



INSTITUTO FEDERAL DA PARAÍBA  
CAMPUS CAMPINA GRANDE  
TECNOLOGIA EM TELEMÁTICA  
DISCIPLINA DE PROGRAMAÇÃO III  
PROF. VICTOR ANDRÉ PINHO DE OLIVEIRA

Programação III

Atividade Prática 3 - Sockets TCP

## Instruções

Leia atentamente as questões abaixo e resolva em sua IDE de preferência. Não esqueça de submeter as questões no Classroom. Lembre-se que a avaliação da disciplina será contínua, prioritariamente por meio de exercícios práticos. E, mais uma vez, fique atento ao prazo. As questões de 1 a 4 valem 1 ponto e a questão 5, 2 pontos. Bom trabalho!

## Exercícios

- 1) Altere a função `recvall` da Listagem Servidor/Cliente Melhorado para que, ao invés de a função receber a quantidade de bytes, receba um caractere que indica o fim da mensagem (por exemplo, uma !). A função deverá retornar a mensagem (em bytes) incluindo o caractere. Lembre-se de converter o caractere (string) para bytes.

A função deverá ter a seguinte assinatura:

```
def recvall(sock, caractere):
```

As mensagens do Servidor e do Cliente vão seguir o seguinte padrão:

```
sock.sendall("Alguma frase que termine com !".encode())
```

ou

```
sock.sendall("Alguma frase que termine com .".encode())
```

ou

```
sock.sendall("Alguma frase que termine com ?".encode())
```

Certifique-se de que tanto o Servidor quanto o Cliente encerra sua mensagem com tal caractere.

- 2) Altere a Listagem Servidor/Cliente Melhorado de modo que o Cliente envie dois números de mesmo comprimento (p.e., 1234 e 4321) para o Servidor e este retorne o menor deles para o Cliente. Os números devem ser lidos pelo usuário.
- 3) Altere a Listagem Servidor/Cliente Melhorado de modo que o Cliente leia do usuário 2 números de até 5 dígitos. Antes de enviar cada número, o Cliente deve primeiro enviar o comprimento do número. O Servidor deverá realizar a soma dos números e retornar o resultado ao Cliente, sem esquecer de antes enviar o comprimento do número.
- 4) Altere a Listagem Servidor/Cliente Upper de modo que o Cliente primeiro envie para o Servidor um caractere (um byte) que indicará o fim da mensagem e depois envie a mensagem terminada por esse caractere. A função do Servidor será devolver a mensagem em caixa alta. O Cliente deverá exibir a mensagem sem a presença desse caractere terminador.
- 5) Que tal combinarmos em um só programa (Servidor/Cliente) os Sockets UDP e TCP? Pois é exatamente o que a questão te proporá a fazer. Aproveite o código Servidor/Cliente Broadcast (da Aula Sockets UDP - Parte 2) de modo que o Cliente continue a usar a função `find_server` para buscar o Servidor em Broadcast (usando, obviamente Socket UDP) e, após encontrá-lo, passem a se comunicar através de Sockets TCP. Modifique o corpo da função `main` de modo que o Cliente envie apenas uma mensagem de olá (pode ser de tamanho fixo) e o Servidor responda com outra mensagem de olá (pode ser de tamanho fixo).

**Dica:** Evite usar acentos na mensagem para não errar na contagem de bytes, já que um caractere acentuado equivalerá a mais de um byte.