

Sistema Agropecuário Distribuído

Otávio Miranda Côrtes
32211BSI015

Julho 14, 2025

Código fonte disponível no GitHub: <https://github.com/otaviomcortes/Sistemas-Distribuidos/tree/main>.

1 Estado atual do sistema agropecuário distribuído

O projeto tem como objetivo simular, de forma distribuída, a operação de uma fazenda que integra atividades de agricultura e pecuária. Nesta Etapa 2, foi implementado um subconjunto funcional do sistema, composto pela Central de Controle (servidor TCP) e pelo Sensor Ambiental (cliente TCP).

A Central de Controle atua como o componente central do sistema, responsável por receber informações de sensores distribuídos na fazenda. Já o Sensor Ambiental simula um dispositivo autônomo que envia, periodicamente, dados de temperatura e umidade.

Ambos os componentes já estão funcionais e integrados, sendo executados como processos independentes e se comunicando via sockets TCP, futuramente mudanças podem ocorrer para tornar ainda mais real e coerente a comunicação entre esses componentes.

Essa implementação está alinhada diretamente com a proposta definida na Etapa 1, na qual foi elaborada a arquitetura inicial do sistema distribuído agropecuário. A partir dessa arquitetura, foi possível iniciar a implementação prática das entidades, respeitando a divisão entre processos e threads conforme suas responsabilidades. O modelo de comunicação adotado, assim como os tipos de interação entre os componentes, foram projetados conforme os princípios de sistemas distribuídos e estão sendo desenvolvidos de forma incremental nesta etapa.

2 Justificativa para a escolha entre processos e threads

A escolha entre utilizar processos ou threads em cada entidade do projeto foi baseada no grau de independência, na necessidade de isolamento e na forma

como cada componente se comporta dentro de um sistema distribuído.

As entidades que representam dispositivos autônomos, como os sensores ambientais, foram implementadas como processos independentes. Isso porque elas simulam equipamentos físicos separados, que operam de forma isolada da Central de Controle e dos demais sensores. Utilizar processos permite que essas entidades sejam executadas em máquinas distintas, o que se alinha à natureza distribuída do sistema.

Por outro lado, a Central de Controle, que precisa gerenciar múltiplas conexões simultâneas, foi implementada como um processo único que utiliza threads internamente. A cada novo sensor que se conecta, uma nova thread é criada para lidar exclusivamente com aquela conexão. Essa abordagem permite que a Central escute e processe dados de diversos sensores ao mesmo tempo, sem bloquear o funcionamento geral do servidor. O uso de threads nesse caso é mais eficiente, já que elas compartilham o mesmo espaço de memória e possuem menor custo de criação do que novos processos.

Essa divisão garante tanto a independência e escalabilidade dos sensores (como processos), quanto a concorrência eficiente na Central (com threads), aproveitando as vantagens específicas de cada modelo de execução no contexto do sistema distribuído proposto.

3 Descrição da comunicação via sockets

A comunicação no projeto ocorre por meio de sockets TCP (Transmission Control Protocol), estabelecendo um canal confiável de troca de dados entre os componentes do sistema. Essa comunicação é central para a integração entre as entidades distribuídas, e foi implementada inicialmente entre dois elementos principais: a Central de Controle e o Sensor Ambiental.

A Central de Controle atua como um servidor TCP, responsável por escutar conexões em uma porta específica (neste caso, a 12345). Ela permanece em estado de espera, aguardando que sensores se conectem. Quando um sensor se conecta, a Central cria uma thread dedicada para aquele sensor, garantindo que múltiplos sensores possam enviar dados ao mesmo tempo, sem interferência mútua.

O Sensor Ambiental, por sua vez, é um cliente TCP que se conecta à Central e envia, em intervalos regulares, mensagens contendo leituras simuladas de temperatura e umidade. Essa comunicação é estabelecida através de um socket cliente que se conecta à porta do servidor. Após estabelecer a conexão, o sensor utiliza um canal de saída (OutputStream) para enviar mensagens de texto.

A escolha pelo protocolo TCP foi motivada pela necessidade de garantir que os dados cheguem com integridade e na ordem correta. Como as leituras dos sensores são informações sensíveis (que podem ser usadas para tomada de decisão, monitoramento e alerta), é fundamental que não haja perda ou corrupção de mensagens, o que é assegurado pelo uso do TCP. O protocolo também facilita a manutenção de uma conexão persistente, que permite o envio contínuo de dados ao longo do tempo, sem a necessidade de reconectar a cada

envio.

As mensagens trocadas são simples e estruturadas em formato de texto, seguindo um padrão que facilita tanto o envio quanto o processamento:

```
Dados enviados: sensor0:temperatura=22;umidade=46
Dados enviados: sensor1:temperatura=23;umidade=53
```

Figure 1: Mensagens enviadas pelo sensor ambiental

Esse formato foi escolhido por ser leve, de fácil leitura e simples de expandir para incluir novos sensores e tipos de dados futuramente (como localização de animais, presença de chuva, estado de equipamentos, etc.).

Em versões futuras do sistema, há a possibilidade de incorporar o protocolo UDP para tipos de dados onde a velocidade é mais importante do que a confiabilidade como atualizações de localização em tempo real ou transmissão de grandes volumes de dados com baixa criticidade. No entanto, para esta etapa do projeto, o foco foi garantir uma comunicação segura, confiável e controlada, por isso o TCP foi adotado exclusivamente.

4 Instruções de como compilar e executar

Para executar o projeto, é necessário que o Java esteja devidamente instalado na máquina, com o compilador (javac) e a máquina virtual (java) configurados no ambiente. O projeto está dividido em dois programas principais: a Central de Controle, que atua como servidor, e o Sensor Ambiental, que funciona como cliente.

Primeiramente, é necessário compilar ambos os arquivos .java. Isso pode ser feito acessando o terminal e navegando até o diretório onde os arquivos estão salvos. Com os comandos de compilação, os arquivos serão convertidos em bytecode .class prontos para execução.

Após a compilação, o primeiro programa a ser executado deve ser a Central de Controle. Ao iniciá-la, ela ficará em modo de escuta, aguardando conexões na porta definida (neste caso, 12345). Uma vez que o servidor esteja em execução, pode-se abrir outro terminal para iniciar o Sensor Ambiental. Este sensor se conecta automaticamente ao servidor, envia leituras simuladas (como temperatura e umidade) em intervalos definidos, e encerra a conexão após 10 envios.

A execução correta dos dois programas demonstrará a comunicação via sockets TCP, com mensagens sendo enviadas do sensor e recebidas pela Central em tempo real. O processo pode ser repetido com múltiplos sensores para simular um ambiente distribuído mais amplo.