

Primeiro no servidor ao receber o nome do arquivo pelo cliente, o servidor verifica a existência do arquivo pela função `arquivo_existe`. Ao não encontrar o arquivo, uma exceção responde ao usuário informando um erro. Caso o arquivo exista, o servidor então calcula o tamanho do arquivo pela função `tamanho_arquivo`, então lê o arquivo byte a byte para determinar seu tamanho. Depois é convertida em string e enviada ao cliente via `sock.sendto`, que envia os dados através do socket UDP.

Já do lado no cliente, após enviar o nome do arquivo para o servidor, o cliente aguarda receber o tamanho do arquivo. Esse tamanho é obtido pela `sock.recvfrom`, na qual escuta para mensagens do servidor. Quando a mensagem recebida for `ERROR`, o mesmo interpreta que houve um erro na solicitação por exemplo: o arquivo não foi encontrado e informa ao usuário via mensagem de erro. Caso contrário, tenta converter essa mensagem em um valor inteiro, porque o tamanho do arquivo deve sinalizar o tamanho que o cliente deve esperar do processo de recebimento dos dados.

Caso o tamanho do arquivo seja maior que zero, o cliente prepara para gravar o arquivo recebido. O mesmo cria um novo arquivo na pasta e inicia um loop de recebimento, onde vai coletar os dados do arquivo até os receber em blocos de 4096 bytes, até que o tamanho total seja alcançado. Isso garante que o cliente não somente tenha ciência do que esperar, mas que a gravação do arquivo aconteça de forma mais organizada e correta.