -** O Aviso de Descumprimento será encaminhado por e-mail ao funcionário infrator e ao chefe imediato na primeira violação cometida, indicando qual a norma que foi violada.

**Art. 14º -** A Advertência ou Suspensão Disciplinar será aplicada por escrito nos casos de infrações de menor gravidade ou na hipótese de reincidência e será registrada na ficha pessoal do funcionário.

**Art. 15º -** A Demissão por justa causa será aplicada nos casos legais e de natureza grave ou nas hipóteses previstas no artigo 482 da Consolidação das Leis do Trabalho.

**Art. 16º -** Infrações que coloquem a segurança pública em risco resultarão em abertura de processo civil ou criminal, a depender da gravidade da infração

**6. Resultados obtidos**

O desenvolvimento do projeto foi considerado satisfatório. Durante a implementação, o sistema de controle dos LEDs funcionou conforme o esperado. Isso contribuiu diretamente para a simulação adequada do controle de tráfego na rua Barão do Rio Branco, objetivo principal do projeto.

A linguagem de programação utilizada foi o C++, que se mostrou eficiente para a manipulação dos pinos digitais do módulo Wemos D1 Mini. O uso do ESP8266 também permitiu a possibilidade de controle remoto e reconfiguração do sistema via rede Wi-Fi, o que abre portas para futuras expansões como monitoramento remoto, coleta de dados e integração com sensores de fluxo de veículos.

**7. Conclusão**

Conclui-se que o projeto atendeu satisfatoriamente aos objetivos propostos para o funcionamento de um controle de trafego, o sistema desmontou estabilidade e eficiência, futuras versões podem incluir sensores para melhorar a autonomia e adaptação ao tráfego real

Além de atingir os objetivos técnicos propostos, o projeto proporcionou uma excelente oportunidade de aprendizado prático promovendo o desenvolvimento de competência em montagem de circuitos, lógica condicional e temporização, além da importância da sincronização entre elementos de um sistema de controle

**8. Referencias**

Disponível em: https://docs.arduino.cc/arduino-cloud/guides/arduino-c/. Acesso em: 14 Junho.

Disponível em: https://www.microsoft.com/pt-br/windows/windows-11?ocid=cmm0rc11irk. Acesso em: 14 Junho.

Disponível em: https://pt.wikipedia.org/wiki/Windows\_11. Acesso em: 14 Junho.

Disponível em: https://pt.wikipedia.org/wiki/C%2B%2B. Acesso em: 14 Junho.

Disponível em: https://docs.arduino.cc/software/. Acesso em: 14 Junho.

**Anexo 1**

#define vm1 D2

#define amrl1 D3

#define vd1 D4

#define vm2 D5

#define amrl2 D6

#define vd2 D7

String funcionarios[2] = { "yuri", "malu" };

String senhas[2] = { "123", "321" };

unsigned long millisTarefa1 = millis();

unsigned long millisTarefa2 = millis();

int qual = 0;

int tempVar;

int msg = 0;

int verde1 = 25000;

int amarelo1 = 4000;

int vermelho1 = 29000;

int verde2 = 25000;

int amarelo2 = 4000;

int vermelho2 = 29000;

int temp1 = verde1;

int temp2 = amarelo1;

int temp3 = vermelho1;

int temp4 = verde2;

int temp5 = amarelo2;

int temp6 = vermelho2;

String tempClear;

String clear = "";

String identidade;

String chave;

String estado1 = "normal";

String estado2 = "normal";

String tempEstado = estado1;

bool resposta = false;

bool usuario = false;

bool senha = false;

bool autorizado = false;

bool confirmado1 = false;

bool confirmado2 = false;

bool sair = false;

void setup() {

  Serial.begin(9600);

  verde1 = verde1;

  amarelo1 = verde1 + amarelo1;

  vermelho1 = amarelo1 + vermelho1;

  vermelho2 = vermelho2;

  verde2 = vermelho2 + verde2;

  amarelo2 = verde2 + amarelo2;

}

void loop() {

  DeclaracaoPinos();

  if (estado1 == "normal") {

    FuncionamentoNormalSemaforo1();

  } else if (estado1 == "oscilando") {

    FuncionamentoOscilandoSemaforo1();

  }

  if (estado2 == "normal") {

    FuncionamentoNormalSemaforo2();

  } else if (estado2 == "oscilando") {

    FuncionamentoOscilandoSemaforo2();

  }

  if (autorizado) {

    Sistema();

  } else {

    Autenticacao();

  }

}

void Sistema() {

  if (Serial.available()) {

    if (resposta == false) {

      if (clear == "prog") {

        clear = Serial.readString();

        clear.trim();

        qual = clear.toInt();

      } else {

        clear = Serial.readString();

        clear.trim();

      }

    }

    if (clear == "prog" || tempClear == "prog") {

      if (qual == 0) {

        Serial.println("Que led sera configurado?\nSemáforo 1\n1-)Verde 2-)Vermelho\n\nSemáforo 2\n3-)Verde 4-)Vermelho\n\nFluxos\n5-)Fluxo padrão 6-)Fluxo acelerado\n\nTrocar para qual estado?\n7-)normal 8-)oscilando\n\n)Opções\n9-)Confimar 10-)Mostrar tempos\n11-)Forçar 12-)sair 13-)Sincronizar 14-)Instruções");

        tempClear = clear;

      }

      if (qual == 1) {

        AlterarVerde1();

      }

      if (qual == 2) {

        AlterarVermelho1();

      }

      if (qual == 3) {

        AlterarVerde2();

      }

      if (qual == 4) {

        AlterarVermelho2();

      }

      if (qual == 5) {

        FluxoPadrao();

      }

      if (qual == 6){

        FluxoAcelerado();

      }

      if (qual == 7) {

        EstadoNormal();

      }

      if (qual == 8) {

        EstadoOscilando();

      }

      if (qual == 9) {

        Confirmar();

      }

      if (qual == 10) {

        MostrarValores();

      }

      if (qual == 11) {

        Forcar();

      }

      if (qual == 12) {

        Sair();

      }

      if (qual == 13) {

        Sincronizar();

      }

      if (qual == 14){

        Instrucoes();

        }

      }

    }

  }

void Instrucoes(){

  Serial.println("O verde deve estar aceso por 20 a 40 segundos");

  Serial.println("O vermelho deve estar aceso por 20 a 80 segundos");

  qual = 0;

  resposta = false;

  tempClear = "";

  clear = "";

}

void Sincronizar(){

  Serial.println("Sincronizando");

  temp6 = temp1 + temp2;

  temp3 = temp4 + temp5;

  qual = 0;

  resposta = false;

  tempClear = "";

  clear = "";

}

void Sair(){

   if (!confirmado1) {

          if (!sair) {

            Serial.println("Sair sem confirmar?");

            sair = true;

            qual = 0;

            resposta = false;

            tempClear = "";

            clear = "";

          } else {

            Serial.println("saindo do modo de configuração");

            autorizado = false;

            identidade = "";

            chave = "";

            qual = 0;

            resposta = false;

            tempClear = "";

            clear = "";

          }

        } else {

          Serial.println("saindo do modo de configuração");

          autorizado = false;

          identidade = "";

          chave = "";

          qual = 0;

          sair = false;

          tempClear = "";

          clear = "";

        }

}

void Confirmar(){

  Serial.println("Confirmado");

  confirmado1 = true;

  confirmado2 = true;

  qual = 0;

  resposta = false;

  tempClear = "";

  clear = "";

}

void AlterarVerde1() {

  if (resposta == false) {

    Serial.println("quantos segundos o sinal verde estará aberto?");

    resposta = true;

  } else {

    tempVar = Serial.parseInt() \* 1000;

    if (tempVar >= 20000 &&tempVar <= 40000){

    temp1 = tempVar;

    tempVar = 0;

    Serial.print(temp1);

    Serial.println(" temp1");

    }else{

      Serial.println("O valor deve ser entre 20 e 40 segundos");

    }

    qual = 0;

    resposta = false;

    tempClear = "";

    clear = "";

  }

}

void AlterarVermelho1() {

  if (resposta == false) {

    Serial.println("quantos segundos o sinal vermelho estará aberto?");

    resposta = true;

  } else {

    tempVar = Serial.parseInt() \* 1000;

    if (tempVar >= 20000 &&tempVar <= 80000){

    temp3 = tempVar;

    tempVar = 0;

    Serial.print(temp3);

    Serial.println(" temp3");

    }else{

      Serial.println("O valor deve ser entre 20 e 80 segundos");

    }

    qual = 0;

    resposta = false;

    tempClear = "";

    clear = "";

  }

}

void AlterarVerde2() {

  if (resposta == false) {

    Serial.println("quantos segundos o sinal verde estará aberto?");

    resposta = true;

  } else {

    tempVar = Serial.parseInt() \* 1000;

    if (tempVar >= 20000 &&tempVar <= 40000){

    temp4 = tempVar;

    tempVar = 0;

    Serial.print(temp4);

    Serial.println(" temp4");

    }else{

      Serial.println("O valor deve ser entre 20 e 40 segundos");

    }

    qual = 0;

    resposta = false;

    tempClear = "";

    clear = "";

  }

}

void AlterarVermelho2() {

  if (resposta == false) {

    Serial.println("quantos segundos o sinal vermelho estará aberto?");

    resposta = true;

  } else {

    tempVar = Serial.parseInt() \* 1000;

    if (tempVar > 20000 &&tempVar <80000){

    temp6 = tempVar;

    tempVar = 0;

    Serial.print(temp6);

    Serial.println(" temp6");

    }else{

      Serial.println("O valor deve ser entre 20 e 80 segundos");

    }

    qual = 0;

    resposta = false;

    tempClear = "";

    clear = "";

  }

}

void EstadoNormal() {

  Serial.println("Alterando para estado normal");

  tempEstado = "normal";

  qual = 0;

  resposta = false;

  tempClear = "";

  clear = "";

}

void EstadoOscilando() {

  Serial.println("Alterando para estado oscilando");

  tempEstado = "oscilando";

  qual = 0;

  resposta = false;

  tempClear = "";

  clear = "";

}

void FluxoPadrao() {

  Serial.println("Alterando para fluxo padrão");

  temp1 = 40000;

  temp2 = 4000;

  temp3 = 45000;

  temp4 = 40000;

  temp5 = 4000;

  temp6 = 45000;

  qual = 0;

  resposta = false;

  tempClear = "";

  clear = "";

}

void FluxoAcelerado(){

  Serial.println("Alterando para fluxo padrão");

  temp1 = 20000;

  temp2 = 4000;

  temp3 = 25000;

  temp4 = 20000;

  temp5 = 4000;

  temp6 = 25000;

  qual = 0;

  resposta = false;

  tempClear = "";

  clear = "";

}

void MostrarValores() {

  Serial.println("TEMPOS");

  Serial.println("Semáforo 1");

  Serial.print((temp1 / 1000));

  Serial.println(" segundos no verde");

  Serial.print((temp2 / 1000));

  Serial.println(" segundos no amarelo");

  Serial.print((temp3 / 1000));

  Serial.println(" segundos no vermelho");

  Serial.println("\nSemáforo 2");

  Serial.print((temp4 / 1000));

  Serial.println(" segundos no verde");

  Serial.print((temp5 / 1000));

  Serial.println(" segundos no amarelo");

  Serial.print((temp6 / 1000));

  Serial.println(" segundos no vermelho");

  qual = 0;

  resposta = false;

  tempClear = "";

  clear = "";

}

void Forcar() {

  Serial.println("Valores sendo alterado agora");

  confirmado1 = true;

  confirmado2 = true;

  Desligatudo();

  millisTarefa1 = millis();

  millisTarefa2 = millis();

  Alteracao1();

  Alteracao2();

  qual = 0;

  resposta = false;

  tempClear = "";

  clear = "";

}

void Desligatudo() {

  digitalWrite(vm1, LOW);

  digitalWrite(vd1, LOW);

  digitalWrite(amrl1, LOW);

  digitalWrite(vd2, LOW);

  digitalWrite(amrl2, LOW);

  digitalWrite(vm2, LOW);

}

void Alteracao1() {

  if (confirmado1) {

    verde1 = temp1;

    amarelo1 = verde1 + temp2;

    vermelho1 = amarelo1 + temp3;

    estado1 = tempEstado;

    confirmado1 = false;

  }

}

void Alteracao2() {

  if (confirmado2) {

    vermelho2 = temp6;

    verde2 = vermelho2 + temp4;

    amarelo2 = temp5 + verde2;

    estado2 = tempEstado;

    confirmado2 = false;

  }

}

void DeclaracaoPinos() {

  pinMode(vm1, OUTPUT);

  pinMode(vd1, OUTPUT);

  pinMode(amrl1, OUTPUT);

  pinMode(vd2, OUTPUT);

  pinMode(amrl2, OUTPUT);

  pinMode(vm2, OUTPUT);

}

void FuncionamentoNormalSemaforo1() {

   if ((millis() - millisTarefa1) < verde1) {

    digitalWrite(vm1, LOW);

    digitalWrite(vd1, HIGH);

  } else if ((millis() - millisTarefa1) < amarelo1 && (millis() - millisTarefa1) > verde1) {

    digitalWrite(amrl1, HIGH);

    digitalWrite(vd1, LOW);

  } else if ((millis() - millisTarefa1) < vermelho1 && (millis() - millisTarefa1) > amarelo1) {

    digitalWrite(amrl1, LOW);

    digitalWrite(vm1, HIGH);

  } else if (millis() - millisTarefa1 > vermelho1) {

    millisTarefa1 = millis();

    Alteracao1();

    }

  }

void FuncionamentoOscilandoSemaforo1() {

  if ((millis() - millisTarefa1) < temp2) {

    digitalWrite(amrl1, HIGH);

    digitalWrite(vd1, LOW);

    digitalWrite(vm1, LOW);

  } else if (millis() - millisTarefa1 < (temp2 + 1000)) {

    digitalWrite(amrl1, LOW);

  } else {

    millisTarefa1 = millis();

    Alteracao1();

  }

}

void FuncionamentoNormalSemaforo2() {

  if ((millis() - millisTarefa2) < vermelho2) {

    digitalWrite(vm2, HIGH);

    digitalWrite(amrl2, LOW);

  } else if ((millis() - millisTarefa2) < verde2) {

    digitalWrite(vm2, LOW);

    digitalWrite(vd2, HIGH);

  } else if ((millis() - millisTarefa2) < amarelo2) {

    digitalWrite(amrl2, HIGH);

    digitalWrite(vd2, LOW);

  } else if ((millis() - millisTarefa2) > amarelo2) {

    millisTarefa2 = millis();

    digitalWrite(amrl2, LOW);

    Alteracao2();

  }

}

void FuncionamentoOscilandoSemaforo2() {

  if ((millis() - millisTarefa2) < temp5) {

    digitalWrite(amrl2, HIGH);

    digitalWrite(vd2, LOW);

    digitalWrite(vm2, LOW);

  } else if (millis() - millisTarefa2 < (temp5 + 1000)) {

    digitalWrite(amrl2, LOW);

  } else {

    millisTarefa2 = millis();

    Alteracao2();

  }

}

void Autenticacao() {

  if (Serial.available()) {

    if (clear != "prog") {

      clear = Serial.readString();

      clear.trim();

    }

    if (clear == "prog") {

      if (!usuario) {

        Serial.println("Digite o usuário");

        usuario = true;

      } else if (!senha) {

        identidade = Serial.readString();

        identidade.trim();

        Serial.println("Digite a senha");

        senha = true;

      } else if (senha && usuario) {

        chave = Serial.readString();

        chave.trim();

        for (int i = 0; i < sizeof funcionarios / sizeof funcionarios[0]; i++) {

          String n = funcionarios[i];

          senha = false;

          usuario = false;

          clear = "";

          if (n == identidade && chave == senhas[i]) {

            autorizado = true;

            Serial.println("autorizado");

            clear = "";

            Serial.println("Antes de fazer alterações, veja as intruções \n\n");

            Instrucoes();

          }

        }

      }

    }

  }

}