Daniel Henrique Charlo, Murilo Della Coletta e Otávio Saturnino

Semáforo Inteligente

Limeira-SP

2026

FASE 1 — LEVANTAMENTO DE REQUISITOS – Murilo

Requisitos Funcionais
• RF01: O semáforo deve priorizar o fluxo com maior volume de veículos. • RF02: Deve existir tempo mínimo e máximo para cada sinal (verde, amarelo, vermelho). • RF03: Em caso de falha de comunicação IoT, o sistema deve entrar em modo seguro (ciclo fixo). • RF04: Veículos de emergência podem receber prioridade manual ou automática. • RF05: O sistema deve funcionar 24 horas/dia, mesmo em condições climáticas adversas. • RF06: Em queda de energia, o semáforo deve seguir protocolo de contingência. • RF07: O sistema deve identificar e tratar inconsistências nos dados recebidos dos sensores (valores fora do padrão ou leituras duplicadas). • RF08: O sistema deve permitir configuração manual dos tempos do semáforo para operação emergencial ou manutenção.

Requisitos Não Funcionais
• RNF01: Disponibilidade mínima de 99% • RNF02: Tempo de resposta ≤ 2 segundos • RNF03: Comunicação segura (autenticação e criptografia MQTT) • RNF04: Sistema compatível com Linux • RNF05: Logs protegidos contra acesso não autorizado • RNF06: Arquitetura escalável para novos cruzamentos

Histórias de usuário:

Motorista: o semáforo deve se adaptar ao fluxo de veículos, para reduzir congestionamentos.

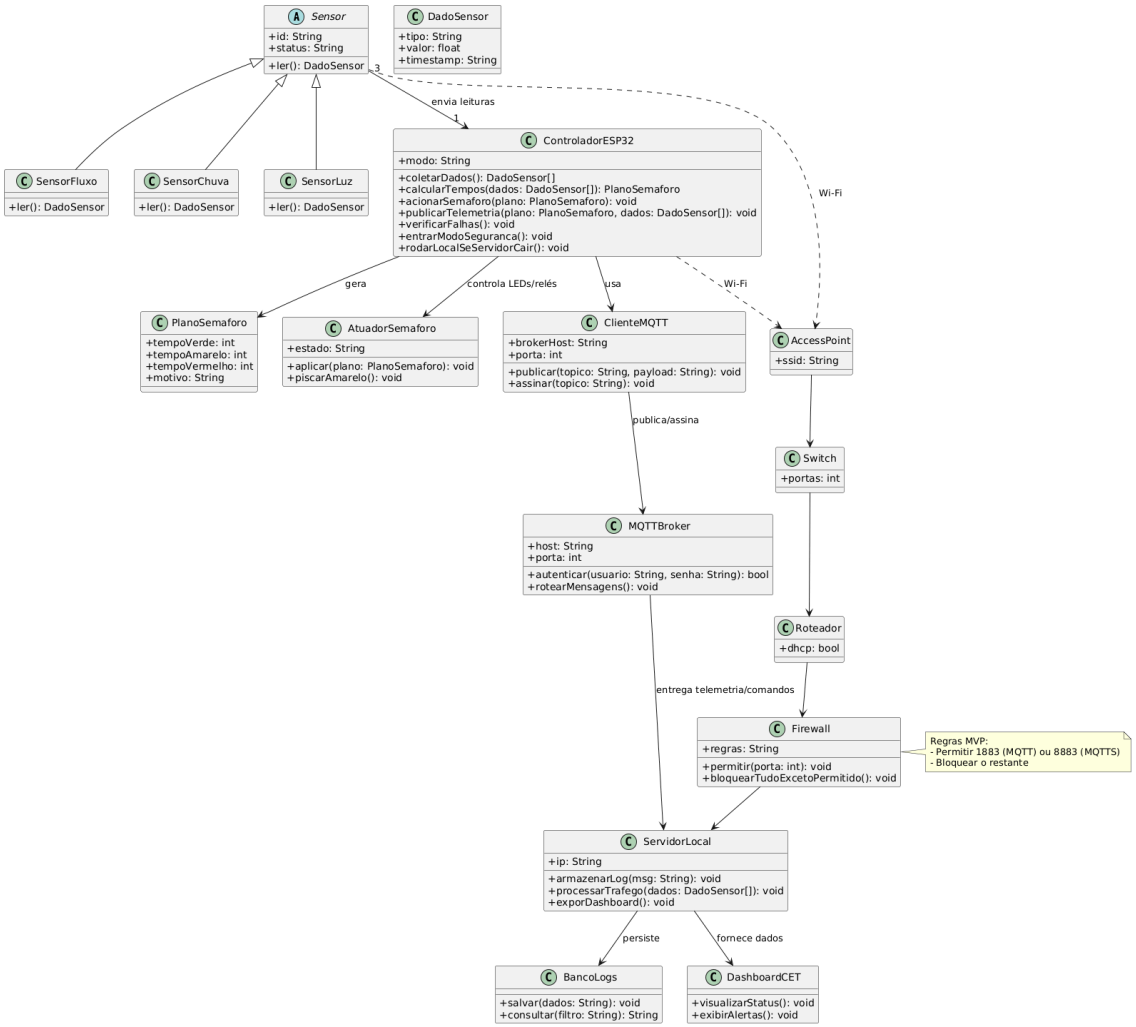
Administrador do sistema: o semáforo deve funcionar mesmo sem internet, para garantir segurança no trânsito.

Agente da CET: quero monitorar o status do cruzamento remotamente, para agir rapidamente em caso de falha.

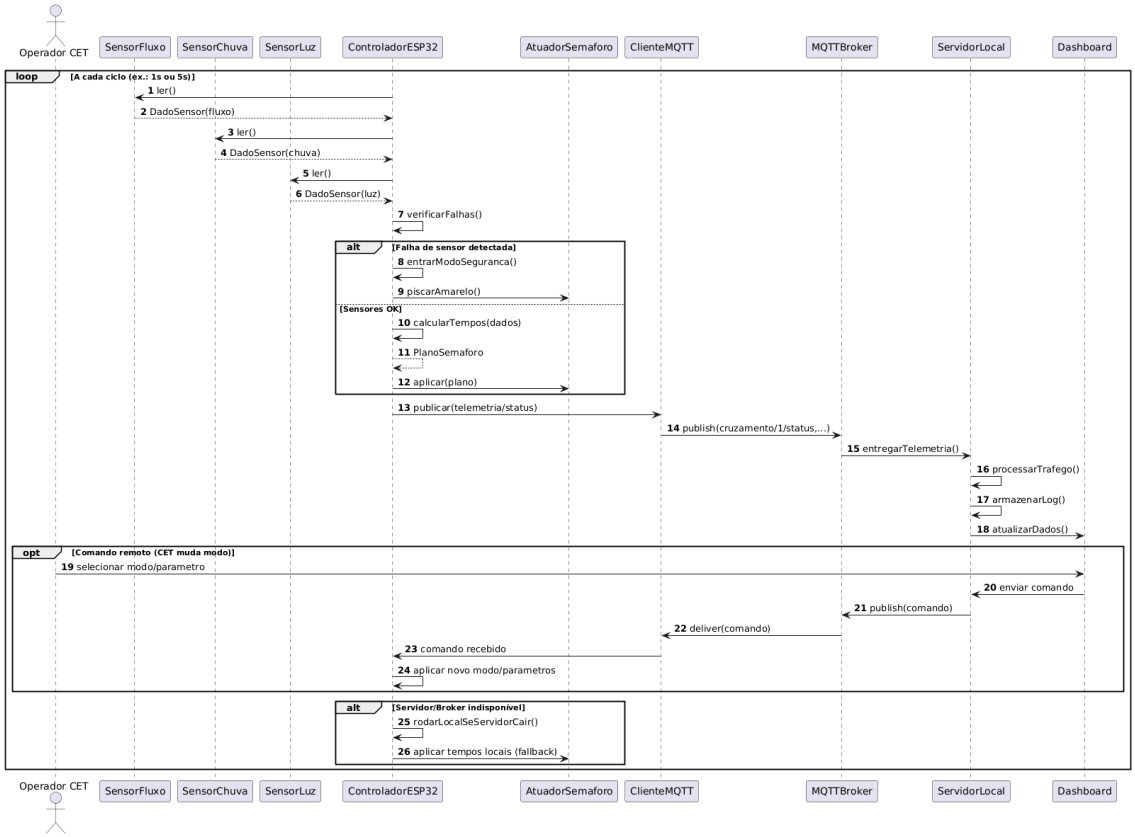
Priorização (MoSCoW):

Prioridade	Requisitos
Must Have	RF01, RF02, RF03, RF05, RF06, RF07, RNF01, RNF02, RNF03
Should have	RF04, RF08, RNF05
Could have	RNF06
Won't have	Nenhum requisito explicitamente descartado nesta fase.

FASE 2 — MODELAGEM DO SISTEMA E ARQUITETURA IoT – Daniel



- Sensores mandam dados para o ControladorESP32
- O Controlador gera um PlanoSemaforo e manda para o AtuadorSemaforo
- O Controlador também usa um ClienteMQTT para publicar/receber mensagens no MQTTBroker
- O ServidorLocal recebe do broker, processa, salva no BancoLogs e mostra no DashboardCET
- A infraestrutura (AP → Switch → Roteador → Firewall → Servidor) mostra por onde trafega a rede e onde a segurança entra.



- ESP32 lê Fluxo, Chuva e Luz
- ESP32 verifica falhas (sensor off / dado inválido)
- Se falhar → modo segurança (pisca amarelo)
- Se estiver OK → calcula tempos (verde/amarelo/vermelho) e aciona semáforo
- ESP32 publica status/telemetria via MQTT
- Broker entrega para o Servidor, que processa, salva logs e atualiza dashboard
- Se o servidor cair → ESP32 continua rodando localmente (fallback)

Equipamentos de rede:

- Roteador: organiza a rede (IPs/DHCP) e faz o roteamento entre os dispositivos (IoT ↔ servidor). Sem ele, a comunicação fica limitada ou bagunçada.
- Switch: conecta tudo por cabo (servidor, firewall, roteador, AP) com mais estabilidade e velocidade do que depender só de Wi-Fi.
- Access Point (AP): fornece Wi-Fi para o ESP32/sensores (quando não dá para passar cabo) e permite ter uma rede/SSID só do IoT.

- Firewall: protege a rede e o servidor. Bloqueia o que não precisa e libera só as portas usadas (ex.: 1883/8883 para o MQTT), reduzindo invasões e acessos indevidos.
- Servidor local: roda o broker (MQTT), processamento, logs e dashboard. Centraliza monitoramento e registro; mesmo se cair, o controlador pode continuar localmente, mas o servidor é essencial para supervisão e histórico.

ESP32 ou Arduino (controlador)

- Escolha: ESP32
- Por quê: já tem Wi-Fi/Bluetooth embutido, então conversa com o servidor/broker sem precisar de módulos extras; é mais prático para o MVP IoT.
- Arduino (quando faria sentido): se fosse um projeto bem simples e todo cabeado/sem rede, ou se já tivesse shield/módulo pronto.

MQTT ou TCP/IP “puro” (protocolo)

- Escolha: MQTT
- Por quê: foi feito pra IoT, é leve e aguenta melhor instabilidade. Dá para publicar/assinar em tópicos, facilita monitoramento e comandos, e você não precisa programar toda a lógica de conexão “na mão”.
- TCP/IP puro (quando faria sentido): se fosse um sistema muito simples (cliente manda dados direto pra uma API) e sem precisar de pub/sub.

Broker MQTT local (ex.: Mosquitto) ou sem broker

- Escolha: ter broker local
- Por quê: ele é o “central” das mensagens. Sensores/controlador não precisam falar direto com cada serviço; isso deixa o sistema mais organizado, escalável e fácil de controlar acesso (ACL).

Topologia Mesh + Cliente-Servidor

- Escolha: Mesh (sensores→controlador) + Cliente-Servidor (controlador→servidor)
- Por quê:
  - Mesh: ajuda quando tem queda/interferência, porque reduz ponto único e melhora alcance/robustez.

- Cliente-Servidor: é o mais simples e padrão pra mandar dados para o servidor e receber comandos (via broker).

Servidor local (no cruzamento) ou só nuvem

- Escolha: servidor local
- Por quê: o cruzamento é crítico: depender de internet é arriscado. Local dá menos latência e mantém logs/monitoramento mesmo se a conexão externa cair.

Rodar algoritmo local no controlador (fallback)

- Escolha: sim, sempre
- Por quê: no cenário o servidor pode cair/ficar lento. O semáforo não pode parar; então o controlador mantém o ciclo local e só usa o servidor pra supervisão/ajustes.

Portas 1883 vs 8883

- Escolha: 1883 no laboratório / 8883 em produção
- Por quê: 8883 (TLS) é o ideal por segurança, mas no MVP de laboratório às vezes usam 1883 por simplicidade (desde que a rede esteja isolada e com firewall).

Se você me disser qual seu grupo quer defender (ex.: “vamos usar Arduino e TCP/IP”), eu adapto essas justificativas pra bater com essa escolha sem perder coerência com o cenário

FASE 3 — SISTEMA OPERACIONAL E SEGURANÇA - Murilo

3.1. Escolher um SO para o servidor do semáforo:

Item	Windows Server	Ubuntu Server
Custo	Licença paga, custo elevado para prefeitura	Gratuito (open source)
Segurança	Boa segurança, porém alvo frequente de ataques	Alta segurança, atualizações constantes e menor superfície de ataque
Suporte a IoT	Suporte limitado e dependente de softwares proprietários	Excelente suporte a MQTT, Node-RED, Python, Docker e IoT em geral

Justificativa:

O Ubuntu Server foi escolhido por apresentar:

- Menor custo (importante para órgãos públicos)
- Melhor integração com soluções IoT
- Grande comunidade de suporte
- Facilidade para automação, monitoramento e segurança
- Estabilidade para operação contínua do semáforo inteligente

3.2. Configurar em laboratório/VM:

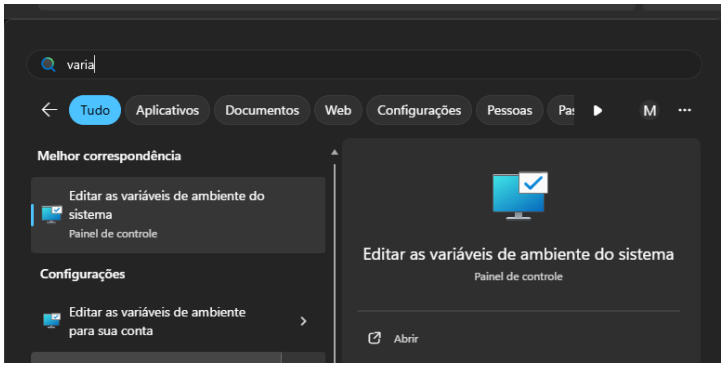
- Criar usuários:
- Configurações → Contas
- Outras pessoas → Adicionar outra pessoa a este PC
- Escolha Não tenho as informações de entrada dessa pessoa
- Adicionar um usuário sem conta da Microsoft
- Defina:
    - Nome de usuário
    - Senha
    - Dica de senha

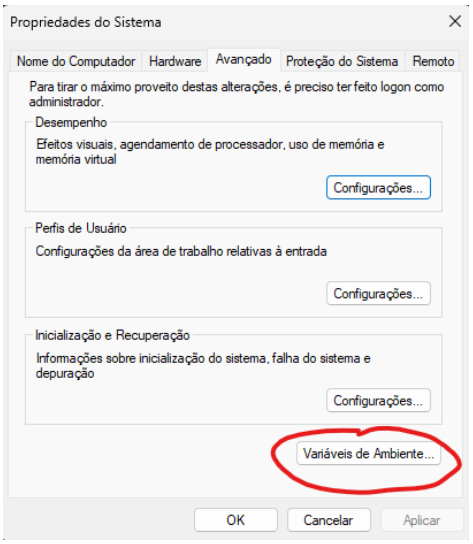
Definir permissões e configurar pastas compartilhadas:

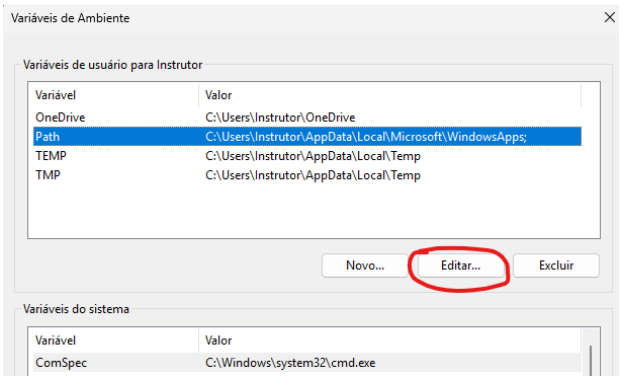
Defina quais programas e pastas cada usuário tem acesso.

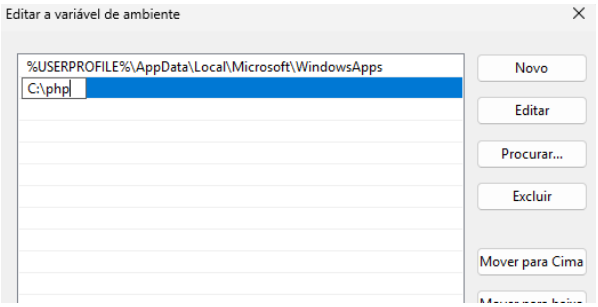
Exemplo: O usuário 1 tem acesso ao VScode, Google Chrome e pastas criadas em sua área de trabalho. O usuário 2 tem acesso a todos os programas e à todos os arquivos do computador

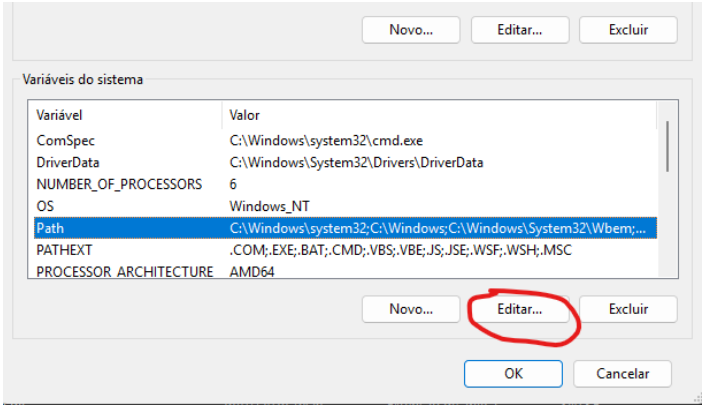
Ajustar variáveis de ambiente:

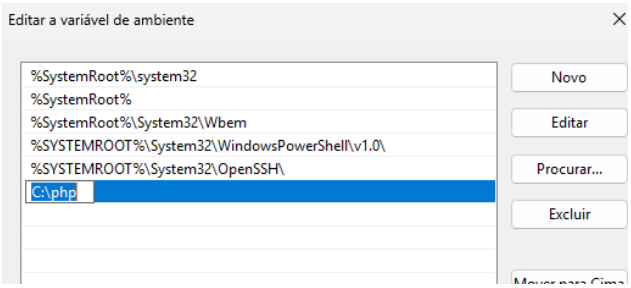



Propriedades do Sistema
Nome do Computador
Hardware
Avançado
Proteção do Sistema
Remoto
Para tirar o máximo proveito destas alterações, é preciso ter feito logon como administrador.
Desempenho
Efeitos visuais, agendamento de processador, uso de memória e memória virtual
Configurações...
Perfis de Usuário
Configurações da área de trabalho relativas à entrada
Configurações...
Inicialização e Recuperação
Informações sobre inicialização do sistema, falha do sistema e depuração
Configurações...
Variáveis de Ambiente...
OK
Cancelar
Aplicar


Variáveis de Ambiente
Variáveis de usuário para Instrutor
Variável
Valor
OneDrive
C:\Users\Instrutor\OneDrive
Path
C:\Users\Instrutor\AppData\Local\Microsoft\WindowsApps;
TEMP
C:\Users\Instrutor\AppData\Local\Temp
TMP
C:\Users\Instrutor\AppData\Local\Temp
Novo...
Editar...
Excluir
Variáveis do sistema
Variável
Valor
ComSpec
C:\Windows\system32\cmd.exe


Editar a variável de ambiente
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\php
Novo
Editar
Procurar...
Excluir
Mover para Cima


Novo...
Editar...
Excluir
Variáveis do sistema
Variável
Valor
ComSpec
C:\Windows\system32\cmd.exe
DriverData
C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS
6
OS
Windows_NT
Path
C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;...
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE
AMD64
Novo...
Editar...
Excluir
OK
Cancelar



Testar navegação via terminal:

- Abrir o Prompt de Comando ou PowerShell no Windows.
- Listar os diretórios disponíveis para verificar o acesso ao sistema de arquivos.
- Navegar até o disco principal e acessar a pasta criada para compartilhamento.
- Verificar se o conteúdo da pasta pode ser visualizado sem erros.
- Testar o acesso à pasta compartilhada utilizando o caminho de rede da máquina.
- Confirmar que o acesso ocorre conforme as permissões configuradas.

Configurar firewall permitindo somente portas usadas (ex.: 1883 para MQTT):

- Acessar o Firewall do Windows com Segurança Avançada.
- Definir a política padrão para bloquear conexões de entrada não autorizadas.
- Criar uma nova regra de entrada.
- Informar o protocolo e a porta utilizada pelo serviço (ex.: porta 1883 para MQTT).
- Permitir a conexão e aplicar a regra ao perfil de rede do laboratório.
- Salvar a configuração e verificar se a regra está ativa.
- Testar o serviço para confirmar que apenas a porta liberada está acessível.

3.3. Criar uma mini Política de Segurança (PSI)

Senha:

- Uso obrigatório de letras maiúsculas, minúsculas, números e caracteres especiais
- Mínimo de 8 caracteres

Acesso:

- Administrador: terá acesso a tudo e poderá gerenciar e alterar valores
- Funcionário: pode visualizar trânsito e valores, mas não pode alterá-los
- Cliente: pode apenas visualizar o nível do tráfego

Backup:

- Backup automático diário dos dados de tráfego
- Backup semanal completo do sistema
- Armazenamento em servidor externo seguro

Procedimentos em Caso de Falha no Sensor:

- Sistema entra em modo seguro (tempo fixo de semáforo)
- Alerta automático enviado à CET
- Registro da falha
- Equipe técnica será acionada

Proteção contra Engenharia Social:

- Treinamento dos operadores
- Proibição de compartilhamento de senhas
- Confirmação de identidade para qualquer solicitação técnica
- Alertas sobre acessos suspeitos