



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Faculdade de Engenharia

Departamento de Sistemas e Computação

Controle de Processos por Computador

Turma 1

Philippe Macêdo Borges Ramos

Trabalho Final

Parte 3: Simulação do Sistema de Controle

Professor: Luigi Maciel Ribeiro

Data de Envio: 05/12/2025

Rio de Janeiro

2025

1 Introdução

Após a definição dos requisitos de funcionalidade, desempenho e segurança e a modelagem teórica do sistema através de diagrama de blocos, o objetivo desta etapa é comprovar a viabilidade técnica da solução proposta. O projeto consiste em um Sistema de Controle Térmico para Servidores, operando em arquitetura distribuída com integração IoT.

Uma correção foi feita no diagrama de blocos apresentada no relatório da PARTE 2, como forma de melhorar o controle do usuário através da Dashboard, foi adicionada a relação em que o usuário consegue enviar uma mensagem de emergência para o módulo da lógica de segurança e assim prevenir danos ao sistema de forma remota. A Figura 1 apresenta o diagrama atualizado.

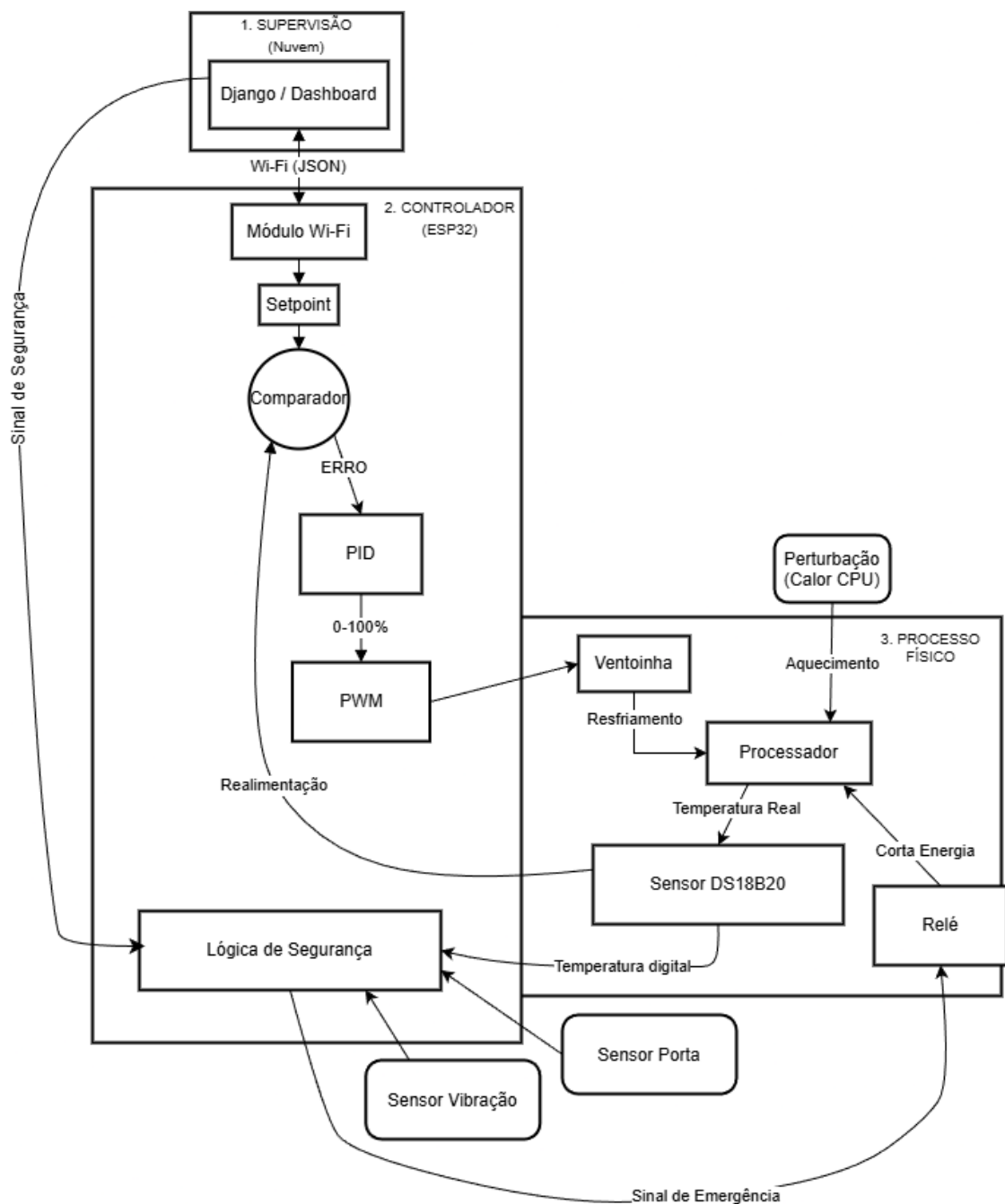


Figura 1: Diagrama de Blocos Atualizado

2 Configuração do Ambiente de Simulação

O sistema de controle térmico foi validado através da integração de duas plataformas de simulação:

1. **Wokwi:** Responsável por simular o microcontrolador ESP32, sensores e atuadores em tempo real;
2. **PythonAnywhere:** Responsável pela hospedagem do servidor Django, banco de dados e interface de supervisão.

2.1 Circuito

O circuito foi montado conforme a modelagem apresentada na PARTE 2, a Figura 2 utilizando:

1. Controlador ESP32:

2. Sensores:

- **DS18B20:** Configurado para atuar como a "carga térmica" na CPU;
- **Sensores Digitais:** Chave deslizante simulando a porta do gabinete e os botões simulando o sensor de vibração e o botão de turbo.

3. Atuadores:

- **LED Azul (PWM):** Representa a ventoinha de resfriamento e a intensidade luminosa varia linearmente com o *Duty Cycle* calculado pelo PID;
- **Módulo Relé:** Atua como o dispositivo de segurança, cortando a energia da carga (LED Amarelo) em situações críticas.

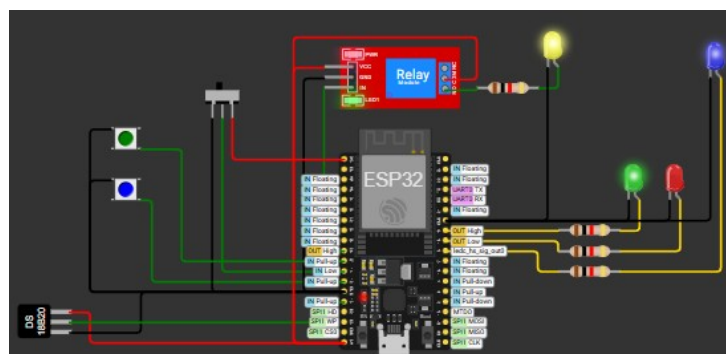


Figura 2: Circuito montado no simulador Wokwi

3 Modelagem do Processo e Algoritmo

Para que a simulação apresentasse um sistema térmico com a fidelidade próximo ao real, foi necessário implementar uma modelagem física via software, pois o simulador não possui termodinâmica nativa.

A modelagem física foi da inércia térmica, onde no firmware, foi implementado uma equação diferencial para simular a capacitância térmica do servidor. A temperatura lida pelo sistema não é o valor direto do slider do sensor térmico, mas sim o resultado da interação entre o calor gerado e o resfriamento:

$$T_{nova} = T_{atual} + \alpha * (T_{setpoint} - T_{atual})$$

O coeficiente de inércia térmica é representado por $\alpha = 0,1$. Isso garante que a temperatura não mude instantaneamente, criando curvas mais suaves e realistas, fundamentais para testar a ação do PID.

O controle nesta simulação foi executado por um algoritmo PID com os seguintes características:

- **Ganhos:** $K_p = 2.0$ (resposta rápida), $K_i = 0.8$ (correção do erro estático), $K_d = 5.0$ (amortecimento);
- **Janela de atuação:** A integral só é ativada quando o erro está dentro de uma faixa controlável, otimizando a estabilidade.

4 Análise dos Resultados

4.1 Desempenho do Controle

O gráfico apresentado na Figura 3 foi obtido no Dashboard e demonstra o comportamento dinâmico do sistema de controle térmico durante uma operação de resfriamento forçado.

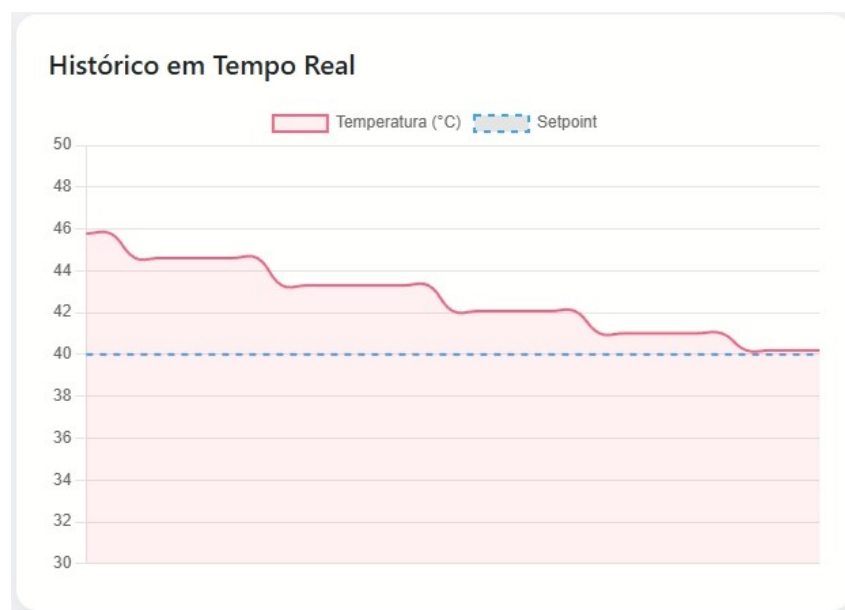


Figura 3: Histórico da temperatura real em relação ao Setpoint

Analisando o gráfico é possível perceber que o setpoint escolhido para a simulação foi de 40°C e esta imagem é um recorte de quando o sistema estava em um estado de aquecido, com a temperatura próxima dos 46°C. Isso simula um cenário de alta carga de processamento na CPU, esta carga gera um erro positivo que é detectado pelo algoritmo PID, em seguida o PID atua sobre a ventoinha aumentando a sua potência. Ainda nesse gráfico é possível observar uma redução da temperatura com o perfil em decidas suaves, refletindo a interação entre a inércia térmica modelada e a ação corretiva do controlador ajustando o PWM para forçar o resfriamento. No fim do período monitorado, a temperatura converge para o setpoint e estabiliza garantindo um resfriamento eficiente e seguro para o hardware.

5 Segurança

O sistema de segurança opera em paralelo ao PID e foi validado através da injeção de falhas no sistema, essas falhas foram:

- **Overshoot Crítico:** Ao forçar a temperatura 20°C acima do setpoint, o sistema identificou o risco térmico e desligou o relé, como apresentado na Figura 4

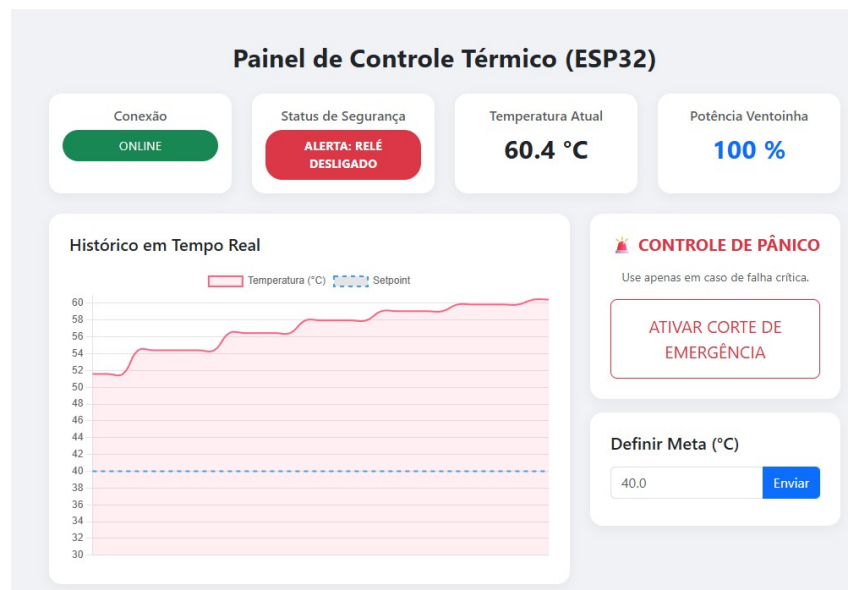


Figura 4: Overshoot Crítico

- **Violação Física:** A abertura do sensor da porta ou acionamento da vibração provocam o corte de energia à carga, como apresentado na Figura 5

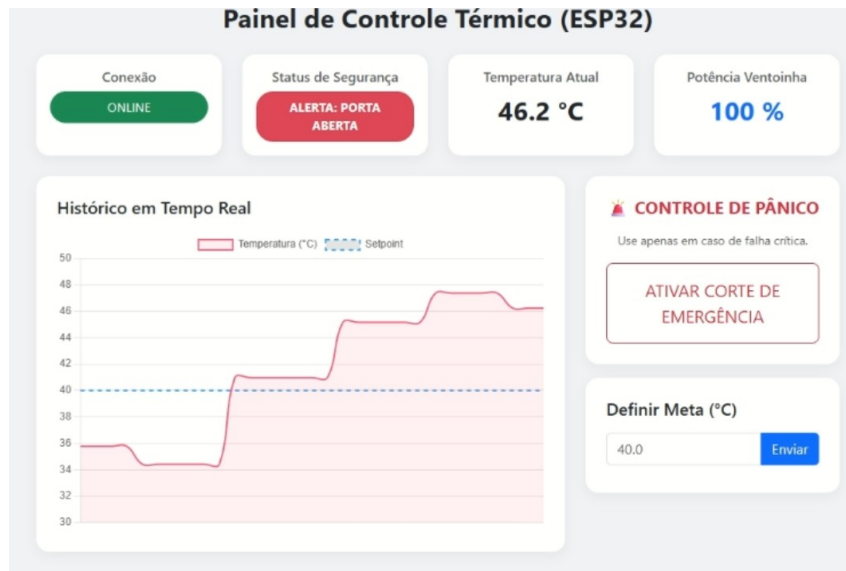


Figura 5: Violação Física

- **Intervenção Remota:** O acionamento do botão de "pânico" no Dashboard foi recebido pelo ESP32 via JSON, demonstrando a eficácia da comunicação IoT descendente, como apresentado na Figura 6



Figura 6: Intervenção Remota

6 Integração IoT e Interface

A integração com o servidor Django provou-se robusta e funcional. Pois, a atualização de telemetria a cada 10 segundos mostrou-se adequada para o monitoramento térmico, não sobrecarregando o microcontrolador nem a rede. Além disso, o Dashboard permitiu não apenas a visualização, mas também a alteração dinâmica do setpoint e o acionamento de emergência, cumprindo o requisito de controle distribuído.

A Figura 7 mostra como ficou o painel admin

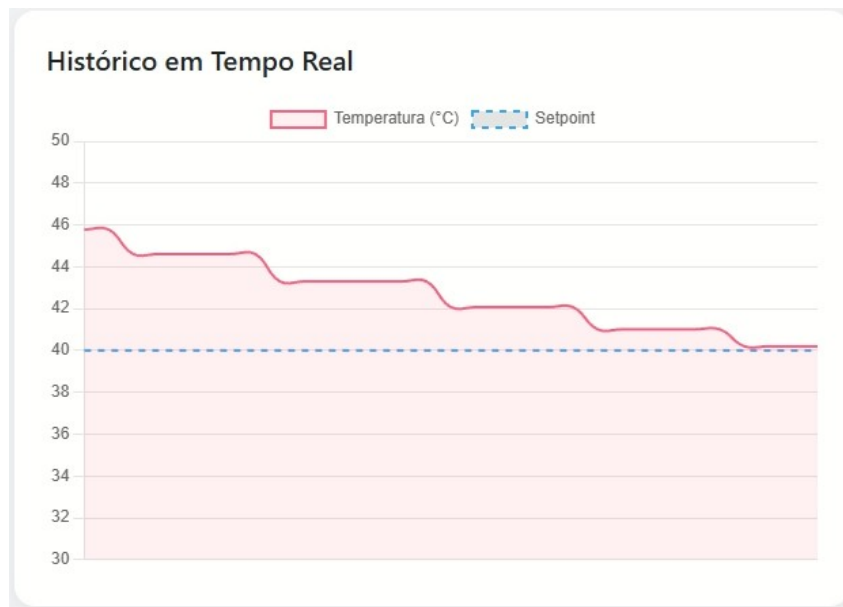


Figura 7: Histórico da temperatura real em relação ao Setpoint

7 Conclusão

O sistema demonstrou-se capaz de manter a temperatura do servidor estável em 40°C diante de perturbações, e as camadas de segurança atuaram de maneira correta para mitigar riscos físicos e operacionais. A arquitetura proposta, separando o controle crítico da supervisão, mostrou-se uma solução segura e escalável.