## Lista de exercícios - Sockets TCP e UDP

Nome: Pedro Maurer

1) Analise os exemplos dados em aula de Sockets TCP, altere os mesmos, e responda as seguintes perguntas:

a. O que acontece quando um cliente tenta se conectar a um servidor inativo/inexistente?

Quando um cliente tenta se conectar a um servidor inativo ou inexistente, geralmente ocorre um erro na tentativa de conexão. Esse erro pode variar dependendo da linguagem de programação ou da biblioteca de rede utilizada, mas geralmente é uma exceção ou um código de erro que indica que a conexão não pôde ser estabelecida. Isso pode acontecer, por exemplo, se o servidor não estiver em execução na porta especificada, se o endereço IP estiver errado ou se o servidor estiver inativo.

b. As operações send e receive são bloqueantes ou não? Se sim, por quê?

As operações `send` e `receive` em comunicações de rede podem ser bloqueantes ou não, dependendo de como são configuradas. Em muitas implementações, essas operações são bloqueantes por padrão, o que significa que o programa cliente ficará parado até que a operação seja concluída. Isso pode ser útil em alguns casos, mas também pode levar a bloqueios se a conexão for perdida ou se houver problemas na rede. No entanto, é possível configurar essas operações para serem não bloqueantes, o que permite que o programa cliente continue sua execução enquanto a comunicação ocorre em segundo plano.

c. Os timeouts são aplicados por default em operações connect, send e receive? É possível alterá-los (ou defini-los, caso não sejam utilizados)?

Os timeouts não são aplicados por padrão em todas as operações de rede, mas é uma prática comum definir timeouts para evitar bloqueios indefinidos. Os timeouts podem ser definidos para operações `connect`, `send` e `receive` para garantir que o programa cliente não fique esperando indefinidamente por uma resposta do servidor. A capacidade de definir e modificar timeouts geralmente depende da biblioteca ou API de rede utilizada. Por exemplo, em muitas implementações de soquetes em C/C++, você pode usar a função `setsockopt` para configurar um timeout para operações de soquete.

d. O servidor, ao chamar o método accept para aguardar conexões, fica esperando para sempre ou existe algum timeout? É possível definir/modificar isso?

Quando um servidor chama o método `accept` para aguardar conexões, essa operação pode ser bloqueante por padrão. O servidor ficará aguardando conexões indefinidamente até que uma conexão seja estabelecida ou ocorra um erro. No entanto, é possível definir um timeout para a operação `accept` em muitas implementações. Isso permite que o servidor aguarde uma conexão por um período limitado e, se nenhuma conexão for estabelecida dentro desse período, o `accept` retornará com um erro ou um valor especial para indicar que nenhum cliente se conectou. A capacidade de definir ou modificar esse timeout dependerá da API ou biblioteca de rede que está sendo usada.