



**UNIVERSIDADE FEDERAL DO MARANHÃO
CAMPUS DE SÃO LUÍS - CIDADE UNIVERSITÁRIA
ENGENHARIA DA COMPUTAÇÃO
SISTEMAS DISTRIBUÍDOS**

TERMO DE ABERTURA DO PROJETO (TAP)

ALUNOS:

**ANA PATRÍCIA GARROS VIEGAS – 2022003512
ANA POLIANA MESQUITA DE JESUS DE SOUSA – 20250013597
GUSTAVO ANTONIO SILVA ROCHA - 20240065473
LEONARDO DOS SANTOS PEREIRA – 20240065464
WELYAB DA SILVA PAULA – 2021035825**

PROFESSOR: LUIZ HENRIQUE NEVES RODRIGUES

São Luís – MA

2025

SUMÁRIO

1 IDENTIFICAÇÃO DO PROJETO	3
2 CONTEXTO E JUSTIFICATIVA	4
3 OBJETIVO DO PROJETO	5
4 ESCOPO DO PROJETO	6
5 PREMISSAS	7
6 RESTRIÇÕES	8
7 CRONOGRAMA (GRÁFICO DE GANTT)	9

1 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

2 CONTEXTO E JUSTIFICATIVA

Com o avanço das tecnologias de informação e comunicação, os sistemas distribuídos tornaram-se uma base essencial para aplicações modernas que exigem escalabilidade, disponibilidade contínua e tolerância a falhas. Tais sistemas são caracterizados por um conjunto de computadores autônomos, interligados por uma rede, que colaboram para atingir um objetivo comum. Ao contrário dos sistemas centralizados, os sistemas distribuídos distribuem processamento, armazenamento e serviços entre múltiplos nós, oferecendo flexibilidade, desempenho e robustez.

O monitoramento de variáveis ambientais em tempo real é um dos campos em que a aplicação de sistemas distribuídos tem ganhado cada vez mais espaço, especialmente com o crescimento da Internet das Coisas (IoT). Neste contexto, sensores distribuídos atuam como nós autônomos que coletam dados do ambiente e os transmitem para servidores ou plataformas centrais, responsáveis por processar, armazenar e visualizar essas informações. Essa arquitetura reflete diretamente o modelo de sistema distribuído: dispositivos independentes que se comunicam entre si, compartilham informações e operam de forma coordenada.

A escolha por desenvolver um Sistema de Monitoramento Distribuído de Temperatura parte de um cenário real e recorrente em diversos setores. Em data centers, por exemplo, o monitoramento térmico é crítico para garantir o funcionamento adequado dos servidores e evitar superaquecimentos que possam comprometer os serviços. Em hospitais, o controle de temperatura em áreas refrigeradas garante a conservação adequada de medicamentos e vacinas. Em ambientes industriais e residenciais, o uso de sensores para acionamento automatizado de sistemas de climatização aumenta a eficiência energética e o conforto dos usuários.

Ao simular esse tipo de sistema com múltiplos clientes (sensores) e um servidor central, o projeto proporciona aos discentes a oportunidade de aplicar, de forma prática, conceitos fundamentais da disciplina de Sistemas Distribuídos, como:

- Comunicação entre nós (via sockets ou protocolos HTTP),
- Arquitetura cliente-servidor,
- Transparência de localização,
- Escalabilidade horizontal (adição de novos sensores),
- Tolerância a falhas (comportamento resiliente mesmo com a perda de nós),
- e sincronização de informações.

Além disso, o projeto é viável tecnicamente, pode ser implementado com ferramentas acessíveis, e estimula o raciocínio lógico, o desenvolvimento em Python e o entendimento prático das camadas de rede. Assim, a proposta se justifica como um exercício acadêmico relevante, aplicável ao mundo real, e profundamente conectado ao escopo da disciplina.

3 OBJETIVO DO PROJETO

3.1 OBJETIVO GERAL

Desenvolver um sistema distribuído funcional, baseado na arquitetura cliente-servidor, que simule o monitoramento de temperatura em tempo real por meio da comunicação entre múltiplos nós clientes (sensores) e um servidor central responsável pelo recebimento, armazenamento e visualização dos dados.

Esse objetivo visa consolidar, na prática, os principais conceitos teóricos da disciplina de Sistemas Distribuídos, proporcionando aos discentes uma experiência de construção e análise de um sistema realista, mesmo que em ambiente simulado, abordando aspectos como descentralização, comunicação em rede, escalabilidade e tolerância a falhas.

3.2 OBJETIVOS ESPECÍFICOS

1. Projetar a arquitetura do sistema distribuído, definindo claramente os papéis do cliente (sensor) e do servidor, considerando aspectos de modularidade, autonomia dos nós e modelo de comunicação síncrona ou assíncrona;
2. Simular sensores distribuídos, utilizando scripts programados para gerar dados de temperatura com base em variações aleatórias realistas;
3. Implementar a comunicação entre os nós do sistema, aplicando técnicas de socket TCP/IP ou requisições HTTP (usando Flask), para garantir a troca eficiente de mensagens entre clientes e servidor;
4. Desenvolver o servidor central, com capacidade de receber, processar e armazenar os dados enviados pelos clientes em tempo real, assegurando a integridade e sincronização das informações;
5. Construir uma interface de visualização dos dados coletados, que permita ao usuário acompanhar, de forma simples e intuitiva, as leituras de temperatura recebidas pelo servidor;
6. Testar a escalabilidade e a tolerância a falhas do sistema, simulando a adição e remoção de sensores, bem como interrupções pontuais na comunicação, para observar o comportamento do sistema diante de eventos típicos em ambientes distribuídos.

4 ESCOPO DO PROJETO

4.1 ESCOPO DELIMITADO

Este projeto contempla o desenvolvimento de um sistema distribuído simulado para monitoramento de temperatura, utilizando uma arquitetura cliente-servidor. O sistema será composto por múltiplos clientes (representando sensores de temperatura simulados) e um servidor central que será responsável por receber, armazenar e exibir os dados coletados em tempo real.

O escopo compreende:

- Modelagem da Arquitetura do Sistema: definição da comunicação entre os nós, das responsabilidades de cada componente e da estrutura de funcionamento distribuído;
- Desenvolvimento dos clientes simuladores: programas que geram valores de temperatura aleatórios periodicamente, funcionando como sensores independentes;
- Implementação do servidor central: programa que escuta as requisições dos sensores, armazena os dados recebidos e os disponibiliza para visualização;
- Visualização das informações: criação de uma interface simples para apresentar os dados coletados, podendo ser em forma de tabela ou gráfico de linha, com atualização periódica;
- Execução e teste do sistema em ambiente local: simulação de funcionamento com múltiplos sensores, testando o comportamento do sistema em condições de normalidade e falhas;
- Aplicação de conceitos de Sistemas Distribuídos, como escalabilidade, autonomia dos nós, comunicação por rede, e transparência de localização.

4.2 FORA DO ESCOPO

Para garantir a viabilidade do projeto no tempo e recursos disponíveis, ficam fora do escopo:

- Integração com sensores físicos de temperatura (hardware);
- Implantação em redes reais externas (como internet ou ambientes de nuvem);
- Persistência de dados em banco de dados estruturado, como MySQL ou PostgreSQL (será utilizado armazenamento simples ou em memória);
- Implementação de camadas de segurança complexas (como criptografia ou autenticação de múltiplos usuários);
- Interface avançada com painéis customizados, dashboards interativos ou aplicação mobile.

5 PREMISSAS

Para o desenvolvimento e execução bem-sucedida deste projeto, foram estabelecidas as seguintes premissas:

1. Simulação dos sensores: Os dados de temperatura serão simulados por scripts desenvolvidos em Python, utilizando geração aleatória com base em faixas realistas. Não será necessário o uso de sensores físicos ou dispositivos externos.
2. Execução local: Todo o sistema será desenvolvido e executado em ambiente local, podendo utilizar o próprio computador pessoal da aluna. O servidor e os clientes poderão rodar em múltiplas instâncias dentro da mesma máquina ou em rede local (LAN).
3. Ferramentas acessíveis: As bibliotecas e linguagens utilizadas (como Python, Flask, sockets e matplotlib) são gratuitas, de código aberto, e compatíveis com os recursos computacionais já disponíveis.
4. Conectividade entre os nós: Assume-se que haverá conectividade estável entre os processos clientes e o servidor durante os testes, seja via localhost ou rede privada simples, permitindo o tráfego contínuo de mensagens.
5. Capacidade de escalabilidade simples: O sistema será capaz de receber dados de múltiplos sensores simultâneos (clientes) sem degradação severa de desempenho, considerando que o número de clientes simulados será limitado a fins acadêmicos.
6. Infraestrutura suficiente: O equipamento disponível (notebook ou desktop da aluna) terá capacidade de executar todas as tarefas de simulação, programação, visualização e testes do projeto, sem a necessidade de upgrades.
7. Funcionamento assíncrono tolerante a falhas: O servidor deverá estar apto a continuar operando mesmo diante da falha de um ou mais sensores clientes, simulando o comportamento resiliente típico de sistemas distribuídos.

6 RESTRIÇÕES

As restrições a seguir delimitam os recursos, prazos e escopo técnico do projeto, influenciando diretamente o planejamento e a execução das atividades:

- Tempo limitado: O projeto deverá ser concluído dentro de um período de aproximadamente 4 meses, conforme o calendário acadêmico da disciplina. Esse prazo exige foco em um escopo enxuto e viável para o semestre.
- Desenvolvimento : Todas as etapas do projeto serão conduzidas por alunos, o que impõe restrições à quantidade de tarefas simultâneas, ao ritmo de desenvolvimento e à complexidade da solução implementada.
- Ambiente de execução restrito: O sistema será implementado e testado apenas em ambiente local (localhost). Não será feita implantação em redes públicas, ambientes de nuvem ou servidores externos.
- Recursos computacionais limitados: O desenvolvimento será realizado em um computador pessoal, com memória e capacidade de processamento compatíveis com aplicações de pequeno porte. Isso restringe o número de clientes simultâneos e a complexidade da visualização dos dados.
- Ausência de sensores físicos e dispositivos reais: O projeto será baseado exclusivamente em simulação de sensores, o que impede a validação em ambientes físicos ou com coleta de dados reais.
- Sem integração com banco de dados externo: O armazenamento dos dados coletados será feito em memória volátil (durante a execução), ou em arquivos locais simples. O uso de bancos de dados relacionais ou não-relacionais está fora do escopo.
- Ausência de mecanismos avançados de segurança: O sistema não implementará camadas de segurança como criptografia, autenticação de usuários ou controle de acesso por níveis, por se tratar de um protótipo acadêmico.

7 CRONOGRAMA (GRÁFICO DE GANTT)



Mês: Abril

- Semana 1: Escolha do tema, conversa com o professor, definição do escopo inicial.
- Semana 2: Pesquisa teórica sobre sistemas distribuídos, arquitetura cliente-servidor e aplicações reais.
- Semana 3: Estudo técnico das ferramentas: Python, sockets, Flask e matplotlib.
- Semana 4: Esboço da arquitetura do sistema, diagrama de funcionamento e validação do planejamento.

Mês: Maio

- Semana 1: Definição dos módulos do sistema: cliente (sensor simulado), servidor e visualizador.

- Semana 2: Elaboração do cronograma técnico e divisão das tarefas de desenvolvimento.
- Semana 3: Implementação do módulo cliente: geração de dados simulados de temperatura.
- Semana 4: Implementação do módulo servidor: recebimento e armazenamento local dos dados.

Mês: Junho

- Semana 1: Desenvolvimento da interface de visualização (gráficos ou tabela simples).
- Semana 2: Integração dos três módulos do sistema (cliente, servidor, visualizador).
- Semana 3: Testes com múltiplos clientes simultâneos; avaliação da consistência dos dados.
- Semana 4: Testes de falhas, simulação de interrupções e validação da escalabilidade.

Mês: Julho

- Semana 1: Redação do relatório final: metodologia, desenvolvimento, testes e resultados.
- Semana 2: Revisão e formatação do relatório conforme ABNT; finalização dos anexos.
- Semana 3: Apresentação oral (ou gravação da apresentação, se aplicável).
- Semana 4: Entrega final do relatório/documentação e encerramento do projeto.