

Relatório Parcial do Projeto: Sistema de Monitoramento Distribuído de Temperatura

IDENTIFICAÇÃO DO PROJETO

Nome do Projeto: Sistema de Monitoramento Distribuído de Temperatura

Curso: Engenharia da Computação

Disciplina: Sistemas Distribuídos

Instituição: Universidade Federal do Maranhão – UFMA

Alunos responsáveis: Ana Patrícia Garros Viegas, Ana Poliana Mesquita de Jesus de Sousa, Gustavo Antonio Silva Rocha, Leonardo dos Santos Pereira, Welyab da Silva Paula.

Professor orientador: Luiz Henrique Neves Rodrigues

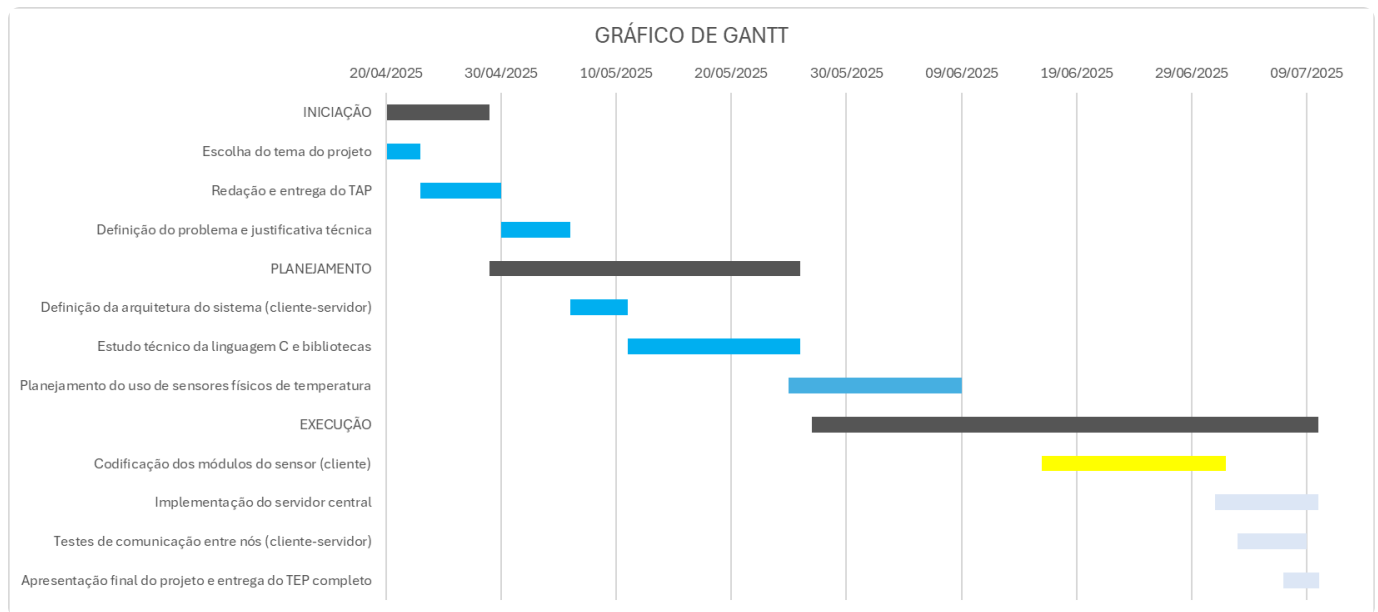
Data de início: 20/04/2025

Data prevista de término: 14/07/2025

1. Introdução

Este relatório parcial detalha o desenvolvimento do projeto "Sistema de Monitoramento Distribuído de Temperatura", realizado como parte da disciplina de Sistemas Distribuídos. O objetivo principal deste projeto é conceber, implementar e testar um sistema capaz de coletar dados de temperatura de múltiplos pontos e centralizá-los para monitoramento. Atualmente, o projeto encontra-se em andamento, com a fase de codificação dos módulos do sensor (cliente) em progresso.

2. Fases do Projeto



2.1. Iniciação

2.1.1. Escolha do Tema do Projeto

2.1.2. Redação e Entrega do TAP (Termo de Abertura do Projeto)

2.1.3. Definição do Problema e Justificativa Técnica

2.2. Planejamento

2.2.1. Definição da Arquitetura do Sistema (Cliente-Servidor)

2.2.2. Estudo Técnico da Linguagem C e Bibliotecas

2.2.3. Planejamento do Uso de Sensores Físicos de Temperatura

2.3. Execução

2.3.1. Codificação dos Módulos do Sensor (Cliente)

3. Próximos Passos

O projeto continuará com a implementação do servidor central, testes de comunicação entre os nós e a apresentação final.

4. Conclusão Parcial

O tema "Sistema de Monitoramento Distribuído de Temperatura" foi selecionado devido à sua relevância prática e teórica para a disciplina de Sistemas Distribuídos. A escolha permitiu explorar conceitos fundamentais como comunicação cliente-servidor, sincronização de dados e resiliência em ambientes distribuídos. Além disso, a aplicação de monitoramento de temperatura possui um vasto campo de uso em diversos setores, desde a indústria até a agricultura, tornando o projeto não apenas academicamente enriquecedor, mas também potencialmente aplicável no mundo real. A complexidade intrínseca de lidar com múltiplos sensores e a necessidade de garantir a integridade e a disponibilidade dos dados em um ambiente distribuído apresentaram desafios que se alinhavam perfeitamente com os objetivos de aprendizado da disciplina.

2.1.2. Redação e Entrega do TAP (Termo de Abertura do Projeto)

O Termo de Abertura do Projeto (TAP) foi elaborado como o documento formal que estabelece a existência do projeto e fornece uma visão geral de seus objetivos, escopo, partes interessadas e requisitos iniciais. A redação do TAP envolveu a descrição detalhada do que o sistema de monitoramento de temperatura distribuído deveria alcançar, incluindo a capacidade de coletar dados de temperatura de diferentes fontes, transmiti-los para um servidor central e permitir a visualização ou análise desses dados. A entrega do TAP marcou o início formal das atividades do projeto, servindo como um guia para as fases subsequentes e garantindo o alinhamento entre a equipe do projeto e os requisitos da disciplina.

2.1.3. Definição do Problema e Justificativa Técnica

O problema central abordado por este projeto é a necessidade de monitorar temperaturas em ambientes distribuídos de forma eficiente e confiável. Em muitos cenários, como grandes instalações industriais, armazéns ou até mesmo em aplicações agrícolas, a temperatura é um fator crítico que afeta a qualidade, segurança e eficiência. A coleta manual de dados de temperatura é inviável em larga escala, propensa a erros e não oferece monitoramento em tempo real. Sistemas centralizados tradicionais podem falhar em ambientes distribuídos devido a problemas de escalabilidade, latência e pontos únicos de falha.

A justificativa técnica para o desenvolvimento de um sistema distribuído reside na sua capacidade de superar essas limitações. Um sistema distribuído permite que múltiplos sensores (clientes) operem de forma autônoma, coletando dados localmente e enviando-os para um servidor central. Isso garante maior robustez, pois a falha de um único sensor não compromete todo o sistema. Além disso, a arquitetura cliente-servidor facilita a escalabilidade, permitindo a adição de novos sensores conforme a necessidade, sem a necessidade de reestruturar todo o sistema. A escolha da linguagem C para a codificação dos módulos do sensor e a implementação do servidor central justifica-se pela sua eficiência, baixo nível de abstração e controle direto sobre o hardware, características essenciais para sistemas embarcados e aplicações de tempo real, como o monitoramento de temperatura. A utilização de sensores físicos de temperatura garante a precisão e a confiabilidade dos dados coletados, fundamentais para a tomada de decisões baseada nas informações do sistema.

2.2. Planejamento

2.2.1. Definição da Arquitetura do Sistema (Cliente-Servidor)

A fase de planejamento foi crucial para estabelecer as bases do sistema de monitoramento distribuído. A primeira etapa consistiu na definição da arquitetura do sistema, optando-se pelo modelo cliente-servidor. Esta escolha foi fundamentada na sua robustez, escalabilidade e na clara separação de responsabilidades entre os componentes. No contexto deste projeto, os 'clientes' são os módulos de sensor, responsáveis pela coleta de dados de temperatura em pontos específicos. Cada cliente opera de forma autônoma, lendo dados de um sensor físico e, em seguida, transmitindo-os para um 'servidor' central. O servidor, por sua vez, é o componente responsável por receber, processar, armazenar e disponibilizar esses dados para visualização ou análise. Essa arquitetura permite que o sistema seja distribuído geograficamente, com clientes localizados em diferentes áreas, todos reportando a um único ponto central. A comunicação entre cliente e servidor foi planejada para ser assíncrona e resiliente, garantindo que a perda temporária de conexão de um cliente não afete a operação geral do sistema. A escolha de um protocolo de comunicação leve e eficiente, como TCP/IP, foi considerada para otimizar o tráfego de rede e garantir a entrega confiável dos dados.

2.2.2. Estudo Técnico da Linguagem C e Bibliotecas

Paralelamente à definição arquitetural, foi realizado um estudo técnico aprofundado da linguagem de programação C e das bibliotecas pertinentes. A escolha do C para a implementação dos módulos de sensor (cliente) e do servidor central foi motivada pela sua eficiência, performance e capacidade de manipulação de baixo nível, características essenciais para sistemas embarcados e aplicações que exigem controle preciso de hardware e recursos. O estudo abrangeu:

- **Fundamentos da Linguagem C:** Revisão de conceitos como gerenciamento de memória (ponteiros, alocação dinâmica), estruturas de dados, manipulação de arquivos e programação de sistemas. Isso garantiu que a equipe tivesse uma base sólida para desenvolver código otimizado e robusto.
- **Programação de Sockets:** Um foco significativo foi dado ao estudo das APIs de sockets em C (BSD Sockets), que são a base para a comunicação em rede. Foram explorados os diferentes tipos de sockets (stream e datagrama), os modelos de programação (bloqueante e não bloqueante), e as funções essenciais para estabelecer conexões (socket(), bind(), listen(), accept(), connect()), enviar e receber dados (send(), recv()). A compreensão aprofundada desses conceitos foi vital para a implementação da comunicação cliente-servidor.
- **Bibliotecas para Sensores:** Pesquisa e avaliação de bibliotecas existentes para interface com sensores de temperatura. Embora o projeto pudesse simular a leitura de sensores, o estudo incluiu a análise de como bibliotecas como `wiringPi` (para Raspberry Pi) ou drivers de baixo nível poderiam ser integrados para leituras reais de sensores como o DS18B20 ou termistores. Isso preparou o terreno para uma eventual implementação com hardware físico.
- **Multithreading e Concorrência:** Para o servidor central, o estudo incluiu conceitos de multithreading (usando `pthread` no Linux) para permitir que o servidor lidasse com múltiplas conexões de clientes simultaneamente. Foram abordados tópicos como criação e gerenciamento de threads, sincronização (mutexes, semáforos) e comunicação entre threads, visando garantir a escalabilidade e a responsividade do servidor.

Este estudo técnico não apenas capacitou a equipe com as ferramentas necessárias para a codificação, mas também permitiu a identificação de potenciais desafios e a formulação de

estratégias para superá-los durante a fase de execução.

2.2.3. Planejamento do Uso de Sensores Físicos de Temperatura

Embora a implementação inicial do projeto pudesse envolver a simulação de leituras de temperatura, o planejamento incluiu a consideração do uso de sensores físicos de temperatura. Esta etapa foi fundamental para garantir que o design do sistema fosse compatível com hardware real e que as futuras expansões pudessem ser realizadas sem grandes reestruturações. O planejamento envolveu:

- **Seleção de Sensores:** Pesquisa sobre os tipos de sensores de temperatura disponíveis no mercado, suas características (precisão, faixa de medição, custo, interface de comunicação) e adequação para um sistema distribuído. Sensores como o DS18B20 (interface One-Wire), termistores (requerem um ADC - Conversor Analógico-Digital) e DHT11/DHT22 (temperatura e umidade) foram avaliados. A escolha recaiu sobre sensores que oferecessem uma boa relação custo-benefício e facilidade de integração com microcontroladores ou computadores de placa única (como Raspberry Pi ou ESP32), que seriam os hospedeiros dos módulos cliente.
- **Interfaceamento com Hardware:** Definição das estratégias para interfacear os sensores com os dispositivos que executariam o código cliente. Isso incluiu a compreensão dos pinouts dos sensores, dos requisitos de alimentação e dos protocolos de comunicação (e.g., One-Wire, I2C, SPI, ou leitura analógica). O planejamento também considerou a necessidade de circuitos de condicionamento de sinal, se aplicável, para garantir leituras precisas e estáveis.
- **Calibração e Precisão:** Discussão sobre a importância da calibração dos sensores para garantir a precisão das leituras. Embora a calibração em si não fosse parte da fase de planejamento, a sua necessidade foi reconhecida e documentada como um requisito para futuras implementações com hardware real. A precisão dos sensores foi um fator chave na seleção, visando atender aos requisitos de monitoramento de temperatura do projeto.
- **Localização e Instalação:** Considerações sobre a localização física dos sensores em um ambiente distribuído. Isso incluiu aspectos como a proteção dos sensores contra intempéries, a otimização da cobertura da área a ser monitorada e a minimização de interferências externas que pudessem afetar as leituras. Embora não fosse uma etapa

de implementação física, o planejamento conceitual da instalação foi importante para validar a viabilidade do sistema em cenários reais.

Este planejamento detalhado do uso de sensores físicos garantiu que o projeto não ficasse restrito a uma simulação puramente teórica, mas que tivesse uma base sólida para uma implementação prática e escalável no futuro.

2.3. Execução

2.3.1. Codificação dos Módulos do Sensor (Cliente)

A fase de execução iniciou-se com a codificação dos módulos do sensor, que representam os clientes na arquitetura distribuída. Cada módulo cliente foi projetado para ser leve, eficiente e autônomo, com a responsabilidade principal de coletar dados de temperatura e transmiti-los ao servidor central. A implementação foi realizada em linguagem C, aproveitando as vantagens de baixo nível e performance que a linguagem oferece. Os principais componentes e funcionalidades desenvolvidos nos módulos cliente incluíram:

- **Leitura do Sensor:** Implementação da lógica para interagir com o sensor de temperatura. Para fins de demonstração e teste, uma função de simulação de leitura de temperatura foi inicialmente utilizada, gerando valores aleatórios dentro de uma faixa realista. No entanto, o código foi estruturado para permitir a fácil integração com drivers de sensores físicos reais (como DS18B20 ou termistores) no futuro, utilizando as interfaces de hardware apropriadas (e.g., One-Wire, I2C, ou ADC).
- **Formatação dos Dados:** Os dados de temperatura lidos foram formatados em um padrão predefinido (e.g., JSON ou um formato de texto simples com delimitadores) para facilitar a interpretação pelo servidor. Essa formatação incluiu, além do valor da temperatura, um identificador único para o sensor (ID do cliente) e um timestamp, garantindo a rastreabilidade e a ordenação dos dados.
- **Comunicação com o Servidor:** Utilização da API de sockets em C para estabelecer e manter uma conexão TCP/IP com o servidor central. O módulo cliente foi configurado para tentar reconectar-se automaticamente em caso de falha na conexão, garantindo a resiliência do sistema. A transmissão dos dados foi implementada de forma a enviar pacotes de dados de temperatura em intervalos regulares, otimizando o uso da rede e a carga no servidor.

- **Tratamento de Erros e Robustez:** Incorporação de mecanismos de tratamento de erros para lidar com falhas de leitura do sensor, problemas de comunicação de rede e outras exceções. Isso incluiu a implementação de retries para envio de dados e logging de eventos para depuração e monitoramento do comportamento do cliente. A robustez foi uma preocupação central, visando garantir que os módulos cliente pudessem operar de forma contínua e confiável em ambientes de produção.

A codificação dos módulos cliente foi realizada de forma modular, permitindo a fácil manutenção e a adição de novas funcionalidades, como a coleta de outros tipos de dados (umidade, pressão) ou a implementação de lógicas de processamento local antes do envio dos dados ao servidor.

4. Conclusão Parcial

Até o momento, o projeto "Sistema de Monitoramento Distribuído de Temperatura" progrediu significativamente, com as fases de Iniciação e Planejamento concluídas e a codificação dos módulos do sensor (cliente) em um estágio avançado. A arquitetura cliente-servidor foi definida, e o estudo técnico da linguagem C e bibliotecas relevantes forneceu a base necessária para a implementação. A codificação dos módulos cliente demonstrou a capacidade de coletar e formatar dados de temperatura, preparando o terreno para a integração com o servidor central. Os próximos passos envolverão a implementação do servidor, testes de comunicação e a finalização do projeto, que promete ser uma solução robusta e eficiente para o monitoramento distribuído de temperatura.