

Relatório de Desenvolvimento – Sistema de Chat Simples

Autores: Hugo M. M. Silva, Joselio F. S. Júnior e Shayra Kelly E. Silva

1. Introdução

Este relatório descreve o desenvolvimento de um sistema distribuído simples do tipo chat, construído como parte do Projeto 1 da disciplina de Sistemas Distribuídos. O objetivo do trabalho foi compreender os fundamentos da comunicação entre processos, a coordenação em sistemas distribuídos e os desafios práticos de implementação de um ambiente funcional.

O sistema foi projetado em um modelo cliente-servidor, utilizando sockets e WebSockets para a comunicação entre nós. O relatório apresenta a arquitetura implementada, as principais decisões de design, as dificuldades enfrentadas durante o desenvolvimento em grupo e os aprendizados adquiridos ao longo do processo.

2. Arquitetura do Sistema

A arquitetura do sistema é composta por três módulos principais:

- **Frontend (cliente):** Desenvolvido em HTML, CSS e JavaScript, com suporte em Node.js. É responsável por fornecer uma interface simples para o usuário, onde é possível enviar e receber mensagens em tempo real.
- **Backend-Gateway:** Implementado em Java com Spring Boot, atua como ponto de entrada das mensagens via WebSocket. Seu papel é intermediar a comunicação entre os clientes conectados.
- **Backend-Processor:** Também implementado em Java com Spring Boot, processa as mensagens recebidas, aplicando regras de negócio e repassando as informações quando necessário.

O modelo de comunicação segue a estrutura cliente-servidor distribuída. Os clientes interagem diretamente com o Gateway, que por sua vez repassa as mensagens para o Processor. A camada de transporte é viabilizada por WebSockets, garantindo comunicação em tempo real. O Gateway atua como ponto central de coordenação, controlando a entrada e saída de mensagens e evitando conflitos de comunicação.

3. Principais Decisões de Projeto

Optamos por desenvolver um sistema de chat por se tratar de uma aplicação prática e representativa de sistemas distribuídos, permitindo simular cenários reais de comunicação semelhantes aos aplicativos de mensagens utilizados atualmente. A escolha pelo uso de sockets e WebSockets mostrou-se a mais adequada, tanto pela consolidação dessas tecnologias quanto pela sua compatibilidade com os objetivos do projeto.

O modelo de comunicação adotado contempla interações síncronas e assíncronas, inspirado em trabalhos anteriores da disciplina que já exploravam a troca de mensagens entre máquinas. Por fim, a coordenação do sistema foi mantida de forma simples e centralizada no Gateway, que atua como mediador das interações, sem a necessidade de mecanismos distribuídos mais complexos.

4. Dificuldades Encontradas

Durante o desenvolvimento, enfrentamos algumas dificuldades, como a configuração em múltiplas máquinas, que exigiu atenção especial para garantir o correto funcionamento dos três nós em diferentes ambientes. Além disso, o trabalho em equipe trouxe o desafio de alinhar distintas ideias e estilos de programação, o que demandou reuniões frequentes e colaboração constante. Outro ponto relevante foi a integração das camadas do sistema (*Frontend*, *Gateway* e *Processor*) que necessitou de diversos testes e ajustes no protocolo de comunicação para assegurar o funcionamento adequado. Apesar desses obstáculos, não houve limitações significativas relacionadas às linguagens ou bibliotecas utilizadas, uma vez que as tecnologias escolhidas atenderam plenamente aos objetivos do projeto.

5. Aprendizados

O desenvolvimento do projeto proporcionou diversos aprendizados relevantes. Aprofundamos nossa compreensão sobre sistemas distribuídos, em especial a forma como a comunicação entre processos pode ser estruturada em diferentes máquinas e os papéis desempenhados por cada nó em uma arquitetura cliente-servidor. Também passamos a compreender melhor os conceitos de comunicação síncrona e assíncrona, identificando suas vantagens, limitações e impactos no desempenho do sistema. Além disso, reforçamos a importância de adotar boas práticas de programação, como a modularização do código, a separação clara das responsabilidades entre as camadas de aplicação, transporte e processamento, bem como a documentação adequada do funcionamento do sistema. Por fim, reconhecemos que, em uma futura versão

do projeto, seria interessante investir em melhorias no desenvolvimento e interface do chat, tornando a experiência do usuário mais amigável e completa com os conhecimentos adquiridos futuramente.

6. Conclusão

O projeto permitiu aplicar na prática conceitos fundamentais de sistemas distribuídos, explorando a comunicação entre processos, a coordenação centralizada e os desafios de integração em um ambiente distribuído.

A implementação de um chat simples mostrou-se uma escolha didática e funcional, evidenciando o potencial e as dificuldades de trabalhar com arquiteturas distribuídas. Além disso, o processo de trabalho em grupo contribuiu para desenvolver habilidades de colaboração, resolução de problemas e adaptação tecnológica.