



## 1. Protocolo de Transporte - [TAN96], pg. 552 - 600

Serviço de transporte	Orientado à conexão (confiável) – para o usuário os erros são transparentes, e ele simplesmente cria um ponto de conexão (soquete) com o destino, sem se preocupar em corrigir os erros, pois é função da camada 4.
	Orientado a datagrama (não confiável) – o usuário utiliza esse serviço quando não tem compromisso com a confiabilidade, ou ela é menos importante, como por exemplo transmissão de voz.

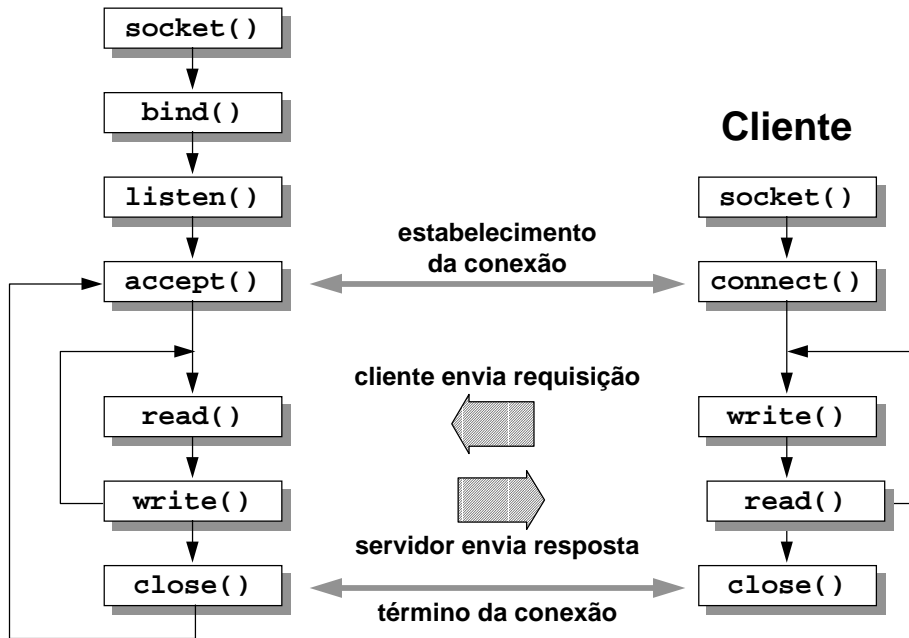
### Primitivas para um serviço simples de transporte

Primitiva	TPDU enviada	Significado
LISTEN	Nenhuma	Bloquear até que algum processo tente se conectar
CONNECT	CONNECTION REQ.	Tentar ativamente estabelecer uma conexão
SEND	DATA	Enviar informações
RECEIVE	Nenhuma	Bloquear até que uma TPDU DATA chegue
DISCONNECT	DISCONNECTION REQ.	Este lado quer encerrar a conexão

### 1.1 API de Sockets

Um *socket* pode ser considerado como um ponto de referência para o qual as mensagens são enviadas e a partir da qual as mensagens podem ser recebidas. Dessa forma, elas estabelecem um canal de comunicação e podem começar a enviar e receber mensagens uma para a outra. Uma estação de trabalho A que deseje se comunicar com uma estação de trabalho B deve saber o número do *socket* aberto pela aplicação que roda na estação B e vice-versa.

A seqüência típica de utilização de *sockets* para comunicação entre um processo cliente e um servidor está ilustrada na figura a seguir. O procedimento é assimétrico, porque um dos processos (o servidor) espera que o outro requisiute a conexão (o cliente).



**Figura: seqüência de chamadas para comunicação orientada a conexão**

Na seqüência da figura acima, o servidor atende apenas um cliente, pois suporta uma única conexão aberta de cada vez. Tal interação é, entretanto, bem mais simples do que a de um servidor que precisa atender múltiplas chamadas concorrentemente. Neste caso, o servidor precisa continuamente verificar novos pedidos de requisição, ler das conexões já estabelecidas e tratar conexões falhas.

## 1.2 Exemplo: aplicativo utilizando sockets

/\*\*/ mostrar programa java