

Definição de Protocolo - Trabalho Redes 2023.2

O objetivo do projeto é a criação de um chat onde o servidor oferece um espaço para que todos os clientes possam trocar mensagens juntos. O **servidor** deve, além de **permitir que os clientes se conectem e se desconectem, receber mensagens e reenviá-las aos clientes** para que todos possam visualizar e ter uma cópia das mensagens do chat. De maneira equivalente, os **clientes** precisam ser capazes de se **conectarem com o servidor, enviar mensagens e receber as informações enviadas pelo servidor**.

Existem algumas decisões-chaves que foram tomadas para garantir o bom funcionamento do par cliente-servidor. Elas são:

1. **Protocolo da camada de transporte:** Este protocolo de chat utiliza o TCP para a comunicação entre o servidor e os clientes. O TCP é responsável por garantir a entrega confiável, ordenada e sem erros das mensagens, bem como o controle de fluxo e o controle de congestionamento na rede.
2. **Porta de Comunicação:** O servidor está configurado para escutar na porta 18000 (`PORTA = 18000`). Essa é a porta pela qual os clientes se conectam ao servidor para iniciar a comunicação.
3. **Endereço do Servidor:** O servidor é vinculado a um endereço IP específico, que neste caso será definido em sala no momento do teste. Antes disso, cada grupo pode se vincular ao próprio "localhost" para momentos de teste. Os clientes usam esse endereço para se conectarem ao servidor.
4. **Formato da Mensagem:** O protocolo define o formato das mensagens trocadas entre os clientes e o servidor, que é codificado em UTF-8 (`FORMAT = 'utf-8'`).
5. **Tamanho do Cabeçalho (HEADER):** O cabeçalho da mensagem tem um tamanho fixo de 64 bytes (`HEADER = 64`). O cabeçalho é usado para informar o servidor sobre o tamanho da mensagem real que será enviada, permitindo que o servidor saiba quantos bytes esperar.
6. **Desconexão:** O valor `":D"` é definido como a mensagem de desconexão (`DISCONNECT = ':D'`). Quando um cliente envia essa mensagem, o servidor entende que o cliente deseja se desconectar.
7. **Conexão e Comunicação:** O servidor usa um socket para aceitar conexões de clientes e iniciar threads para lidar com cada conexão. O loop no `'handler'` processa a comunicação com o cliente, recebendo mensagens, verificando o cabeçalho para determinar o tamanho da mensagem e enviando mensagens de volta para o cliente.