

# Trabalho Prático 2 - Redes de Sensores Sem Fio (RSSF)

Thaís Ferreira da Silva - 2021092571

Janeiro 2025

## 1 Introdução

As Redes de Sensores Sem Fio (RSSF) são uma tecnologia essencial para a coleta de dados em tempo real em ambientes diversos, encontrando aplicação em áreas como monitoramento ambiental, automação industrial e até biomedicina. Este trabalho propõe a implementação de um sistema que simula uma rede composta por sensores distribuídos em um espaço delimitado, que se comunicam utilizando a interface de sockets POSIX e o protocolo TCP. O servidor gerencia os tópicos associados aos diferentes tipos de sensores, enquanto os clientes representam os sensores, enviando medições e ajustando seus valores com base nos vizinhos mais próximos. Este projeto explora conceitos fundamentais de redes de computadores, como comunicação cliente-servidor, threads, e gerenciamento dinâmico de conexões, consolidando os conhecimentos teóricos por meio de uma abordagem prática.

## 2 Implementação

O sistema foi desenvolvido em C, utilizando a interface POSIX de sockets. A estrutura segue um modelo cliente-servidor, no qual o cliente atua como sensor e envia sua medição a cada intervalo de tempo, e o servidor recebe a medição e distribui para todos os outros sensores de mesmo tipo.

É importante reforçar que toda a base da comunicação cliente-servidor foi feita de acordo com a implementação deixada nas referências do trabalho [1].

Temos no total 4 tipos de arquivos desenvolvidos:

- common: contendo as funções utilizadas em comum pelo servidor e pelo cliente
- client: contendo as funções relacionadas com o cliente ou sensor
- server: contendo as funções relacionadas com o servidor
- Makefile: contendo as instruções para compilar automaticamente os outros arquivos utilizando 'make' para criar e 'make clean' para deletar

## 3 Desafios, dificuldades e imprevistos do projeto

Durante o desenvolvimento do projeto, surgiram diversos desafios e imprevistos que exigiram ajustes no planejamento e soluções criativas. Neste capítulo, serão apresentados os principais obstáculos enfrentados e como eles foram superados, destacando o aprendizado adquirido ao longo do processo.

### 3.1 Planejamento das estruturas a serem utilizadas

De longe, eu acredito que a parte mais complicada do trabalho foi a de planejamento. A utilização das threads gerou a necessidade de uma alocação dinâmica do servidor, para ser capaz de suportar vários clientes conectados ao mesmo tempo, e juntamente uma nova estrutura responsável por associar o socket do cliente aos seus dados armazenados no servidor.

Já no cliente optei por uma alocação estática armazenando os dados relevantes do sensor, os dados de todos os seus vizinhos, e o ponteiro dos 3 sensores mais próximos na vizinhança.

Um ponto importante de se comentar é que inicialmente o cliente foi inteiramente construído para armazenar apenas os 3 sensores mais próximos, e depois adaptado para armazenar todos os sensores da vizinhança devido a minha falta de atenção.

Os problemas relacionados a esta parte envolveram algumas falhas de segmentação e dados armazenados incorretamente.

### 3.2 Fluxo do cliente

Outro desafio interessante de se comentar foi a elaboração do fluxo de como o cliente funciona. Vi no fórum que existem diferentes formas de se receber e armazenar os dados do sensor no cliente e no servidor e optei por fazer alguns tratamentos por conta própria.

Considerando que o servidor pode me mandar apenas as mensagens de interesse do sensor, ou todas as mensagens até que o sensor envie a primeira mensagem, adicionei uma verificação do tipo na thread de processamento no cliente. Além disso, escolhi implementar a abordagem onde o sensor só mandará a primeira medida quando passar o tempo de espera necessário para o seu tipo.

Já no servidor, encaminho as mensagens para os sensores interessados no tópico de seu mesmo tipo apenas após o primeiro envio de sua mensagem. É nesse momento que o servidor cadastra aquele sensor, e o adiciona à lista de interesse do seu tipo.

### 3.3 Legibilidade do código

Por último, optei por melhorar a organização e legibilidade do meu código num geral se comparado com o primeiro trabalho do semestre. Principalmente no cliente, onde ocorrem diversas verificações frequentes, optei pela utilização de diversas funções diferentes e separadas para facilitar a compreensão do fluxo de execução. Isso foi extremamente útil na thread de processamento da mensagem do servidor, pois o código estava muito extenso.

As complicações relacionadas a essa mudança surgiram principalmente na modificação da forma como os parâmetros eram enviados, que passou do envio dos valores da mensagem, para o envio do endereço da mensagem.

## 4 Conclusão

Por fim, gostaria de concluir que o desenvolvimento do trabalho prático sobre o RSSF foi mais simples do que o primeiro trabalho, devido à semelhante base de conexão usando o TCP e o IPv4 e IPv6. No entanto o que complicou foi a adição de threads devido à necessidade de uma conexão simultaneamente de vários sensores com o servidor.

Além disso, acredito que um dos fatores determinantes para o baixo número de erros encontrados ao testar o meu código sozinho ou com o código dos meus colegas, foi novamente a criação de

uma lista de tarefas previas e checkpoints necessários de se concluir para um bom funcionamento do sistema num geral.

## References

- [1] ÍTALO CUNHA. Introdução à programação em redes. <https://www.youtube.com/playlist?list=PLyrHOCFXIM5Wzmbv-1C-qvoBejsa803Qk>, Ago 9, 2020.